DESIGN AND IMPLEMENTATION OF A NETWORK FORENSICS SYSTEM FOR LINUX

Hong-Ming Wang
National Kaohsiung Normal University
Kaohsiung, Taiwan
alexwang24@gmail.com

Chung-Huang Yang
National Kaohsiung Normal University
Kaohsiung, Taiwan
chyang@computer.org

Abstract—Technological advances of the Internet not only facilitate human life, but also give opportunities to attackers more easily conduct the activities of network intrusion and destruction. Network forensics is a forensic science and an important technology for network security realm. In this paper, we develop a network forensics system for Linux, which is used to collect and protect evidences when the cyber crime occurred. It consists of a live system, a friendly graphical launch menu, strengthen PyFlag software, and integrate required tools of system and network. This system can expand its volatile, report presentation functionalities, and provide investigator to perform network forensics work quickly and correctly. The result of the forensics in this system can not only preserve evidences of the cyber crime, but also help organizations and institutions to understand the whole context of network security incidents and to strengthen the network host defense and security policy.

Keywords : Network Forensics; Computer Forensics; Digital Evidence; PyFlag

I. INTRODUCTION

Internet shortens the physical distance barrier, so that people can easily share information with each other in real time. The more applications in Internet, the more people rely on the actions. According to the Computer Crime and Security Survey [1] showed that 64% respondents of information security and information technology professionals in United States had dealt with malware events (50% in 2008), 29% dealt with denial of service events (21% in 2008), 23% dealt with bots events (20% in 2008), these statistics reflected the rising on cyber crime in past, and the more serious is more than the past.

In order to assist investigating the fact of cyber crime or network intrusion, this study focuses on a compromised Linux host network forensics. We demonstrate network forensics work of a victim host by collecting the evidences, such as storage image, related communication packets, logs and records. We analyze these evidences and try to discover criminal behaviors, present the obtained evidences and findings. Our purpose is to keep victim’s proper preservation of the criminal evidences, in addition to find out the incident truth and perform security enhancements.

II. RELATED WORK

This study follows the correct forensic procedures, collects and preserves the original evidences from the victim, analyzes the evidences and then yields the forensic report which is helpful to clarify the truth.

The following terms related network forensics and forensic tools will be discussed.

A. Network Forensics

Network forensics is not a new term or science, is defined by Ranum [2] as “the capture, recording, and analysis of network events in order to discover the source of security attacks or other problem incidents.” Network forensics involves capturing network packets, monitoring and examining network traffic, it is saved by acquisition of digital evidence, and enables investigator have a full understanding on information security incidents and solve the problems.

Garfinkel [3] proposed two approaches to implement network forensics system, the first approach is called “catch it as you can”, because of it needs capturing all network packets, it requires large storage to save. Another is called “stop, look, and listen”, due to it needs monitoring to examine all traffic, it requires good performance processor to process.

Currently academic or educational institutions have not yet developed their own dedicated network forensics system. In present, they still use commercial forensic software such as: EnCase, FTK, etc. The researches of their network forensics systems or tools, most of the features only emphasize a part of implementation, or use with particular forensic software tools. These research may be classified into several categories [4-8], see Tab. 1.

B. Digital Evidence

Digital evidence is defined by [9] as “Information of probative value that is either stored or transmitted in a binary form.” It is not traditional physical evidence, and has some properties: (1) easy to copy and modify, (2) not easy to confirm the source and its integrity, and (3) humans can not directly perceive and understand the content [10].

It is very important that forensic investigator should guarantee the integrity of digital evidence, and prove that the acquisition of digital evidence from the original exhibit. Therefore, investigators obtain digital evidence should abide by the three forensic methods and principles: (1) the collection of evidence must be proved deriving from the original, (2) do not change or damage obtaining original evidence, and (3) do not change or destroy evidence cases when we conducted an analysis [11].

As for evidence transmission through the network produced has some time’s characteristics. It is necessary to understand the importance of evidence gathered from the
various timing and restrictions. To make in accordance with its time-point regards as the implementation of network forensics work necessary guidelines.

