FORENSIC EVIDENCE IDENTIFICATION AND MODELING FOR ATTACKS AGAINST A SIMULATED ONLINE BUSINESS INFORMATION SYSTEM

Manghui Tu
Department of Computer Information Technology and Graphics
Purdue University
Calumet, IN

Dianxiang Xu
College of Business and Information Systems
Dakota State University

Eugene Butler
Department of Computer Science
University of Minnesota at Morris

Amanda Schwartz
College of Business and Information Systems
Dakota State University

ABSTRACT

Forensic readiness of business information systems can support future forensics investigation or auditing on external/internal attacks, internal sabotage and espionage, and business fraud. To establish forensics readiness, it is essential for an organization to identify which fingerprints are relevant and where they can be located, to determine whether they are logged in a forensically sound way and whether all the needed fingerprints are available to reconstruct the events successfully. Also, a fingerprint identification and locating mechanism should be provided to guide potential forensics investigation in the future. Furthermore, mechanisms should be established to automate the security incident tracking and reconstruction processes. In this research, external and internal attacks are first modeled as augmented attack trees based on the vulnerabilities of business information systems. Then, modeled attacks are conducted against a honeynet that simulates an online business information system, and a forensic investigation follows each attack. Finally, an evidence tree, which is expected to provide the necessary contextual information to automate the attack tracking and reconstruction process in the future, is built for each attack based on fingerprints identified and located within the system.

1. INTRODUCTION

With continuing advances in internet technology, information systems have played more and more important roles in moving businesses toward online practices (De Aalst, Van Hee, Van De Werf, Kumar, & Verdonk, 2009; Romney & Steinbart, 2008). Online business offers convenience and flexibility to customers, employees, and partners. With lower costs than traditional methods, this method provides a highly profitable channel for businesses (Romney & Steinbart, 2008). However, due to the untrustworthy nature of the internet environment and the sophisticated business processes involved, online businesses also face severe security challenges. Over the past few years, millions of sensitive data records have been compromised (Ramzan, 2008; RSA Security, 2008) and a large number of frauds have been committed (Gu, Liang, & Wang, 2005; Larson, 2008; Lendez & Korevec, 1999; Singleton, Singleton, Bologna, & Lindquist, 2006). For a business, these security breaches not only result in substantial financial and operational losses, but also greatly hurt the confidence of customers, business partners and stakeholders (Hoffman, 2007; Seltxer, 2006). It is evident that cyber crime and fraudulent activity against online businesses will continue to thrive (Ramzan, 2008; Robb, 2008; RSA Security, 2008; Zhang & Guan, 2008). Meanwhile, over the last decade, government and industry bodies around the world have issued many laws and regulations to ensure the availability, integrity, and confidentiality of business data and the IT infrastructures. These mandates place a lot of pressure on businesses and organizations to implement programs to ensure compliance with laws and regulations. Therefore, securing data and IT infrastructures is critical to online business and should be addressed appropriately.

Many intrusion/fraud prevention, detection, and tolerance mechanisms have been deployed by organizations and companies doing online business in order to secure their IT infrastructures and the sensitive data stored in information systems (Fratto, 2008; RSA Security, 2008; Williamson, 2006). However, the number of data breach incidents has still risen over the past few years (CENZIC, 2008; RSA Security, 2008). It is evident that even with the state-of-the-art security prevention, detection, and tolerance mechanisms, the risks to online business cannot be completely excluded. Consequently, intrusion/fraud deterrence, such as digital forensics investigation, has been recognized as a complement to traditional security protection techniques and provides another dimension of protection for the critical infrastructures of these vulnerable businesses (Endicott-Popovsky & Frincke, 2004; Siponen & Oinas-Kukkonen, 2007; Straub, 1990; Valentine, 2007).

Digital forensics is the process of investigating computer devices and associated storage media to determine whether they have been used to commit a crime and/or gain unauthorized access (Casey, 2011; Tan, 2001). Digital forensics involves the process of preservation, acquisition, analysis, discovery, documentation, and presentation of evidence (Casey, 2011). The success of digital forensics is highly

dependent on forensics readiness (Espiner, 2008; Endicott-Popovsky, Frincke, & Taylor, 2007; Tan, 2001; Valentine, 2007), e.g., the availability of forensically-sound evidence that is able to stand up to legal scrutiny and that can be investigated in an efficient and effective way (Endicott-Popovsky et al., 2007; Tan, 2001). Forensic readiness is an increasingly important topic in forensic investigation and information assurance research (Carrier & Spafford, 2003, 2004; Endicott-Popovsky et al., 2007; Rowlinson, 2004; Tan, 2001; Tang & Daniels, 2005; Wilson & Wolfe, 2003; Yasinsac & Manzano, 2001). Existing research efforts focus on the organization-level framework design for forensics readiness, such as policy design, implementation, and management. However, they did not address the investigation of security incidents in information systems (Poolsapassit & Ray, 2007).

The overall goal of this research is to provide technical guidance to effectively and efficiently investigate security incidents that take place in online business information systems. However, there are a few challenges that need to be addressed. First, the fingerprints left by attacks in information systems remain unclear to digital forensics and security professionals, and the fingerprints that are needed to reconstruct the corresponding attack incidents should be determined (Poolsapassit & Ray, 2007). Second, many attacks and frauds remain undetected due to the lack of sophisticated detection mechanisms (Espiner, 2008; Endicott-Popovsky et al., 2007; Valentine, 2007). Third, many forensics investigations are not conducted due to the cost of identifying, locating, and processing the vast amount of the information in the system (Jeyaraman & Atallah, 2006; Endicott-Popovsky et al., 2007; Tan, 2001; Valentine, 2007). This research effort addresses the first challenge and provides foundations to address the other two challenges in digital forensics investigation. A systematic approach will be developed to identify and locate the fingerprints that are needed to reconstruct the attacks studied. An evidence model will be developed based on the indentified fingerprints for each attack. Evidence models can be used to guide forensics investigation in the future and to provide the contextual information that is needed for the automation of security incident tracking and investigation.

The remainder of the paper is organized in the following manner: System modeling and Methodology are described in Section 2. Attack generation and evidence acquisition processes are presented in Section 3. Results analysis and evidence tree building process are presented in Section 4. Section 5 discusses how to utilize the fingerprints located to identify and reconstruct attacks. Section 6 gives a brief literature review and Section 7 states the conclusion of the paper.

2. SYSTEM MODELING AND METHODOLOGY

The overview of the research methodology is shown in Figure 1. Throughout this research, threats and attacks will be modeled as augmented attack trees for online business information systems (Mauw & Oostdijk, 2005; Poolsapassit & Ray, 2007; Saini, Duan, & Paruchuri, 2008; Schneier, 1999). Attacks are then conducted

against an online business information system that is simulated by a honeynet. Forensics investigation will be conducted following each attack and fingerprints are identified, located, and manually reconstructed to determine whether the attack itself can be reconstructed. If the attack or fraud cannot be reconstructed successfully, the attacking and forensics investigation process will be repeated with enhanced evidence logging. If the attack or fraud is reconstructed successfully, the fingerprints of each attack operation will be identified. The metadata of the fingerprints of each attack operation, such as log name, format, location, timestamps, and security features, etc. are composed into nodes, which become child nodes of the leaf nodes in the augmented attack tree. This entire process will finally result in the creation of an evidence tree for each attack studied.

