121st ASEE Annual Conference & Exposition
Indianapolis, IN
June 15-18, 2014

Paper ID #8420

Security Incident Tracking in Virtualized Linux Environment

Dr. Manghui Tu, Purdue University Calumet

Assistant Professor, Computer Information Technology, Purdue University Calumet, USA. He received his Ph.D. degree of computer science from the University of Texas at Dallas in December 2006. His research interests include distributed computing, information security, and computer forensics.

Mr. Shiming Xue, Purdue University Calumet

SHIMING XUE Address: 6943 Wicker Ave E-mail: xues@purduecal.edu Hammond, IN, U.S 46323 Cell: +1 (765) 404-9776

EDUCATION

Purdue University Calumet, Hammond, IN Jul 2014(expected) Bachelor of Science in Computer Info Technology Department of Computer Info Tech Graphic Courses: Integrative Programming, Networking Technologies, Discrete Mathematics for IT, Applied Database Tech, Data Communication and Networking.

Projects and Activities: Created a customer/employee management system with windows server 2008 R2 Created the database and few Web Pages that help to update, insert, and delete data for a customer and employees. Created a Minesweeper with the C# by Visual studio 2010 Created a minesweeper game, which can set mines' number. Created a high school grade management system with the Basic C by Visual Studio 2010 Created a database for a high school which could help the school manage their students' grades

PROFESSIONAL EXPERIENCE Huaqing Computer LLC, Pujiang, China June 2007- July 2007 Intern Consultant Maintained WINDOWS/UNIX operation system for customers and initiated new computers by installing the computer operation systems Vista/XP and by setting up the working environment. Diagnosed and fixed the problems on both software and hardware of the computers for customer supporting.

Sony Ericsson Mobile Communication (China) Ltd., Beijing, China Dec2011-Jan 2012 Intern Consultant Checked the credit files and preparation of disbursement checks for different departments Checked monthly trading volume, and provided the trading volume to the department header Practiced leadership skills and assisted the department header to leader the team Guided my group to complete financial statement

SKILLS

COMPUTER: SQL Server 2008 R2: Create database, Create tables, Insert data to the tables, create store procedures.

Oracle 10g: Create database, Create tables, Insert, Update data to the tables

Visual Studio 2010 Create C# program, Create Web Pages with aspx, can create normal web application to finish the

Microsoft Excel: Insert data, Create bar charts or pie charts for the data, use the formula to complete the data inserting and updating

Microsoft Visio: Crete the flowchart for processing a project

Virtual machine 7.0: Run the Virtual system: Windows Server 2003sq

OTHERS: English (Fluent), Chinese (Mother Tongue)

AWARDS AND HONORS Semester Honors and Dean's List (Distinguished student awards) FALL 2010

Security Incident Tracking in Virtual Linux Environment

Abstract

Virtualized environment provides a heaven for malicious and criminal activities. It is expected that illegal activities in virtualized environments will be increased as virtualization gains its popularity. Meanwhile, numerous digital security and privacy laws and regulations have put business and organizations under obligations to prepare for auditing and legal investigations. Therefore, businesses must prepare for the responsiveness to unforeseen security incidents in virtualized environments. To establish forensics readiness for businesses and organizations, it is essential to identify what fingerprints are relevant and where they can be located, and whether all the needed fingerprints are available to reconstruct the incidents successfully. Also, fingerprint identification and locating mechanisms 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. All these rely on the knowledge of security attacks and the fingerprints left by them. In this research, we will explore potential security exploitations and their corresponding fingerprints left in the virtualized Linux environment. Attacks are modeled as augmented attack trees and then are conducted against a simulated virtualized environment, which is followed by a forensic investigation. Finally, an evidence tree is built for each attack based on fingerprints identified within the system. With evidence tree, it is possible to identify sensitive fingerprints for each attack. Also, the evidence tree is expected to provide contextual information needed for automating forensics investigation of a security incident.

1 Introduction

Virtualization technology has been playing a key role in server virtualization and cloud computing[6, 10, 12]. It can be leveraged to achieve business benefits since it can provide huge benefits in many aspects such as reducing IT administrative time, facilitating data backup and recovery, being adaptive to business change, business continuity and disaster recovery, etc[10, 12]. However, it also provides a heaven for malicious code to be executed, untrustworthy data to be processed, or illegitimate data to be stored by services inside the virtual environment[1, 2, 3, 12, 16, 17]. Even though intensive efforts have been put to secure the virtualized environments, it is evident that cyber crime and fraudulent activity against virtualized environment will continue to thrive[3, 12]. With rapid increase of the adoption of virtualization technology, we can expect to witness the increase of illegal activities in the cyber space. These security incidents not only result in substantial financial and operational losses, but also greatly hurt the confidence of customers, business partners and stakeholders. Meanwhile, over the last decade, government and industry bodies around the world have issued many laws and regulations, for example, *The Basel Committee on Banking Supervision* (*Base II Accord*), *The Health Information Portability and Accountability Act of 1996* (*HIPAA*), *Payment Card Industry Security Standard(PCISS), Sarbanes-Oxley Act (SOX), The Federal Information Security Management Act of 2002* (*FISMA*), and some others, to ensure the availability, integrity, and confidentiality of business data and the IT infrastructures. These mandates have placed pressures on businesses and organizations to take actions to ensure compliance with laws and regulations. Consequently, to minimize financial loss and be compliant with laws and regulations, businesses that host virtualized services must be prepared for the responsiveness to unforeseen security incidents.

Many intrusion/fraud prevention, detection, and tolerance mechanisms have been deployed by organizations and companies that have adopted cloud computing and virtualization technologies in order to secure their IT infrastructures and the sensitive data stored in information systems[3, 12]. However, security incidents cannot be totally eliminated as shown by researchers[3, 6]. It is evident that even with the state-of-the-art security prevention, detection, and tolerance mechanisms, the risk introduced by virtualization and especially cloud computing technologies cannot be completely avoided[3, 12]. 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[13, 14, 19, 22, 23].