Table 1. Network Forensics Research Issue

<table>
<thead>
<tr>
<th>Category</th>
<th>Research Domain</th>
<th>Forensic Tools</th>
<th>Information Gathering</th>
<th>Forensic Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corey et al. 2002</td>
<td>Event and Log Analysis</td>
<td>Intrusion Detection System or Firewall</td>
<td>Network Traffic, System &amp; Event Logs</td>
<td>Network Forensics</td>
</tr>
<tr>
<td>Nikkel 2006</td>
<td>Network Packets Analysis</td>
<td>Hardware Device (PNFEC)</td>
<td>Network Packets</td>
<td>Network Forensics</td>
</tr>
<tr>
<td>Yasinse &amp; Manzano 2003</td>
<td>Honeytraps Analysis</td>
<td>Honeytraps or Honeynets</td>
<td>Network Attacks Evidence</td>
<td>Network Forensics</td>
</tr>
<tr>
<td>Meghanathan et al. 2009</td>
<td>The Combination of Forensic Tools</td>
<td>Web, Email, Network Packets &amp; IP tracer</td>
<td>Web, Email History, Network Packets</td>
<td>Network Forensics</td>
</tr>
<tr>
<td>Cohen 2008</td>
<td>Integrated Forensic Tool</td>
<td>PyFlag Forensic Tool</td>
<td>Network Packets, Disk image Logs</td>
<td>Network, Disk, and Memory Forensics</td>
</tr>
</tbody>
</table>

1) Volatile Information: Such information will disappear after the computer shuts down, the digital evidence, of course, will be destroyed which is being transmitted. This information includes system time, the current network connection status, and transmission of communication packets, running processes, opened files, users logged on, etc. Because they are such important information for cyber crime clues and evidences, according to Pogue et al. [12] view, the implementation of the forensic work need to pay attention to the evidence of a valid time.

2) Non-volatile Information: Such information will not disappear when the computer shuts down. These were usually stored with electromagnetic records on computer disk devices and network communication equipments. This information include system files, document files, user log files, system logs, service logs, audit logs, browser history, social communication records, etc.

C. Procedure of Network Forensics

Due to network forensics focuses on finding out the truth of illegal or criminal activities from evidences, therefore we need to explore digital forensics process and regard it as the procedure of network forensics basis.

According to the digital forensics process, the first Digital Forensics Research Workshop [13] produced seven steps of process are: (1) Identification, (2) Preservation, (3) Collection, (4) Examination, (5) Analysis, (6) Presentation, and (7) Decision. Reith et al. [14] proposed a framework includes nine components are: (1) Identification, (2) Preparation, (3) Approach strategy, (4) Preservation, (5) Collection, (6) Examination, (7) Analysis, (8) Presentation, and (9) Returning evidence. Kent et al. [15] proposed forensics process consists of four phases are: (1) Collection, (2) Examination, (3) Analysis, (4) Reporting. From the views of scholars and experts above mentioned, we summarize the process should be fully prepared before forensic work over the original digital evidence acquisition, analysis of the results should take appropriate protective measures, forensic history is necessary to record the steps, and the final forensic report must be output to show findings of network incident with cybercrime relationship. Despite scholars proposed forensic process and stages may be a bit different, but overall, the implementation of the action and should pay attention to forensic issues is similar. In this study, we use Kent’s proposition for our network forensic process.

D. Network Packets Capture

The ongoing network packets with other hosts’ communication on the network may keep proof of cybercrime. In order to capture this important system information which is fade off easily, we can use the tool to capture network packets to obtain volatile evidence.

This study uses the packet capture software Tcpdump [16], is free software based on BSD license, the source is from the Van Jacobson et al. at Lawrence Berkeley Laboratory Network Research Group services written. Tcpdump is executed in command mode, and Linux system must use Libpcap library to extract network packets. After it analyzes the output format, it will help us to examine and understand the packet header and content. This system uses this capture tool to implement acquiring the important real-time network packets evidence of a compromised host.

E. Live DVD/USB

When the victim has been unable to respond or has been shut down, to not change any status of the host, we use Live DVD/USB system to boot up. The system contains acquisition of victim's computer disk image evidence and then proceeds to the next phase of network forensics process.