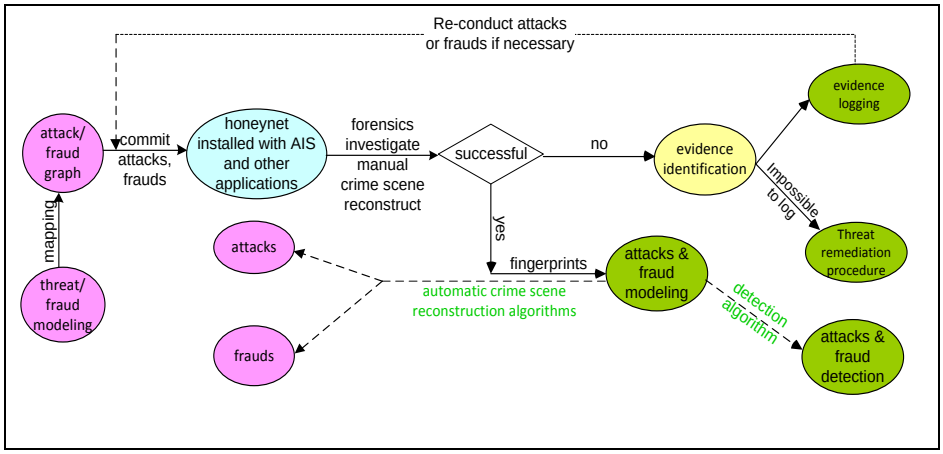

Figure 1 The overview of the research process on forensics readiness

2.1 Threat Modeling and Attack Generation

The attack tree approach that is first proposed by Schneier (1999) is used to systematically analyze security threats. Attacks are modeled and represented by a tree structure where the root node represents the final goal, other interior nodes represent subgoals, and leaf nodes are attacking approaches to achieve the final goal (Poolsapassit & Ray, 2007). Children of a node in the tree can be one of the two logical types: *AND* and *OR*. To reach the goal, all of its *AND* children, or at least one of its *OR* children, must be accomplished. Attack trees grow incrementally by time and they capture knowledge in a reusable form. First, possible attack goals must be identified. Each attack goal becomes the root of its own attack tree. Construction continues by considering all possible attacks against the given goal. These attacks form the *AND* and *OR* children of the goal. Next, each of these attacks becomes a goal and their children are generated. Figure 2 shows an example of an attack tree of the inside threat, "*achieving the root privilege*". In such an attack, the attacker is a regular user and has a lower access privilege to the target (which needs root privilege), and conducts a series of attacking operations to achieve the root privilege as the system user. Note that links that are connected with a line represents

the "AND" relationship among the states or sub-goals, which are working together to achieve the same parent goal.

External threats are modeled using attack trees and attacks are then further modeled as augmented attack trees (Poolsapassit & Ray, 2007). An augmented attack tree is built from the attack tree by including the attack operations as child nodes to the leaf nodes of the original attack tree. To ensure the coverage of **external threats**, a two dimension table (shown in Table 1) is used to enumerate all potential threats. A row of the table represents a vulnerability of the honeynet identified at the previous step, and a column of the table represents a type of external threat classified using the Microsoft STRIDE model (Swiderski & Snyder, 2004), i.e., denial of service, repudiation, information disclosure, spoofing, tampering, and elevation of privilege.

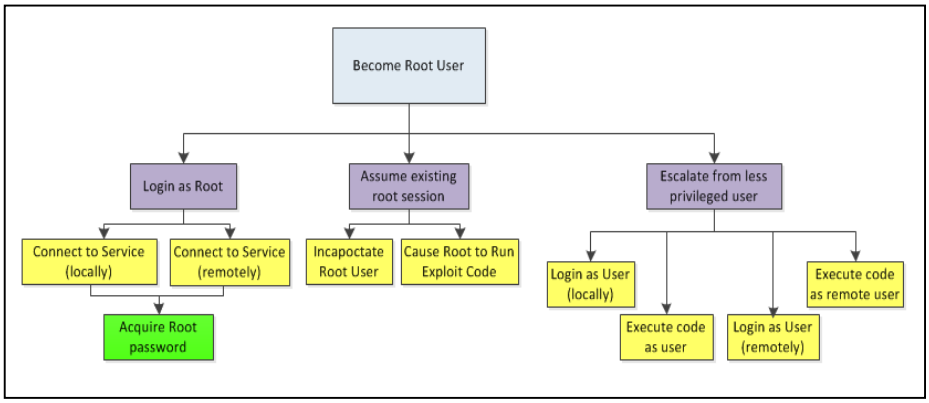

Figure 2 An attack tree of an internal threat "*achieving the root privilege*"

Table 1 The enumeration table for external threats of the honeynet

system vulnerabilities	spoofing	tampering	DOS	repudiation	information disclosure	Privilege escalation
IIS					X	X
ftp			X			
…						

Internal threats include espionage, sabotage, and privilege or resource abuse. Insiders usually have a pre-defined goal and target (Cappelli & Trzeciak, 2008); for example, accessing or copying sensitive information, destructing critical services, degrading the security configuration of the system. To reach the targets, insiders need to have known or unknown paths (Cappelli & Trzeciak, 2008), such as appropriate access privilege, privilege escalation to achieve appropriate access, or exploiting vulnerabilities to crash critical services. To conduct internal attacks, insiders may or may not need to access the target, they may or may not have the appropriate access privilege in advance, or they may or may not need to exploit

system vulnerabilities. Therefore, the threat modeling techniques proposed for external threats might not be sufficient for internal threat modeling. In this research, internal threats are first enumerated based on both attack targets and access paths as shown in Table 2. Two categories of internal threats are considered here. **Case One**, the insider conducts an attack to destroy valuable assets or escalate the privilege to access sensitive assets; and **Case Two**, the insider accesses sensitive assets with desired privilege for industry espionage purposes. In this paper, **Case One** internal threats and attacks will be systematically analyzed and modeled using attack trees and augmented attack trees, similar to the external threats and attack modeling described above. **Case Two** internal threats are first enumerated and modeled through the identification of access paths such as USB, email, and CD ROM, etc., then they are identified through the linking between the access paths and access target. They are then modeled using attack trees, similar to those for external attacks.

Table 2 the enumeration table for internal threats of the honeynet

target	no access privilege	privilege escalation	crash services
email services	X	X	
web services	X	X	
…	X	X	
sensitive assets 1	X	X	
…	X	X	
vulnerabilities 1		X	X
…		X	X

2.2 A Honeynet Simulating an Online Business Information System

The system designed for this research is a third generation Honeynet, as shown in Figure 3. It consists of two major parts: a set of honeypots and a single honeywall controlling the entire honeynet. The honeypots simulate some of the necessary functioning components of the online business, e.g. a web server, a file server, a printing server, an email server, and an open source business information system, such as the CeBuSoft Accounting Information System 1.01 (CeBuSoft, 2013). The honeywall acts like an invisible bridge between the honeypots and the outside world, and it can intercept all traffic between the honeynet and the outside. The honeynet uses a public IP address that is reserved specifically for this research and it becomes part of the DSU campus network and is controlled by the boarder router outside of the DSU firewalls intended to attract external attacks. For the research described in this paper, the attacks and forensics examination will mainly involve two windows honeypots.

