Website and Network Security Techniques against Brute Force Attacks using Honeypot

Arif Nursetyo
Department of Informatics Engineering
Dian Nuswantoro University
Semarang, Indonesia
setyonurarif@gmail.com

De Rosal Ignatius Moses Setiadi
Department of Informatics Engineering
Dian Nuswantoro University
Semarang, Indonesia
moses@dsn.dinus.ac.id

Eko Hari Rachmawanto
Department of Informatics Engineering
Dian Nuswantoro University
Semarang, Indonesia
eko.hari @dsn.dinus.ac.id

Christy Atika Sari
Department of Informatics Engineering
Dian Nuswantoro University
Semarang, Indonesia
atika.sari s@dsn.dinus.ac.id

Abstract— **The development of the internet and the web makes human activities more practical, comfortable, and inexpensive. So that the use of the internet and websites is increasing in various ways. Public networks make the security of websites vulnerable to attack. This research proposes a Honeypot for server security against attackers who want to steal data by carrying out a brute force attack. In this research, Honeypot is integrated on the server to protect the server by creating a shadow server. This server is responsible for tricking the attacker into not being able to enter the original server. Brute force attacks tested using Medusa tools. With the application of Honeypot on the server, it is proven that the server can be secured from the attacker. Even the log of activities carried out by the attacker in the shadow server is stored in the Kippo log activities.**

Keywords— Honeypot, Medusa, Network Security, Brute Force, Web Security

I. INTRODUCTION

The web is indispensable in commercial and non-commercial activities as a complex platform for creating sophisticated distributed applications with varying security requirements when searching websites by entering Uniform Resource Locator (URL) into the browser window, the browser will first check the URL scheme to determine protocol [1]. The rapid progress in internet-based technology available on the internet has increased the comfort of human life. But the internet also has weaknesses because there are many threats carried out by irresponsible individuals and organizations.

Attackers can work by entering into a computer network that is available to steal personal and company information and data. This is information that can be very critical and sensitive, such as personal data on social security or bank account information [2]. Safeguarding data and information can be done in various ways, from direct protection to data or information using cryptographic and steganography techniques [3], [4], or by protecting access through the network on the data such as using Honeypots.

Since honeypots first emerged as a network security concept that needs to be configured and managed by network administrators for the dynamic honeypot. Honeypots continue to be developed to be intelligently used on internet networks without requiring configuration and maintenance which has the main advantage for honeypot technology in modern network security [5]. A honeypot is a tool used to commit network fraud in providing an illusion to an attacker, which has the basic goal of getting alignment in a honeypot when the system encounters problems with an attacker [6]. The attacker is only in an illusion system and the attacker's information and methods used are stored with both in the server activity log, so that security can find out the position of the attacker who displays a valid IP address. Some authors have submitted various authentication protocols for multi-server environments over the past decade.

Reddy et al. [7] prove that the proposed protocol reaches the authentication property simultaneously safely using Burrows-Eternal-Needham (BAN) logic. This logic is widely used with the formal security process of the proposed protocol when verifying using an Automated Validation Protocol and Internet Security Application (AVISPA) tool which is widely accepted in showing protocols that can withstand active and passive attacks including replay and management attacks. in-the-middle.

Studiawan et al. [8] which models the log as a graph and proposes to absorb the k-click into the auth cluster. the log file helps the system administrator manage the process by checking the results of experiments which show that this approach is suitable for grouping the initial log of an SSH brute-force attack because of a brute-force attack on an SSH service that still persists in a server environment due to methods that have not yet implemented graph theory to analyze the authentication log in recording this attack.

There are various types of malware capture systems available now such as malware detection systems that are used with other security systems such as Intrusion Detection System (IDS) and firewall to make the entire network safer[9]. IDS is used to monitor, analyze network-related data, and detect dangerous activities. The IDS algorithm used by srcIP is detected as a brute force attack with the number of trials entered on the dstIP where the attack indicates that adstIP was detected as an attack by srcIP on a certain date so

using TOPASE can extract DBF victims correctly when the IDS log is entered mainly some kind of DBF[10]. In this study, the process of capture and detection of malware carried out by brute force on the server through the website domain using Medusa tools to secure the server by creating a shadow server which has a very important task in a secure infrastructure.

