TELKOMNIKA, Vol.11, No.2, June 2013, pp. 399~408
ISSN: 1693-6930
accredited by DGHE (DIKTI), Decree No: 51/Dikti/Kep/2010

Attack and Vulnerability Penetration Testing: FreeBSD

Deris Stiawan[1], Mohd. Yazid Idris[2], Abdul Hanan Abdullah[2]
[1]University of Sriwijaya, Palembang, Indonesia
[2]Universiti Teknologi Malaysia, Johor, Malaysia
e-mail: deris@unsri.ac.id[1], yazid@utm.my, hanan@utm.my[2]

Abstrak

Sistem keamanan komputer telah menjadi perhatian utama selama beberapa tahun terakhir. Serangan, ancaman atau gangguan, terhadap sistem komputer dan jaringan telah menjadi peristiwa yang umum terjadi. Di sisi lain, ada beberapa perangkat sistem dan alat-alat yang tersedia untuk membantu mengatasi ancaman serangan. Saat ini, serangan cyber merupakan sebuah topik penelitian utama dan ini sudah tidak terelakkan. Dalam makalah ini kami menyajikan beberapa langkah untuk dapat menembus ke dalam sistem operasi FreeBSD, beberapa alat dan langkah-langkah baru untuk menyerang digunakan dalam penelitian ini, probe untuk pengintaian, menebak password melalui percobaan berulang-ulang, berusaha untuk mendapatkan akses istimewa dan membanjiri mesin korban untuk mengurangi ketersediaan layanannya. Serangan-seragan ini semua dieksekusi dan dilakukan di dalam jaringan data set yang diberi nama Intrusion Threat Detection Universiti Teknologi Malaysia (ITD UTM). Kami berharap hasil riset ini dapat menjadi acuan bagi praktisi untuk mempersiapkan sistem mereka dari serangan internet.

Kata kunci: *ancaman, dataset, serangan,sistem deteksi secara dini*

Abstract

Computer system security has become a major concern over the past few years. Attacks, threats or intrusions, against computer system and network have become commonplace events. However, there are some system devices and other tools that are available to overcome the threat of these attacks. Currently, cyber attack is a major research and inevitable. This paper presents some steps of penetration in FreeBSD operating system, some tools and new steps to attack used in this experiment, probes for reconnaissance, guessing password via brute force, gaining privilege access and flooding victim machine to decrease availability. All these attacks were executed and infiltrate within the environment of Intrusion Threat Detection Universiti Teknologi Malaysia (ITD UTM) data set. This work is expected to be a reference for practitioners to prepare their systems from Internet attacks.

Keywords*: attack, dataset, intrusion detection system, threat*

1. Introduction

According to CSI/FBI 2011 annual report; it was reported that there are increasing the numbers of types and volume of attacks. These results are similar to the survey conducted by CERT 2011 [1], which conduced serious concern. From the analysis and prediction by [2] [3] [4], there are explosion of security threats in recent years, such as Trojan, virus, worms, adware, spyware and DoS which are continuing to grow, multiply, evolve toward the future in the cyber war. On the other hand, attackers can exploit and penetrate system without the owner's knowledge or consent. For some instance, via implant virus/Trojan in the web and send it in disguise/camouflage technique to valid mail are easy steps to infect the target. In the attacker's perspective, there are some steps and scenario for penetrating the victim, the vulnerability of operating system and actively running application provide the opportunity to be penetrated. Mentioned by [5], there are some favorite operating systems in Internet as a cloud server, such as Windows Server, Linux and FreeBSD. Work performed by [6] [7], they test bed some operating systems and measurement of performance on each.

The purpose of this study is to find the detailed information and the vulnerability to gain full access. This paper focuses on the attack scenario in FreeBSD server. The network environment that was employed in this study is called the Intrusion Threat Detection Universiti Teknologi Malaysia (ITD UTM). ITD UTM [8], deployed with following the standard of DARPA

Received January 15, 2013; Revised March 5, 2013; Accepted March 25, 2013

MIT. Although, this data set still suffers from some of the problems discussed by [9] and may not be a perfect representative of existing real networks. ITD UTM answered the lack of availability of public data sets for network-based IDSs. The problem of availability new data set was described by [10] in 2012.