Live DVD/USB technology is suppressed into the bootable USB stick or DVD with specified operating system and the necessary kit. It does not need pre-installing on the computer's hard drive, and we can use it to boot up the operating system, especially it will not undermine the existing operating system and data installed on disk. It is particularly suitable for the usage of network forensics system. Currently, there are many Linux distribution vendors have introduced Live DVD/USB system, the known such as: Knoppix, Fedora, Ubuntu, Debian, etc. In this study, we conduct Debian Live [17] system which uses Live-helper and Live-magic tools to customize our own system and necessary tools suite.

F. PyFlag

PyFlag [18] was originally designed by the Australian Department of Defense, followed by release of GPL license. Originally designed as a database-driven digital forensics analysis tool, and then developed into involving network forensics capabilities. PyFlag innovative approach is to
integrate all the forensic perspectives into the same case, to allow investigators to use disk technology such as keyword indexing and hash processing network data generated [8].

PyFlag available features include loading many different logs file format capabilities, implementation of disk image and memory forensic analysis, and network packets forensics by using Tcpdump tool. Features briefly are as follows:

1) Network packets forensics: Analyzes Tcpdump network packets captured, and supports multiple network protocols such as: HTTP, SMTP, POP, MSN, Yahoo Messenger, etc.

2) Log analysis: Supports many logs format, such as: Apache log, IIS log, custom format log, etc.

3) Disk/File forensics: Supports for hard disk image and file formats such as: Zip, PCAP, OLE2, Reg File, IE Cache, etc.

4) Memory forensics: Has a basic function for Windows system only. Since there is a problem in this functionality on PyFlag current version, we need to apply other mechanism to fix this deficiency.

Although PyFlag forensic tool had these advantages, it did not implement Chinese environment, lacked of volatile information integration and presentation, did not describe the usage of the evidence, and the result of forensic report was simple. Hence, we need to enhance PyFlag software functionalities to archive our requirements.

III. SYSTEM DESIGN

A. System Design

In this study, we use Debian Live Linux as a system development platform, enhance PyFlag network forensics tool, implement the evidence presentation of volatile information and forensic report, and integrate tools of system and network. This system can be booted up from the live network forensics system and running. We can also mount the necessary tools on a real operational system to archive the assigned work. The flow chart of network forensic system is shown in Fig. 1.

B. System Procedures

Network forensics system process can be divided into two stages of operation, as described as follows:

1) Phase I - Preserve evidence work: First, when the victim host system can be started and operated, we must mount external system and network tools to obtain important evidences as soon as possible, such as volatile information and current network packets of transmission. If the victim system does not start properly or shut down, then booting up this live system to perform disk image making, and then shut down the victim host. In this stage, the data collected will be used for further analysis of the second stage.

2) Phase II - Analyze forensic work: Use another safe and reliable computer and launch this system. Importing data collected from the victim, including the important volatile information, network packets data, disk image files, web services, log files, system log files, etc., finishes filtering and analyzing suspicious records to identify incident of network intrusion or attack, and further yields network forensic report.

IV. SYSTEM IMPLEMENTATION

In this study, our main objective focuses on developing a Chinese network forensics system, it is working from Live DVD/USB, allows to perform network forensics work due to Linux platform victim suffer from network intrusion. In addition, this system consists of self-developed graphical launch menu, and compiles the required tools, such as system, network, and data acquisition. It implements open source software PyFlag volatile information integration and forensics report presentation, uses the MySQL database back-end to accelerate the search speed of the evidence, facilitates investigators smooth execution of the network forensics work.

A. Graphical Launch Menu

In order to facilitate forensic investigators to operate this system, this study uses PyGTK language implementing graphical launch menu. The implementation of this menu indicates how to preserve relational important evidences, and these critical evidences will save on the external mounted drive or disk to avoid false data writing and damage suffered its original state. Graphical launch menu is shown in Fig. 2.