II. Basis of Theory

A. Bruteforce Attack

Brute force attacks are to illegally get username and password pairs by trying all existing partners to enter network services and one of the security threats for network administrator services [11]. The attack carried out using brute force can be seen in Fig. 1. which explains that the attempted attack is carried out as much as possible until the attack is successfully carried out with the process at the counter = n.

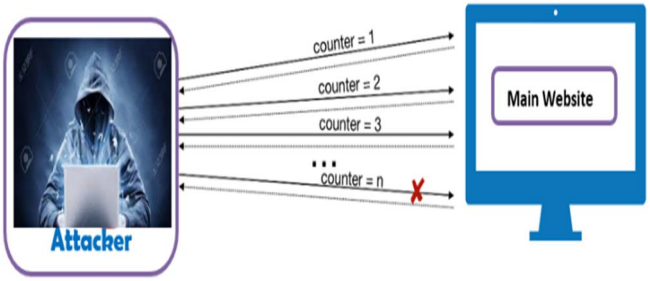

Fig. 1. Bruteforce Architecture

B. Honeypot

Honeypot is a security system that functions to create a shadow server or a trap server[12]. When an attacker tries to attack or access the server as if, the attacker has entered and accessed the server, but in fact that accessed by the attacker is a shadow server formed by the honeypot [13].This allows the server admin to know the attacker's info in the form of an IP address, date of the attack, and activities that the attacker is doing on that server [14]. Where the working principle of the honeypot is described as shown in Fig. 2.

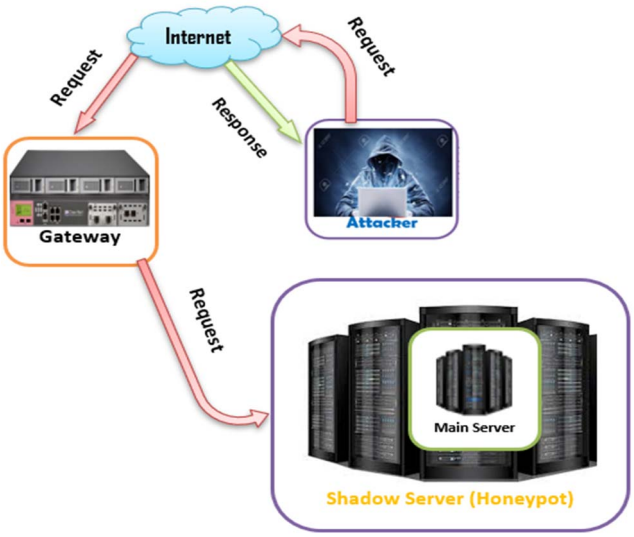

Fig. 2. Working principle of the honeypot

As in Fig. 2. that the working principle of Honeypot includes:

1. The first process is a process carried out by attackers in carrying out attacks by making requests via the internet.
2. The second process is the process when the internet provides a response based on the request activity that has been carried out by the attacker.
3. The third process is a follow-up action after the attacker has successfully connected his activities through the internet to access the gateway which is a device that connects one computer network or more computer networks with different communication media so that information when the computer network is switched will be different from different network media.
4. This fourth process occurs when the activities carried out by the attacker managed to get the identity of the device such as the user who has a username and password that is connected by a computer network. In this process when the server has not been installed by a honeypot, the attacker can do any activity on the main server (webserver). But when the honeypot is installed, the attacker will only do shadow server activities, so the attacker will fail when adding files, cracking, and installing a backdoor.