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[7, 21, 22]. Digital forensics involves the process of preservation, acquisition, analysis, discovery, documentation, and presentation of evidence[7, 22]. The success of digital forensics is highly dependent on forensics readiness[21, 22], 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[15, 21, 22]. Forensic readiness is an increasingly important topic in forensic investigation and information assurance research[4, 5, 15, 21, 22, 25]. 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 computer information systems[13].

The overall goal of this research is to provide technical guidance to effectively and efficiently investigate security incidents that take place in virtualized Linux systems. There are a few challenging issues need to be addressed for virtualized Linux environments. First, the fingerprints left by attacks and frauds that are needed to reconstruct the incidents of attacks remain unclear to digital forensics and security professionals[13]. Second, many security incidents remain undetected due to the lack of sophisticated identification mechanisms. Third, many security incident investigations are not conducted due to the unaffordable effort needed to process the huge amount of the information kept in the system[15, 22]. As discussed in our previous work[22], one of the key is the establishment of evidence models for attacks. In this research, we will first examine potential security vulnerabilities virtualized Linux environment and attacks are then conducted against the virtualized Linux environment. Second, forensics investigation will be followed to identify and locate their corresponding fingerprints left in system. With fingerprints identified and located, evidence model will be developed for each attack. We expect that evidence models can be used to guide forensics investigation in the future. Furthermore, evidence models are expected to provide the contextual information for an incident tracking software, which may have the potential to automate the incident tracking in a virtualized computing environment or the cloud.

2 System Modeling and Methodology

2.1 Research Method

The overview of research method is presented Fig. 1 and described below.

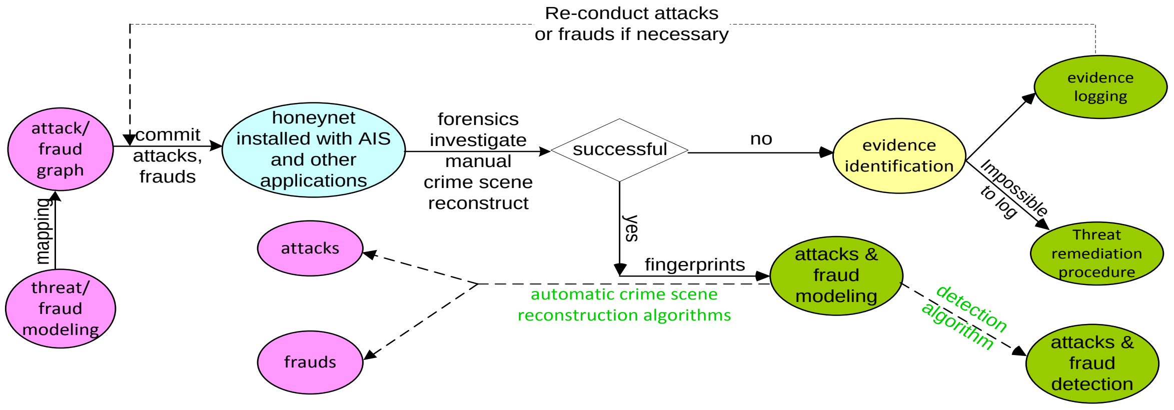


Fig. 1. The overview of the research process on forensics readiness

First, virtual Linux environments will be designed to simulate different online business environment, which may be composed of the needed functional components of online businesses, such as web servers, file servers, email servers, as well as open source business information systems. Second, a systematic approach will be developed to model attacks and inside activities. The vulnerabilities of virtual Linux environments will be assessed and the corresponding threats will be identified. Threats will be systematically modeled by using attack trees[13, 20, 22] based on a set of known attacks exploiting the identified vulnerabilities existed in the online information system. Individual attacks are then modeled as augmented attack trees, on which each leaf node represents a specific attack operation. Third, the modeled attacks and inside activities will be conducted against the virtualized Linux environment and forensics investigation will be followed. Finally, once fingerprints are identified and located, they are used to manually reconstruct the crime scene 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 will then become child nodes of the leaf nodes in the augmented attack tree. This entire process will finally result in an evidence tree for each attack studied. Fingerprints of sensitive operations of the evidence trees will be identified as incident identifiers and the evidence tree can provide the contextual information to reconstruct security incidents automatically.

2.2 Threat Modeling and Attack Generation

The attack tree approach first proposed by Schneier[18] is used to systematically analyze security threats and has been introduced as a systematic threat modeling approach in some research[20]. 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[8, 9, 13]. 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. Fig.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, 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.

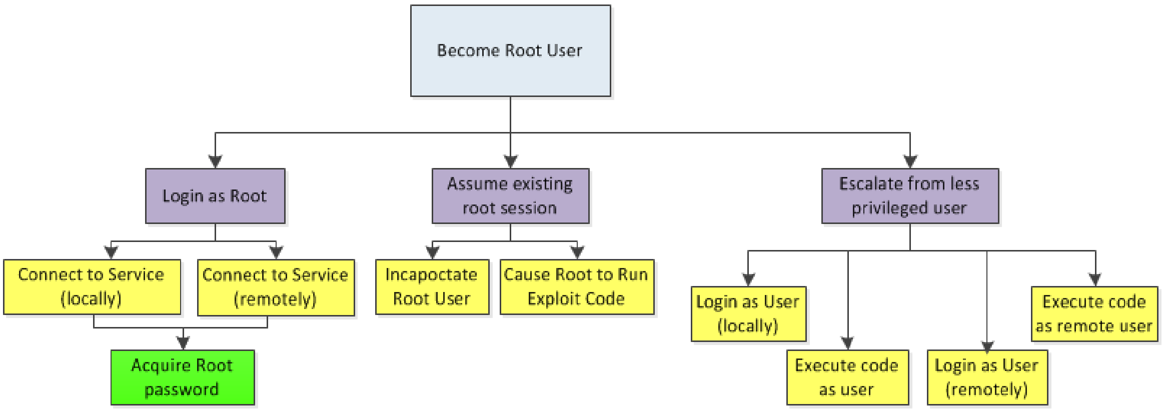

Become Root User
Login as Root
Assume existing root session
Escalate from less privileged user
Connect to Service (locally)
Connect to Service (remotely)
Incapoctate Root User
Cause Root to Run Exploit Code
Login as User (locally)
Execute code as remote user
Acquire Root password
Execute code as user
Login as User (remotely)