The reminder of this paper is organized as follows. The network environment, attack scenario, their specification are described in section 2. They way to escalate privilege process in order to gaining the root of user that walking on FreeBSD are described in Section 3. The result and analysis in this experiment are presented in section 4. Finally, a conclusion and suggestion for future work are given in section 5.

2. Attack Scenario

There are attack stages to follow the scenario of this experiment: (1) Gathering information is collecting data to obtain detail information of target, information of operating system, IP Address, network resources, type of hardware, version of hardware firmware and topology is necessary and useful. Reconnaissance is the main focus in this step known as footprinting, (2) Scanning, there are some security holes in any operating system that can be used as an opportunity to get into the system. A map of the different services running on it can be retrieved.

Furthermore, in stage (3) Vulnerability is a hole or weakness of a system that can be explored further. The types of error become vulnerability, such as: boundary condition error, access validation error, input validation error, and failure to handle exceptional conditions. Once a potential system has been identified and information has been gathered, those can be exploited one by one to find the weaknesses. Finally, (4) Penetration, weak point of the system may further penetrate. Penetration steps must be executed carefully and slowly.

In this step, update information are obtained from security community to find out ways to execute and if exploitation is necessary, wherein the process of scanning and penetration carried out gradually and alternately. In Figure 1, the ways the hosts were used to get connected are shown, and its specifications are depicted in Table 1.

