

## BAB 3

### METODE PENELITIAN

#### 3.1 ALAT DAN PARAMETER YANG DITELITI

Pada penelitian yang berjudul “Analisis Keamanan Jaringan Terhadap Serangan *Brute Force* Dan *DoS Attack* Menggunakan *Honeypot* Kippo dan *Honeypot* Dionaea” ini menggunakan beberapa perangkat lunak guna membantu melakukan penelitian seperti *Virtual Box* untuk menjalankan sistem *Honeypot* serta beberapa *tools* untuk melancarkan serangan *Brute Force* dan *DoS Attack* untuk mengumpulkan data.

##### 3.1.1 Perangkat Keras

Perangkat keras yang dibutuhkan dalam penelitian ini terdiri dari satu laptop yang digunakan untuk menjalankan perangkat lunak *Virtual Box* dengan spesifikasi laptop seperti pada tabel 3.1.

**Table 3. 1 Spesifikasi Laptop**

Sistem Operasi	Windows 11 Pro
Processor	Intel(R) Core (TM) i5-8265U CPU @ 1.60GHz 1.80 GHz
RAM	4 GB
Harddisk	500 GB

##### 3.1.2 Perangkat Server

*Server* yang dibutuhkan dalam penelitian ini terdiri dari beberapa aplikasi (*software*) yang memiliki masing-masing fungsi yang digunakan dalam penelitian ini, berikut beberapa perangkat lunak yang dibutuhkan seperti pada tabel 3.2

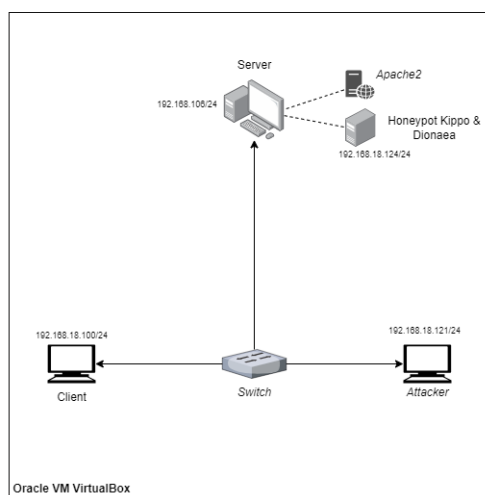
**Table 3. 2 Software yang digunakan**

Software	Fungsi
<i>Virtual Box</i>	Digunakan untuk membuat lingkungan virtual yang terisolasi dari sistem operasi utama sehingga dapat menginstal dan menjalankan sistem operasi dan aplikasi tanpa

<i>Software</i>	<b>Fungsi</b>
	memengaruhi sistem operasi utama.
Nmap	Digunakan untuk memindai <i>port port</i> yang ada di suatu jaringan dan menampilkan <i>port</i> apa saja yang terbuka dan dijalankan.
LOIC	Digunakan sebagai alat serangan DoS <i>Attack</i> .
SQLMAP	Digunakan sebagai melancarkan serangan database URL
<i>Apache2</i>	Membangun web <i>server</i> .
Medusa	Digunakan sebagai melancarkan serangan <i>Brute Force</i> .
Kippo Graph	Digunakan untuk menampilkan hasil log dari serangan <i>Brute Force</i> dalam bentuk grafik pada web <i>server</i> .
Dionaea	Digunakan untuk menampilkan hasil log dari serangan DoS <i>Attack</i> .

### 3.1.3 TOPOLOGI JARINGAN

Untuk topologi jaringan pada penelitian tugas akhir ini terdapat beberapa perangkat dan alur seperti gambar 3.1



**Gambar 3. 1 Topologi Jaringan**

Berdasarkan gambar 3.1 topologi jaringan dibuat dalam jaringan Oracle VM VirtualBox yang dirancang terdiri dari beberapa perangkat yaitu PC *server*, PC *client* dan PC *attacker*. Pada penelitian ini semua PC terhubung melalui jaringan *Switch* agar saling terhubung dalam satu *network* yang sama. Dalam hal ini, IP Address dibuat menjadi IP *Privat* menggunakan *Bridged Adapter*. Pada PC *server* terdiri dari 2 *server* yaitu *server* asli dan *server honeypot*. *Server* asli berisikan web *server* *apache2* dan *server honeypot* berisikan *kippo* dan *dionaea*. *Honeypot dionaea* akan mendeteksi serangan *dos* dan *sql* yang masuk pada port 8080, sedangkan *honeypot kippo* akan mendeteksi serangan *brute force* yang masuk pada port 22. PC Untuk pengalamatan Alamat IP dapat dilihat pada tabel 3.3 dibawah ini.

