Implementation of a modern security systems honeypot Honey Network on wireless networks

Conference Paper · May 2017

DOI: 10.1109/YEF-ECE.2017.7935647

4 authors:

Hibatul Wafi
Syarif Hidayatullah State Islamic University Jakarta
1 PUBLICATION **2** CITATIONS
SEE PROFILE

Andrew Fiade
Syarif Hidayatullah State Islamic University Jakarta
7 PUBLICATIONS **5** CITATIONS
SEE PROFILE

Nashrul Hakiem
Syarif Hidayatullah State Islamic University Jakarta
39 PUBLICATIONS **57** CITATIONS
SEE PROFILE

Rizal Broer Bahaweres
National Research University Higher School of Economics
46 PUBLICATIONS **62** CITATIONS
SEE PROFILE

Implementation of a Modern Security Systems Honeypot Honey Network on Wireless Networks

Hibatul Wafi, Andrew Fiade, Nashrul Hakiem, Rizal Broer Bahaweres
Deptartment of Informatics, Faculty of Science and Technology
UIN Syarif Hidayatullah Jakarta, Indonesia
hibatulwafi3@gmail.com, {andrew_fiade, hakiem, rizalbroer}@uinjkt.ac.id

***Abstract* - Today's Internet technology is not free from problems. Unfortunately, it is used by people who have no right to steal important data. The National ICT Training and Research Center (PUSTIKNAS) Ministry of Communication and Information Technology the Republic of Indonesia is one of the government agencies that often suffer experimental intrusion by outsiders. One solution is to implement a honeypot system that can be used to improve the security of the Internet network and can detect intrusion attempts or attacks. A honeypot system with a Modern Honey Network (MHN) has been developed to make deployment and management easier and to make the honeypot more secure. The honeypots used in this study are the Kippo, Glastopf, Dionaea honeypots. With testing of the honeypot system in the form of an intrusive attack, it can be concluded that the honeypot system managed to outwit the attacker by opening ports on a server that turned the port into a hoax and was cleared. In addition, the suspicious activity was recorded and the attacks detected.**

Keywords - Honeypot, Modern Honey Network, Kippo honeypot, Glastopf honeypot, Dionaea honeypot

I. INTRODUCTION

Technology is something that is a rapidly growing entity and arguably already at a stage of concern. Along with the rapid development of these technologies, the greater is the threat of interference in the performance of these technologies. One of these is Internet technology, which currently cannot be separated from the many problems or security holes faced today. There are many instances where such security holes have been exploited by unauthorised persons to steal important data. Even a beginner can easily penetrate into a system using the existing available attack tools. The attacks occur because the party that was attacked often does not realise the importance of network security to be applied to their systems.

The technology used to ward off such attacks has actually been around for some time, but its development has reached the limit. The use of such technology is now inefficient because it cannot guarantee the accuracy of its security. One of the techniques is Intrusion Detection Systems (IDS) used to detect attacks. However, IDS itself has drawbacks, namely when the traffic in the network is very high, then the system will find it difficult to tell which packets are normal and which packets are an anomaly. In addition, IDS capability is limited to knowing the incoming attacks in the form of alerts, in the absence of follow-up [1].

National ICT Training and Research Center (PUSTIKNAS) is a government agency under the supervision of the Ministry of Communications and Information Technology. In conducting network security against infiltration attempts by unauthorised persons, the National ICT Centre still uses IDS. According to the result of interviews between the authors and one of the Network Administrators, IDS has begun to be ineffective and inefficient. Traffic is dense and a lot of data occurs in large amounts that must be processed by the network IDS which results in PUSTIKNAS often not detecting network attacks against this institution, especially if the Network Administrator of the National ICT Centre does not check the incoming network packets carefully. The amount of data processed in the PUSTIKNAS network is also not small, so the IDS is used to process the data which leads to the activity on the network being disrupted. Therefore, a solution is needed that can increase the level of system flexibility of the PUSTIKNAS network security, by building a honeypot security system that can be used to secure the network and to equip a Network Security Management system [2].

A honeypot allows for follow up on an incoming attack by acting as a decoy in a computer network in order to trick the attacker and also to collect malware. The system is made to appear exactly the same as the main system to be protected, so that when there is a penetration then it would seem to the attacker that he/she has managed to get into the system, when in fact the attacker has entered a trap [3].