C. Medusa

```
Medusa [-h host|-H file] [-u username|-U file] [-p password|-P file] [-C file] -M module [OPT]

-h [TEXT]    : Target hostname or IP address
-H [FILE]    : File containing target hostnames or IP addresses
-u [TEXT]    : Username to test
-U [FILE]    : File containing usernames to test
-p [TEXT]    : Password to test
-P [FILE]    : File containing passwords to test
-C [FILE]    : File containing combo entries. See README for more information.
-O [FILE]    : File to append log information to
-e [n/s/ns]  : Additional password checks ([n] No Password, [s] Password = Username)
-M [TEXT]    : Name of the module to execute (without the .mod extension)
-m [TEXT]    : Parameter to pass to the module. This can be passed multiple times with a
               different parameter each time and they will all be sent to the module (i.e.
               -m Param1 -m Param2, etc.)
-d           : Dump all known modules
-n [NUM]     : Use for non-default TCP port number
-s           : Enable SSL
-g [NUM]     : Give up after trying to connect for NUM seconds (default 3)
-r [NUM]     : Sleep NUM seconds between retry attempts (default 3)
-R [NUM]     : Attempt NUM retries before giving up. The total number of attempts will be NUM
-c [NUM]     : Time to wait in usec to verify socket is available (default 500 usec).
-t [NUM]     : Total number of logins to be tested concurrently
-T [NUM]     : Total number of hosts to be tested concurrently
-L           : Parallelize logins using one username per thread. The default is to process
               the entire username before proceeding.
-f           : Stop scanning host after first valid username/password found.
-F           : Stop audit after first valid username/password found on any host.
-b           : Suppress startup banner
-q           : Display module's usage information
```

Fig. 3. Medusa syntax

Medusa is a trigger that functions so that in brute-force techniques in use quickly, parallel, and modular in entering a system through the network. Medusa works in support of as many services as possible that allow remote authentication. This test can be performed against multiple hosts, users or passwords concurrently. To operate Medusa requires this syntax rule by default which must be met because when the compilation process every line of the script will be checked to ensure that the programming language commands to do the compiler are able to be completed correctly. Fig. 4 shows some examples of Syntax which are mostly operated on Medusa.

III. Building Security on Server using Honeypot

Before building a honeypot, several types of concepts are needed to play a role in building, forming, and training some of the models needed as a step in system requirements.

This honeypot is very important to be an additional device to minimize attacks that occur in our system. There are several elements found in honeypot in general, including monitoring or logging tools; alerting mechanism; keystroke logger; packet analyzer; forensic tools. The honeypot collects a little data but with a high value that allows rapid analysis and response. The simplicity of using a honeypot makes it easy to configure its utilization, although there are also complex ones for research purposes. The simpler the honeypot, the smaller the risk, such as when the volume of data is not as much as a log on a firewall system or IDS. Fig.4. Show the Honeypot architecture.

For the process of simulation and assault tests using a Honeypot will be described as Fig. 5. Tests are carried out on the Kali Linux operating system. The first time the attacker finds out the IP address of the target server using ping on the website address. Next, do the scanning service on the target server with Nmap and the ssh service on the standard port. Then the attacker performs brute force on the ssh service to find the password. In Kali Linux, there are several Bruteforce Hydra and Medusa tools, where Medusa will be used as brute force tools by the attacker. The brute force process is done by reading the password list in the /home/pass.txt folder. This file is a collection of many texts or words and when the password has been found, the attacker tries to access the server via SSH. At this stage, illustrated in Fig. 6.

