Design and Implementation of Honeypot Systems Based on Open-Source Software

Chao-Hsi Yeh and Chung-Huang Yang
Graduate Institute of Information and Computer Education
National Kaohsiung Normal University, Taiwan, R.O.C.
Stanley5402@gmail.com
chyang@computer.org

***Abstract*—A honeypot is a type of information system that is used to obtain information on intruders in a network. When a honeypot is deployed in front of a firewall, it can serve as an early warning system. When deployed behind the firewall, it can serve as part of a defense-in-depth system and can be used to detect attackers who bypass the firewall and the intrusion detection system (IDS) or threats from insiders. Honeyd is an open-source honeypot; however, it uses a command-line interface and its configuration is difficult for beginners. The purpose of this study is to use the open-source tool to construct a graphic user interface (GUI) for honeyd. For the sake of portability and easy deployment, the whole system will be installed in a live USB stick. The end user can create a honeyd template by using the GUI or the result of the Nmap scan of a target computer. Moreover, the system will provide a log-review interface and real-time SMS functionality. Finally, we deployed the designed system in a campus network and presented an analytic result of a 60-day period with a Web-based data analysis system.**

I. Introduction

Network administrators usually use a firewall and an intrusion detection system (IDS) to protect their network. The firewall can control the inbound and outbound traffic according to the type of service requested, the user name, and the IP address of packets. The IDS can be deployed between the local area network and the Internet or any other important gateway for detecting suspicious packets [1]. However, sometimes administrators might forget to update the firewall rules. Moreover, the IDS system that uses anomaly detection has a high false-positive ratio [2]. The use of a honeypot can overcome the inherent deficiencies of the IDS and firewall. More importantly, we can treat it as a platform for security education in a university [3]. If a honeypot is deployed in front of a firewall, it can be treated as an early-warning system. If we deploy it behind the firewall, it can serve as part of a defense-in-depth system and can be used to detect attackers who bypass the firewall and IDS or threats from insiders [4].

II. System Architecture

The architecture of the honeypot system is shown in Fig. 1. The primary module can be divided into the honeyd, honeyd management console, template generation module, IP binding module, alert module, IP trace module and data analysis module.

- Honeyd

Honeyd [5] is an open-source low-interaction honeypot system that was developed by Dr. Niels Provos in April 2002. Honeyd can successfully emulate the IP stack of various operating systems according to the Nmap and Xprobe fingerprint files.

- Honeyd management console

Using this module, users can define various paths and parameters. Moreover, users can run Nmap to scan a target computer through the GUI. The results of the scan can be used by a parser to extract the type of OS and open ports of the system. Then, users can select a similar type of OS to create a honeyd template on the basis of a comparison with the Nmap fingerprint database.

- Template generation module

Users can use this module to select the type of OS from the Nmap fingerprint database and set up the corresponding statuses and services for TCP, UDP, and ICMP ports.

- IP binding module

It is used to bind a honeyd template with an IP address. A single honeyd configuration file can include several templates that are already bound with an IP address.

- Alert module

When the system boots, it initiates a process to monitor the snort alert file; when there is a change in the size of this file, the system will send a message to the cell phone of the system administrator.

- IP trace module

This module is based on InetAddressLocator [6]—a modified open-source Java-based software program. It uses a local copy of the WHOIS database to perform fast and accurate lookups of country codes.

- Data analysis module

The Data analysis module is written in C language and users can query the analytic result through a Web interface which is constructed with ZK [7].

