https://doi.org/10.48009/1_iis_2011_366-372

Issues in Information Systems

Volume XII, No. 1, pp. 366-372, 2011

A FRAMEWORK AND DEMO FOR PREVENTING ANTI-COMPUTER FORENSICS

Chris B. Simmons, University of Memphis, cbsmmons@memphis.edu
Danielle L. Jones, University of Memphis, dljones9@memphis.edu
Lakisha L. Simmons, Indiana State University, lakisha.simmons@indstate.edu

ABSTRACT

Computer forensics has been researched extensively within industry and academia. Anti-computer forensics conceals and deletes computer data. Currently, organizations rely on computer investigators or fraud specialists to conduct audits to unveil misconduct. There is an increased need to develop a preventative anti-computer forensics framework to assist organizations who are not privy to a specialist. In this paper, we propose a preventative anti-computer forensics framework and brief demonstration. This framework will assist organizations with strengthening their approach to thwarting tools and procedures conducted by criminals.

Keywords: Computer Forensics, Anti-Computer Forensics

INTRODUCTION

Computer forensics tools allow investigators to gain intelligence about a computer's activities, regarding the recovery of deleted files and activities. As well, anti-computer forensics has been thoroughly researched involving concealing data from a computer investigation. A preventative measure to anti-computer forensics still remains a topic requiring further research. Anti-forensics is defined as a collection of tools and techniques used to avoid detection [25]. Anti-computer forensics can be defined as the use of a variety of armaments to obfuscate data, not only via technology but also human aspects. One of the pressing issues of a computer forensics investigator is the constant evolution of anti-computer forensics tools.

Investigators are to diligently conduct a computer forensics investigation to ensure that evidence is not impaired when admitting results into any legal matter. The aim is to provide a framework that will assist an organization in setting policies and procedures to thwart anti-forensics activity that would hinder a computer forensics investigation. If ineffective recovery processes are used, the potential for mutable evidence and contamination is likely [4].

In this paper, we propose a preventative anti-computer forensics (PACF) framework for defending against anti-forensics tools and methods. Satpathy and Mohapatra [18] highlighted one of the major issues to collecting efficient data is to determine how rapidly one can collect and normalize digital evidence from various sources including firewalls, hosts, routers, and vulnerability repositories. Our framework is poised to assist this issue and consists of an iterative process involving acquisition, analysis, presentation, deterrence, and baseline. Each phase within the PACF framework entails specific information that can be used in an iterative process to assist with preventing anti-computer forensics processes and tools.

The evaluation of this framework will consist of a literature review of the current anti-forensics frameworks and their commonalities therein. We will document the analysis of our framework using a simple scenario of an anti-forensics tool to see if our framework is capable of capturing the process of the technology and/or human hacker. Cronin [1] stated that a committed resource must be able to defend information warfare strategies to protect themselves against theft, disruption, distortion, denial of service, or destruction of sensitive information assets. Berinato [2] highlighted the current state of investigating computer crimes involves tools used by the investigators. Computer crimes have made a recent increase in tools used to circumvent detection during the investigation. Moreover, current computer forensics tools are insufficient in capturing and analyzing large amounts of data where an investigator is able to determine a pattern. This is a shocking discovery as it is important to identify any possible contamination or loss of data that has occurred during a computer forensics investigation [3]. In this paper we present a preventative anti-computer forensics framework to assist organizations and academia towards thwarting anti-forensics tools and procedures conducted by criminals.

This paper is organized as follows: In section 2 we provide a literature review on computer forensics, anti-computer forensics, and computer security. In section 3 we provide an overview of a preventative anti-computer forensics framework. In section 4 we present an example of our proposed solution using a pre-computer forensics investigation and in section 5 we conclude our paper and provide insight for future work.