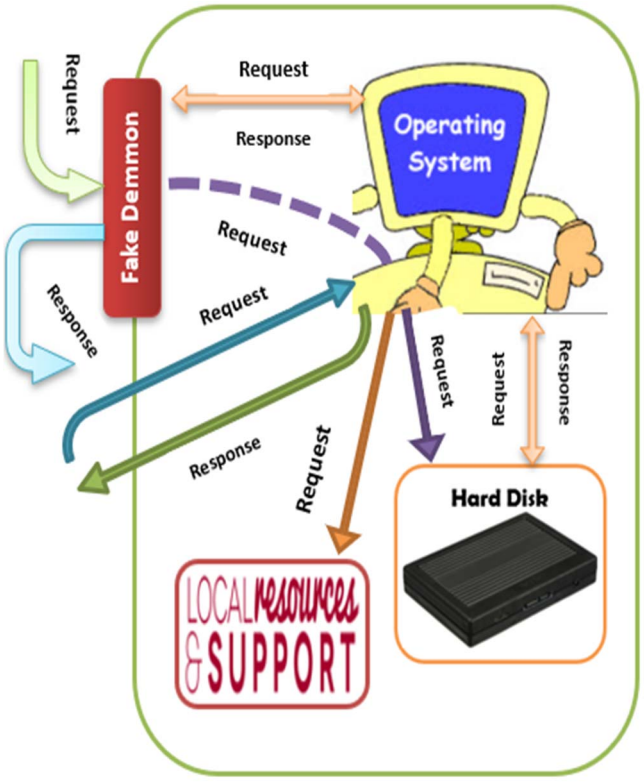

Fig. 4. Honeypot Architecture

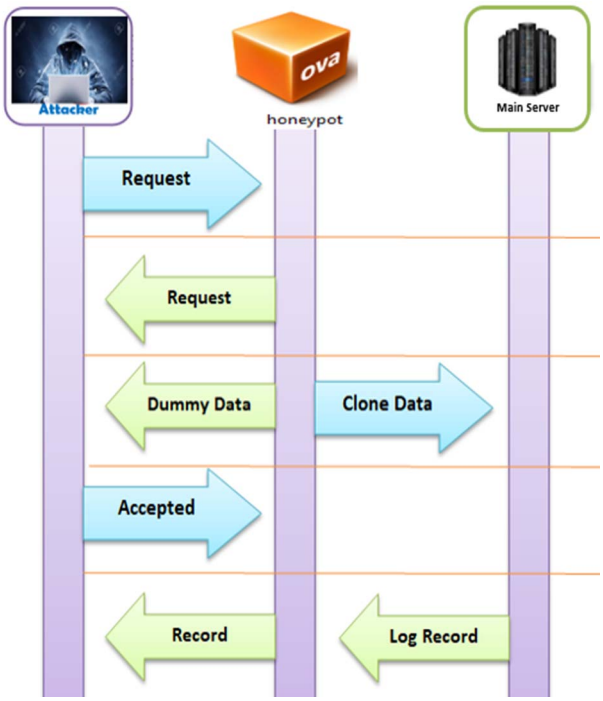

Fig. 5. Honeypot workflow

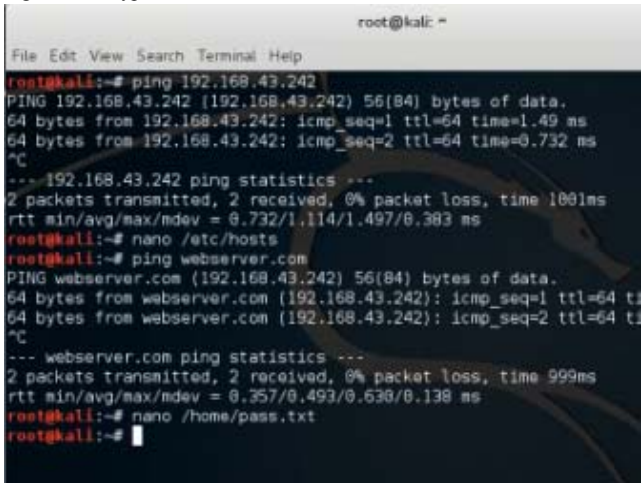

Fig. 6. Getting web IP address and prepare password list for Medusa

When the attacker will enter the Honeypot server it will point to the shadow server. When entering the shadow server, the attacker will fail in carrying out activities such as creating files and installing a backdoor. With this, all attacker activities are detected by the Kippo log so that the admin can find out all the activities carried out by the attacker.

IV. Evaluation and Testing

A. Attacking Results Before using Honeypot

After the website server address is known by pinging the website address on the world wide web. The attacker will try to control the server with a brute force process. After obtaining access rights to the server, the attacker can control the website easily. Fig. 7 shows the process of attacking a server that is not protected by Honeypot. Easily IP server with the address 192.168.43.242 changed its content like Fig. 8.