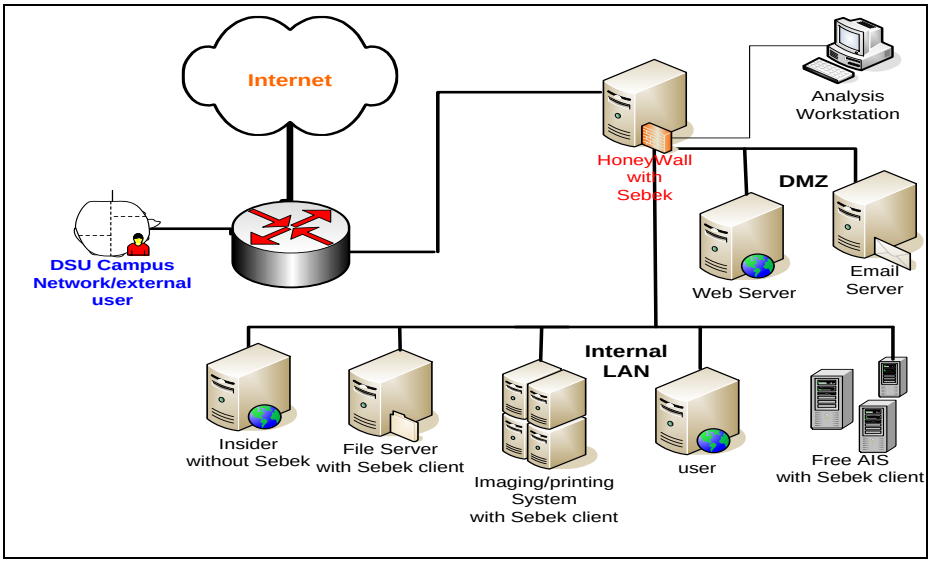

Figure 3 The honeynet that simulates the online business information systems

The honeypots run copies of Microsoft Windows XP with Service Pack 2 x86 without patches. They are, however, installed with the Sebek kernel module downloaded from the Honeynet Project (2013). This module allows activity monitoring on the honeypots, such as console command logging, without compromising the honeypot by providing clues that would tip off a potential attacker. The honeypots are connected to a Linksys eight port switch alongside the honeywall's *eth1* interface. The honeywall computer is connected to a wall port and, from there, a cable modem on *eth0*. To capture and analyze traffic, the honeywall computer is installed with *Roo*, which is a Honeynet Project Linux distribution to provide a set of tools for an administrator to manage the network. One of these tools is the web interface called walleye, which can be used to both change the configuration of the honeywall as well as analyze the data that passes through it. The minimal requirements for honeywall CDROM are: Intel x_86 Pentium class CPU, 512MB Memory, Minimum 10GB HDD and 2 network interface cards (3 if you want to use the remote management). The default configuration of the network interface cards was used. Therefore, eth0 was connected to the wall port which was in turn connected to a cable modem and Eth1 was connected to a Linksys eight port switch which was further connected to the two honeypots. The honeywall transparently logs all communications between its *eth0* and *eth1* interfaces as well as all Sebek traffic from the honeypot computers. Also, it can provide traffic and honeypot information to the honeynet administrator through the Walleye web interface, which allows the administrator to change honeywall settings and to review logged communications. Connections can be sorted by date, originator, recipient, and service. Traffic is also downloadable in the form of PCAP files.

3. THE ATTACKS AND FINGERPRINTS ACQUISITION

We simulate external attacks and internal attacks against online business in the following way: external attacks are executed from a computer outside of the honeynet without knowledge of the system security credentials, such as an account password. Internal attacks are executed on one of the honeypots locally or remotely through a machine that is remotely logged in the honeynet. In this paper, a few attacks will be conducted against the honeynet, including two external attacks, two **Case One** internal attacks, and two **Case Two** internal attacks. Before each attack, the system is restored using the image of the original system. After each attack is conducted, the affected honeypot is powered down and the hard disk is taken out and put into a write blocker, which is connected to a USB port of the laptop computer with a Backtrack 4 live DVD. A forensic sound image is obtained through the use of Backtrack 4 live DVD's *dcfldd* program to make a bit-for-bit copy of the */dev/hda1* device. Note that volatile fingerprints in memory were not collected. Backtrack 4 live DVD is a well known, free ethical hacking tool that can provide great flexibility and well-developed exploits to users, making the attacking jobs much easier.

3.1 ATTACKS GENERATION

Two external denial of service attacks are conducted as described below.

Attack A is a Denial of Service attack targeting the *Filezilla* administrator user interface. The attack is accomplished by sending an excessively long USER command to the FTP Server that runs the Administration Interface (*FileZilla Server Interface.exe*). After the stack is overwritten by the long string, an exception is generated. The attack is launched using Metasploit's exploit "*auxiliary/dos/windows/ftp/filezilla_amin_user*". Once the attack is completed, the victim computer is locked until the administration interface is forced to shut down. The modeled attack tree is described in Figures 4 (a) and (b). Note that a successful attack is composed of only a part of the attack tree.

Attack B is a denial of service targeting the FTP Server's vulnerable PORT Command. The attack is launched using Metasploit's exploit "*auxiliary/dos/windows/ftp/filezilla_server_port*". This attack works by sending a malformed "*PORT*" command combined with a "*LIST*" command. To execute this command, the server attempts to write to a NULL pointer, which will generate an exception. Once this attack is successfully executed, no client can connect to the server.

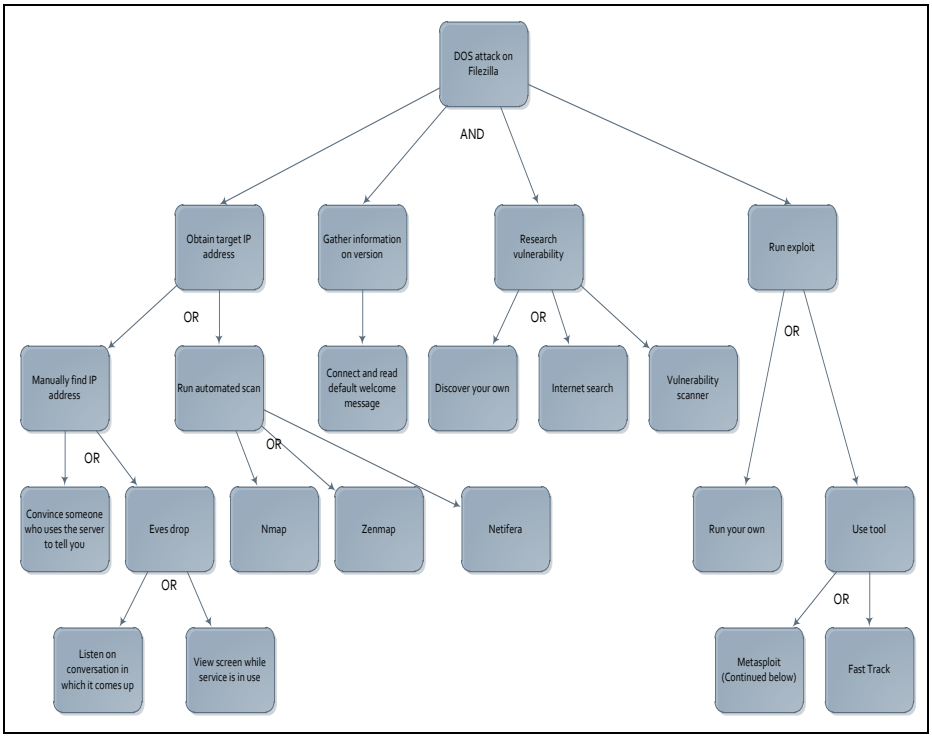

Figure 4(a) The left part of the attack tree specific to Attack A and Attack B

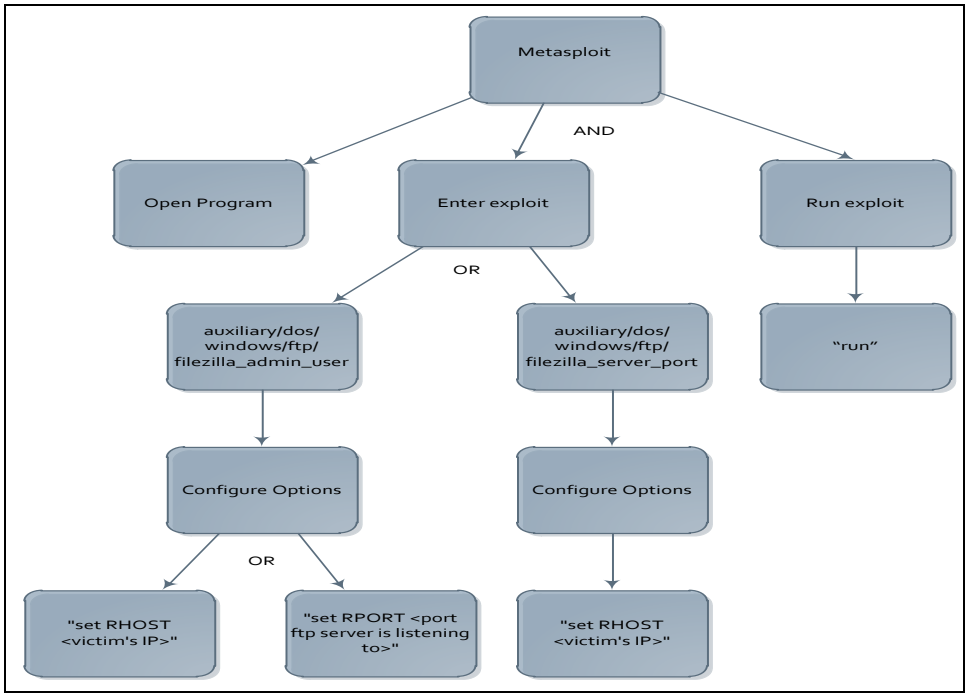

Figure 4 (b) The right part of the attack tree specific to Attack A and Attack B

Two Case One internal attacks are conducted as described below.