LITERATURE REVIEW

In this section we provide a review of the current state of anti-computer forensics and computer security relating to the discovery of pertinent information

Rogers [6] proposed four categories involving anti-forensics tactics: data hiding, artifact wiping, trail obfuscation, and computer forensics tools and process attacks. Harris [6] extended Rogers' categories to propose an alternative four categories: evidence destruction, evidence hiding, evidence source elimination, and evidence counterfeiting. Harris [6] described the destruction of evidence as dismantling evidence leading to its ineffectiveness while conducting an investigation. Hiding evidence relates to the malicious activity of removing data from plain sight [6]. Hiding evidence can be circumvented with a diligent and thorough investigation. Elimination of evidence sources involves preventing data from being stored initially. This is a crucial blow to investigators, as it will take more of a process or procedure to deter this anti-forensics category. Guirgus et al. [9] mentioned the use of eliminating evidence through the use of a burn before reading framework to combat live forensics. Counterfeiting evidence is the act of forging evidence to appear a particular way different from the true evidence.

Guirgus et al. [9] proposed a burn before reading framework that is used to combat live forensics investigations. This framework is developed to defeat computer forensics investigations, through triggers which can alert and alter once an investigation begins. The metasploit anti-forensics project (MAFIA) [24] consists of an arsenal of three anti-forensics tools: Timestomp, Slacker, and Sam Juicer. These tools can disrupt forensics investigations. Sam Juicer [24] provides the ability to dump hashes from the SAM without writing to the disk. Slacker [24] uses the slack space of the NTFS file system to hide files from an investigation, but with some level of understanding and due diligence, files are retrievable. Timestomp [24] allows a computer hacker to disrupt the New Technology File System (NTFS) timestamp values, which are modified, accessed, created, and entry modified.

Harris [6] highlighted three aspects that require further research when preventing anti-forensics software and processes. The three aspects consisted of the human element, dependence on tools, and physical/logical limitations. The human element seems the most difficult to overcome, as it depends on the investigators' level of expertise. Dependence on tools suggests investigators depend on tools that are subject to attacks, which may produce flaws in the results. Physical limitations pertain to the storage format of an investigation and logical limitations involve storage space limitations as well as time and money. Harris [6] highlighted the need for forensics tools to perform in a manner that conceals the running of the forensics investigation tool. Anti-forensics software is designed to locate the use of computer forensics software. Once the investigation is initiated, anti-forensics software is initiated to obfuscate pertinent data to an investigation. Providing a means of dismantling a computer forensics investigation into small sub-sections may conceal the use of the tool.

Tang [15] proposed two algorithms to reconstruct falsified computer logs for computer forensics investigations. Han et al. [20] highlighted the importance of frequent pattern documentation for data indexing, classification, clustering, etc. An accurate cyber attack classification is pertinent for damage assessment and recovery. Web logs are used to locate potential attack vectors within a particular application. This provides insight into the applicability of our proposed PACF framework that enables the prevention of human and technical aspects when preventing and proving the corruption of data.

Furthermore, Ninget al. [21] proposed three utilities to facilitate correlating a large dataset of Intrusion Detection System (IDS) related alerts. These utilities are adjustable graph reduction, focused analysis, and graph decomposition. This resulted in the correlation consequences of earlier events with prerequisites of later events. Ning et al. [22] extended this work further to construct attack scenarios using hyper-alert type representing prerequisite and consequences for each alert type. Wu et al. [19] used correlation techniques to propose a graph

similarity-based approach to event analysis. Within this work, identifying anomalies fall into two categories, signature based and statistical based. The statistical based approach compares the techniques of known attacks, security policies, undesirable states, and appropriate configurations [19]. The correlation techniques contain a set of carefully selected attributes to locate the patterns of different events. The current state of computer forensics, as a whole, can use anomaly based forensics software to correlate aspects of data ensuring the most up-to-date information is captured. The PACF framework intends to compensate for human as well as technical misconduct.