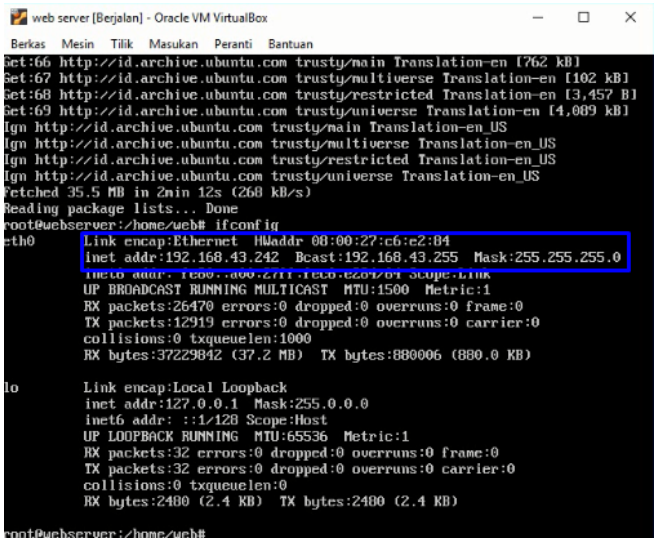

Fig. 7. Getting web IP address and attack it

As in the picture, it has been stated that there is an internet address which means that the address of the website is as testing material. We can see the appearance of the website before it is attacked as like fig 8.

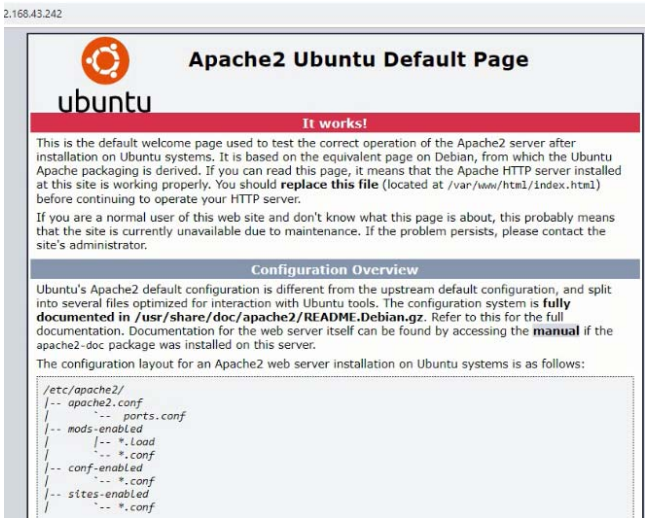

Fig. 8. website appearance before being attacked

As in the appearance of the website in Fig. 8, the coding of the website program can be seen as in Fig. 9

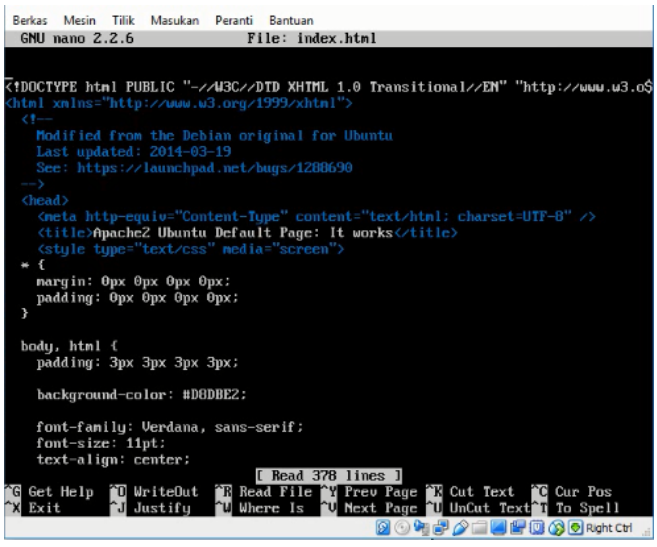

Fig. 9. Website coding before being attacked

Changes that occur when the website is like in fig. 8 when a website that has been attacked but has not used protection from the honeypot, it will produce as in fig. 10

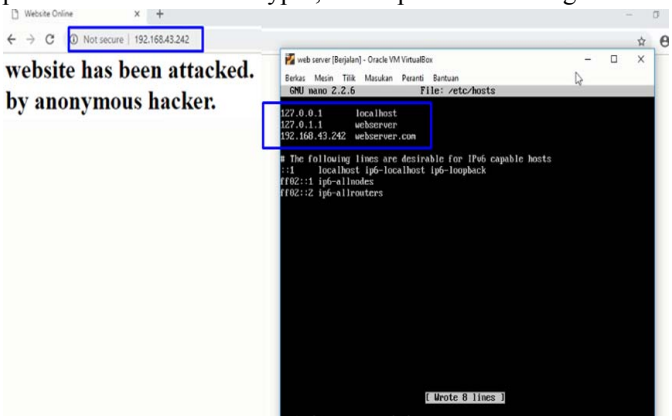

Fig. 10. Attack result without Honeypot

Please note that using the Medusa brute force process conducted by Medusa takes 1 hour 30 minutes to be able to enter the server and damage the data. Before logging out of the server the attacker managed to install a backdoor so he had a special door to enter the server without the need to repeat the initial steps.