Attack C involves manual privilege escalation using the Windows' command line and Windows Task Scheduler's scheduling service. The attack can be accomplished by login as a regular user on the system. In this research, the default user account created during the Windows installation process is actually chosen here. Then, the *cmd.exe* process is lunched from the run box. Once the Command Prompt is active, the command "*at XX:YY /interactive cmd.exe*" is typed (*XX:YY* defines the time when the *cmd.exe* process should be launched, and it is calculated by taking the current time plus the specified length of the time period in minutes). After such period of time passed, the Windows Task Scheduler creates an instance of *cmd.exe* process in interactive mode, causing a new command prompt window to appear on the screen. This command prompt window runs as the Local System User and has the title "*C:\WINDOWS\svchost.exe*". Then, having access to a command prompt running with higher permissions, the *explorer.exe* process will be shut down. Once the *explorer.exe* process is successfully shut down, the command "*cd ..*" is typed into the new system-level command prompt, followed by the command "*start explorer.exe*". This will launch the *explorer.exe* process with the privileges of a system user. The attack operations are shown by the internal nodes of the evidence tree presented in Figure 7.

Attack D is accomplished in a more sophisticated manner. The attacking machine is a Dell Latitude D810 laptop computer, running Ubuntu 9.10, Karmic Koala,

connected to a wireless network that is part of the honeynet. The attack is conducted using Metasploit's (version 3.4.2-dev [core:3.4 API:1.0] exploit "*windows/browser/ms10_002_aurora*", a server-based Internet Explorer memory corruption attack, and the payload "*windows/meterpreter/reverse_tcp*", a reflective injection attack that runs the *meterpreter* service on the target machine. Once the exploit is running, the attacker simply needs to navigate to the malicious page, the browser freezes, and a successful intrusion is accomplished. The next step involves connecting and gaining system user access from the external computer. Once the *meterpreter* service is connected to the malicious computer from the target computer, its session could be opened with the command "*sessions -i* 1". The actual privilege escalation is achieved by using the priv, which is a "privilege" *meterpreter* extension. One simply needs to load it with "*use priv*", and then use a named pipe impersonation attack by "*getsystem -t* 1". After this is accomplished, the server process is running with system user permissions and the privilege escalation is successfully accomplished. The attack operations are shown by the internal nodes of the evidence tree presented in Figure 8.

Two Case Two internal attacks are conducted as described below, both of which utilize removable media (USB drive and CD-ROM) as the access path to steal sensitive business information.

Case Two Internal Attack E is a typical industrial espionage inside attack**.** In such an attack, the attacker has all the needed privileges to access sensitive data and to access the USB ports which are required to perform the user's duty. However, those sensitive data should not be copied to personal USB devices since this may result in potential information leakage. To perform such an attack, the user is logged into system with all needed privileges, navigate to sensitive data, copy and paste the sensitive data into the USB device. The USB device is then removed and the user is logged out of the system later. The attack is shown in Figure 5.

Case Two Internal Attack F is a typical industrial espionage inside attack, similar to Attack E described above**.** In such an attack, the attacker has all the needed privileges to access sensitive data and to access the CD-ROM Drive, which is needed to perform the user's duty. However, those sensitive data should not be copied to CD-ROM since this may result in potential information leakage. To perform such an attack, the user can log into system with all needed privileges and navigate to sensitive data, and then burns the sensitive data onto a CD-ROM. The CD-ROM is then removed and the user is logged out of the system. The attack is shown in Figure 6.

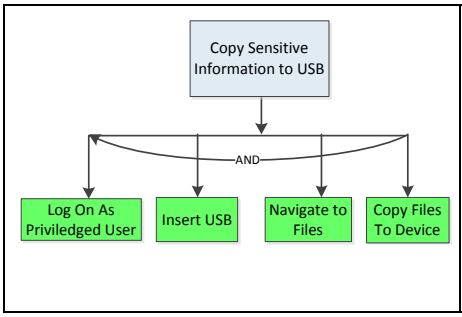

Figure 5 Case Two Internal Attack E (USB copy)

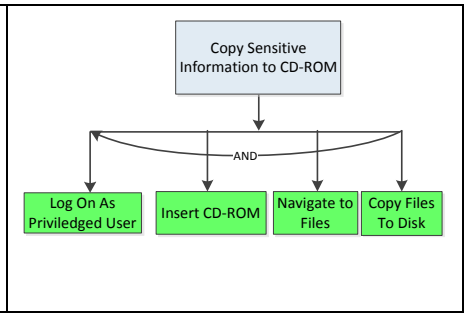

Figure 6 Case Two Internal Attack F (CD-Rom copy attack)

3.2 FINGERPRINT ACQUISITION

Fingerprints are retrieved from three sources in the honeynet. The first source is the hard disk of the compromised honeypots. After each attack was successfully conducted, a bit-by-bit copy of the compromised honeypot hard disk is created using the *dcfldd* program from the Backtrack toolkit. Then, AccessData's Forensic Toolkit FTK is used to process the image of hard disk. The *disk index* feature has the ability to search every fingerprint item whether it is recognized by the file system or not (Note that other forensic fingerprint processing tools can also provide a similar index feature). This index feature can greatly improve the efficiency of searching for fingerprints relevant to the attack. For example, for Attack D, FTK can search the disk image for all items with the IP address of the hostile server in a few seconds since all related fingerprints are indexed. For Attack C, FTK can search the disk image for all items with particular console command used. The second source of fingerprints is the logs maintained in the system such as the event log, the security event log, and Internet Explorer history entries, which can either be searched by using FTK or searched manually without using any tools since these types of fingerprints are easily readable within Windows. Once relevant events and history information are discovered, they are recorded into the augment attack tree to reconstruct the evidence tree for such modeled attack. Note that a piece of fingerprint is said to be *relevant* if, and only if, it is the fingerprint left by an operation of the attack studied. In the studied system, there are many other processes running, each of which will have different operations on the system. Therefore, they will leave significant amounts of fingerprints in the system. Since these processes are not part of the attack studied, the fingerprints they left are not relevant to the attack. Note that the second source of fingerprints can also be obtained from the forensics disk image. The third source of fingerprint is the honeywall's records of incoming and outgoing connections. Note that not all fingerprints retrieved from honeywall are visible in a regular network without proper configuration. Using the Walleye web interface from the computer acting as the honeywall's management interface, it is easy to isolate the connections made

within the attack's time frame and to review each connection to determine its relevance to the attack. Packets captured from each communication are made available in PCAP format by the web interface, and are downloaded and reviewed to ensure that the inferences made about the content of those communications are factual (not hidden, disguised, or modified).

4. RESULTS AND ANALYSIS

The results of the two external DoS attacks (A and B) can be found in Table. 3. This table is composed of the fingerprints that are relevant to each of the two attacks and can be found in some important logs in the honeynet. The content of the log files can be either searched manually or searched by using FTK.