PREVENTATIVE ANTI-COMPUTER FORENSICS (PACF) FRAMEWORK

In this section we highlight the basic components of our preventative anti-computer forensics framework. It has five basic components: acquisition, analysis, presentation, deterrence, and baseline. The major goals are described from a data fusion [18] and anomaly [7] point of view:

- Enable a descriptive protocol to prevent human related activities involving anti-forensics technology
- Enable a decisive description of all anomalous activities to construct a complete path to an attack and infrastructure restore
- Predict, correlate, analyze, and disseminate adversary actions and impact
- Ensure digital information is free of omissions and capable of being presented as evidence
- Enable suitable evidence for prosecution.

With these goals in mind, each component is described below for further discussion.

Baseline

The Baseline component provides the historical data for the machine learning process. Karlsen [7] highlighted the use of machine learning methods to autonomously learn the normal behavior based on a set of historical data. Historical data is represented as learned behavior used to compare instances against the baseline to alert abnormal behavior. Satpathy and Mohapatra [18] proposed a data fusion based technique for an effective forensics tool. In the proposed PACF framework, we use the baseline phase to define the standard image all machines should not vary significantly from. This base image will be the bases for future computer forensics investigations.

Acquisition

The Acquisition component is used from Carrier [14] involving the accumulation of digital evidence for later analysis. This phase provides a mechanism to collect all pertinent information that may provide data used in a forensics investigation. Satpathy and Mohapatra [18] highlighted that one of the major issues to collecting efficient data is to determine how rapidly one can collect and normalize digital evidence from various sources including firewalls, hosts, routers, vulnerability repositories, etc. During the acquisition phase our framework can be used to ensure that the data captured is free of omissions through various bit-by-bit images of the disk taken at random intervals. Through comparison of the images to the baseline, it can be surmised if anti-forensics activity has occurred.

Analysis

The Analysis is used from Carrier [14] to analyze the captured data to identify applicable information for evidence. Using our approach this data can come from various sources and correlated to make meaningful evidence. Computer forensics is a reactive process, as opposed to our framework being a proactive process. Acquiring one or more images allows for a comparative analysis of the bit-by-bit to the baseline untampered system. If the percentage of change is greater than the company's standard, then it can be hypothesized that anti-forensics activity has occurred, and thus further analysis is necessary.

Presentation

The Presentation component is used from Carrier [14] to report results of a computer forensics investigation without omission. We further extend the presentation to include dispersing the information within the organization to make employees feel comfortable with preventing anti-forensics activities. This aspect involves a social implication. Stahl [12] defines forensics computing as a social observance, which interacts with various facets of society and how the society uses technology. Regarding the entrance of the resultant data presented as evidence, it must pass the "Daubert Test" which is a US Supreme Court ruling involving the admission of expert testimony in Daubert vs. Merrell Dow Pharmaceuticals [17].

Deterrence

Deterrence is derived from the Presentation Component's social observance which seeks to understand the motivation of a hacker or fraudster and to motivate and educate ethical machine usage. Knowing the objective of a hacker enables an organization to devise deterrence policies for both internal and external occurrences. When dealing with deterrence it involves staying abreast of the potential vulnerabilities, whether technically or human based. Massi et al. [23] proposed a Botnet detection and mitigation strategy used to assist an investigator with modeling botnets by creating flowcharts, conducting simulation of network traffic, and study botnet topologies and their lifecycle. Awareness training allows a company to clearly define what is and is not acceptable machine usage and state consequences of such behavior. If actions are not within bounds, then scaling back a user's access and requiring authorization for processes that cause modification to system files could be used as further deterrence. This information assists an organization with deterrence procedures and policies.

Figure 1 highlights the iterative process within the PACF framework. This process can be used in organizations to prevent the use of anti-computer forensics processes and tools. The initial phase is the analysis phase that begins the process of anti-computer forensics prevention. We constantly analyze the data to prepare for presentation. Once we have presented the data omissions free, we then can implement a deterrence mechanism. Following the deterrence phase, we move forward with creating a baseline of all the activities that were captured and disseminated within the organization and/or computer forensics software. We envision this framework to be an iterative process to capture information where investigators are privy to the latest information involving anti-computer forensics techniques.