II. BASIC THEORY

A. Honeypot

A honeypot is a source of information that is usually designed with the aim of detecting and trapping any attempt to penetrate into an experimental system [4]. A system consisting of

978-1-5090-4639-3/17/$31.00 ©2017 IEEE

several honeypots is called a honeynet. If the attacker breaks into the system or server, then the honeypot that resembles the original server will be assaulted by the attack, while the actual system remains safe and untouched as a server behind the honeypot. For those who are not experienced attackers, they tend to think that they have easily managed to hack the system / server. However, all actions, tools, and techniques used in the attack have been recorded for study by the System Administrator (Sysadmin) concerned through the data and information presented by the honeypot [5].

The role of the honeypot is not solving a problem faced by the server, but it contributes in terms of overall security based on the level of involvement. The level of involvement measures the degree of interaction of an attacker with the information system which consists of three types, namely a low-involvement honeypot, medium- and high-involvement honeypot.

The low-involvement honeypot is the most easily installed and maintained because of the simple design and basic functionality. A medium-involvement honeypot provides greater interaction capabilities when compared with the low-involvement honeypot but its functionality is still less than the high-involvement honeypot.

A high-involvement Honeypot uses honeyspot technology (a mixture of 'honeypot' and WiFi 'hotspot') that provides a lot of information about the attackers but it takes time to obtain it. The purpose of the high-involvement honeyspot is to provide access to a real operating system to attack where there is no limit set [6]. The following is the process of how honeypots can capture the attack as a sequence of communication networks between the attacker and the victim.

The most important issue is to configure a network honeypot-honeypot in large organisations by using programs based on the current server and a network scanner. The solution to the challenge of cost-effectiveness in analysing the volume of important data for the large amounts is by applying an aggregation process that integrates current and a data network honeypot. This solution is integrated into a complete framework to facilitate deployment of the honeypot and to analyse the attacks. Software solutions that differ from this framework are shown in Figure 1 [7].

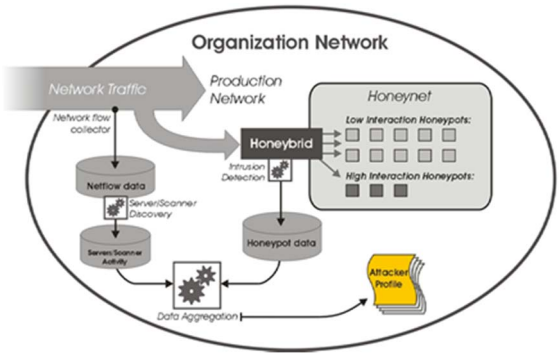

Fig 1. Honeypot Usage

Packets received by a network are searched for any suspicious activity. If the packet is not suspicious, then the packet is processed and enters the goto production network, which generates a response packet from the server. If the packet is suspicious, then the packet is divided into two stages. The first packet is collected so as to form a flow of data from a network and is processed by the honeypot servers. The packet is processed by a honeybrid in which there is a honeynet (a collection of honeypots). The incoming packet is processed and considered as an intrusion.

Packets that have been processed by the honeypot server are considered as a suspicious activity and are then collected. Once processed by the honeypot, the packet will be incorporated into the data honeypot. Packs of honeypot data are processed into a set of data that has been processed from the honeypot-honeypot server, synchronised and the data considered as an assault.

B. Modern Honey Network (MHN)

A Modern Honey Network (MHN) is a management system that allows multiple sensors in a honeypot-honeypot such as: Kippo, Dionaea, Glastopf and others to create a network of active defence and function as a whole within a few minutes.

MHN uses a set of sensors to collect data related to the attack on the network. MHN analyses the attack and the attack parameters are mapped into a World Map view while maintaining the large amount of information about the attack, which makes it very visual and intuitive [8].

Honeypots are not yet considered as an overall defence, mainly because of the complicated process management that is provided for the security of a company. MHN is an open source project to create "Active-Defence" that make more use of honeypots and is widely used for enterprise security [9].

C. Kippo Honeypot

Kippo is a medium-medium involvement Secure Shell (SSH) honeypot designed to log attacks and, most importantly, the whole of the interaction performed by the attacker.