External Attack A targeting the administration interface (described in Figure 4 (a) and (b) and section 3.1) yields rich fingerprints, most of which can be found in the *Filezilla* log file at the application level. This particular attack works by sending four thousand user requests such that the length of a succeeding request is longer than the previous request. This makes the log file very difficult to read. However, it does indicate the IP address where the requests originated as well as the time of the request. In a denial of service attack, the attacker's IP address is often the most valuable piece of fingerprint identified. This is because that once the attacker's IP address is identified, a person can block that particular IP address (or a block of IP addresses) from connecting to the target server. The event log also contains a security event with timestamps which records the time when the administrative interface crashed. The firewall log includes a record of the attacker's IP address and the timestamps of the attack. The firewall log has a larger file size than other logs, which makes it more difficult to locate fingerprints. The honeywall log contains similar information to the firewall log, but is much easier to read and locate.

Table 3 fingerprints of the two external attacks in part of the important logs within the honeynet

Types and Locations of Fingerprint				
	Event log	Firewall log	Filezilla log	Walleye
Attacks	A	A, B	A, B	A , B
Time of the Attack logged	A	A , B	A, B	A, B
Logged IP of the attacker		A , B	A, B	A, B
Protected from Tampering				A, B

Attack B (described in Figure 4 (a), Figure. 4 (b), and section 3.1) is similar to Attack A, but it is much more difficult to identify the fingerprints of Attack B than

those of Attack A, since Attack B connects to the *Filezilla* sever only once per execution. The *Filezilla* log is able to log the connection causing the DoS attack, but there are too many connections to the *Filezilla* server from other legitimate users. Thus, it becomes very difficult to locate and identify the connection made by Attack B. Other relevant fingerprints can be found in similar locations as those of Attack A, except that the event log does not log any fingerprint for Attack B. The firewall and the honeywall can successfully log the timestamps and IP address of the attacking computer of Attack B.

The results of the two **Case One internal attacks** (Attack C and Attack D) can be found in Figures 7 and 8, each represents an evidence tree built based on the corresponding augmented attack tree. The goal of both attacks is to exploit vulnerabilities of the system in order to escalate from a regular user to the one with system user permissions.

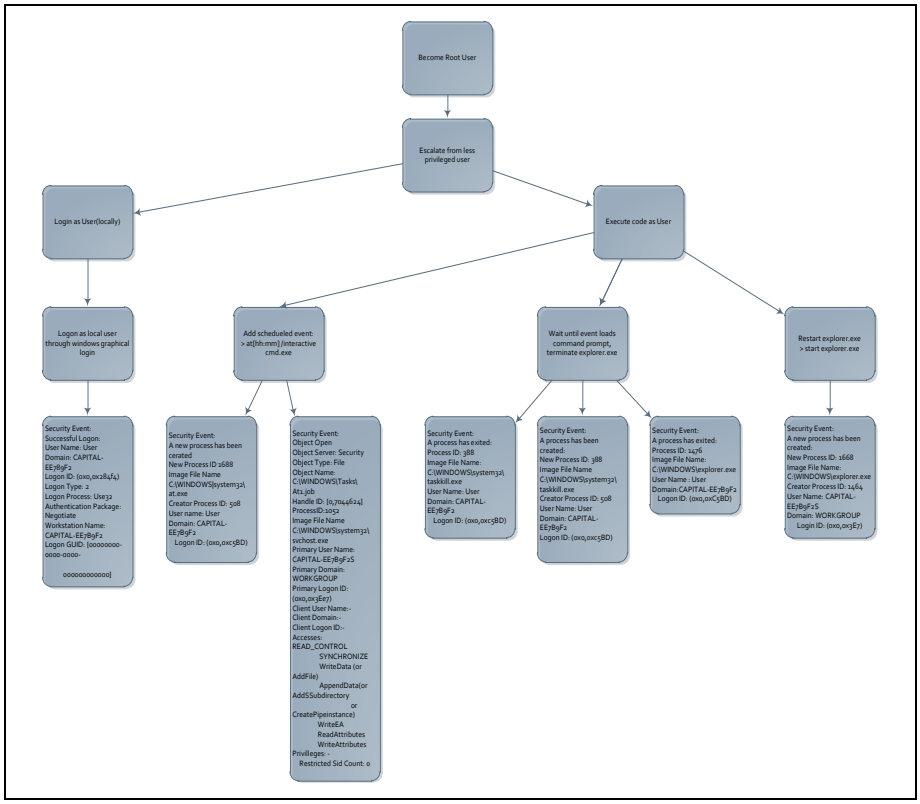

Figure 7 A part of the evidence tree for Case One Internal Attack C

Internal Attack C begins with a simple operation of logon as a regular user on the honeypot (workstation *CAPITAL-EE7B9F2*). In this case, the user was named as *User*. This operation generates a security event of type "Successful Logon" that is logged by the system's event log. The second operation generates a scheduled event

(to open an interactive command prompt) and two other events, e.g., the launch of a new instance of the process "*C:\WINDOWS\system32\at.exe*", and the open of the file "*C:\WINDOWS\Tasks\At1.job*", which is the job file that the scheduler would execute at the time when the execution command is entered (for example, to launch a *cmd.exe* process). The third operation is to terminate the process, "*explorer.exe*", which generates three events to be logged by the security event log. The fingerprints of the three events can track the creation of an instance of the process "*C:\WINDOWS\system32\taskkill.exe*," the termination of the *explorer.exe* process, and the termination of the Windows task manager. The final step is the restart of the *explorer.exe* process in a command prompt that was scheduled to be launched in a previous step. This operation creates another event logged by the security event log. The process, "*C:\Windows\explorer.exe*", is created not by the user *User* but actually by the system user *CAPITAL-EE7B9F2$*. Note that the system user *CAPITAL-EE7B9F2$* is also the user that is responsible for the *Object Open* action on *At1.job* and the launch of the *cmd.exe* process. The scheduling event itself (the one created by the operation that scheduled the task), however, is the property of *User* who is just a regular user. Thus, a line could be drawn between the actions of *User* and the subsequent actions of the system user *CAPITAL-EE7B9F2$*. The fingerprints left by all of these operations are used to build the leaf nodes of the evidence tree for this attack (shown in Figure 7), which can be useful in locating relevant fingerprints as well as to automate the tracking and reconstruction of Attack C.