B. Attacking Results After using Honeypot

Update the repository before running the honeypot using the command "apt-get update", then install ssh and git using the command "apt-get install ssh git", as Fig. 9.

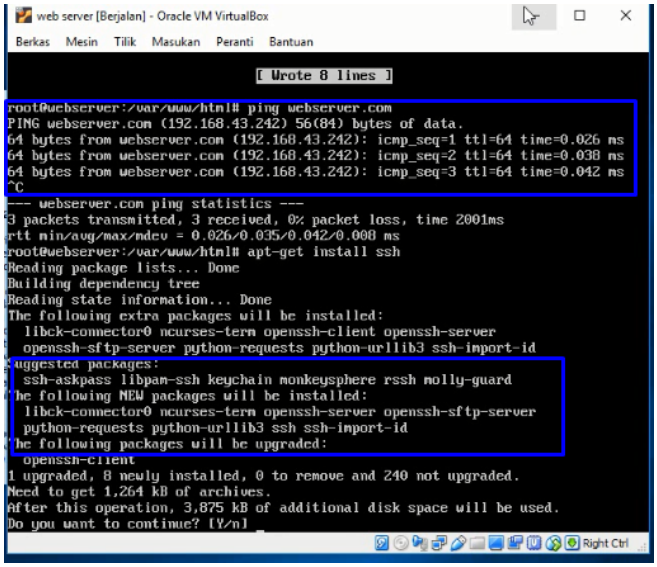

Fig. 11. The SSH and GIT installation

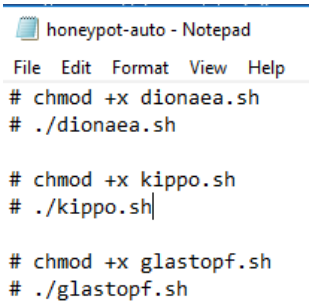

Fig. 12. Adding execute permissions

Then the next step is to go back to the installation download directory by adding execute permissions and running the install script, as Fig. 10. Next, change the SSH port 65534 to the horse's mouth, and false SSH port (Honeypot) to port 22. Then do restart servicing as in Fig. 11.

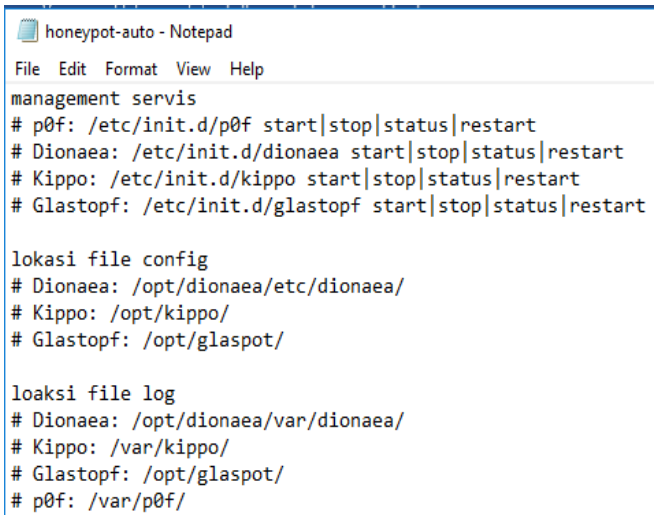

Fig. 13. Restarting service

Next, brute force the ssh login password with medusa for example 192.168.43.242. With syntax medusa -u root -P / home / passwordlist.txt -h 192.168.43.242 -M ssh like Fig. 12.