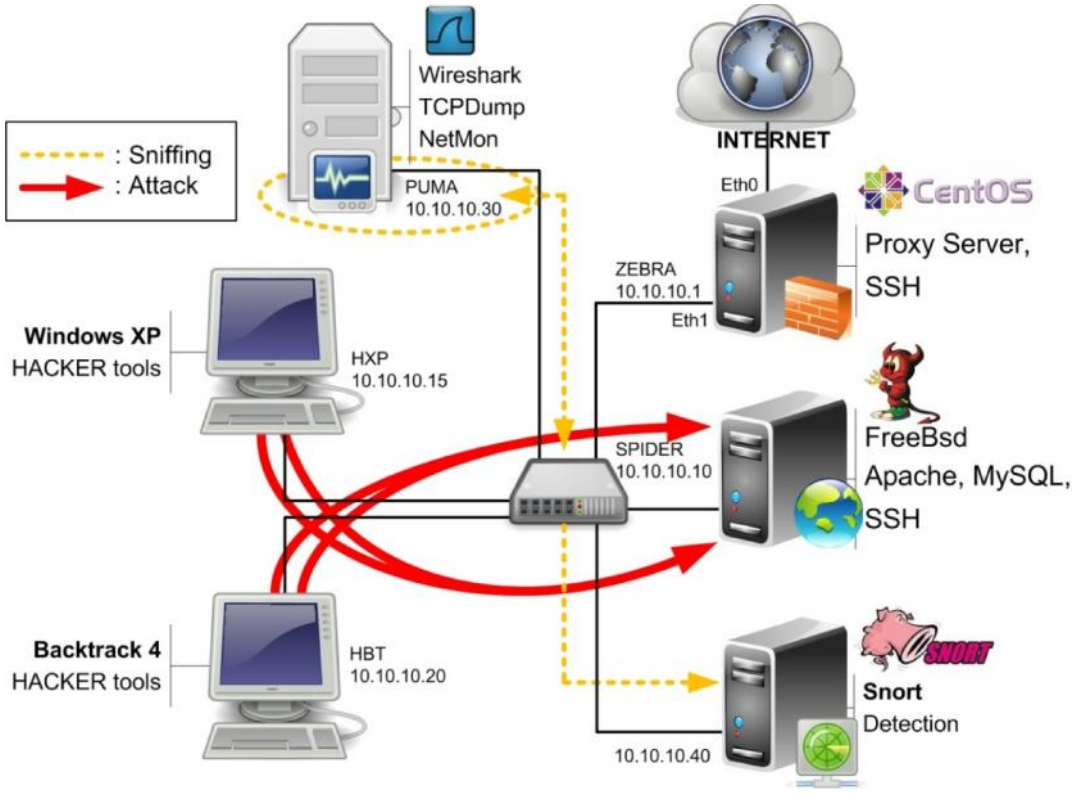


Figure 1. Test bed environment

Table 1. Hardware Specification

Machine Types	Specification	Configurations
CentOS 4.8 IP : 10.10.10.1	Pentium 4 with processor 1 Ghz , RAM 1 GB and two Gigabit Ethernet	IPTables for masquerade NAT as a proxy server. DNS and Mail Server its daemon running well. Configuration : only outgoing network traffic is allowed through the proxy (normal access is performing the Web 2.0 activity)
FreeBSD 7.2 IP : 10.10.10.10	Intel® Pentium 4 with processor 2.0 Ghz , RAM 1 GB.	Installed of Apache, MySQL, PHP, and SSHD server
Fedora Core 14 IP : 10.10.10.30	Intel® Xeon CPU 2.0 Ghz Processor, 2 GB RAM , 1 TB Hardsik	TCPDump and Wireshark for Network sniffing develop with network management, also packet generator. Configuration : tcpdump -w eth0, iptraff eth0
Snort IP : 10.10.10.40	Intel® Pentium 4 with processor 1 Ghz, RAM 1 GB.	Running Snort 2.8.5.2 (Build 121), PCRE ver 8.12 2011-01-15. Configuration : alert of threat/attack
Backtrack 4 IP : 10.10.10.20 Windows XP SP3 IP : 10.10.10.15	Intel® Pentium 4 with processor 2.8 Ghz , RAM 1 GB.	Running and execute some script/ application Configuration: attack tools: probes, DoS, Man in the middle, poison, buffer overflow, Rootkit & Trojan, Password guessing.

Table 1 shows detailed specification and configuration of host in Figure 1. There are several applications daemons that ran and installed in FreeBSD, such as the Apache, MySQL and PHP for running web server, and SSHD for secure communicate server and client via SSH.

Meanwhile, the penetration procedure is conducted and illustrated further below:

Step 1 : Probes

- Attackers trying to network mapping the victim machine via Hping2 and Xprobe2
- Attackers machine 10.10.10.15 via Nessus start the reconnaissance the host 10.10.10.10
- Attackers scanning the victim machine via NStealth
- Attackers scanning the victim machine via Nmap
- Attackers scanning the HTTP Reconnaissance via Nikto
- Attacker scanning the network 10.10.10.10 via attack machine 10.10.10.15 used GFILanGuard
- The attack machine 10.10.10.20 probes the web server of the target via HTTPrint
- Attackers find open port to potential penetration, Port 22 (SSH), 80 (HTTP) and 3306 (MySQL)
- Attackers try various guesses to seek exploits

Step 2 : Brute Force

- Attackers attempts attack the host 10.10.10.10 by SSH brute-force
- Attackers change the dictionary password to guessing password
- Attackers failed the pass

Step 3 : Escalating Privilege

- Attackers try to enter via login user “admin”
- Attacker attempted remote exploitation
- The alert numbers of attempts “sensepost.exe”, “c99shell.php” command shell attempt

	- The alert numbers of attempts “/wwwboard/passwd.txt access”, “/cgi-bin/ access”, “/cgi-bin/ls access”, “cmd.exe access”, “/etc/passwd” , “/~root acces”, “/etc/shadow access” - Attackers attempted to implant malware and create backdoor - Attackers login and exploit to escalate privileges
Step 4 : DoS	- Attackers attempts against the network via TCP/UDP flooding - Attackers launch ICMP flooding - Attackers attempts sending TCP SYN via Trinoo - Attackers flooding packets using forged source addresses

2.1. Collecting Data

Data collection is one of the most important steps in designing intrusion detection/prevention system as it will affect the whole design, implementation and result of process. In this experiment, there are some stages are conducted; (1) two weeks for installation of each server with daemon application, (2) one week, gathering information and scanning. (3) three weeks of doing collecting data to find vulnerability from security of community, and (4) five weeks spare time is taken to attempt and penetration the systems. There are some differences in the results obtained in the first and second data collection. The first data are collected directly on the server, regardless of the network broadcast. Conversely, in the second data are collected using Hub terminal that also captured the broadcast network

In this case, TCPdump is used to sniff the real-traffic. It uses the libcap library to capture packets and has ability to consider the properties of an ideal as a packet sniffer. TCPdump working to execute in command line, the command is *`tcpdump –w nameofile.pcap -i eth0`* for write pcap file. On the other hand, for Identifying and recognising threat, Snort is used. It can perform protocol analysis, content matching, capable to configure as a sniffer, packet logger and network detector. Snort produce alert to identify threat, below is a sample of alert derived from attack 10.10.10.10.