**Tabel 3. 3 Pengalamatan IP**

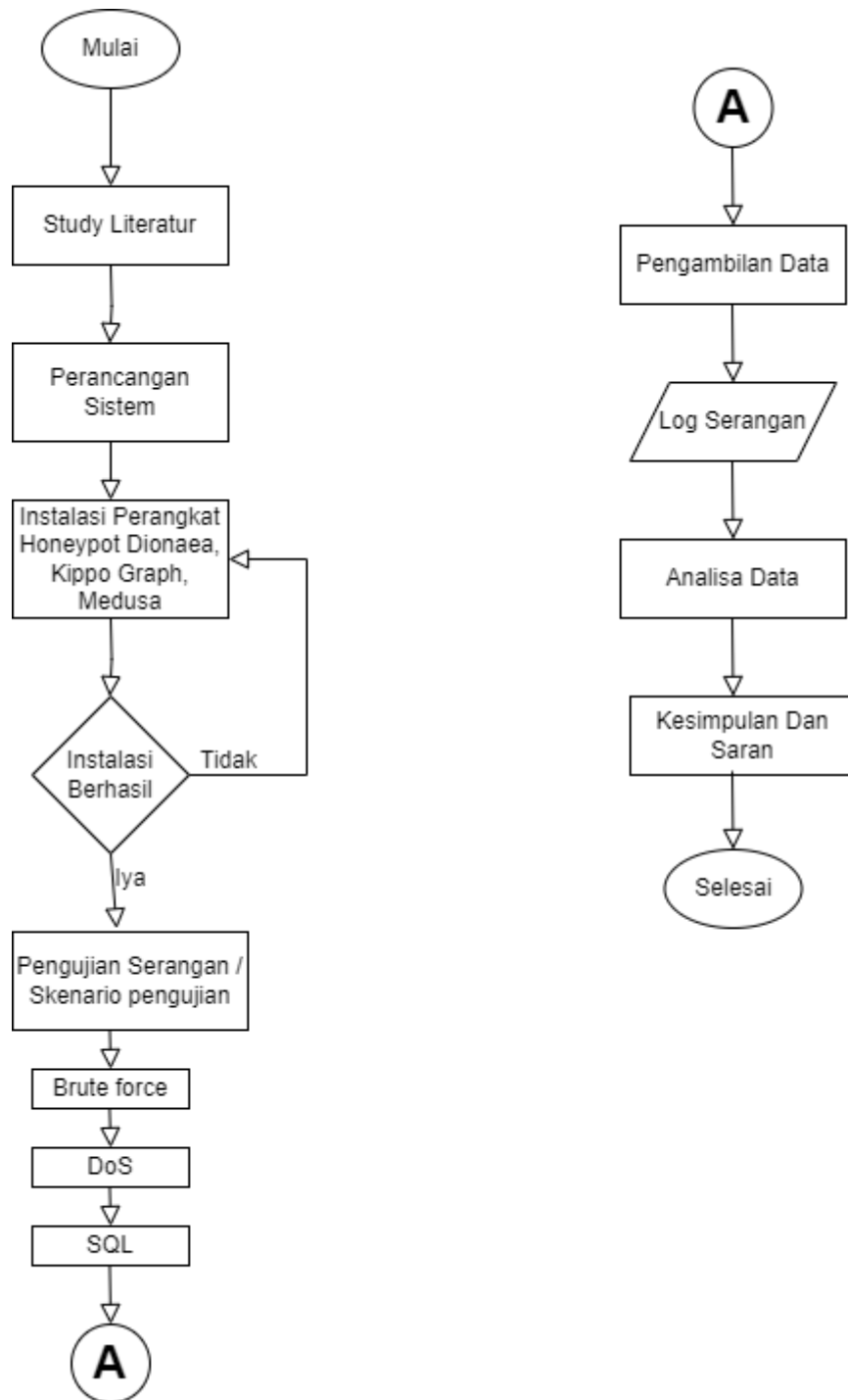
<i>Hardware</i>	<i>Alamat IP</i>
<i>Server Palsu</i>	192.168.18.124/24
<i>Server Asli</i>	192.168.18.106/24
<i>PC Attacker</i>	192.168.18.121/24
<i>PC Client</i>	192.168.18.100/24

*Server* palsu dibangun dengan menggunakan IP 192.168.18.124/24, *server* asli menggunakan IP 192.168.18.106/24, PC penyerang menggunakan IP 192.168.18.121/24 sedangkan PC *client* menggunakan IP 192.168.18.100/24.

### **3.2 DIAGRAM ALUR PENELITIAN**

Diagram alur penelitian ini merupakan beberapa tahap yang diawali dengan menentukan jenis sistem keamanan jaringan tiruan yang menyerupai *server* asli beserta jenis serangan yang akan di uji pada penelitian. Selanjutnya mengetahui apakah sistem tersebut bekerja dengan baik dan dapat memonitoring saat terjadinya serangan pada *server*.

Tujuan dari perancangan ini yaitu untuk mendapatkan hasil dari uji coba sistem pertahanan keamanan jaringan terhadap serangan *cyber* dengan menggunakan dua jenis keamanan jaringan yaitu *honeypot dionaea* dan *kippo*.



**Gambar 3. 2 Alur Penelitian**

Pada gambar 3.2 merupakan penjelasan dari setiap proses yang akan dilakukan saat melakukan penelitian berdasarkan gambar diatas:

1. Studi literatur

Studi literatur bertujuan untuk membandingkan, mengevaluasi atau menyelidiki perkembangan pemahaman topik terbaru. Untuk sumber

dapat diperoleh dari buku, karya ilmiah, laporan, internet serta dokumen-dokumen yang ada.

## 2. Perancangan Sistem

Perancangan sistem dilakukan untuk Menyusun dan mempersiapkan sistem yang akan dijalankan dalam penelitian agar tercapai tujuan yang diharapkan.

## 3. Instalasi perangkat

Setelah mendapatkan sistem yang dibutuhkan selanjutnya melakukan instalasi perangkat pada *server* yang akan dibangun yaitu honeypot *dionaea* dan *kippo*.

## 4. Skenario penyerangan

Skenario penyerangan dilakukan apabila sebelumnya instalasi sistem keamanan sudah berhasil terpasang pada *server*, jenis serangan yang dilakukan pada penelitian ini ada 3 yaitu serangan *brute force*, dos dan sql. Tiga jenis serangan tersebut merupakan serangan yang umum dilakukan oleh *cyber* untuk masuk dan mengacau *server* yang ada.

## 5. Pengambilan data

Pengambilan data dilakukan setelah berhasil melancarkan skenario penyerangan terhadap *honeypot kippo* dan *dionaea*. Pengambilan data berupa hasil serangan *brute force*, dos dan sql.

## 6. Analisis data

Dari data yang didapat selanjutnya analisa terhadap *honeypot*, apakah *honeypot* sudah berhasil bekerja sesuai harapan, apakah *honeypot* memenuhi spesifikasi yang diharapkan, dan bagaimana *honeypot* bekerja.

## 7. Kesimpulan

Kesimpulan dari semua proses Langkah perancangan *honeypot kippo* dan *dionaea*, poin penting apa saja yang perlu diperhatikan, serta evaluasi bagaimana proses perancangan kedepannya. Kesimpulan juga berisi tentang saran terkait penelitian ini.

### 3.3 METODE PENELITIAN

Dalam penelitian ini metode yang digunakan yaitu Analisis *Log* karena *Honeypot* dapat memantau *log* aktivitas untuk mencari tanda-tanda serangan, seperti banyaknya upaya *login* yang gagal, permintaan yang tidak biasa atau pola aktivitas yang mencurigakan dapat menandakan serangan yang sedang berlangsung.