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

2.3 A Simulated Virtualized Linux Environment and Tools

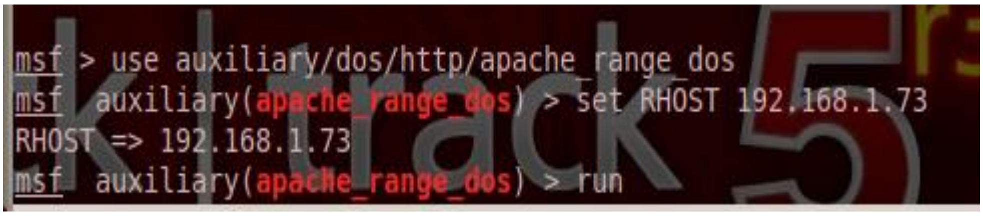
A desktop PC is used to host the Ubuntu 10 virtual Linux environment, which requires large sized memory and hard disk. A wireless router will be used for network setup and a thumb drive will be used for data collection. The attack tool used for this research is BackTrack 5 R3, which is a tool suite installed on the Ubuntu Linux distribution aimed at digital forensics and penetration testing use. A major package provided by Backtrack is the Metasploit Framework, which is a suite of penetration tools that can be used for system and network penetration and security testing. A BackTrack 5 R3 Bootup Linux machine is setup as the attacking system.

External attacks and internal attacks[11, 22] against the virtualized Linux environment are simulated in the following way. To study external attacks[22], the attack machine with backtrack 5 and the victim machine with Linux virtual machine are deployed to different networks, and the attack machine is assumed to have no knowledge of the victim machine and the system security credentials, such as an account password, of the network the victim machine resides. To study internal attacks[22], the attack machine with backtrack 5 and the victim machine with Linux virtual machine are deployed to the same network. Therefore, the attack machine has legitimate accesses to a subset of resources of the victim network. To study inside activities[11], the attack machine with backtrack 5 and the victim machine with Linux virtual machine are deployed to the same network, and the attack machine has legitimate accesses to all the resources hosted in the virtual environment.

3 Security Attacks

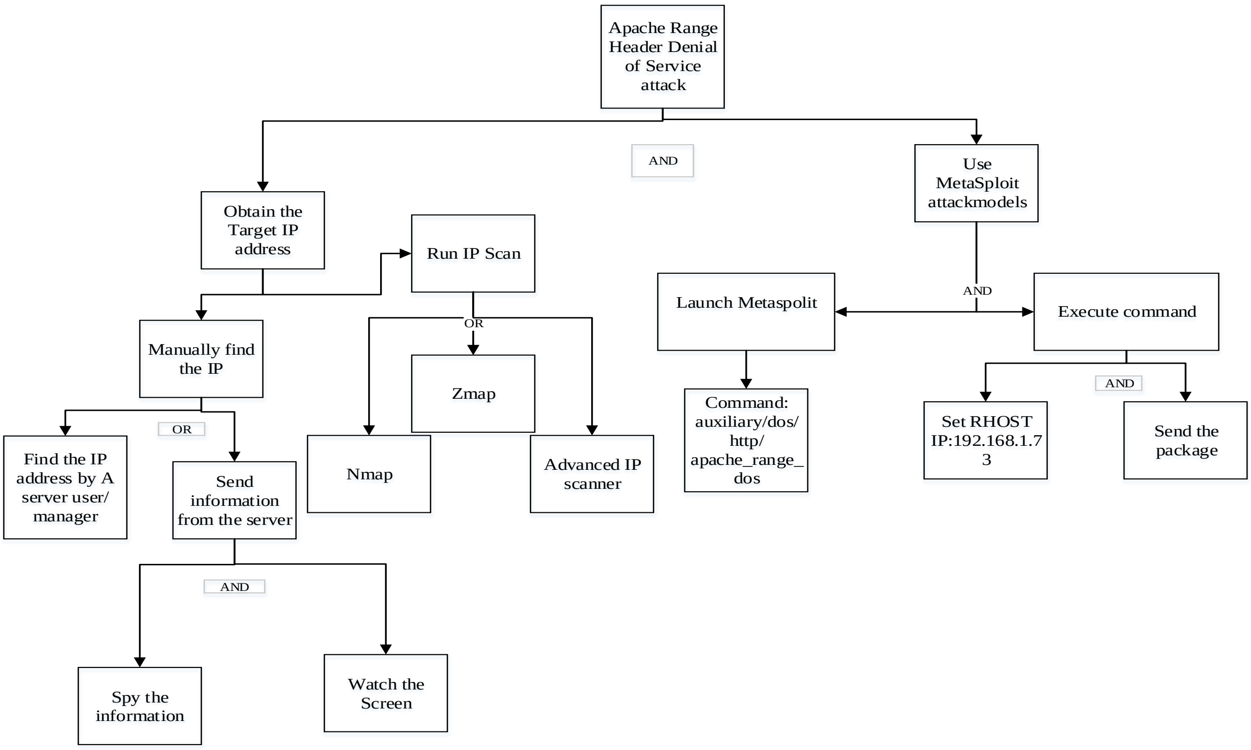
3.1 Attack I

Attack I is the Apache Range Header Denial of Service Attack, which targets on the Apache Web service installed in the virtualized Linux environment. In this research, Attack I is conducted against virtulized Linux server by using the MetaSploit console. The configuration and execution of the attack is shown in Fig. 3 and the attack modeling is shown in Fig. 4.



```
msf > use auxiliary/dos/http/apache_range_dos
msf  auxiliary(apache_range_dos) > set RHOST 192.168.1.73
RHOST => 192.168.1.73
msf  auxiliary(apache_range_dos) > run
```

Fig. 3: Configuring the Apache DoS attack.