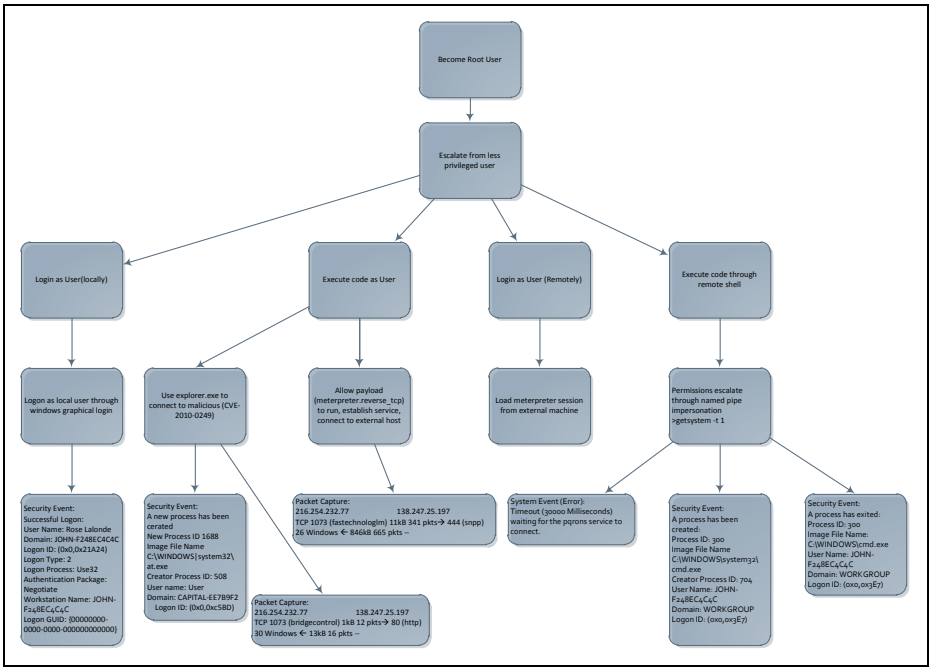

Figure 8 A part of the evidence tree for Case One Internal Attack D

Attack D is executed in much the same way as Attack C, with the login of the regular user *Rose Lalonde* on a different honeypot (workstation *JOHN-F245EC4C4C* with an IP address of *2xx.xxx.xxx.x7*). After login, the attacker attempts to visit the malicious site "http://1xx.xxx.xx.xx7/exploit". This operation is recorded in Internet Explorer's history and can be found both in "*C:\Documents and Settings\Rose Lalonde\Local Settings\Monday\1xx.xxx.xx.xx7*" and in the *index.dat* file. This operation is also captured by honeywall's transparent bridge. During this communication, the honeypot machine sends 12 packets to the malicious site, and receives 16 packets from the malicious site. After the payload has been executed, the attacker reconnects to the malicious machine on port 444, on which the *meterpreter* process is running. Once the remote intrusion has been completed, privilege escalation is attempted. Fingerprints of the attack come in the form of a system event of type "*Error*," warning of a timeout of 30 seconds while "waiting for the *pqrons* service to connect." Subsequently, when the attacker opens a remote shell, a new instance of the command prompt, "*C:\Windows\system32\cmd.exe*", is created by user *JOHN-F245EC4C4C$*, which is the system user of the honeypot workstation. The fingerprints of this attack are logged by firewall logs and internet history, linking the user *Rose Lalonde* to the actions generated by *JOHN-F245EC4C4C$*. This is because the successful completion of this attack has to go through the communications between an external server and a local honeypot machine within the honeynet. The fingerprints left by all of these operations are used to build the evidence tree (shown in Figure 8), which can be useful in locating relevant fingerprint as well as to automate the tracking and reconstruction of Attack D.

The results of Attack E and Attack F are shown in Figures 9 and 10. Each figure contains an augmented threat tree that represents the vulnerability exploited, the steps needed to exploit it, the attacker's operations, and the fingerprint generated by those operations. The final goal of both attacks is to steal sensitive information from a business information system with desired system permissions.

Operations conducted on a Windows machine may leave some forensic traces in the registry, some are persistent for a long time and some are volatile. If a piece of registry fingerprint is coupled with information from the event logs and file systems, the insider attack may be tracked and reconstructed. Based on our observation, relevant fingerprints can be located in machine's *System* hive, *Software* hive, the user's *NTuser.dat* hive, the *setupapi.log* that keeps a history of all devices installed via plug and play, and the Security event log.

Attack E is a classic industrial espionage inside attack that is accomplished by copying sensitive data to a personal USB device. The inside attack is conducted on 7/29/2011. Based on information in the registry, at 1:03:39 AM, a *Centon* USB device with a serial number of *6AFA4AAD80* was attached to the machine. At 1:04:34 AM, the attack was logged into the system and left fingerprints in the security event log. Based on additional fingerprints in the registry, the USB device

with serial number *6AFA4AAD80* can be linked with the disk with driver letter *E*. Examining the *RecentDocs* registry key with the tool *RegExtract* shows that _USBSTOR.sql, Removable Disk (E:), _USB.sql, and a file named "highly sensitive things" which is flagged in the honeypot as a sensitive file, were recently accessed. At 1:14:44 AM, User synchronized the document titled with "highly sensitive things", with the Removable Disk (E:). All these fingerprints are shown in Figure 9.

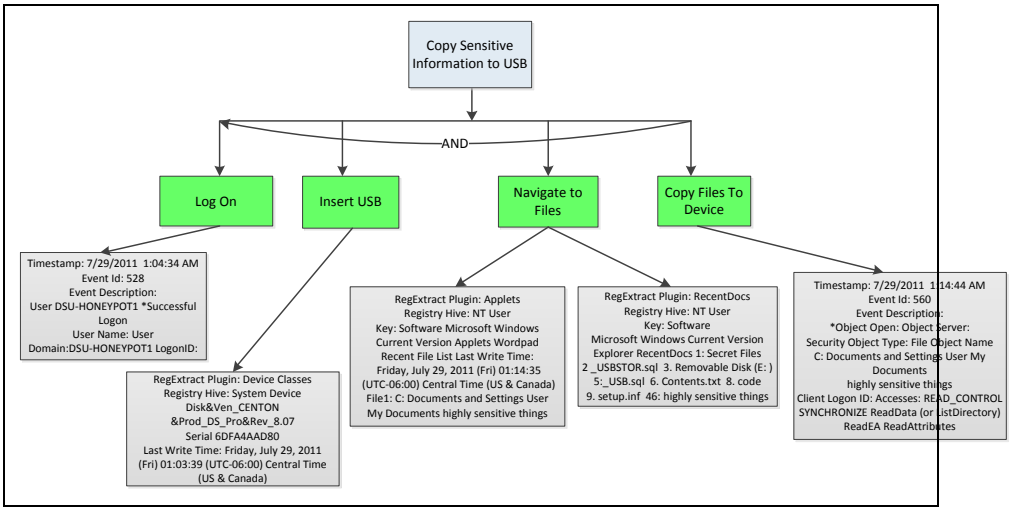

Figure 9 A part of the evidence tree for **Case Two Internal Attack E**

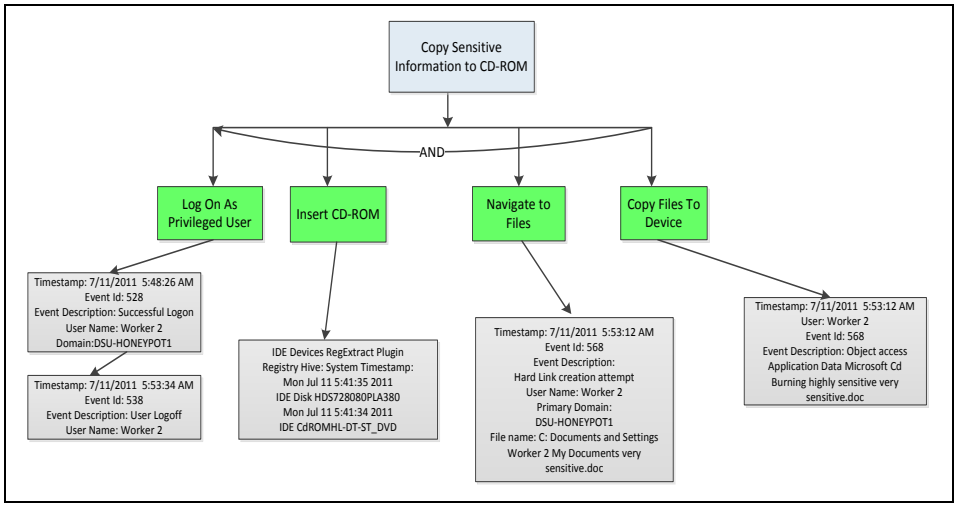

Figure 10 A part of the evidence tree for **Case Two Internal Attack F**

Attack F is a classic industrial espionage inside attack by copying sensitive data to a personal CD-ROM device. The inside attack is conducted on 7/29/2011. Based on fingerprints in the *security event log*, user *Worker 2* logged into the system at

5:46: 26, and attempted to create a hard link with "highly sensitive very sensitive" at 5:53:12. Analysis of the IDE Device Class registry shows that a CD ROM was documented at 5:47:34, a minute after *Worker 2* logged on to the system. Finally, the user *Worker 2* is found to burn the file "highly sensitive things" to the CD ROM at 5:53:12. All these fingerprints of Attack F can be found in Figure 10.