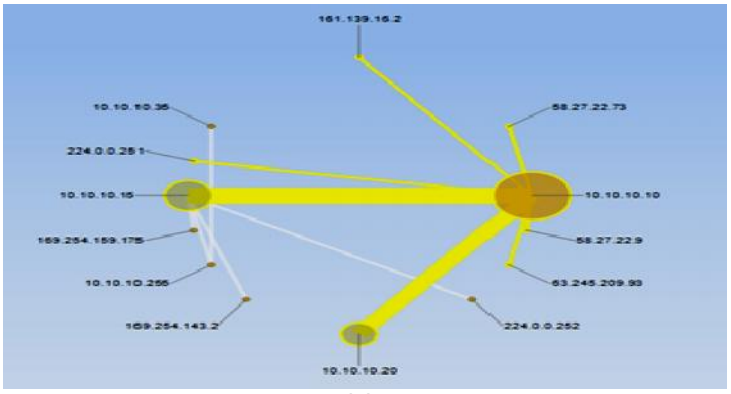
```
[**] [1:621:7] SCAN FIN [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-09:41:03.957453 10.10.10.20:45115 -> 10.10.10.10:99
TCP TTL:37 TOS:0x0 ID:32240 IpLen:20 DgmLen:40
*******F Seq: 0x7E4B11E5  Ack: 0x0  Win: 0x800  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS27]

[**] [1:249:8] DDOS mstream client to handler [**]
[Classification: Attempted Denial of Service] [Priority: 2]
11/18-09:29:49.299721 10.10.10.15:60123 -> 10.10.10.10:15104
TCP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:48 DF
******S* Seq: 0x155B2B19  Ack: 0x0  Win: 0x1000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2000-0138][Xref
=> http://www.whitehats.com/info/IDS111]
```

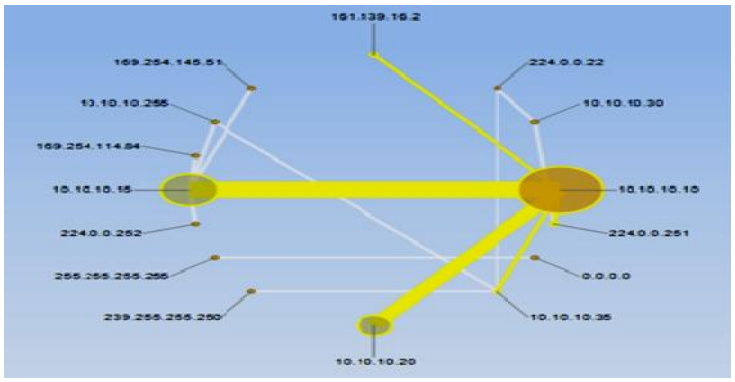
3. Experiment

In this section, several experiments that have been carried out were described. Figure 2 shows pattern visualisation of probes and penetration stages of FreeBSD, this pcap was compiled by the Cascade pilot software [11] . This tool has been introduced previously by [12], [13] and [14] they used it to captured, compare and analyze online traffic.

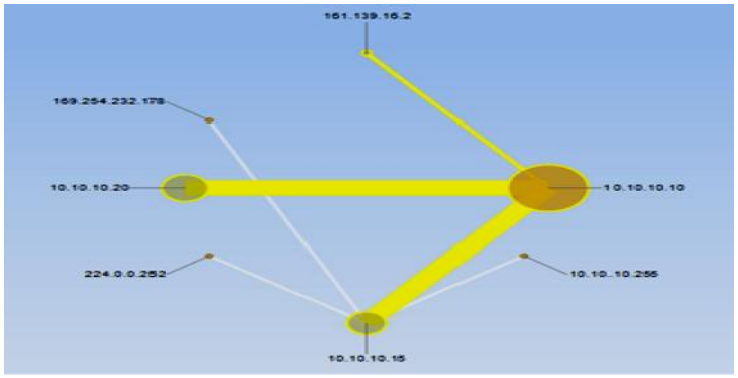
Figure 2 shows pattern of probes for reconnaissance and penetration attack to the victim’s machine. Meanwhile, scanning tools such as Nessus, Nstealth and GFILanGuard several times to connect to their servers, it is necessary to ensure and compare if there any updates of existing vulnerabilities in its database, as shown in Figure 2 (a) - (b).