### 3.4 KONFIGURASI SISTEM

Pada penelitian ini konfigurasi sistem dilakukan untuk menunjang keperluan yang dibutuhkan oleh sistem sebelum melakukan pengujian.

#### 3.4.1 KONFIGURASI *APACHE2*

Konfigurasi *Apache2* dilakukan agar web *apache2* dapat berjalan dengan optimal. Untuk memperbarui semua *environment* yang terinstall pada linux menjadi yang terbaru, dapat dipergunakan perintah.

```
# sudo apt update
```

setelah semua paket sudah diperbarui selanjutnya, menginstal *apache2*, dapat dipergunakan perintah.

```
# sudo apt install apache2
```

setelah *apache2* terinstal, selanjutnya mengaktifkan layanan *apache2* dan agar layanan dapat berjalan otomatis setiap dinyalakan dapat dipergunakan perintah.

```
# sudo systemctl start apache2 sudo systemctl enable apache2
```

setelah mengaktifkan layanan *apache2*, kemudian kita dapat memeriksa status *apache2* untuk memastikan *apache2* berjalan dapat dipergunakan perintah.

```
# sudo systemctl status apache2
```

jika *apache2* berjalan dengan baik, kita akan melihat pesan yang menunjukkan status *active* atau *running*.

```
redi@redi-VirtualBox:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor pres
   Active: active (running) since Wed 2023-05-17 14:05:31 WIB; 4min 42s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 4465 (apache2)
    Tasks: 55 (limit: 2279)
   Memory: 5.0M
      CPU: 52ms
   CGroup: /system.slice/apache2.service
           └─4465 /usr/sbin/apache2 -k start
             └─4467 /usr/sbin/apache2 -k start
               └─4468 /usr/sbin/apache2 -k start

Mei 17 14:05:31 redi-VirtualBox systemd[1]: Starting The Apache HTTP Server...
Mei 17 14:05:31 redi-VirtualBox apachectl[4464]: AH00558: apache2: Could not r
Mei 17 14:05:31 redi-VirtualBox systemd[1]: Started The Apache HTTP Server.
lines 1-16/16 (END)
```

**Gambar 3. 3 Tampilan Running Apache2**

Pada gambar 3.3 line 4 menunjukkan bahwa status dari *server apache2* sudah berjalan. Selanjutnya merubah *hostname server*, disini penulis merubah nama *hostname server* yang sebelumnya Bernama redi-VirtualBox menjadi RediServer dengan perintah.

```
# sudo hostnamectl set-hostname RediServer
```

Setelah berhasil merubah *hostnameserver* selanjutnya edit file */etc/hosts* supaya mengganti nama lama dengan nama yang baru yaitu RediServer dengan perintah.

```
# sudo nano /etc/hosts
```

```
redi@RediServer: ~
GNU nano 6.2 /etc/hosts
127.0.0.1    localhost
127.0.1.1   RediServer
192.168.18.99 RediServer

# The following lines are desirable for IPv6 capable hosts
::1        ip6-localhost ip6-loopback
fe00::0    ip6-localnet
ff00::0    ip6-mcastprefix
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
```

**Gambar 3. 4 Tampilan hostname server**

Pada gambar 3.4 line 1-3 menunjukkan IP dan *hostname* yang digunakan pada penelitian ini. Tahap selanjutnya *reboot* system supaya *hostname* bisa berubah, dengan perintah.

```
# sudo reboot
```

```
redi@RediServer:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.18.99 netmask 255.255.255.0 broadcast 192.168.18.255
    inet6 fe80::192:bfca:9cba:7c11 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:92:bd:7f txqueuelen 1000 (Ethernet)
    RX packets 48 bytes 13818 (13.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 104 bytes 12676 (12.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 120 bytes 10130 (10.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 120 bytes 10130 (10.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

**Gambar 3. 5 Tampilan *Hostname Server Asli***

Pada gambar 3.5 line 1 dapat dilihat bahwa *hostname* berhasil kita ubah dan line 3 menunjukkan ip yang digunakan pada *server*.

Setelah selesai, selanjutnya penulis menginstall php pada apache2 dan mariadb dapat digunakan perintah.

```
# sudo apt-get install mysql
# sudo apt-get install php
# sudo apt-get install phpmyadmin
```

Kemudian buka direktori html dan install DVWA pada direktori html agar DVWA berjalan pada *server* apache2

```
# cd /var/www/html
# sudo git clone https://github.com/digininja/DVWA.git
```

Selanjutnya masuk ke direktori DVWA config dan copy file config dapat digunakan perintah.

```
# sudo cp config.inc.php.dist config.inc.php
```

Selanjutnya edit konfigurasi file yang kita copy tadi dapat digunakan perintah.

```
# sudo nano config.inc.php
```



```
red@redserver: /var/www/html/DVWA/config
GNU nano 2.5.3 File: config.inc.php Modified

# Database variables
## WARNING: The database specified under db_database WILL BE ENTIRELY DELETED $
# Please use a database dedicated to DVWA.
#
# If you are using MariaDB then you cannot use root, you must use create a dedi$
# See README.md for more information on this.
$_DVWA = array();
$_DVWA[ 'db_server' ] = '127.0.0.1';
$_DVWA[ 'db_database' ] = 'dvwa';
$_DVWA[ 'db_user' ] = 'root';
$_DVWA[ 'db_password' ] = 'root';
$_DVWA[ 'db_port' ] = '3306';

# ReCAPTCHA settings
# Used for the 'Insecure CAPTCHA' module
# You'll need to generate your own keys at: https://www.google.com/recaptcha/$
$_DVWA[ 'recaptcha_public_key' ] = '6LdtFqMnAAAAACKY7HGA3_m1NK4s78Fbez9x4Kkb';
$_DVWA[ 'recaptcha_private_key' ] = '6LdtFqMnAAAAAD0W6knNZsqrI-bMsJ70vAm2o';

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^_ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