5. DISCUSSION

Once fingerprints of each individual attack have been identified and located, an evidence tree can be built to guide the tracking and reconstruction of such attack. Take Attack C for example. The most sensitive operation of this attack is the start of the *explore.exe* process with the system user *CAPITAL-EE7B9F2$'s* privilege since such a process (*explore.exe*) is usually run by a regular user instead of the system user *CAPITAL-EE7B9F2$*. Once such an event is logged in the system and is identified by a monitoring agent, an alert can be issued to indicate that a privilege escalation attack may have been launched.

Evidence trees are expected to be the key to automate the tracking and reconstruction of both external and inside attacks since the fingerprints defined in the evidences trees can provide contextual information to guide the forensic investigation of corresponding attacks. Take Attack C for example. Once the system has identified that the *explore.exe* process is running and the system user *CAPITAL-EE7B9F2$* is logged in, a privilege escalation attack alert should be issued. Now it is critical to determine how such an attack has been conducted and who has conducted it. Based on the evidence tree of Attack *C* (Figure 8), a sequence of operations including the opening of the file *At1.job*, the launch of the process *cmd.exe*, and the launch of the process *explore.exe* are correlated with the system user *CAPITAL-EE7B9F2*, while the process *at.exe* is run by the regular user *User*. Therefore, the regular user *User* can be correlated with Attack *C*. However, when multiple users share the same system, there are many issues to be addressed in order to reconstruct the attack and correlate such an attack to a specific user. If each user schedules a task in the system, then it will need to determine which scheduled task starts the command prompt (*cmd.exe*). This information can be identified with its corresponding scheduled job which can be found in "*C:\WINDOWS\Tasks\Atx.job*" (*x* represents the schedule creation sequence such that a smaller value of *x* means earlier creation of such job. Also, the creation sequence of *Atx.job*, timestamps of the *at.exe* process can be used together to link the job file to a specific user). If two users create the same type of task, i.e., each start a instance of *cmd.exe* process, then it would be extremely difficult to correlate this attack to a specific user since the start of *explore.exe* process leaves no other information in the system. To correlate such an attack to a specific user under this situation, more contextual information is needed, for example, the operations the attacker will do after the user obtained the system user's privilege.

Overall, even though sensitive operations of an attack can be used as the identity of

an attack, the fingerprints left in the system alone may not be sufficient to reconstruct the corresponding attack without the help of other contextual information. In a computer system, commands executed in a command prompt are usually not recorded, therefore, what the attacker has exactly done to system remains unknown to investigators. Also, sensitive system operations are usually executed by the system user instead of a regular user, therefore, there is a missing link between the regular user's (the insider) activities and the system operations. Hence, other contextual information is needed to successfully reconstruct the attack, and such information is exactly what evidence trees can provide to investigators. Taking Attack *C* for example, once the chain linking operations of the launch of the *at.exe* process, the opening of the file *Atx.job*, *the* start of *the process cmd.exe*, and the start of the process *explorere.exe* is established, then Attack *C* can be successfully identified, tracked, and reconstructed automatically.

6. RELATED WORK

Forensics readiness has recently been a big research concern in digital forensic investigation and information assurance (Carrier & Spafford, 2003; Endicott-Popovsky et al., 2007; Rowlinson, 2004; Tan, 2001; Tang & Daniels, 2005; Wilson & Wolfe, 2003; Yasinsac & Manzano, 2001). Existing research efforts focus on organization-level framework design such as policy or management. None of them has addressed the details of the technology part of forensics readiness, e.g. mechanisms of the application and system event logging, fingerprint storage and archiving, and evidence-handling procedures. In this research, a formal forensics investigation is conducted for each category of frauds and intrusions against a honeynet simulating an online business information system. The research will allow security and digital forensics professionals to fully understand what fingerprints are available, what fingerprints are necessary but not available based on current settings, how to log the needed fingerprints, how long fingerprints should be preserved in logs, and the detailed procedures to appropriately handle evidences.

Honeynet has recently been applied to the fields of cyber security protection and network forensic investigation (Chen, Laih, Pouget, & Dacier, 2005; Khattab, Melhem, Mosse, & Znati, 2006; Krasser, Grizzard, & Owen, 2005; Levine, Grizzard, & Owen, 2004; Levine, Labella, Owen, Contis, & Culver, 2003; Spitzner, 2003a, 2003b; Todtmann, Riebach, & Rathgeb, 2007; Watson, 2007), due to its cost-effectiveness for information assurance education and research. Honeynet is sometimes deployed along with the target information system to divert attacks (Watson, 2007). It can also be deployed as a standalone system to improve employee's security awareness (Krasser et al., 2005; Levine et al., 2003; Levine et al., 2004), mitigate the impact of attacks (Khattab et al., 2006), provide early response to external attacks (Todtmann et al., 2007), obtain statistical data for attack analysis (Chen et al., 2005), understand the general mechanisms of attacks (Chen et al., 2005; Pouget & Dacier, 2004), and help to detect insider threat (Spitzner,

2003a). A different approach is taken to apply the honeynet technology in this research. Instead of using a honeynet to attract external intrusions, attacks and fraudulent activities are performed on the honeynet to simulate both external intrusions and internal attacks against online businesses.

7. CONCLUSION

In this paper, a systematic approach is proposed to develop the forensics readiness to fight against attacks and frauds that are committed to online business information systems. The approach mainly focuses on identifying, locating, and modeling evidences for external and internal attacks. Threat models are developed for the online business information systems using attack trees, and then these threat models are mapped to augmented attack trees by including individual attack operations. A total of six modeled attacks, two external DoS attacks, two Case One internal attacks, and two Case Two internal attacks are conducted against a honeynet that simulates an online business information system. Forensics investigations are conducted immediately after each attack is committed, and fingerprints are then identified, collected, and mapped to an evidence tree.

The resulted evidence trees can provide essential information for attack investigation, by answering at least the following three key questions: what information is relevant to the attack studied, where related fingerprint items can be located, and what information each piece of fingerprint can indicate. An evidence tree provides a mechanism to correlate attack operations with the fingerprints they produce, which can provide guidance in manual forensic investigation and provide the contextual information that is needed for the automation of attack tracking and reconstruction.

Future efforts will involve the analysis of additional avenues of attacks against the online business information system in order to gain a complete view of valuable evidence identification, locating, and logging mechanisms. The eventual goal is to develop a systematic mechanism to automate the attack tracking and reconstructing in online business environments.

ACKNOWLEDGEMENTS

This work is supported in part by NSF under grant CNS 1004843. We would also like to thank NSF REU DSU Site student fellows Mikal Ustad and Tom SwiftBird for their contributions to this paper, and Kacie Fodness for her hard work on the grammatical editing.

REFERENCES

Cappelli, D., Moore, A., Trzeciak, R., and Shimeall, T. (2009). Common sense guide to prevention and detection of inside threats, 3rd edition. White Paper of CMU CyLab.

Carrier, B. & Spafford, E. (2003, Fall). Getting physical with the digital investigation process. International Journal of Digital Evidence, 2(2). Retrieved from http://www.cerias.purdue.edu/ssl/techreports-ssl/2003-29.pdf

Carrier, B. & Spafford, E. (2004, July). An event-based digital forensic investigation framework. In Proceedings of Digital Forensic Research Workshop. Retrieved from http://www.digital-evidence.org/papers/dfrws_event.pdf

Casey, E. (2011). Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet (3rd edition). Burlington, MA: Elsevier.