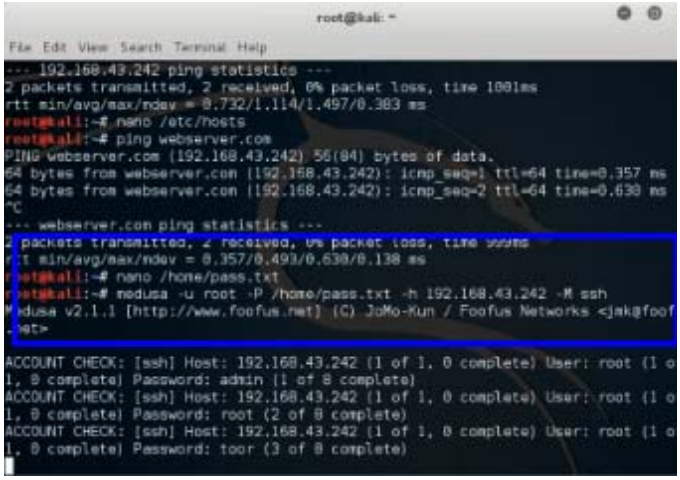

Fig. 14. Medusa brute force process

After getting the password then continue with logging into the system to see the log on the server that has installed the honeypot. Attackers try to do activities by creating additional files but fail. For example, as in Fig. 13.

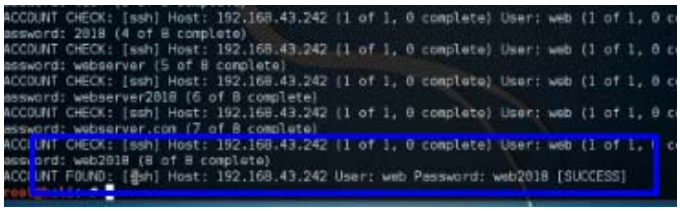

Fig. 15. Attackers try to do add file

Then the attacker tried to install the backdoor, but it still failed too, as shown in Fig.14. Honeypot records the log using Kippo as shown in Fig.15. At the time the attacker also managed to install a backdoor to be able to enter the system again easily. But after the honeypot is installed then the attacker tried to install the backdoor, but it still failed too, as shown in Fig.16.

When installing honeypot, the attacker only performs activities on the shadow server where the attacker cannot copy or damage data on the existing server, even the attacker's activities are stored in the Kippo log as shown in Figure 17.