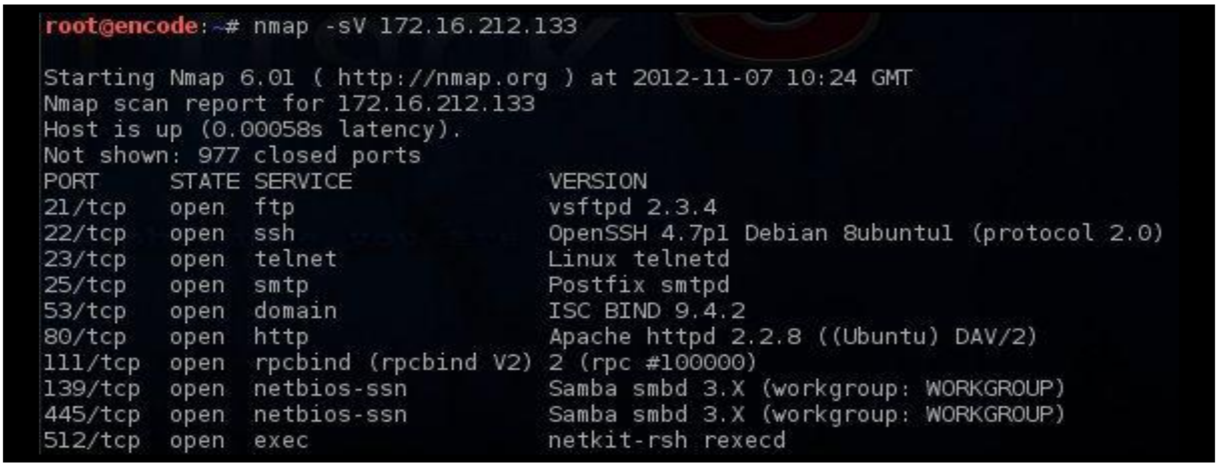
Apache Range Header Denial of Service attack
AND
Obtain the Target IP address
Use MetaSploit attackmodels
Run IP Scan
Launch Metaspolit
AND
Execute command
Manually find the IP
OR
Zmap
Command: auxiliary/dos/ http/ apache_range_ dos
AND
Set RHOST IP:192.168.1.73
Send the package
OR
Find the IP address by A server user/ manager
Send information from the server
Nmap
Advanced IP scanner
AND
Spy the information
Watch the Screen

Fig. 4: The attack tree modeling of Apache DoS attack.

For a small business, Apache web service is usually its first choice to host company website due to its low cost and low demand of knowledge and skills for deployment. Virtualization can further help to reduce the deployment cost and space requirement. A few vulnerabilities exists for Apache web service in Linux environment and Apache Range Header Denial of Service attack (apache_range_dos) is one of the most common attacks exploiting those vulnerabilities. This attack uses a range header that expresses multiple overlapping ranges, which forks processes and causes excessive memory and CPU resource consumption.

3.2 Attack II

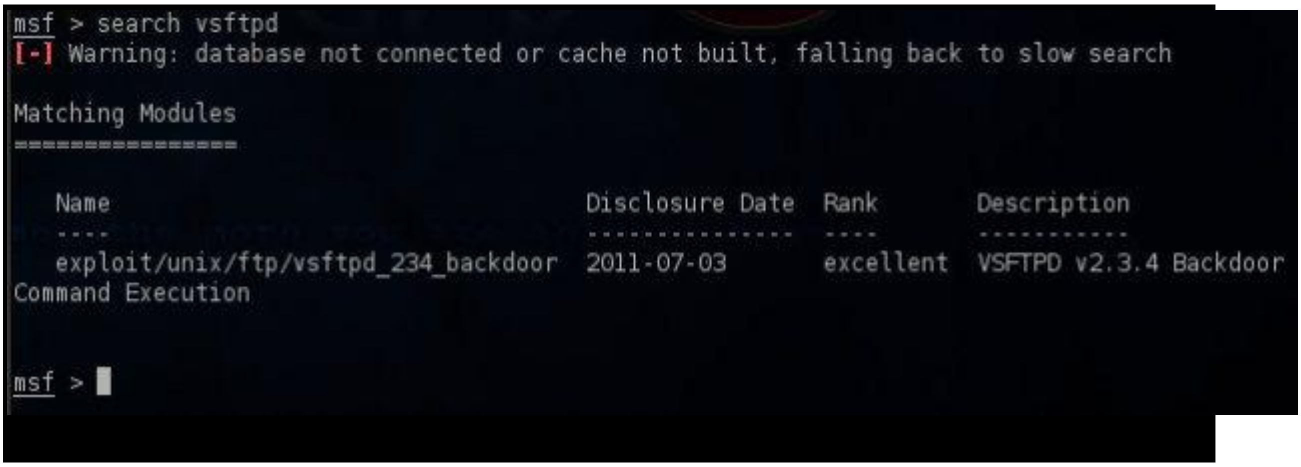
Attack II is a privilege escalation attack[24], which exploits a malfunctioning version of FTP server (version 2.3.4, named VSFTPD) installed in Linux environment. To detect whether there exists such VSFTPD service installed, a service fingerprinting scan is conducted by using Nmap (shown in Fig. 5).



```
root@encode:~# nmap -sV 172.16.212.133

Starting Nmap 6.01 ( http://nmap.org ) at 2012-11-07 10:24 GMT
Nmap scan report for 172.16.212.133
Host is up (0.00058s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE            VERSION
21/tcp   open  ftp                vsftpd 2.3.4
22/tcp   open  ssh                OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet             Linux telnetd
25/tcp   open  smtp               Postfix smtpd
53/tcp   open  domain             ISC BIND 9.4.2
80/tcp   open  http               Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind (rpcbind V2) 2 (rpc #100000)
139/tcp  open  netbios-ssn        Samba smbd 3.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn        Samba smbd 3.X (workgroup: WORKGROUP)
512/tcp  open  exec               netkit-rsh rexecd
```