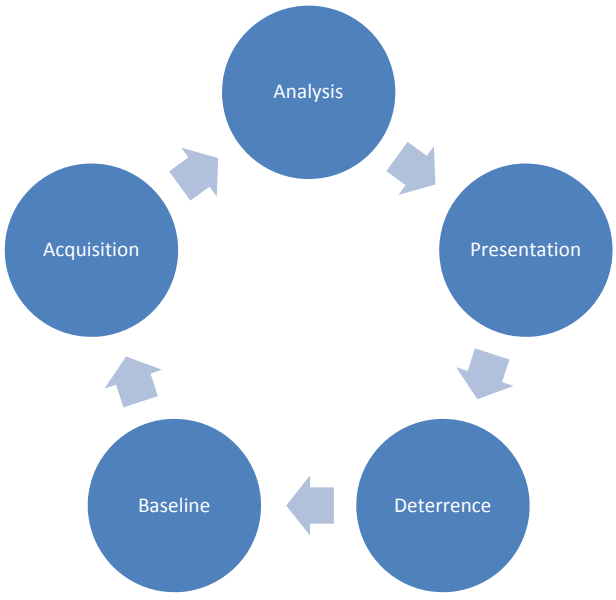

Figure 1: Preventative Anti-Computer Forensics (PACF) Framework

Volume XII, No. 1, pp. 366-372, 2011

METHODOLOGY OF DEMO AND REPORT

In this section we present our methodology of the anti-forensics demonstration and preliminary discussion of the viability of using the PACF framework. Berinato [2] discussed in his report the call to improve computer forensics toolsets as penetration tools such as the Metasploit Framework are often used to perform anti-computer forensics.

We conducted an experiment to study the technology associated to anti-computer forensics. The experiment's goal was to successfully launch an attack on a machine and check to see if the framework proposed could help prevent or identify wrong doing. The main tool used to run this experiment was Metasploit which is a tool for the development and execution of exploits against a target machine, the culprit for many anti-forensics and evasion tools.

The Experiment Methodology

Open source software was used to conduct the exploitation associated to the PACF framework. The concurrent version system (CVS) was used to capture the baseline of the target system [26]. The CVS server pointed to several critical folders located on the target system. The Wireshark Network Protocol Analyzer [27] was used to acquire the necessary traffic data for analysis. Scripts were developed using Microsoft's LogParser [28] as sensors running in five minute intervals for notification and analysis of changes involving the target system. LogParser was further used to parse the WireShark capture files to correlate the raw network traffic with changes occurring on the target system. For presentation the results retrieved from LogParser was sent to a MySQL database table for analysis of specific events. Human intervention is currently required to implement deterrence policies to thwart the current attack and/or prevent any future attack from the presented results.

In the demo, two dedicated machines were used on a closed network, both running Microsoft XP SP3. These systems contained the Metasploit Framework installed as well as WireShark. Both systems also contained screen capture software, SnagIt [29], to demonstrate the exploit in progress. The attacker system did not have an antivirus program installed due to its interference with Metasploit's unsafe contents. The experiment is based on available metasploit video tutorials found at YouTube.