**Gambar 3. 6 Konfigurasi DVWA**

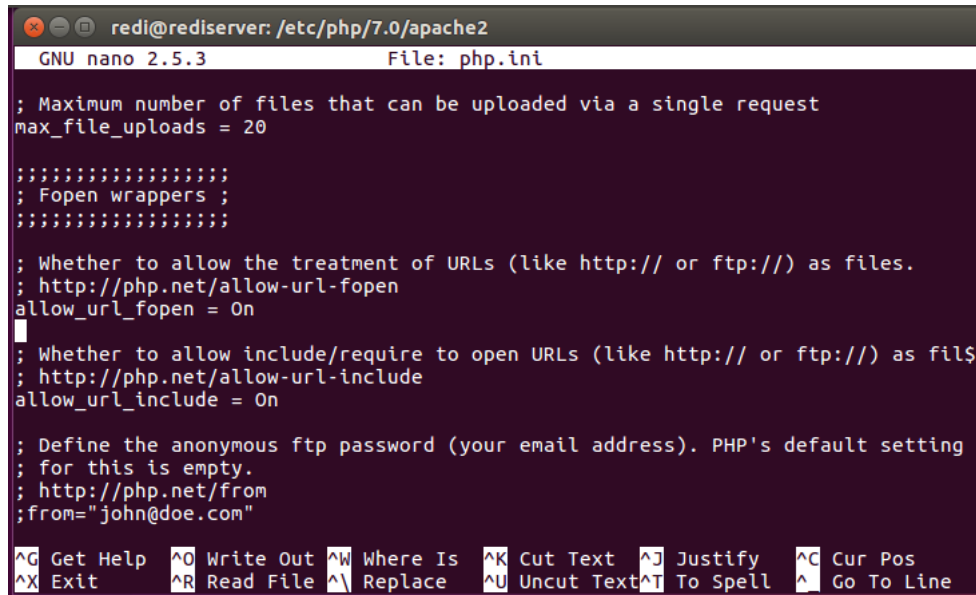
Pada gambar 3.6 line 9 kita harus mengubah db\_database sesuai dengan nama yang akan kita gunakan pada contoh ini menggunakan dvwa sebagai database, pada line 10-11 kita harus merubah db\_user dan db\_password sesuai dengan kita buat sebelumnya pada contoh ini penulis menggunakan user root dan password root. Pada line 17-18 kita harus mengisi recaptcha atau cookies yang dibuat pada <https://www.google.com/recaptcha/>

Selanjutnya memberikan hak izin file dapat digunakan perintah.

```
# sudo chgrp www-data hackable/uploads
# sudo chgrp www-data config
# sudo chmod g+w hackable/uploads
# sudo chmod g+w config
```

Selanjutnya edit direktori php pada apache2 untuk mengaktifkan url DVWA dapat digunakan perintah.

```
# cd etc/php/7.0.3/apache2/
# sudo nano php.ini
```



```
redl@rediserver: /etc/php/7.0/apache2
GNU nano 2.5.3 File: php.ini
; Maximum number of files that can be uploaded via a single request
max_file_uploads = 20

;;;;;;;;;;;;;;;;;;;;;;;;;
; Fopen wrappers ;
;;;;;;;;;;;;;;;;;;;;;;;;;

; Whether to allow the treatment of URLs (like http:// or ftp://) as files.
; http://php.net/allow-url-fopen
allow_url_fopen = On

; Whether to allow include/require to open URLs (like http:// or ftp://) as files
; http://php.net/allow-url-include
allow_url_include = On

; Define the anonymous ftp password (your email address). PHP's default setting
; for this is empty.
; http://php.net/from
;from="john@doe.com"

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

**Gambar 3. 7 Konfigurasi php.ini**

Pada gambar 3.7 line 10 dan 11 kita harus merubah allow\_url yang sebelumnya off menjadi on agar bisa diakses oleh orang lain. Selanjutnya simpan perubahan tersebut.

### 3.4.2 KONFIGURASI HONEYPOT KIPPO GRAPH

Konfigurasi *honeypot* kippo graph digunakan untuk membangun *honeypot* atau perangkat palsu yang meniru *server* SSH. Kippo dapat merekam setiap upaya login palsu atau percobaan lain yang mencurigakan dan akan menyimpannya dalam format *log* yang dapat dianalisis.

Pertama unduh kippo graph, untuk mengunduh kippo graph dapat digunakan perintah.

```
# git clone https://github.com/desaster/kippo.git
```

kemudian masuk ke direktori kippo kemudian edit file konfigurasi yang perlu disesuaikan sebelum dapat digunakan. Untuk salin file konfigurasi dapat digunakan perintah.

```
# cd honeydrive/kippo
# cp kippo.cfg.dist kippo.cfg
```

Setelah Kippo Graph terpasang dan di salin, selanjutnya disini penulis membuka file kippo untuk mengedit port pada kippo.cfg perintah yang digunakan seperti dibawah ini.

```
#sudo nano kippo.cfg
```

Edit file *config* kippo, dengan merubah *ssh\_port* menjadi 22 dan *hostname* menjadi *RediServer*, supaya si penyerang akan mengira bahwa *hostname* tersebut merupakan *hostname* asli *server*.

```
# Port to listen for incoming SSH connections.
#
# (default: 2222)
ssh_port = 22

# Hostname for the honeypot. Displayed by the shell prompt of the virtual
# environment.
#
# (default: svr03)
hostname = rediserver

# Directory where to save log files in.
#
# (default: log)
log_path = log

# Directory where to save downloaded (malware) files in.
#
```

**Gambar 3. 8 Tampilan Edit File Config Kippo**

Pada gambar 3.8 line 4 kita akan merubah *ssh\_port kippo* sama seperti *port ssh* yang digunakan oleh *server* agar penyerang mengira *port ssh* tersebut merupakan *port ssh* asli. Pada line 9 kita juga dapat merubah *hostname kippo* menjadi *hostname* yang digunakan sama *server* asli.

Selanjutnya jika ingin melihat atau menambahkan *username* dan *password* dapat menggunakan perintah.