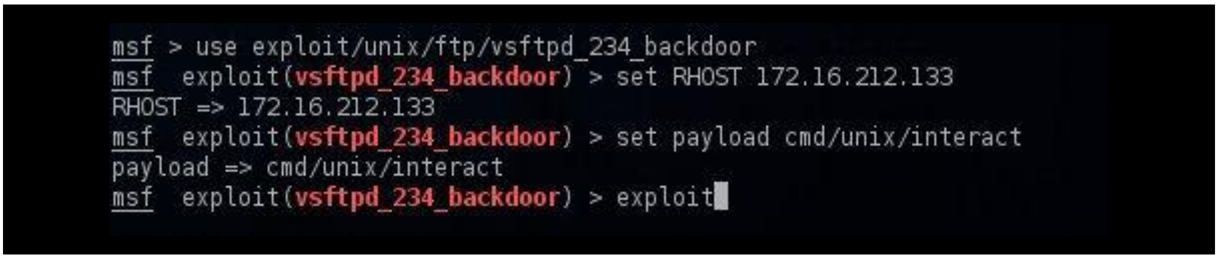
Fig.5. The discovery of VSFTPD Service through Nmap

Once the malfunctioning version of the FTP server has been identified, the open door of the FTP server can be detected by using Metasploit (shown in Fig. 6). Then, Attack II is conducted as shown in Fig. 7, and the attack model of Attack II is shown in Fig. 8.



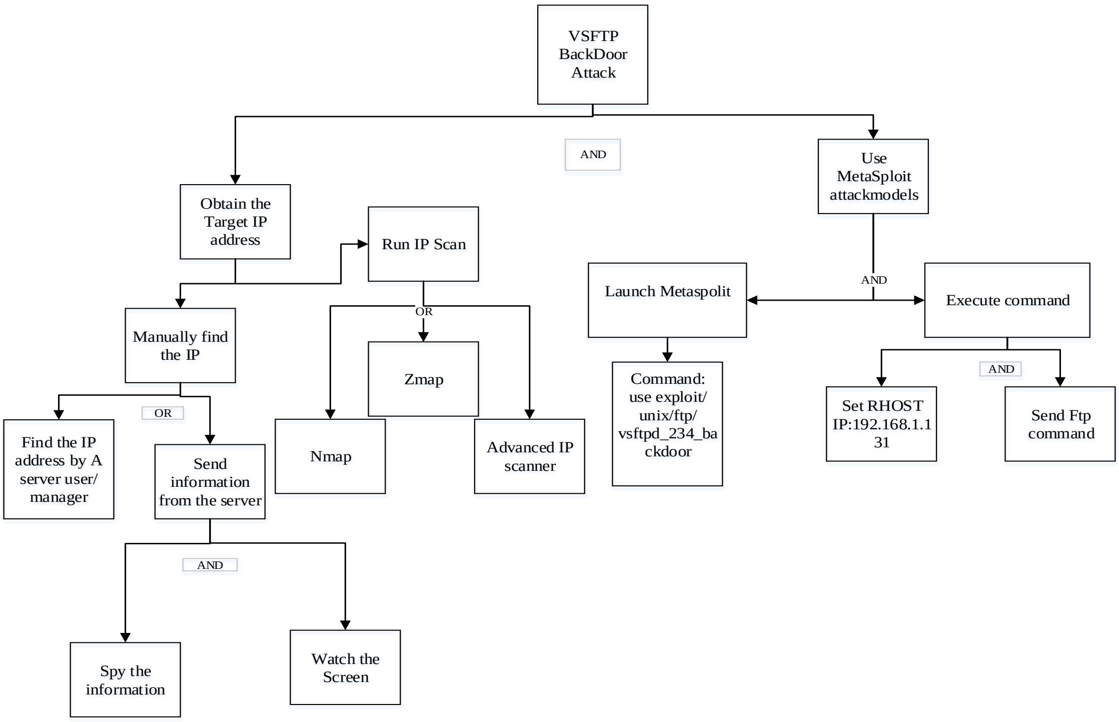
```
msf > search vsftpd
[-] Warning: database not connected or cache not built, falling back to slow search

Matching Modules
================

   Name                               Disclosure Date  Rank       Description
   ----                               ---------------  ----       -----------
   exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03    excellent  VSFTPD v2.3.4 Backdoor
Command Execution


msf > 
```

Fig.6. The discovery of the backdoor module by using Metasploit.



```
msf > use exploit/unix/ftp/vsftpd_234_backdoor
msf  exploit(vsftpd_234_backdoor) > set RHOST 172.16.212.133
RHOST => 172.16.212.133
msf  exploit(vsftpd_234_backdoor) > set payload cmd/unix/interact
payload => cmd/unix/interact
msf  exploit(vsftpd_234_backdoor) > exploit
```

Fig.7. The conduction of attack II by using Metasploit.



VSFTP BackDoor Attack
AND
Obtain the Target IP address
Use MetaSploit attackmodels
Run IP Scan
Launch Metaspolit
AND
Execute command
Manually find the IP
OR
Zmap
Command: use exploit/ unix/ftp/ vsftpd_234_backdoor
AND
Set RHOST IP:192.168.1.131
Send Ftp command
OR
Find the IP address by A server user/ manager
Send information from the server
Nmap
Advanced IP scanner
AND
Spy the information
Watch the Screen

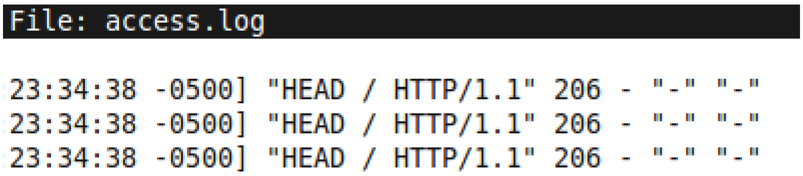
Fig. 8: The attack tree modeling of Attack II.

4. Results and Analysis

The logs at the victim machine were checked for each operation and the fingerprints were identified and located. With fingerprints of all operations identified and located, evidence tree were built for the attack. With the evidence tree, sensitive fingerprints for the attack may be identified and they can be used to detect or identify the existence of such an attack. Also, the evidence tree can provide contextual information to intelligent incident tracking software to help locate and identify fingerprints of each operation of the attack. Thus, incident tracking and reconstruction can be automated.