The Exploit

First, a payload was written as an executable file to open a back door on the target system. This file was housed on the remote target machine, which is required for this attack to work. This file could possibly be transferred through social engineering via an email attachment.

msf> **windows/meterpreter/reverse_tcp LHOST = [MY IP ADDRESS] LPORT = [MY PORT NUMBER] X > myfile.exe**

Second, the exploit selected was the multi-handler listener which allows the opening of a remote shell session through a reverse tcp connection.
msf> **use exploit/multi/handler**
msf exploit(handler) > **set PAYLOAD windows/meterpreter/reverse_tcp**

Thirdly, the exploit requires the attacker set the host IP address before the exploit can begin.
msf exploit(handler) > **set LHOST [MY IP ADDRESS]**
msf exploit(handler) > **exploit**

Now, the attacker has control of a shell session on the remote system. This experiment clears the event logs.
meterpreter> **clearev**

This demo was a simple experiment to uncover the simplicity of using anti-computer forensics tools to conceal wrongdoing. While this experiment was being run on the two separate systems, WireShark captured the packets for

both systems' network card traffic and SnagIt captured the screen activity. The packet dumps were analyzed by LogParser to identify sensor specific information that the packet dump provided. Along with the packet dump, log parser was able to identify the exact file which was installed and capture the before and after of the event logs, which are stored in the database. This information is used for the presentation of anti-computer forensics activities. The results highlight a file change and deletion of event logs. This information is presented to the computer forensicist or security specialist to infer a malicious activity has taken place. In this experiment, the concurrent version system was used to recover the vital information from the baseline. Currently, in the experiment, the deterrence policy is the responsibility of human intervention. Any decision decided by the specialist will be added to the baseline for preventing future anti-computer forensics activities.

CONCLUSION

This paper is an attempt to provide an intuitive evaluation of the anti-computer forensics tools and processes that are used to tamper evidence. We assume the collected anti-computer forensics frameworks have a prevailing impact on many organizations. We believe providing a preventative anti-compute forensics framework to thwart malicious activities will assist organizations with a level of confidence in forensics investigations.

As attackers become more creative in devising attack against computer forensics investigations, it is imperative that organizations formulate procedures that assist with defending against anti-forensics activities. For this reason, we proposed a preventative anti-computer forensics framework which uses an anomaly detection framework to assist organizations with thwarting against anti-forensics tools and procedures.

We are aware of potential new anti-computer forensics manifestations; an anomaly based preventative anti-computer forensics based framework is needed to detect signatures along with statistics that provide information regarding criminal activity in an organization. PACF will provide a defender with the information to make an educated decision to defend against anti-computer forensics. Creative approaches to defending attacks will become available and providing an extensible framework will assist with capturing fraudulent computer activities in an enhanced timeframe. We believe PACF provides a foundation for new preventative anti-computer forensics research to grow as defenses become more sophisticated. In future work, PACF will be the basis for the development of a knowledge management system to track human and computer methods involving anti-computer forensics methods to disseminate in an organization.

REFERENCES

1. Cronin, B. and Crawford, H., "Information warfare: Its Application in military and civilian contexts", Information Society, volume 15, pp. 257-263, 1999.
2. Berinato, Scott. "How Online Criminals Make Themselves Tough to Find, Near Impossible to Nab," Retrieved March 20, 2011.
3. Boddington, R., Hobbs, V. J., & Mann, G. "Validating digital evidence for legal argument," The 6th Australian Digital Forensics Conference, 2008.
4. Bell, G. B., Boddington, R. "Solid State Drives: The Beginning of the End for Current Practice in Digital Forensic Recovery?," *Journal of Digital Forensics*, 5(3), 2010.
5. http://www.forensicswiki.org/wiki/Anti-forensic techniques Retrieved February 27, 2011.
6. Harris, R. (2006). Arriving at an Anti-Forensics Consensus: Examining How to Define and Control the Anti-Forensics Problem. *Proceedings of the 2006 Digital Forensics Research Workshop. Digital Investigation*, *3*(S), 2006.
7. Karlsen, K. "Profile Based Intrusion Detection for Internet Based Banking," Master Thesis, Norwegian University of Technology, 2008.
8. Olsson, J. and Boldt, M. "Computer Forensics timeline visualization tool," Journal of Digital Investigation, 6(1), 2009.
9. Mina Guirguis, Jason Valdez, Bassam El Lababedi and Joseph Valdez. "Burn Before Reading: A Stealthy Framework for Combating Live Forensics Investigations". In Proceedings of 4th APWG eCrime Researchers Summit, October 2009.