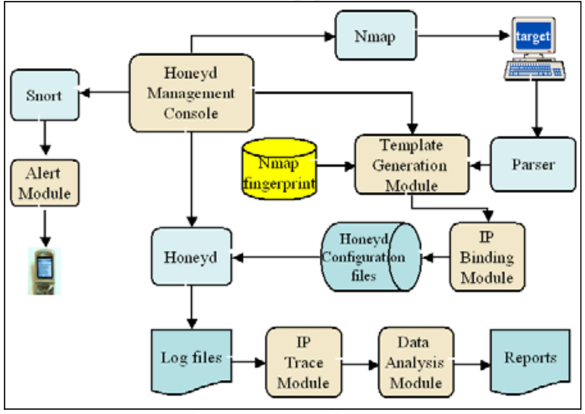


Fig. 1.System Architecture

ISI 2008, June 17-20, 2008, Taipei, Taiwan
1-4244-2415-3/08/$20.00 ©2008 IEEE

III. Implementation

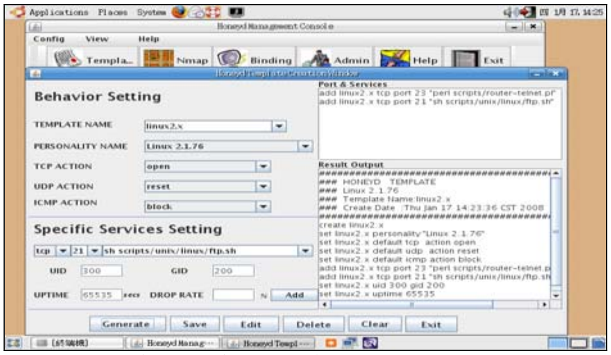
The system is implemented using Eclipse and Jigloo. Eclipse is an open-source project; the latest version can be found at [8]. Jigloo is an Eclipse plug-in that enables us to build sophisticated GUIs rapidly.

A. *Live USB*

A live USB is implemented using Ubuntu 7.04. Ubuntu can be easily and directly installed through a GUI. More importantly, it provides a very strong security control service. Finally, we installed it in a 4GB USB stick; the installation procedures can be found at [9].

B. *Operation*

Fig.2 shows the creation of a template through the GUI. When "Template" is selected from the honeyd console, the template generation screen will appear, users can select the desired behavior and services from the combo box and click the "Generate" button.



Behavior Setting

TEMPLATE NAME: linux2.x
PERSONALITY NAME: Linux 2.1.76
TCP ACTION: open
UDP ACTION: reset
ICMP ACTION: block

Specific Services Setting

tcp 21 sh scripts/unix/linux/ftp.sh
UID 300 GID 200
UPTIME 65535 secs DROP RATE % Add

Port & Services

```
add linux2.x tcp port 23 "perl scripts/router-telnet.pl"
add linux2.x tcp port 21 "sh scripts/unix/linux/ftp.sh"
```

Result Output

```
##########################################
###  HONEYD  TEMPLATE
###  Linux 2.1.76
###  Template Name:linux2.x
###  Create Date  :Thu Jan 17 14:23:36 CST 2008
##########################################
create linux2.x
set linux2.x personality "Linux 2.1.76"
set linux2.x default tcp  action open
set linux2.x default udp  action reset
set linux2.x default icmp action block
add linux2.x tcp port 23 "perl scripts/router-telnet.p
add linux2.x tcp port 21 "sh scripts/unix/linux/ftp.sh
set linux2.x uid 300 gid 200
set linux2.x uptime 65535
```

Fig.2 Template generation screen

C. *System Deployment*

In order to verify the effectiveness of the system and obtain information about attacks, we deployed the designed system in a campus network. This experiment used honeyd to simulate two virtual honeypots—a Linux 2.x system and a Solaris 2.x system.The Linux honeypot was created by using the GUI and running simulated services (e.g., telnet, ftp). The Solaris honeypot was generated from the result of the Nmap scan.

D. *Data Analysis*

The Linux honeypot logged 7814 attacks on the TCP connection over a 60-day period at an average of 5.4 attacks per hour. Over the same period, the Solaris honeypot logged 10732 attacks on the TCP connection, averaging 7.5 attacks per hour.

If we sort the origins of the attacks by "Campus Network" and "Internet," the numbers of attacks on the TCP connection are 9999 and 8547, respectively. Fig.3 lists the top 10 attacked ports.

According to the data in Fig.3, the most attacked port is 445.TCP port 445 (CIFS) of the Windows system is designed to access sensitive or un-encrypted data. Hence, it has always been a target for intruders or worms to establish a connection for initiating an attack. Finally, Table 1 and Table 2 list the analytical results of ISP and location.