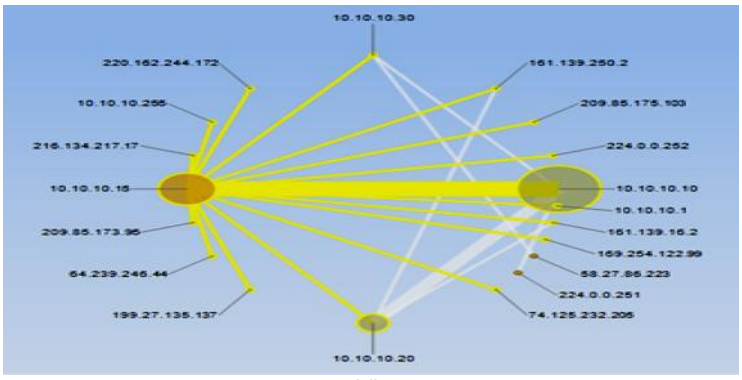
(a)



(b)



(c)



(d)

Figure 2. Scan/probes (a) Free BSD 1st run and (b) 2nd run test,
(c) Attack penetration Free BSD 1st run and (b) 2nd run test

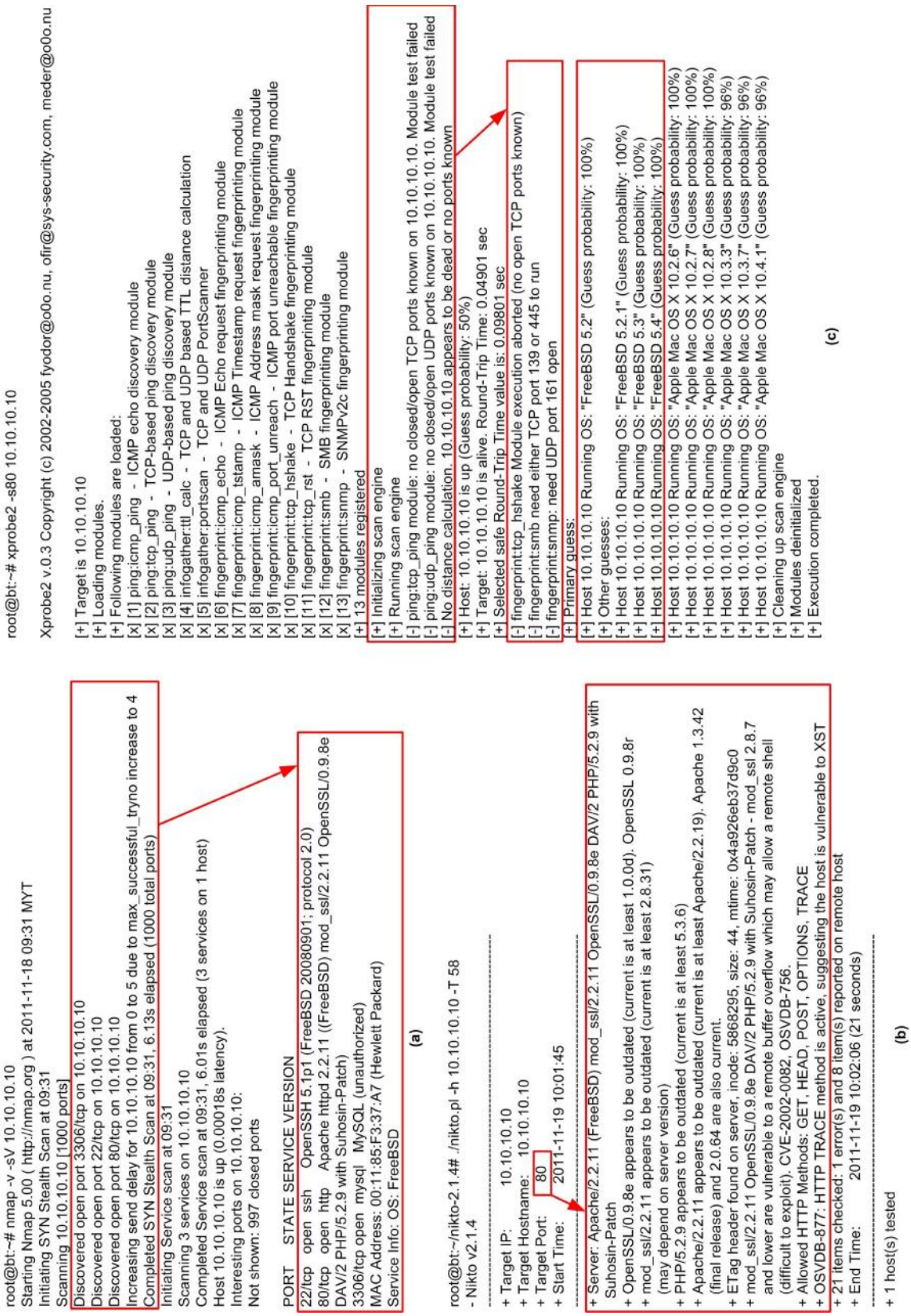


Figure 3. Sample probes stages, (a) Nmap, (b) Nikto and (c) Xprobes

3.1. Probes

In this stage, several tools and scenarios to gather information and findings known as vulnerabilities are mixed and combined the right tools to get the expected results. Some of the measures were adopted to enable these tools can complement each other.

Probing FreeBSD produces only little information and very limited, although using multiple scanning methods and performed a total of two times but the results obtained remain the same. This operating system is highly selective to provide additional information. Multiple requests some tools are ignored from the Hping2 and Xprobe2 command. Figure 3 shows the results of probes from Nmap, Nikto and Xprobes.

3.2. Vulnerability

In the previous step, there are some information obtained to confirming active service with its port addressing. In the attacker's perspective, vulnerability is an opportunity to exploit. This section describes ways to find and collecting hole from vulnerability database: CVE for update information and knowledge. In creating and grouping information, CVE assisted and supported by a few vendors and other security communities. It opens to any security product or service to allow it to cross-link and be associated with the CVE-compatible with other products. CVE is a one-off vulnerability database as a collection of records containing technical descriptions of vulnerabilities in computer systems. Performed work by [15] confirmed the security information provider which tracks the new vulnerabilities and publishes alerts to a wide community of subscribers.