978-1-5090-4639-3/17/$31.00 ©2017 IEEE

Kippo is a honeypot that emulates the SSH service. Emulation involves the formation of a SSH session and, in the case of successful authorisation of users, also the shell interaction with the user. The honeypot can be used to register violent attacks aimed at obtaining user passwords with regard to SSH service.

D. Glastopf Honeypot

Glastopf is low involvement honeypot web application which is capable of emulating thousands of web vulnerabilities to gather data from attacks that target web applications. The principle is very simple as Glastopf returns fire using an attacker's response who perhaps expects to be able to exploit web applications.

E. Dionaea Honeypot

Dionaea is software that offers network services that can be exploited. The action is to trap or exploit malware that attacks the tissue, and its main purpose is to obtain a copy of the malware [10].

F. Proxmox

Proxmox VE is an open source complete virtualisation management solution for servers. It is based on a Kernel-based Virtual Machine (KVM) virtualisation and container-based virtualisation and manages virtual machines, storage, virtual networks, and High Availability (HA) Clustering. The enterprise-class features and an intuitive web interface are designed to help the user increase the use of existing resources and to reduce hardware costs and administration time.

III. Results and Discussion

A. System Topology

Figure 2 shows the flow of the arrested honeypot attacks carried out in this study:

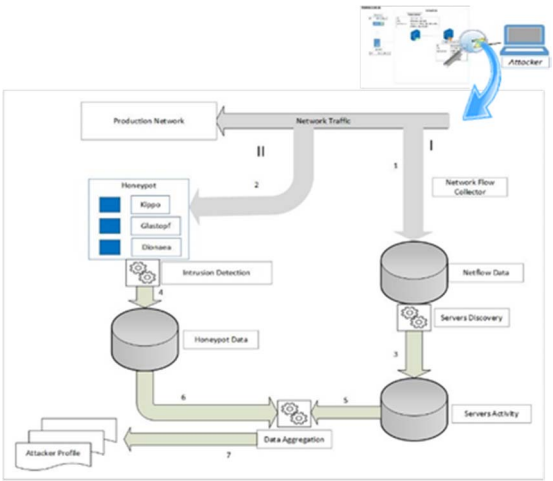

Fig 2. Attack Detection Flow

Figure 2 can be explained is as follows:

1. Packets received by a network are searched for any suspicious activity. If the packet is not suspicious, then the packet is processed and enters into the production network where the packet generates a response from the server. If the packet is suspicious, then the packet is divided into two stages. The first packet is collected so as to form the flow of data from a network server and is processed by the MHN. The packet is processed by the honeypots installed by the researchers. The Kippo honeypot detects the presence of an attack and follows the infiltration into the SSH port. Glastopf detects the presence and follows the infiltration into the HTTP port. Dionaea detects the presence of the intrusion into a particular port.
2. The packet is processed by the honeypots installed by the researchers which are the Kippo, Dionaea, and Glastopf honeypots. The incoming packet is processed and considered as an intrusion.
3. Packets that have been processed by the MHN server are marked as suspicious activity and then collected.
4. Once processed by the honeypots, the packet is incorporated into the honeypot data.
5. The packet from the MHN server is processed into a set of data.
6. The packet from the honeypots is processed into a set of data.
7. The data that has been processed from the MHN and the honeypot server are synchronised and the data considered as an assault.

Based on the results of the analysis to build the honeypot system, the required components in the form of hardware are server, laptop client, and computer attacker. While the software components are described in Table 1.

TABLE 1. Software Components

No	Software	Specification
1	Proxmox VE 2.3	CPU = 4 x Intel®Xeon® CPU E5420 @ 250 GHz (1 Socket) RAM = 5.83 GB Harddisk = 33.47 GB
2	Ubuntu server 14.03 LTS	RAM = 2.00 GB Processors = 2 cores Harddisk = 32 GB
3	Modern Honey Network (MHN)	RAM = 4.00 GB Processors = 2 cores Harddisk = 40 GB

The topology system to be designed is a small scheme which is added to an existing network topology. In other words the authors do not change the existing network scheme at PUSTIKNAS, but add to it instead. Figure 3 shows the topology which has been applied in the research.