CENZIC. (2008). Cenzic Application Security Trends Report -- Q1 2008. Retrieved from http://www2.cenzic.com/downloads/Cenzic_AppSecTrends_Q1_2008.pdf

CeBuSoft. (2013). CeBuSoft Accounting Information System. Retrieved from http://cebusoft-accounting-information-system.software.informer.com

Chen, P., Laih C., Pouget E., & Dacier M. (2005). Comparative survey of local honeypot sensor to assist network forensics. In Proceedings of the 1st International Workshop on Systematic Approach to Digital Forensics Engineering.

De Aalst, W., Van Hee, K., Van De Werf, J., Kumar, A., & Verdonk, M. (2009). Conceptual model for on line auditing. Retrieved from http://www.personal.psu.edu/axk41/olat09.pdf

Endicott-Popovsky, B., & Frincke, D. (2004). Adding the fourth "R". In Proceeding of the 2004 IEEE Workshop on Information Assurance.

Endicott-Popovsky, B., Frincke, D.A., and Taylor, C.A. (2007, May). A theoretical framework for organizational network forensic readiness. Journal of Computers, 2(3), 1-11.

Espiner, T. (2008, Dec). Businesses urged to devise digital-forensics plans. ZDNet Web site. Retrieved from http://www.zdnet.com/businesses-urged-to-devise-digital-forensics-plans-3039569682/

Fratto, M. (2008). 2008 security survey: we're spending more, but data's no safer

than last year. InformationWeek Security. Retrieved from http://www.informationweek.com/security/management /2008-security-survey-were-spending-more/208800942

Gu, L., Liang, J., & Wang, J. (2005, December). Theoretical framework and method of detecting accounting fraud. Journal of Modern Accounting and Auditing, 1(7), 66-71.

Hoffman, P. (2007, January 25). RSA survey reports low level of trust in online banking security. eWeek News. Retrieved from http://www.eweek.com/c/a/Security/RSA-Survey-Reports-Low-Level-of-Trust-in-Online-Banking-Security/

The Honeynet Project. (2013). Retrieved from http://www.honeynet.org

Ingols, K., Lippmann, R., & Piwowarski, K. (2006). Practical attack graph generation for network defense. In Proceedings of 22nd IEEE Annual Computer Security Applications Conference.

Ingols, K., Chu, M., Lippmann, R., Webster, S., & Boyer, S. (2009). Modeling modern network attacks and countermeasures using attack graphs. In Proceedings of the 25th IEEE Annual Computer Security Applications Conference.

Jeyaraman, S. & Atallah, M. (2006). An empirical study of automatic event reconstruction systems. Journal of Digital Investigations, 3S, 108-115.

Jha, S., Sheyner, O., & Wing, J. (2002). Two formal analyses of attack graphs. In Proceedings of the Computer Security Foundations Workshop, pp. 45-59.

Khattab, S., Melhem, R., Mosse, D., & Znati, T. (2006). Honeypot back-propagation for mitigating spoofing distributed denial-of-service attacks. In Proceedings of the 20th Parallel and Distributed Processing Symposium (IPDPS 2006), 25-29 April 2006.

Krasser, S., Grizzard, J., & Owen, H. (2005). The use of honeynets to increase computer network security and user awareness. Journal of Security Education, 1(2/3), 23-37.

Larson, C. (2008, February). Accounting Fraud and Institutional Investors. PhD Dissertation, University of Michigan.

Levine, J., Grizzard, B., & Owen, H. (2004). Using honeynets to protect large enterprise networks. IEEE Security and Privacy, 2(6), 73-75..

Levine, J., Labella, R., Owen, H., Contis, D., & Culver, B. (2003). The use of

honeynets to detect exploited system across large enterprise networks. In Proceedings of the 2003 IEEE Workshop on Information Assurance.

Mauw, S. & Oostdijk, M. (2005). Foundations of attack trees. In Won, D., Kim, S., eds., International Conference on Information Security and Cryptology – ICISC 2005. Volume 3935 of LNCS, Springer, 186–198.

Moore, A., Cappelli, D. & Trzeciak, R. (2008). The "big picture" of insider IT sabotage across U.S. critical infrastructures. Software Engineering Institute, Carnegie Mellon U niversity.

Poolsapassit, N. & Ray, I. (2007). Investigating computer attacks using attack trees. In IFIP International Federation for Information Processing, Vol. 242. Advanced Digital Forensics III.

Pouget, F. & Dacier, M. (2004). Honeypot-based Forensics. In Proceedings Of AusCERT Asia Pacific Information technology Security Conference 2004(AusCERT2004).

Ramzan, Z. (2008, December 24). Security trends of 2008 and predictions for 2009. Net Security News. Retrieved from http://www.net-security.org/article.php?id=1194

Romney, M. & Steinbart, P. (2008). Accounting Information Systems, 11th ed. ISBN: 0136015182. Prentice Hall.

Robb, D. (2008, February 8). Top 5 security trends. Enterprise Planet News. Retrieved from http://www.enterpriseitplanet.com/security/features/article.php/3726926

Rowlinson, R. (2004, Winter). A Ten Step Process for Forensic Readiness. International Journal of Digital Evidence, 2(3). Retrieved from http://www.utica.edu/academic/institutes/ecii/publications /articles/A0B13342-B4E0-1F6A-156F501C49CF5F51.pdf

RSA Security. (2008). 2008 CSI Computer Crime & Security Survey. Retrieved from http://i.zdnet.com /blogs/csisurvey2008.pdf .2008

Saini, V., Duan, Q., & Paruchuri, V. (2008, April). Threat modeling using attack trees. Journal of Computing Sciences in Colleges, 23(4), 124-131.

Schneier, B. (1999, December). Attack trees: Modeling security threats. Dr. Dobb's Journal, 24(12), 21-29.

Seltxer, L. (2006, December 4). Is online banking too dangerous? eWeek News.

Retrieved from http://www.eweek.com/c/a/Security/Is-Online-Banking-Too-Dangerous/

Siponen, M., & Oinas-Kukkonen, H. (2007). A review of information security issues and respective research contributions. TDatabase for Advances in Information Systems, 38(1), 60-80.

Spitzner, L. (2003a). Honeypots: catching the insider threat. In Proceedings of the 19PthP Annual Computer Security Applications Conference.

Spitzner, L. (2003b, March). The Honeynet Project: Trapping the hackers. IEEE Security and Privacy, 1(2), 15-23.

Straub, D.W. (1990, September). Effective IS security: An empirical study. Information System Research, 1(3), 255-276.

Swiderski, F. & Snyder, W. (2004). Threat modeling (Microsoft Professional). Microsoft Press.

Tan, J. (2001, July 17). Forensics readiness. Retrieved from http://isis.poly.edu/kulesh/forensics/forensic_readiness.pdf

Tang, Y. & Daniels, T. (2005). A simple framework for distributed forensics. In Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops.

Todtmann, B., Riebach, S., & Rathgeb, E. (2007). The honeynet quarantine: reducing collateral damage caused by early intrusion response. In Proceedings of the 6th International Conference on Networking.

Valentine, A. (2007). Art of preserving digital evidence. Available at HTUhttp://www.onlinebankingreview.com.au/DigitalEvidence.phpUTH.

Watson, D. (2007, January). Honeynets: A tool for counterintelligence in online security. Network Security, 2007(1), 4-8.

Williamson, G. (2006, Fall). Enhanced authentication in online banking. Journal of Economic Crime Management, 4(2). Retrieved from http://utica.edu/academic/institutes/ecii/publications/articles /51D6D996-90F2-F468-AC09C4E8071575AE.pdf

Wilson, W. & Wolfe, H. (2003, June). Management strategies for implementing forensic security measures. Information Security Technical Report, 8(2), 55-64.

Yasinsac, A. and Manzano, Y. (2001). Policies to enhance computer and network forensics. In Proceedings of the 2001 IEEE Workshop on Information

Assurance and Security.

Zhang, L. & Guan, Y. (2008). Detecting click fraud in pay-per-click streams of online advertising networks. In Proceedings of 28th IEEE International Conference on Distributed Computing Systems.