According to [16], the provider or vendor need time to make a patch release after exploit is found. Consequently, there was delay in time between an exploit release with patch and signature release. In this experiment conducted, there are some vulnerable of FreeBSD, as shown in Table 2.

Table 2. Active/running daemons on FreeBSD

Port	Protocol	Services	Information	CVE
22	TCP	SSH	Remote login protocol	CVE 2011-0539, CVE-2007-1365
80	TCP	HTTP	Running of World Wide Web HTTP Services	CVE-2009-0037
3306	TCP	MySQL	Running on MySQL databases	CVE-2009-0819, CVE-2004-0628
5353	UDP	MDNS	Multicast DNS protocol, in this port running the Zeroconf (zero configuration networking) and DNSExt (DNS Extension).	CVE-2009-0758
55081	TCP	Unknown	Open	-

1. CVE 2011-0539, CVE-2007-1365 effect buffer over flow : Allows remote attackers to execute arbitrary code via fragmented
2. CVE-2009-0037 : Vulnerability Curl & Libcurl, which resulted in the attacker can allows (1) trigger arbitrary requests to intranet servers, (2) read or overwrite arbitrary files via a redirect to a file: URL, or (3) execute arbitrary commands via a redirect to an scp: URL.
3. CVE-2009-0819, CVE-2004-0628 : the attacker can allow remote to cause a denial of service (crash) and possibly execute arbitrary code via a long scramble string
4. CVE-2009-0758, is multicast packet storm, this attack can allow remote attackers to cause a network bandwidth and CPU consumption) via a crafted legacy unicast mDNS query packet that triggers a multicast packet storm.

Main results of these stages are (i) analyzing detailed information from the probes process, (ii) to find and collecting hole from vulnerability data source. The detailed analysis of vulnerability will increase the success of the penetration step.

3.3. Penetration

During the tests performed, FreeBSD is stable operating system that makes the target powerful and hard to be penetrated, as seen from the number of attempts that failed. The stages of penetration are follows.