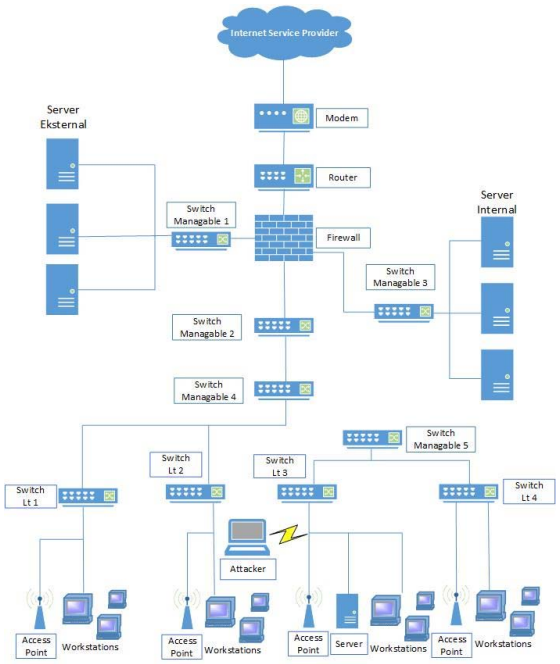

Fig 3. Reprocessed Network topology

Based on the image above there is one Server. The server is used as a virtual server in the first month. The researchers decided to put the server under floor of Switch 3 because that is where the server will obtain the IP address, and at that point the server is also allowed to be attacked by other computers connected to the internal network.

B. Configuration of the MHN

The researchers first installed the components required for this configuration. Once all components were installed, the researchers checked with the command:

```
sudo supervisorctl status
```

The researchers wanted to install the services on a local network honeypot and the network could then be converted into a public network. To do this, the researchers had to edit the configuration of Mnemosyne. Then edit the section [normaliser] to:

```
ignore_rfc1918 = False
```

This configuration allowed the services honeypot in the network to interact with other servers in the internal network. Then to reset the configuration that was changed, the researchers restarted the MHN packet. To complete the installation of the MHN, the researchers ran the script:

```
sudo ./install_mhnserver.sh
```

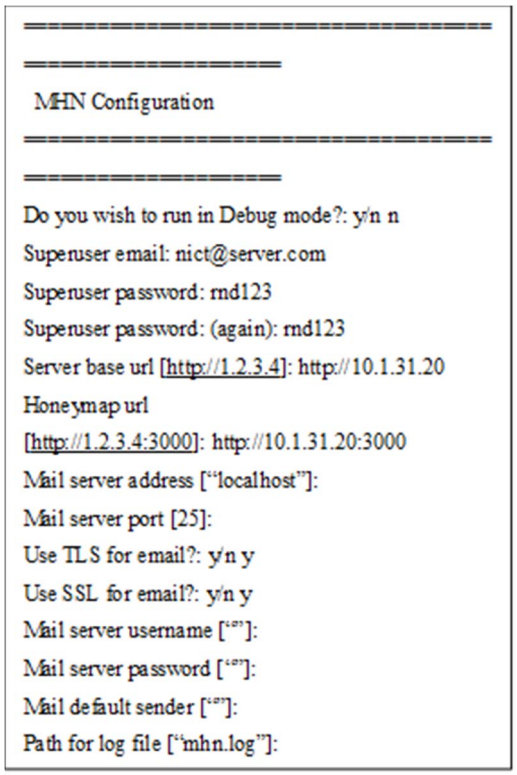

Fig 4. MHN Configuration

After the installation was complete, a window appeared for configuration of the MHN. The researchers completed the configuration according to the reference made in this study. Figure 4 represents the MHN configuration which was set up by the researchers.

Table 2 provide a description of each configuration.

TABLE 2. MHN CONFIGURATION

Configuration	Remark
Debug mode	The user interface is implemented in a computer program that allows the user to view or manipulate the data contained in the program
Superuser email	Email managerial user MHN
Superuser password	Password managerial user MHN
Server base URL	MHN Server IP
Honeymap URL	honeymap IP
Mail server address	Mail server address
Mail server port	Mail Server Port
TLS for email	Transport Layer Security Protocol that

	protects the privacy of the interaction between applications and users
SSL for email	Secure Sockets Layer protocol, security technology standard that is applied between the web server and browser
Mail server username	username mail server
Mail server password	password mail server
Mail Default Sender	Default email sender
Path for log file	Log file location

After the installation was completed, the researchers then accessed the server via an IP MHN web browser and logged in as a superuser.