```
#sudo nano userdb.txt
```

```
root:0:123456

[ Read 1 line ]
^G Get Help  ^O WriteOut  ^R Read File  ^V Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^N Next Page  ^U UnCut Text ^T To Spell
```

**Gambar 3. 9 Username Dan Password**

Pada gambar 3.9 line 1 merupakan *username* dan *password* yang digunakan oleh *kippo*. Kita dapat merubah *username* dan *password* sesuai dengan kita inginkan akan tetapi penulis tidak merubah apapun hanya memakai pengaturan *default* dari *kippo*.

Selanjutnya menjalankan *kippo* dengan perintah.

```
# ./start.sh
```

Setelah semua selesai untuk melihat *kippo* berjalan secara *interactive* dapat menggunakan perintah.

```
# tail -f log/kippo.log
```

```
honeydrive@honeydrive:/honeydrive/kippo$ ./start.sh
Starting kippo in the background...

Loading dblog engine: mysql
honeydrive@honeydrive:/honeydrive/kippo$ tail -f log/kippo.log
2023-10-18 16:04:56+0100 [-] File "/usr/lib/python2.7/dist-packages/twisted/internet/posixbase.py", line 495, in listenTCP
2023-10-18 16:04:56+0100 [-] p.startListening()
2023-10-18 16:04:56+0100 [-] File "/usr/lib/python2.7/dist-packages/twisted/internet/tcp.py", line 980, in startListening
2023-10-18 16:04:56+0100 [-] raise CannotListenError(self.interface, self.port, le)
2023-10-18 16:04:56+0100 [-] twisted.internet.error.CannotListenError: Couldn't listen on 0.0.0.0:22: [Errno 98] Address already in use.
2023-10-18 16:05:25+0100 [-] Log opened.
2023-10-18 16:05:25+0100 [-] twistd 13.2.0 (/usr/bin/python 2.7.6) starting up.
2023-10-18 16:05:25+0100 [-] reactor class: twisted.internet.epollreactor.EPollReactor.
2023-10-18 16:05:25+0100 [-] HoneyPotSSHFactory starting on 2222
2023-10-18 16:05:25+0100 [-] Starting factory <kippo.core.honeypot.HoneyPotSSHFactory instance at 0xb634944c>
```

### Gambar 3. 10 Tampilan Kippo Secara Interactive

Pada gambar 3.10 line 19-23 memberitahu adanya koneksi baru yang ingin mengakses *server* *kippo*. Selanjutnya penulis ingin menguji apakah *kippo* sudah berhasil dan ingin melihat dari sudut pandang penyerang. Penulis menggunakan *puTTY* sebagai penyerang dan untuk masuk ke dalam *server*, berikut merupakan tampilan dari *puTTY* yang berhasil mengakses *server* tiruan.

```
192.168.18.100 - PuTTY
login as: root
Keyboard-interactive authentication prompts from server:
| Password:
| End of keyboard-interactive prompts from server
root@RediServer:~# ls
root@RediServer:~# ls -all
drwxr-xr-x 1 root root 4096 2023-05-26 08:50 .
drwxr-xr-x 1 root root 4096 2023-05-26 08:50 ..
-rw-r--r-- 1 root root 140 2013-04-05 12:52 .profile
drwx----- 1 root root 4096 2013-04-05 13:05 .ssh
drwx----- 1 root root 4096 2013-04-05 12:58 .aptitude
-rw-r--r-- 1 root root 570 2013-04-05 12:52 .bashrc
root@RediServer:~# █
```

**Gambar 3. 11 Tampilan Putty**

Pada gambar 3.11 line 5 dapat dilihat bahwa penyerang berhasil mendapatkan *hostname* yang dibuat oleh *kippo*.

### 3.4.3 KONFIGURASI *HONEYPOT* DIONAEA

Konfigurasi *honeypot* Dionaea digunakan untuk membangun *honeypot* atau perangkat palsu yang rentan terhadap serangan seperti FTP, HTTP, SMB dan lain-lain. Dionaea dapat merekam setiap upaya yang mencurigakan dan menyimpannya dalam format *log* yang dapat dianalisis.

Pada tahap ini update atau perbarui semua *environment* yang terinstall pada linux menjadi yang terbaru, dapat dipergunakan perintah.

```
# sudo apt update
```

Setelah selesai sistem sudah terupdate semua, Langkah selanjutnya menginstall dependensi Dionaea dapat dipergunakan perintah.

```
# sudo apt install build-essential cmake libssl-dev libffi-dev python-dev python2.7-dev python3-dev python-pip python3-pip libpcap-dev libglib2.0-dev libyaml-dev libtool autoconf automake git
```

selanjutnya intall Dionaea dengan perintah.

```
# git clone https://github.com/rubenespadas/DionaeaFR
```

setelah berhasil terinstall, jalankan direktori dionaea dapat dipergunakan perintah.

```
# cd /honeydrive/DionaeaFR
```

untuk mengatur konfigurasi Dionaea dapat dipergunakan perintah.

```
#sudo nano setting.py
```

Setelah semua selesai terinstall, tahap selanjutnya kita hanya perlu menjalankan Dionaea yaitu dengan memasukkan perintah.

```
# python manage.py runserver 192.168.18.124:8080
```

```
honeydrive@honeydrive:/honeydrive/DionaeaFR$ python manage.py runserver 192.168.18.154:8080
Validating models...