4.1. Evidence of Attack I

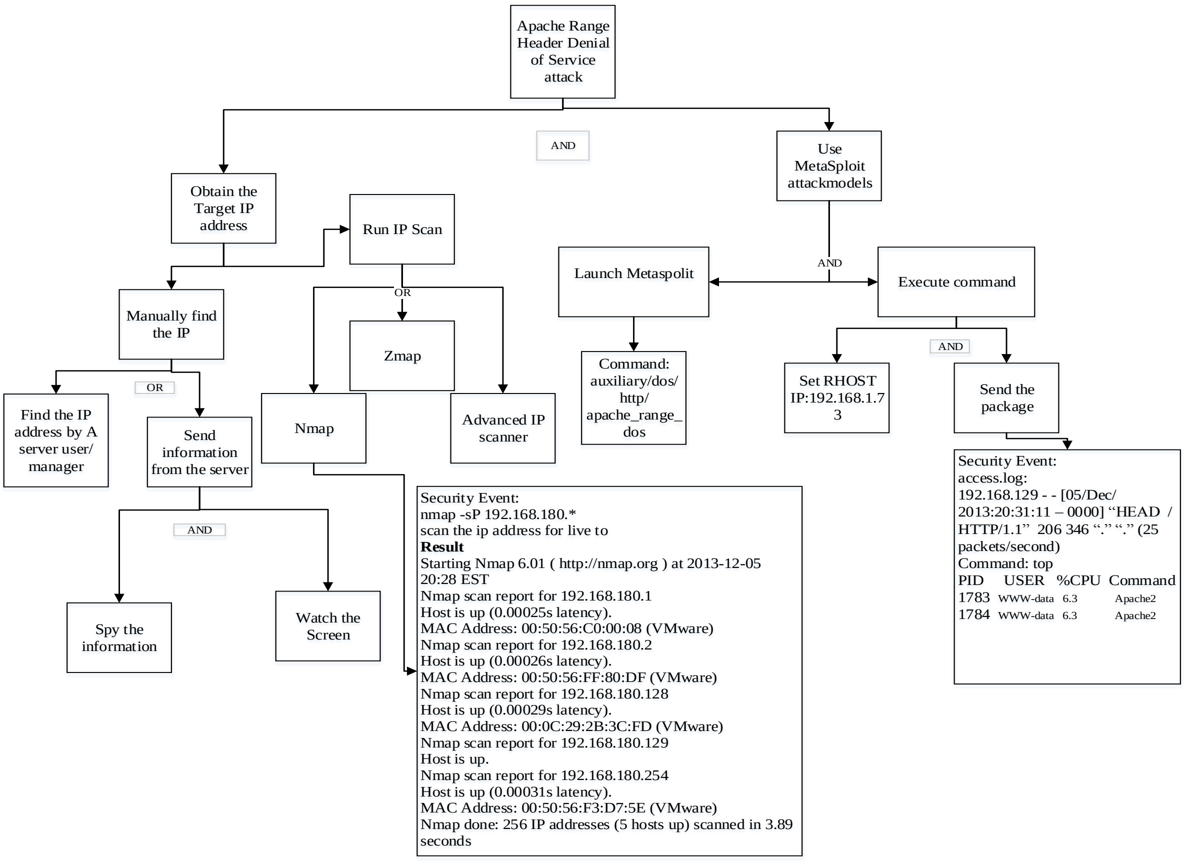
To identify and locate fingerprints of Attack I, the Access.log file on the target machine was examined. The Access.log file records all HTTP requests that have been sent to the server. For each of packet received at the server side, a corresponding response with a code of 206 is generated, which indicates that the server has partially responded to the HTTP GET request. To enable this, the request must have included a range of header field. In Attack I, the range header field contains overlapping ranges, which causes the server to fork the apache process and eventually exhausts all the resources at the server side. The logged evidence is shown in Fig. 9.



```
File: access.log

23:34:38 -0500] "HEAD / HTTP/1.1" 206 - "-" "-"
23:34:38 -0500] "HEAD / HTTP/1.1" 206 - "-" "-"
23:34:38 -0500] "HEAD / HTTP/1.1" 206 - "-" "-"
```

Fig. 9.Raw fingerprints in the Access.log file

The fingerprints of each operation of Attack I was identified and was then composed into a node, which becomes the child node of the corresponding operation node in the attack tree of Attack I. The resulted evidence tree is shown in Fig. 10. Since most of operations of Attack I are performed on the attacking machine, only two operations, i.e., the Nmap scanning operation and the exploit operation, will leave fingerprints on the target machine. For a successful implementation of Attack I, the victim machine can log a large number of packets sent from the attacker with the same user name, IP address of the attack machine, and timestamps, etc. The existence of a large number of HTTP packets sent from the same user can be considered as the sensitive fingerprints to identify Attack I.


Apache Range Header Denial of Service attack
AND
Obtain the Target IP address
Use MetaSploit attackmodels
Run IP Scan
Manually find the IP
OR
Zmap
Nmap
Advanced IP scanner
OR
Find the IP address by A server user/ manager
Send information from the server
AND
Spy the information
Watch the Screen
Launch Metaspolit
AND
Execute command
Command: auxiliary/dos/ http/ apache_range_ dos
AND
Set RHOST IP:192.168.1.7 3
Send the package