C. Honeypot Implementation

To install the services honeypot, the researchers accessed the MHN server as a web application from a browser by typing the IP 10.1.31.20. After successful login then the deployed menu was accessed. Here the researchers found different forms of service honeypot to be applied to other operating systems. The researchers then chose the service honeypot to be installed, used the script in the deploy command and typed in the terminal operating system that was to be installed in the honeypot service. Here the researchers used webserver as the operating system. Once installed, the researchers made sure the services honeypot had certain ports connected using the TCP connection.

The service-service honeypot researchers installed and configured the Kippo, Glastopf and Dionaea honeypots.

Once entered into the MHN menu deployed on the server the researchers chose the service honeypot, copied the script at the command deployed in the dialog box and pasted in the Ubuntu console, then clicked the Update button to save it to the server. The installation went without any problems.

The researchers executed the honeypot configuration file by entering the deploy IP MHN key after typing the server name. Once installed, the researchers restarted the server.

Usually Kippo connects on port 2222, because port 22 cannot be accessed as it requires the right permissions from the root user. However, using the deployed script, Kippo listens on port 64 222 and SSHD listens on port 2222, which means that in order to manage the server the researcher must connect to the 2222 by configuring the kippo.cfg file located in the folder / opt / Kippo. Hence the researchers typed nano /opt/kippo/kippo.cfg

```
[honeypot]
ssh_port = 64222
ssh_addr = 127.0.0.1
reported_ssh_port = 22
```

In such a configuration the researchers included ssh_port = 64 222 which indicated that the honeypot was installed on port 64 222. The researchers inserted ssh_addr = 127.0.0.1 which indicated the address used by SSH to the honeypot was a localhost and the researchers included reported_ssh_port = 22, which showed that in the event of acts of intrusion the SSH port to port 22 would be displayed as a destination port following the intrusion on the web interface Kippo.

The Glastopf honeypot was connected to port 80 http port as this service was configured by the honeypot researchers. The configuration file was located in /opt/glastopf/glastopf.cfg. Therefore, the researchers used the command nano /opt/glastopf/glastopf.cfg, following the display of these configurations:

```
[webserver]
host = 0.0.0.0
port = 8080
uid = nobody
gid = nogroup
proxy_enabled = False
```

In the above configuration, the researchers included host = 0.0.0.0 because the honeypot servers previously defined used 10.1.31.20. Furthermore, the ports were mounted on the honeyport 8080 because the port configured in the Apache config file was 80, but even so the honeypot still showed port 80 as the destination port in the web interface honeypot Glastopf in the case of acts of infiltration. The researchers included the value of a user ID on a web server that was uid = nobody and group ID on a web server that was gid = nogroup indicating that all acts of infiltration were directed at IP 10.1.31.15 and port 80 on all users and all groups who were paired honeypot systems such as Glastopf i.e. the web server, was displayed in the web interface Glastopf honeypot. Recently researchers included a proxy_enabled value = False because the researchers did not configure the proxy honeypot system fitted with the web server.

The last Honeypot was the Dionaea Honeypot Service honeypot which was connected to 10 ports. The researchers configured this honeypot service. The configuration file was located in /etc/dionaea/dionaea.conf. Hence the researchers used nano /etc/dionaea/dionaea.conf.

IV. TESTING AND EVALUATION

The researchers conducted tests consisting of infiltration and attacks on the wireless networks. Steps taken by the authors to perform the tests were as follows:

A. Port Scanning

This test used nmap which was directed to the IP web server that was used in this test as 10.1.31.15. This test was conducted on the honeypot-honeypot installed on the web server. In the first test, the Kippo SSH Honeypot was used to see the action undertaken by the attacker intrusion that led to the SSH port.

After testing, the Kippo SSH Honeypot used by the researchers showed a log of activities undertaken by the attacker IP in the form of an experiment with SSH port scans.