0 errors found
October 18, 2023 - 15:08:32
Django version 1.6.5, using settings 'DionaeaFR.settings'
Starting development server at http://192.168.18.154:8080/
Quit the server with CONTROL-C.
```

**Gambar 3. 12 Tampilan Dionaea**

Pada gambar 3.12 dapat kita lihat pada line 8 dionaea berhasil menjalankan *server* dari ip 192.168.18.124:8080 dan setiap kegiatan serangan *malware* akan langsung terekam oleh *Dionaea*.

### **3.5 PERANCANGAN SKENARIO**

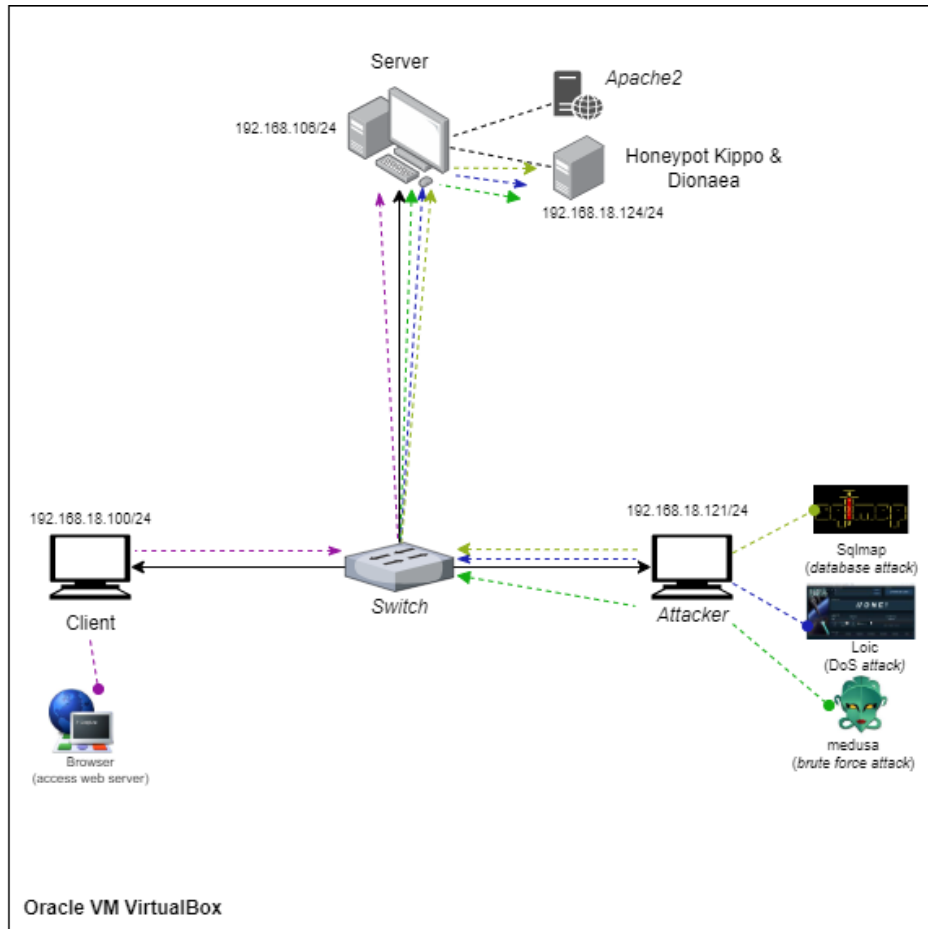
Skenario dilakukan saat semua perangkat yang dibutuhkan berhasil terpasang. Terdapat dua skenario pada penelitian ini yaitu pengujian sistem dan skenario tiap serangan.

#### **3.5.1 Skenario Pengujian**

Skenario pengujian sistem adalah skenario sistem yang di rancang sebagai jalannya proses penelitian yang terdiri dari dua skenario. Skenario pertama yaitu penyerang akan menyerang *server* tiruan dan skenario kedua penyerang akan menyerang *server* asli akan tetapi *server* asli akan diperkuat saat penulis berhasil mendapatkan info penyerang dari *server* tiruan.

##### **3. 5.1.1 Skenario Pertama**

Pada skenario pertama, PC *server* sudah berisikan *server honeypot*, kemudian *server honeypot* akan diaktifkan untuk mendeteksi semua koneksi yang masuk pada *port 22* dan *port 8080*. Untuk alur dari skenario pertama dapat dilihat pada gambar 3.15.

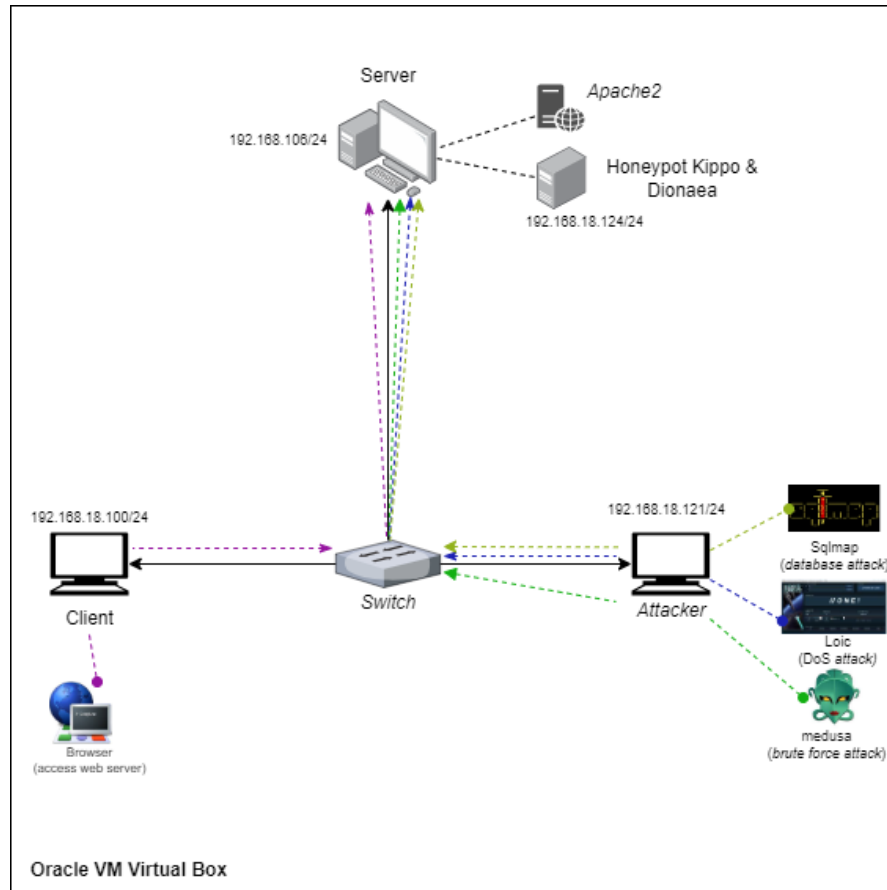


**Gambar 3. 13 Skenario Pertama (Tanpa Blok IP)**

Pada gambar 3.13 alur pengujian skenario pertama dimulai dengan anak panah berwarna ungu pada PC *Client* yang mengakses web *server* menggunakan url `http://192.168.18.106` melalui *web browser* secara *idle*, selama proses penyerangan oleh PC *attacker* baik *brute force*, DoS dan SQL. Pada saat bersamaan PC *attacker* melakukan serangannya mulai dari *brute force* yang ditandai oleh anak panah hijau dengan menggunakan jaringan internet yang sama pada *client* dan *server* menuju *honeypot*. Selanjutnya serangan DoS yang ditandai dengan warna anak panah biru dan serangan terakhir serangan *Sqlmap* ditandai dengan warna anak panah kuning. Setiap serangan yang dilancarkan oleh *attacker* *honeypot* akan memonitoring dan merekam setiap serangan yang dilancarkan dan disimpan dalam bentuk file log.