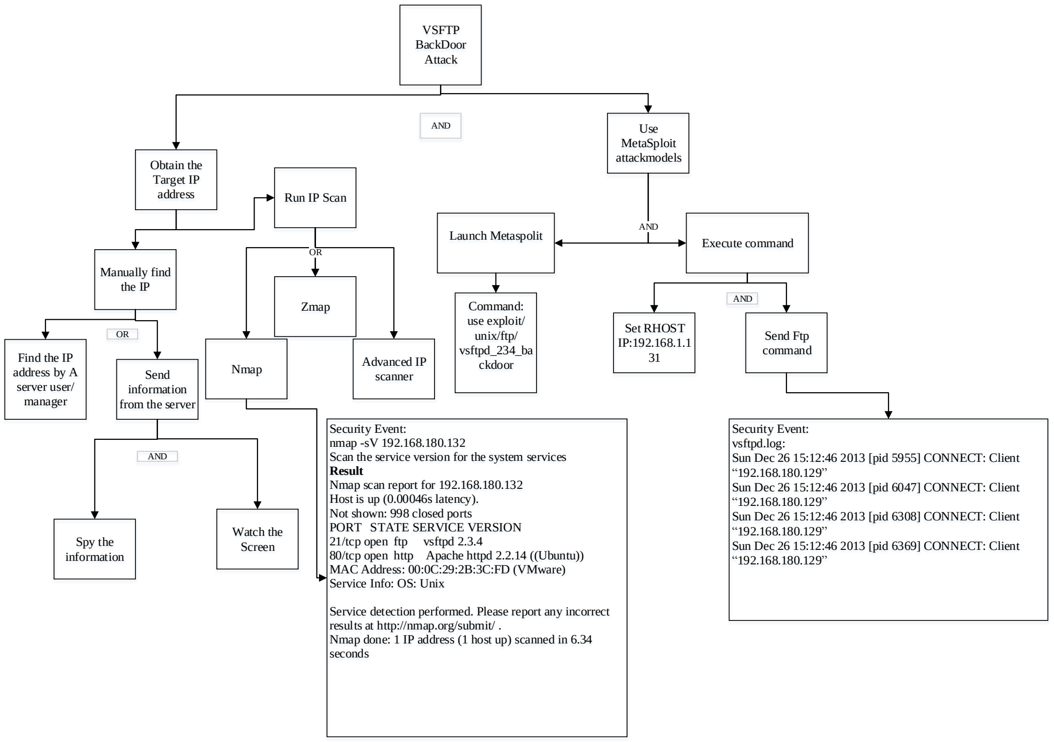
```
Security Event:
nmap -sP 192.168.180.*
scan the ip address for live to
Result
Starting Nmap 6.01 ( http://nmap.org ) at 2013-12-05
20:28 EST
Nmap scan report for 192.168.180.1
Host is up (0.00025s latency).
MAC Address: 00:50:56:C0:00:08 (VMware)
Nmap scan report for 192.168.180.2
Host is up (0.00026s latency).
MAC Address: 00:50:56:FF:80:DF (VMware)
Nmap scan report for 192.168.180.128
Host is up (0.00029s latency).
MAC Address: 00:0C:29:2B:3C:FD (VMware)
Nmap scan report for 192.168.180.129
Host is up.
Nmap scan report for 192.168.180.254
Host is up (0.00031s latency).
MAC Address: 00:50:56:F3:D7:5E (VMware)
Nmap done: 256 IP addresses (5 hosts up) scanned in 3.89
seconds
```

```
Security Event:
access.log:
192.168.129 - - [05/Dec/
2013:20:31:11 – 0000] “HEAD  /
HTTP/1.1”  206 346 “.” “.” (25
packets/second)
Command: top
PID    USER    %CPU   Command
1783  www-data  6.3        Apache2
1784  www-data  6.3        Apache2
```

Fig. 10. Evidence tree of the Attack I

4.2. Evidence of Attack II

To identify and locate fingerprints of Attack II, the vsftpd.log file on the target machine was examined. The Access.log file logs all packets that have been sent to the victim machine. The fingerprints of each operation of Attack II was identified and was then composed into a node, which becomes the child node of the corresponding operation node in the attack tree of Attack II. The resulted evidence tree is shown in Fig. 11. Since most of operations of Attack II are performed on the attacking machine, only two operations, i.e., the Nmap scanning operation and the ftp exploit operation, will leave fingerprints on the target machine. For a successful implementation of Attack II, the victim machine can log the privilege escalation packets that have been sent from the client, as well as IP address of the attack machine, timestamps, etc. The existence of FTP packets sent from the same user, however, cannot be considered as the sensitive fingerprints to identify Attack II, since they could be fingerprints left by a legitimate ftp user. Therefore, more fingerprints of the same operation of Attack II, e.g., the launch of command processor with system user privilege, should also be identified. The combination of the two pieces of fingerprints, i.e., ftp connect from a remote user and the launch of the cmd.exe process by the system user, can be considered as the sensitive fingerprints to identify attack II.


VSFTP BackDoor Attack
AND
Obtain the Target IP address
Use MetaSploit attackmodels
Run IP Scan
OR
Manually find the IP
OR
Find the IP address by A server user/ manager
Send information from the server
AND
Spy the information
Watch the Screen
Zmap
Nmap
Advanced IP scanner
Launch Metaspolit
AND
Execute command
Command: use exploit/ unix/ftp/ vsftpd_234_backdoor
AND
Set RHOST IP:192.168.1.131
Send Ftp command

```
Security Event:
nmap -sV 192.168.180.132
Scan the service version for the system services
Result
Nmap scan report for 192.168.180.132
Host is up (0.00046s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 2.3.4
80/tcp open  http    Apache httpd 2.2.14 ((Ubuntu))
MAC Address: 00:0C:29:2B:3C:FD (VMware)
Service Info: OS: Unix

Service detection performed. Please report any incorrect
results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.34
seconds
```

```
Security Event:
vsftpd.log:
Sun Dec 26 15:12:46 2013 [pid 5955] CONNECT: Client
“192.168.180.129”
Sun Dec 26 15:12:46 2013 [pid 6047] CONNECT: Client
“192.168.180.129”
Sun Dec 26 15:12:46 2013 [pid 6308] CONNECT: Client
“192.168.180.129”
Sun Dec 26 15:12:46 2013 [pid 6369] CONNECT: Client
“192.168.180.129”
```

Fig. 11. Evidence tree of the Attack II.