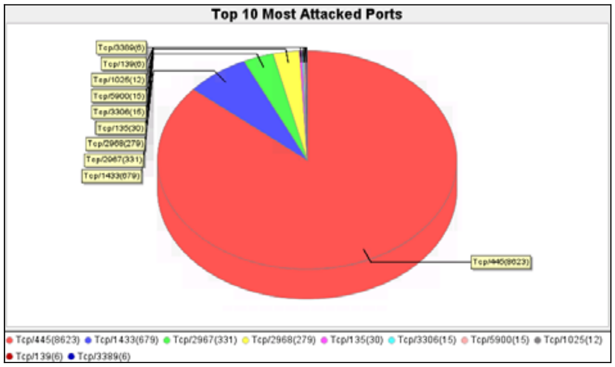
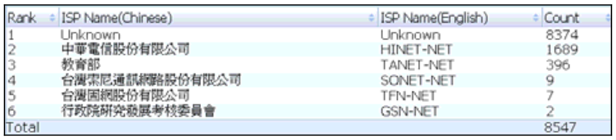


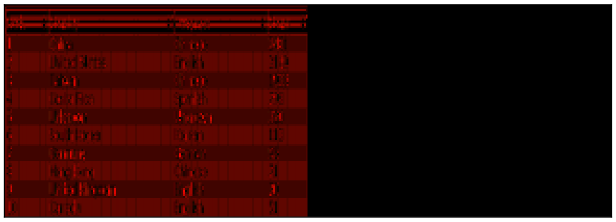
Fig.3 Attacks from campus network

TABLE 1.Isp Analysis Of Internet Attacks



Rank	ISP Name(Chinese)	ISP Name(English)	Count
1	Unknown	Unknown	8374
2	中華電信股份有限公司	HINET-NET	1689
3	教育部	TANET-NET	396
4	台灣索尼通訊網路股份有限公司	SONET-NET	9
5	台灣固網股份有限公司	TFN-NET	7
6	行政院研究發展考核委員會	GSN-NET	2
Total			8547

TABLE 2 Location Analysis Of Internet Attacks



[illegible]	[illegible]	[illegible]	[illegible]
1	China	Chinese	[illegible]
2	United States	English	[illegible]
3	Taiwan	Chinese	[illegible]
4	Costa Rica	Spanish	[illegible]
5	Ukraine	Russian	[illegible]
6	South Korea	Korean	[illegible]
7	Germany	German	[illegible]
8	Hong Kong	Chinese	[illegible]
9	United Kingdom	English	[illegible]
10	Canada	English	[illegible]

IV. Conclusion

In this study, we have developed a GUI that incorporated with the network exploration or security auditing utility—Nmap. We utilized either its OS fingerprinting and TCP port-scanning capability from its scanning results or a combo box and a text field to create a honeyd template. Besides, We also utilized ZK to construct a Web-based interface which can let users query the analytic result through browser. Finally, we installed the system in a portable live USB stick. The honeypot system can be quickly deployed by a System administrator, even with limited knowledge of it.

Reference

[1] J. G. Levine, J. B. Grizzard, and H. L. Owen, "Using honeynets to protect large enterprise networks," *Security & Privacy Magazine, IEEE,* vol. 2, pp. 73-75, 2004.

[2] R. A. Kemmerer and G. Vigna, "Intrusion detection: a brief history and overview," *Computer,* vol. 35, pp. 27-30, 2002.

[3] Lanoy, A., and Romney, G.W.: "A Virtual Honey Net as a Teaching Resource", Information Technology Based Higher Education and Training, 2006. ITHET'06. 7th International Conference on, 2006, pp. 666-669

[4] L. Spitzner, "Honeytokens: The Other Honeypot.," in *Internet: http://www.Securityfocus. com/infocus/1713*, 2003.

[5] Honeyd, http://www.honeyd.org/, 2008.

[6] InetAddressLocator, http://sourceforge.net/projects/ javainetlocator/, 2008

[7] ZK, http://www.zkoss.org/, 2008.

[8] Eclipse, http://www.eclipse.org/downloads/, 2008

[9] Billypan, "Billypan's Blog," http://www.wretch.cc/blog/ billypan101 /, 2007.