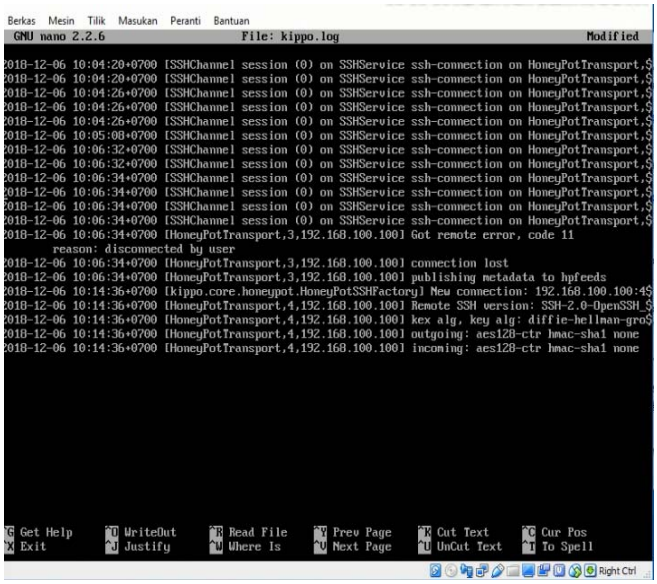

Fig. 16. Attackers try to do create a backdoor

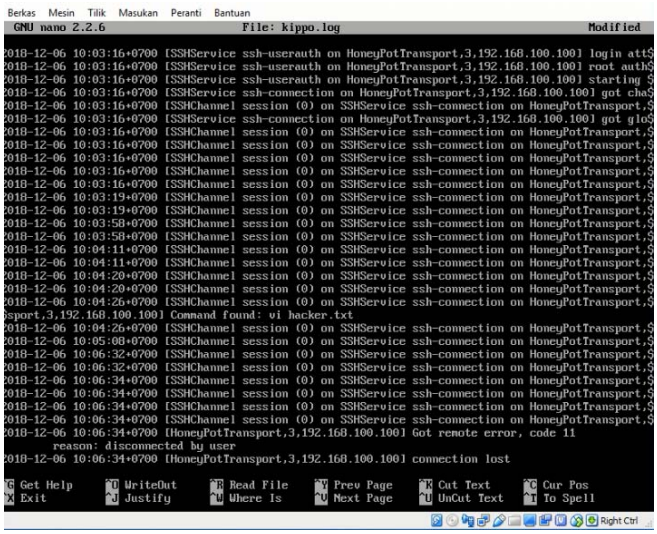

Fig. 17. Kippo Log

V. CONCLUSIONS

From the research conducted, it proves that the process of breaking into a website server using Medusa takes 1 hour 30 minutes. Without the Honeypot, the attacker can enter into the server, the attacker can destroy the existing data. The attacker also managed to install a backdoor to be able to enter the system again easily. But with the installation of a honeypot, the attacker only performs activities in the shadow server where the attacker cannot copy or damage the data on

the existing server, even the attacker's activities are stored in the Kippo log.

REFERENCES

[1] M. F. Haque, M. B. A. Miah, and F. Al Masud, "Enhancement of Web Security Against External Attack Md. Fazlul Haque Mohammad Badrul Alam Miah," no. August 2017.

[2] M. M. Najafabadi, T. M. Khoshgoftaar, C. Calvert, and C. Kemp, "Detection of SSH Brute Force Attacks Using Aggregated Netflow Data," in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, 2015, pp. 283–288.

[3] D. R. I. M. Setiadi, "Payload Enhancement on Least Significant Bit Image Steganography Using Edge Area Dilation," *Intl J. Electron. Telecommun.*, vol. 65, no. 2, pp. 295–300, 2019.

[4] M. N. M. Najih, D. R. I. M. Setiadi, E. H. Rachmawanto, C. A. Sari and SetiaAstuti, "An Improved Secure Image Hiding Technique Using PN-Sequence Based on DCT-OTP," in *International Conference on Informatics and Computational Sciences (ICICoS)*, Semarang, 2017.

[5] D. Fraunholz, M. Zimmermann, and H. D. Schotten, "An adaptive honeypot configuration, deployment, and maintenance strategy," in *2017 19th International Conference on Advanced Communication Technology (ICACT)*, 2017, pp. 53–57.

[6] S. Sharma, "Detection and analysis of network & application layer attacks using Maya Honeypot," in *2016 6th International Conference - Cloud System and Big Data Engineering (Confluence)*, 2016, pp. 259–262.

[7] A. G. Reddy, E.-J. Yoon, A. K. Das, V. Odelu, and K.-Y. Yoo, "Design of Mutually Authenticated Key Agreement Protocol Resistant to Impersonation Attacks for Multi-Server Environment," *IEEE Access*, vol. 5, pp. 3622–3639, 2017.

[8] H. Studiawan, B. A. Pratomo, and R. Anggoro, "Clustering of SSH brute-force attack logs using k-clique percolation," in *2016 International Conference on Information & Communication Technology and Systems (ICTS)*, 2016, pp. 39–42.

[9] P. D. Ali and T. G. Kumar, "Malware capturing and detection in dionaea honeypot," in *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*, 2017, pp. 1–5.

[10] S. Honda, Y. Unno, K. Maruhashi, M. Takenaka, and S. Torii, "TOPASE: Detection of brute force attacks used disciplined IPs from IDS log," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2015, pp. 1361–1364.

[11] S. S. Kolahi, K. Treseangrat, and B. Sarrafpour, "Analysis of UDP DDoS flood cyber attack and defense mechanisms on Web Server with Linux Ubuntu 13," in *2015 International Conference on Communications, Signal Processing, and their Applications (ICCSPA'15)*, 2015, pp. 1–5.

[12] D. Lavrov, V. Blanchet, S. Pang, M. He, and A. Sarrafzadeh, "COR-Honeypot: Copy-On-Risk, Virtual Machine as Honeypot in the Cloud," in *2016 IEEE 9th International Conference on Cloud Computing (CLOUD)*, 2016, pp. 908–912.

[13] R. M. Campbell, K. Padayachee, and T. Masombuka, "A survey of honeypot research: Trends and opportunities," in *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, 2015, pp. 208–212.

[14] J.-H. Park, J.-W. Choi, and J.-S. Song, "How to Design Practical Client Honeypots Based on Virtual Environment," in *2016 11th Asia Joint Conference on Information Security (AsiaJCIS)*, 2016, pp. 67–73.