B. Volatile Information Acquisition

The victim volatile information, such as: the running processes, the status of network connections, users logged on, opened files, network transmission packets, etc., is important information may be sufficient to prove implied an incident of cybercrime or network intrusion. The evidence will disappear when the system shutdown. This operation must take advantage of the system still come out to capture as soon as possible, store on disk or USB additional space to save this critical volatile evidence.
Figure 2. Graphical launch menu

To collect various types of volatile information, the proper mounted external program will be executed. Use the `ps` program to obtain system running process, the `netstat` program to obtain the host network connection status, the `w` program has made a user login information, the `ls` program to capture network transmission packets, here only demonstrates `tcpdump` result shown in Fig. 3.

C. Verification Code Mechanism (SHA256)

In order to acquire the victim evidences completely with creditability, important and volatile information were used for SHA256 one-way hashing verification code on a vulnerable system to ensure the acquisition of evidence from the original is integrity without alteration. Fig. 4 shows the result of files hashing.

D. Create Disk Image

In the forensics process work, to save the victim host disk data completely, and avoid human operation error resulting in destruction of important information, we must use the live system startup and launch the AIR (Automated Image & Restore) [19] tool to obtain the disk image. To assure that original volume contents of the vulnerable system is exact, the system on disk drive will be imaged by computing SHA256 verification code, used to examine the integrity of original evidences. Create disk image using the AIR tool is shown in Fig. 5.

E. Enhanced PyFlag Forensic Software

To support PyFlag forensic software for investigators with local needs and convenient operation, this system includes Traditional Chinese operating environment, solves the original non-English language interface problems, implements volatile evidence integration, enhances forensics report presentation, and makes investigators work effectively in network forensics. Enhanced PyFlag forensic software is required to start on the browser, in URL field enter "http://localhost:8000/", shown in Fig. 6.

F. Comparison with Existing Forensics Systems

This system helps investigators to archive most of network forensics operation. In order to present advantages of our system, we compare famous forensic systems with functionalities, shown in Tab. 2.

G. Network Forensics Lab

In this study, we give a case example: a network security incident for network forensics lab, the forensic operation described as follows:
Table 2. Comparison with existing forensics systems

<table>
<thead>
<tr>
<th>Functionalities</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EnCase</td>
</tr>
<tr>
<td>Chinese Environment</td>
<td>✓</td>
</tr>
<tr>
<td>Retrieve Volatile Evidence</td>
<td>✓</td>
</tr>
<tr>
<td>Create Disk Image</td>
<td>✓</td>
</tr>
<tr>
<td>Image Verification</td>
<td>✓</td>
</tr>
<tr>
<td>File System Forensics</td>
<td>FAT 16/32</td>
</tr>
<tr>
<td></td>
<td>NTFS</td>
</tr>
<tr>
<td></td>
<td>EXT 2/3</td>
</tr>
<tr>
<td>Deleted Files Recovery</td>
<td>✓</td>
</tr>
<tr>
<td>Keyword Search</td>
<td>✓</td>
</tr>
<tr>
<td>Email Search</td>
<td>✓</td>
</tr>
<tr>
<td>View Browser Records</td>
<td>✓</td>
</tr>
<tr>
<td>Network Packets Analysis</td>
<td>✓</td>
</tr>
<tr>
<td>Log Analysis</td>
<td>✓</td>
</tr>
<tr>
<td>Password Cracker Tools</td>
<td>✓</td>
</tr>
<tr>
<td>Forensic Report</td>
<td>✓</td>
</tr>
</tbody>
</table>

1) Import different sources of evidence: According to collect digital evidence from the victim host includes: volatile information, disk image, network packets, system and service logs, they are remitted to the same case as further analysis. Here only demonstrate importing volatile information evidence shown in Fig. 7.

2) Data analysis: Forensic investigators can use some filter conditions, such as: keywords or suspicious IP address, or the other time lines, as far as possible filter the problem files, records and information from importing sources.

3) Generate forensic report: The system can automatically compile forensic analysis result completed in the process of filling in the text record, then yield network forensics report. According to the result of forensic case is classified into five parts: (1) Victim hardware information, (2) Volatile evidence information, (3) Disk image evidence information, (4) Log evidence information, (5) Findings report includes: I-node, timeline and IP address. The report is shown in Fig. 8.
forensics techniques and tools for Windows platform into this study. We hope to make functionalities of this system more complete and suitable for investigators’ requirements and use.

REFERENCES