1. Starting reconnaissance via probes. Unfortunately, this operating system is very selective to provide additional information. Multiple requests some tools are ignored from the Hping2 and Xprobe2 command.
2. Attempt to guess passwords with an account users; administrator/admin/root. Unfortunately, the experiment to guessing the password does not work well, even more than that the system are closed and disconnect the connection after failing to guess three times which is a default security configuration. This stage, length of password characters is very important.
3. Moreover, longer characters usually cannot be cracked in a reasonable time. Brute force attack on password longer than five characters is rarely successful. Hydra, Medusa, Brutus and BruteSSH are a tool that is used to try a brute force attack, and all failed. Experiment conducted with Brutus and hydra failed to execute the task, also Medusa as are to be several times in order to guess the password with dictionary attacks. Even though the configuration of the operating system is allowed to default.
4. Finally, the system response delay value slightly up compared to prior to this attack. Expect the attacker, who attacked the system less responsive to service requests from the user, or even system crashes or is expected collapse. DoS attack experiments conducted simultaneously from two attacker's machine within an hour. These attacks can be anticipated, by limiting access to reversal handshake of the UDP and ICMP. The attacker hopes the system will crash, or may simply by unable to perform ordinary functions.

4. Results and Analysis

As well as with the experimental poisoning and sniffing data traffic from the server to the network has failed due to the toughness of this operating system. This operating system has a unique architecture thus not all hacking tools are able to work well.

This operating system is tough, very selectively provide information when interrogated and of the number of attempts to only probes, password guessing, web injection and DoS were successfully performed, while the rest, implant malware, rooting, backdoor, and man in the middle attack is not successful. Figure 4 shows results from this experiment conducted, (a) total packet and timing packet history on server, (b) utilization of protocol used, (c) top 10 protocol running and (d) visualization of utilization to shown the DoS attack.

5. Conclusions and Future Work

It is seen that penetration tests are a useful measure to check the reliability of network infrastructure. This paper presented penetration of FreeBSD operating system. The penetration was executed from different sides, used some tools and new mechanism of attack. On the other hand, TCPDump produces raw data, there are several issues to solve in future work, such as: (i) how to extract the data to analyzed, (ii) how to test the validity of data, (iii) how to classify the threat and normal access, and (iv) compares FreeBSD with other operating systems.