5. Discussion

5.1. Incident Identification and Reconstruction

To identify security incidents from the large amount of normal logged activities, the key is to find operations as the incident identifier for each incident. In Fig. 11, the identifier of this incident can be (*cmd.exe* ∨ *root user*) ∪ (*client IP connect in* vsflpd.log)). With the evidence tree and sensitive fingerprints of Attack II, an intelligent incident tracking software can detect and reconstruct Attack II in the following way. 1). The tracking software continuously examines the fingerprints located in the system and compare them with the sensitive fingerprints stored in the evidence model repository. 2). Once sensitive fingerprints match the incident identifier of Attack II, Attack II is identified. 3). All fingerprints of Attack II will be identified, located, and retrieved based on the contexture information defined in the evidence tree of Attack II. 4) The retrieved fingerprints will be correlated to reconstruct Attack II.

5.2. Educational Activities Enabled

Many undergraduate and graduate students have been involved in the research partially presented in this paper. The students will conduct security design, programming, testing of various applications installed, as well as ethical hacking and incident investigation in the virtualized Linux environments. All these will provide students with in-depth knowledge and skills in cloud computing and information assurance. More important, research results can be naturally integrated with the existing Computer Information Technology curriculum, which can benefit students in the CIT program at Purdue University Calumet (PUC) and students in the programs that have partnerships with PUC in the Midwest.

6. Conclusion

In this paper, a systematic approach has been proposed to develop the forensics readiness to fight against attacks and inside activities committed in virtualized Linux environments. This approach focuses on identifying, locating, and modeling fingerprints of external and internal attacks. Threat models were developed by using attack tree approach, and then these threat models were mapped to augmented attack trees by adding individual attack operations of corresponding attacks, which resulted in attack trees. In this paper, a total of two modeled attacks, an external attacks and an internal attack, were conducted against a virtualized Linux system. Forensics investigations were followed immediately after each attack, and fingerprints were then identified, collected, and mapped to an evidence tree for the attack. The resulted evidence trees are expected to provide essential information for attack investigation, by answering at least the following three key questions: what information is relevant to the studied attack, where related fingerprint items can be located, and what information each piece of fingerprint can indicate. Also, an evidence tree can provide the contextual information to correlate attack operations by examine the fingerprints they produce. Furthermore, the contextual information provided to an incident tracking software may have the potential of automating attack reconstruction.

References

[1] Biggs, S. and Vidalis, S. (2009). Cloud Computing: The Impact on Digital Forensic Investigations. In Proceeding of *the International Conference on Internet Technology and Secured Transactions*, pp. 1-6.

[2] Biggs, S. and Vidalis, S. (2010). Cloud Computing Storms: *IJICR* **1**(1), pp. 61-68.

[3] Brooks, T., Caicedo, C. and Park, J. (2012). Security challenges and countermeasures for trusted virtualized computing environments. In *Proceedings of World Congress on Internet Security (WorldCIS-2012*). pp. 117-122.

[4] Carrier, B. and Spafford, E. (2003). Getting physical with the digital investigation process. *International Journal of Digital Evidence*, **2**(2). pp. 1-20.

[5] Carrier, B. and Spafford, E.. (2004). An event-based digital forensic investigation framework. In *Proceedings of Digital Forensic Research Workshop.* pp.1-12.

[6] Carpenter, M., Liston, T. and Skoudis, E. D. (2007). Hiding virtualization from attackers and malware*. IEEE Privacy and Security.* Issue May/June. 2007, pp. 62-65.

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

[8] Jeyaraman, S. and Atallah, M. (2006). An empirical study of automatic event reconstruction systems. *Journal of Digital Investigations, Elsevier, 3S*. pp. S108-S115.

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

[10] Menascé, D. A. Virtualization: concepts, applications, and performance modeling. In *Proc. 31th Int. Computer Measurement Group Conf.*, pp. 407-414, 2005.

[11] Moore, A., Cappelli, D. & Trzeciak, R. (2008). The "big picture" of insider IT sabotage across U.S. critical infrastructures. *Advances in Information Security*. **39**, pp. 17-52.

[12] Pearce, M., Zeadally, S., Hunt, R. (2013). Virtualization: issues, security threat, and solutions. *Journal of ACM Computing Survey,* **45**(2), pp. 17:1-17:39.

[13] Poolsapassit, N. and Ray, I. (2007). Investigating computer attacks using attack trees. In *IFIP International Federation for Information Processing*, **242**. *Advanced Digital Forensics III*. pp. 331-343.

[14] Popovsky B. and Fringke, D. (2004). Adding the fourth "R". *In Proceeding of the 2004 IEEE Workshop on Information Assurance.* pp.442-443.

[15] Popovsky, B. E. Frincke, D. and Taylor, C.(2007). A theoretical framework for organizational network forensic readiness. *Journal of Computers*. **2**(3), pp. 1-11.

[16] Ruan, K., Carthy, J., Kechadi, T. and Crosbie, M. (2011). Cloud forensics: An overview. *Advances in Digital Forensics*. **VII**. pp 35-46.

[17] Ruan, K, Carthy, J & Kechadi, T. (2011). Survey on cloud forensics and critical criteria for cloud forensic capability: a preliminary analysis. *Technical Report*. University College Dublin.

[18] Schneier, B. (1999). Attack trees: modeling security threats. *Dr. Dobb's Journal*. [online]. Available: https://www.schneier.com/paper-attacktrees-ddj-ft.html.

[19] Siponen, M. and Oinas-Kukkonen, H. (2007). A review of information security issues and respective research contributions. *Database for Advances in Information Systems*. **38**(1). pp.60-80 .

[20] Swiderski, F. and Snyder, W. (2004). Threat modeling. *Microsoft Professional. Microsoft Press*.

[21] Tan, J. (2001). Forensics readiness. [online]. Available: http://www.arcert.gov.ar/webs/textos/ forensic_readiness.pdf..

[22] Tu, M., Xu, D., Butler, E., and Schwartz, A. (2012). Locating and identifying forensic evidence for attacks against online business information systems by using honeynet. *Journal of Digital Forensics, Security, and Law*. **7**(4), pp. 73- 97.

[23] Valentine, (2007). A. Art of preserving digital evidence. [online]. Available: http://www.onlinebankingreview.com.au/DigitalEvidence.php.

[24] The vsftpd exploit. (2012). [online]. Available: http://pentestlab.wordpress.com/2012/11/08/vsftpd-exploitation/.

[25] 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*. pp. 289-295.