10. Berinato, Scott. "The Rise of Anti-Forensics," Retrieved March 20, 2011.
11. Ridder, C. K. "Evidentary Implications of Potential Security Weaknesses in Forensic Software," International Journal of Digital Crime and Forensics, 2009.
12. Newsham, T., Palmer, C., Stamos, A., & Burns, J. "Breaking forensics software:Weaknesses in critical evidence collection," Retrieved from www.isecpartners.com/files/iSEC-Breaking_Forensics_Software-Slides.BH2007.pdf, May 2007.
13. Stahl, B. C., "Forensic Computing in the Workplace: Hegemony, Ideology, and the Perfect Panopticon?" Journal of workplace Rights, 13(2), 2008.
14. CARRIER B.: "Open Source Digital forensics Tools: The Legal Argument," Research Report, Oct. 2002.
15. Tang, M., Fidge, C. "Reconstruction of Falsified Compute Logs for Digital Forensics Investigations," AISC '10 Proceedings of the Eighth Australasian Conference on Information Security, 2010.
16. Kessler, G.C. and Fasulo, M. " The case for teaching network protocols to computer forensics examiners," In *Proceedings of the Conference on Digital Forensics, Security and Law*, April 18-20, Arlington, VA, pp. 115-137, 2007.
17. Supreme Court of the United States. *Daubert v. Merrell Dow Pharmaceuticals*
18. Satpathy, S. and Mohapatra, A. "A Data Fusion Based Digital Investigation Model as an Effective Forensic Tool in the Risk Assessment and Management of Cyber Security Systems,"The 7th International Conference on Computing, Communications and Control Technologies, 2009.
19. Qishi Wu, Yi Gu, Xiaohui Cui, Praneeth Moka, Yunyue Lin. "A Graph Similarity-Based Approach to Security Event Analysis Using Correlation Techniques," GLOBECOM 2010.
20. J. Han, J. Pei, Y. Yin, and R. Mao. Mining frequent patterns without candidate generation: a frequent-pattern tree approach. Data Mining and Knowledge Discovery, 8(1), pp. 53–87, Jan. 2004.
21. P. Ning, Y. Cui, and D. S. Reeves. Analyzing intensive intrusion alerts via correlation. In Proc. of the 5th Int'l Symposium on Recent Advances in Intrusion Detection (RAID 3002), October 3002.
22. Ning, P., Cui, Y., Reeves, D.: Constructing attack scenarios trough correlation of intrusion alerts. In: CCS '02: Proc. 9th ACM Conference on Computer and Communication Security, ACM Press (3002) 245-254.
23. Massi, J., Panda, S., Rajappa, Gl, Selvaraj, S., Revankar, S. "Botnet Detection and Mitigation," Proceedings of Student-Faculty Research Day, CSIS, Pace University, 2009.
24. Metasploit Anti-Forensics Project: Timestomp, Slacker, SAM Juicer, http://www.google.com/support/chrome/bin/answer.py?answer=95464&hl=en-US, Retrieved on March 22, 2011.
25. Simson Garfinkel. Anti-Forensics: Techniques, Detection, and Countermeasures. *International Conference on i-Warfare and Security (ICIW)*, Naval Postgraduate School, Monterey, CA, March 2007.
26. CVS Suite 2009, http://www.march-hare.com/cvspro/, Retrieved on July 1, 2011.
27. Wireshark Network Protocol Analyzer, http://www.wireshark.org/download.html, Retrieved on July 1, 2011.
28. Microsoft Log Parser 2.2, http://www.microsoft.com/download/en/details.aspx?id=24659, Retrieved on July 1, 2011.
29. SnagIt Screen Capture, http://www.techsmith.com/snagit/default.asp, Retrieved on July 1, 2011