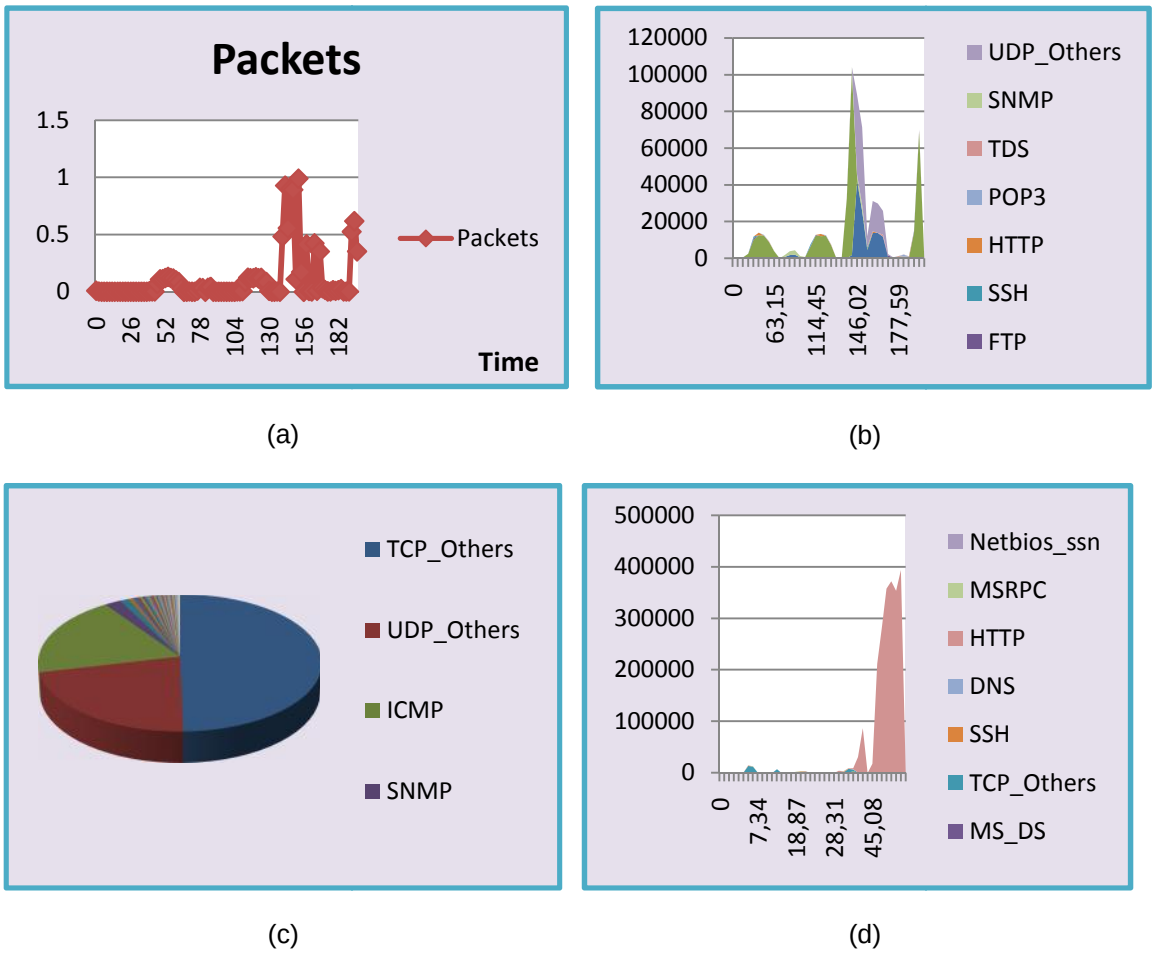


Figure 4. (a) Packet history of attack on FreeBSD, (b) Utilization of protocol,(c) Shown of top 10 protocol used and (d) Utilization of FreeBSD

References

[1] CERT-IST. Cert-IST 2011 annual review regarding flaws and attacks. 2012.
[2] G Kenneth. Cyber Weapons Convention. *Computer Law & Security Review*. 2010; 26: 547-551.
[3] S Mansfield-Devine. DDoS: threats and mitigation. *Network Security*. 2011; 5-12.
[4] S David. The state of network security. *Network Security*. 2012; 14-20.
[5] S Lakka, *et al.* Competitive dynamics in the operating systems market: Modeling and policy implications. *Technological Forecasting and Social Change*. 2013; 80: 88-105.
[6] V Visoottiviseth, N Bureenok. Performance Comparison of ISATAP Implementations on FreeBSD, RedHat, and Windows 2003. 2008: 547-552.
[7] A Alhomoud, *et al.* Performance Evaluation Study of Intrusion Detection Systems. *Procedia Computer Science*. 2011; 5: 173-180.
[8] D Stiawan, *et al.* (2012, *Intrusion & Threat Detection Universiti Teknologi Malaysia Dataset*. Available: http://pcrg-utm.org/dataset/.
[9] J McHugh. Testing Intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. *ACM Transactions on Information and System Security*. 2000; 3: 262-294.
[10] CA Catania, CG Garino. Automatic network intrusion detection: Current techniques and open issues. *Computers and Electrical Engineering*. 2012; 38: 1062-1072.
[11] CP software. 2012, *Riverbed® Cascade® Pilot software*. Available: http://www.riverbed.com/us/products/cascade/cascade_pilot.php.
[12] N Hubballi, *et al.* An Active Intrusion Detection System for LAN Specific Attacks. In: T Kim, H Adeli. Editors. *Advances in Computer Science and Information Technology*. vol. 6059. Heidelberg: Springer Berlin; 2010: 129-142.

[13] N Hubballi*, et al.* LAN attack detection using Discrete Event Systems. *ISA Transactions*. 2011; 50: 119-130.
[14] Martin Zaefferer*, et al.* Intrusion Detection: Case Study. Master of Engineering Automation and IT, Faculty for Informatics and Engineering, University of Applied Sciences Cologne. Gummersbach. 2012.
[15] N Mansourov, D Campara. Chapter 6 - Knowledge of vulnerabilities as an element of cybersecurity argument. 2011: 147-170.
[16] H Gascon*, et al.* Analysis of update delays in signature-based network intrusion detection systems. 2011; 30: 613–624.