B. DoS

Denial of Service (DoS) attacks in this study used the ping of death, TCP Flood, SYN Flood, Smurf Attack, Fraggle Attack, and DDoS. In this experiment the targeted host attacked was 10.1.31.15 as the IP used for the web server residing on the server. The attack came from within the network which was routed to the IP. This research used a honeypot to detect the intrusions occurring in the form of a Denial of Service.

In the SYN flood attacks, these were detected by the MHN server in the same time so that it could be concluded that this was a flooding attack. When the researchers ran the services on the web server, although many requests were received, access to the web services server did not become slow.

The first test applied the ICMP protocol used for the Smurf attack and the second test used the UDP protocol which was used for the Fraggle attack. Although the IP of the source IP address, however it detected as the computer Attacker IP by the honeypot was 10.1.30.72.

C. DDoS

The researchers performed a test of the web servers that were installed with a honeypot via port 80. There are 3 methods are available to test as TCP, UDP, or HTTP. At the time of testing the method used by the researchers was HTTP because it was the only open port on the computer used by the researchers. The thread stating the number of data packets that the researchers submitted gave a number of 1000 pieces. This was used to test the honeypot system that was designed by using the pressed button 'IMMA CHARGIN MAH LAZER' listed at the top right corner of the image.

The honeypot system showed indications of such attacks and received attack indications from the attacker seen on the main form of MHN. In this test, incoming data packets to the environment were not too big and did not deter services such as Web Server.

D. Brute force Attack

In this test, the researchers conducted brute force attacks aimed at the SSH port. Brute force can be applied to the SSH port open as port 22. This can be done using tools that already exist for the Linux operating system such as Hydra, and using a password dictionary with a few examples of usernames and passwords created by the authors. In this attack the authors used the honeypot Kippo that simulated port 22.

The result showed that honeypot Kippo used in the test could detect the brute force attack well and recorded the activity of an attacker who successfully entered into the system.

V. CONCLUSION

Once the honeypot management system with MHN was implemented, all the honeypots managed to outwit the attackers by opening ports on a server that turned these ports into a hoax and were cleared to record the suspicious activities of the attackers. Although the action of intrusion into the system was not optimal, but the results were displayed on the web interface which could complement each other in providing information to administrators for further action.

Honeypots Kippo, Dionaea and Glastopf managed to deceive the attackers by opening ports in servers that are often targeted by attackers. Web-application honeypots need to be developed further in view of PUSTIKNAS which has many web applications.

REFERENCES

[1] S. A. Budiman, C. Iswahyudi, and M. Sholeh, "Implementasi Intrusion Detection System (IDS) Menggunakan Jejaring Sosial Sebagai Media Notifikasi," in *Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST)*, 2014.

[2] J. Gondohanindijo, "IPS (Intrusion Prevention System) Untuk Mencegah Tindak Penyusupan/Intrusi," *Maj. Ilm. Inform.*, 2012.

[3] B. Tambunan, W. S. Raharjo, and J. Purwadi, "Desain dan Implementasi Honeypot dengan Fwsnort dan PSAD sebagai Intrusion Prevention System," *1 / Vol.5 / Sept. 2013*, vol. 0, no. 0, pp. 1–7, 2013.

[4] R. C. Joshi and A. Sardana, *Honeypots: A New Paradigm to Information Security*. Science Publishers, 2011.

[5] I. P. A. E. Pratama, *Handbook Jaringan Komputer : Teori dan Praktek Berbasiskan Open Source*. Informatika, 2014.

[6] C. S. Bayu, "Analisis Penerapan Jaringan Keamanan Menggunakan IDS dan Honeypot," Universitas Dian Nuswantoro, 2014.

[7] S. G. Joseph, "Advanced Honeypot Architecture for Network Threats Quantification," *Int. J. Sci. Eng. Appl. Sci.*, no. 15, pp. 2395–3470, 2015.

[8] S. M. Jigneshkumar, "Modern Honey Network," *Int. J. Res. Advent Technol.*, 2016.

[9] T. M. Eastep, "Netfilter Overview," *ShoreWall*, 2003. [Online]. Available: http://shorewall.org/NetfilterOverview.html. [Accessed: 07-Feb-2017].

[10] A. Muhammad, "Implementasi Honeypot Dengan Menggunakan Dionaea Di Jaringan Hotspot Fizz," *Politek. Telkom*, 2011.