### 3.5.1.2 Skenario Kedua

Pada skenario kedua, penyerang akan menyerang *server* asli akan tetapi penulis akan memperkuat *server* asli dengan cara memblokir IP penyerang yang tercatat dari hasil *log* yang didapatkan dari skenario pertama. Adapun alur dari skenario kedua dapat dilihat pada gambar.



**Gambar 3. 14 Skenario Kedua (penyerang menyerang *server* asli)**

Pada gambar 3.14 alur skenario kedua sama halnya pada alur skenario pertama. Namun yang membedakan antara skenario pertama dan kedua PC *server* sudah memblokir IP dan *port* yang dilalui oleh *attacker* yang didapatkan dari hasil monitoring dan log pada *server honeypot*. Sehingga penyerang gagal saat ingin masuk ke *server* asli, begitupun saat penyerang ingin melancarkan serangan *brute force*, dos dan sql. Saat pengujian serangan *brute force* pada penelitian ini akan ada dua tipe pemblokiran ip pertama dilakukan secara otomatis menggunakan *fail2ban* yang akan dipasang pada *server* tiruan dan yang kedua dilakukan secara manual pada *server* asli dengan cara melihat hasil dari *log kippo*. Ketika keduanya dijalankan maka salah satu dari *kippo* atau *fail2ban* tidak akan mendapatkan data dari



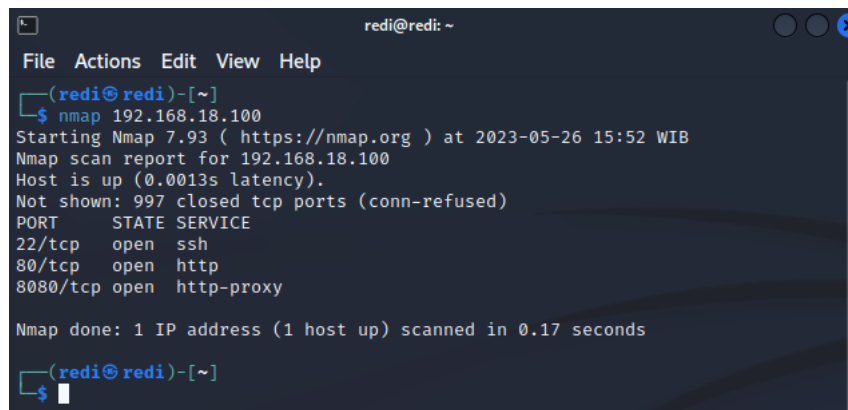
penyerang. Pada skenario kedua ini *client* juga akan mengakses *server* dan penulis ingin melihat perbedaan antara *PC client* dan *PC attacker*.

### 3.5.2 Skenario Penyerangan

Pada penelitian ini akan menguji tiga skenario penyerangan terhadap *server* yang sudah terpasang *honeypot*. Serangan ini akan direkam dan disimpan dalam *log file* oleh *honeypot*. Sebelum melakukan serangan pertama, penulis menggunakan Nmap atau *port scanning* untuk melihat *port port* pada jaringan yang terbuka pada *server target*. *Port scanning* adalah teknik yang digunakan untuk menemukan saluran komunikasi pada suatu jaringan dan kemudian menyimpan informasi yang dapat digunakan untuk melakukan serangan. Dengan menggunakan ekstrusi *log data*, pola distribusi serangan *port scanning* dapat divisualisasikan, yang menunjukkan bagaimana serangan terjadi pada jaringan *server*. Visualisasi ini dapat membantu administrator menjaga jaringan mereka agar tetap aman [20]. Dalam situasi ini, penyerang akan mencoba melakukan serangan *port scanning* menggunakan kondisi yang sudah terpasang *honeypot*.

Pada *OS Kali Linux* untuk melakukan *scanning port* dapat dipergunakan perintah berikut:

```
# nmap <ip target>
```



```
redi@redi: ~  
File Actions Edit View Help  
redi@redi ~  
└─$ nmap 192.168.18.100  
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-26 15:52 WIB  
Nmap scan report for 192.168.18.100  
Host is up (0.0013s latency).  
Not shown: 997 closed tcp ports (conn-refused)  
PORT      STATE SERVICE  
22/tcp    open  ssh  
80/tcp    open  http  
8080/tcp  open  http-proxy  
  
Nmap done: 1 IP address (1 host up) scanned in 0.17 seconds  
redi@redi ~  
└─$
```

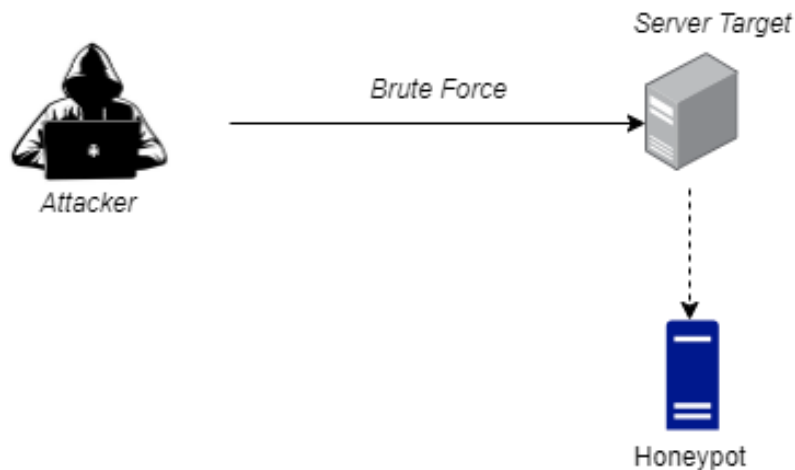
**Gambar 3. 15 Tampilan Nmap**

Pada gambar 3.15 *line 1 tools* yang digunakan untuk melancarkan serangan *scanning* menggunakan nmap dan ip yang akan diserang yaitu 192.168.18.100 dan nmap berhasil mendapatkan data *port* yang terbuka seperti pada line 7-9 yaitu *port 22 ssh*, *port 80 http* dan *port 8080 http-proxy*. penelitian kali ini pengujian sistem akan menyerang *port* yang terbuka untuk *port 22* dimana akan digunakan

sebagai penyerangan *brute force*, *port* 80 untuk melancarkan serangan DoS. Sedangkan untuk *sqlmap* akan menggunakan *url* yang dibuat oleh *dionaea*. selanjutnya penulis melancarkan tiga skenario penyerangan. Berikut tiga serangan yang akan dilakukan:

### 3.5.2.1 Skenario Pertama

Skenario pertama melancarkan serangan *brute force* bertujuan untuk melakukan *remote control* dengan cara mencoba memasukan *username* dan *password* yang sama dengan target *server* yang dituju. Pada saat melakukan serangan *brute force* penulis akan menyiapkan *username* dan *password* yang akan di ujicoba pada *server* target. Hasil dari penyerangan *brute force* ini untuk mengetahui integritas dari sistem yang di implementasi. Pada penelitian ini penulis menggunakan *medusa* dalam melancarkan serangan *brute force*.



**Gambar 3. 16 Serangan *Brute Force***

Pada gambar 3.16 merupakan alur serangan *brute force* dari *attacker* saat menyerang *server*. Pada server target akan dipasangkan *honeypot* untuk *monitoring* atau merekam *log* aktifitas *attacker*.

Untuk menyiapkan *username* yang akan diserang dapat menggunakan perintah:

```
# crunch <min> <max> karakter -o <nama file yang ingin disimpan>
```

```
(redi@redi)-[~/Desktop]
└─$ crunch 3 5 root -o username.txt
Crunch will now generate the following amount of data: 1971 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 351
crunch: 100% completed generating output
```

**Gambar 3. 17 Username.txt**

Pada gambar 3.17 menjelaskan tentang pembuatan daftar kata sandi yang akan digunakan pada serangan *brute force* yang akan di simpan pada halaman desktop, perintah *crunch* digunakan untuk membuat daftar kata sandi yang diawali dengan minimal 3 kata dan maksimal 5 kata. Pembuatan kata yang akan dibuat yaitu *root* kemudian perintah *-o* digunakan untuk menentukan berkas yang akan dikeluarkan dengan diberi nama *username.txt*. kemudian *crunch* berhasil membuat kata-kata yang akan digunakan sebanyak 351 kata.


```
(redi@redi)-[~/Desktop]
└─$ crunch 5 7 1234567 -o passwd.txt
Crunch will now generate the following amount of data: 7512729 bytes
7 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 957999
crunch: 100% completed generating output
```

**Gambar 3. 18 Password.txt**

Pada gambar 3.18 sama seperti penjelasan sebelumnya akan tetapi perbedaannya penulis membuat minimal 5 kata dan maxsimal 7 kata dan pembuatan kata dengan angka 1234567 kemudian diberi nama *passwd.txt* kemudian *crunch* berhasil membuat kata-kata yang akan digunakan sebanyak 957999 kata.

Setelah selesai membuat *username* dan *passwd* untuk menyerang *server* target. Selanjutnya menyerang *server* dengan perintah yang dipergunakan untuk melakukan serangan *brute force* dengan medusa adalah sebagai berikut:

```
medusa -h <IP Addres Target> -u <Username server> -p <password server> -n <port target> -m ssh
```



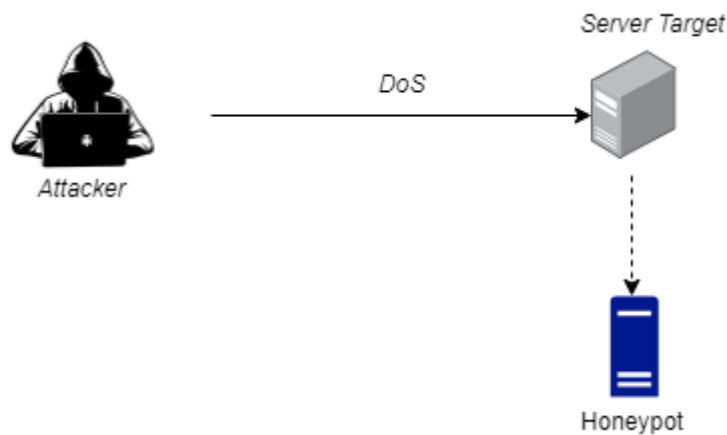
```
redi@redi: ~
File Actions Edit View Help
(redi@redi)-[~]
$ medusa -h 192.168.18.100 -u /home/redi/Desktop/username.txt -P /home/redi/Desktop/passwd.txt -n 22 -M ssh
```

**Gambar 3. 19 Serangan Medusa**

Pada gambar 3.19 *tools* atau alat untuk uji coba penyerangan menggunakan *medusa*, perintah *-h* merupakan perintah untuk menentukan ip yang akan diserang, contoh pada penelitian ini ip yang diserang yaitu 192.168.18.100, perintah *-U* merupakan perintah yang digunakan untuk menentukan *user*, pada contoh kali ini untuk menentukan *user* yang akan digunakan terletak pada file yang sebelumnya sudah kita buat di halaman *desktop* dengan nama *username.txt* dan perintah *-P* digunakan untuk menentukan perintah *password*, contoh pada penelitian ini file yang akan digunakan terletak di halaman *desktop* dengan nama *passwd.txt*. perintah *-n* merupakan perintah untuk menentukan port yang akan diserang, pada contoh ini port yang dituju yaitu 22 dan perintah *-M* merupakan perintah untuk memanggil Module yang akan diserang, contoh module yang diserang merupakan module *ssh*.

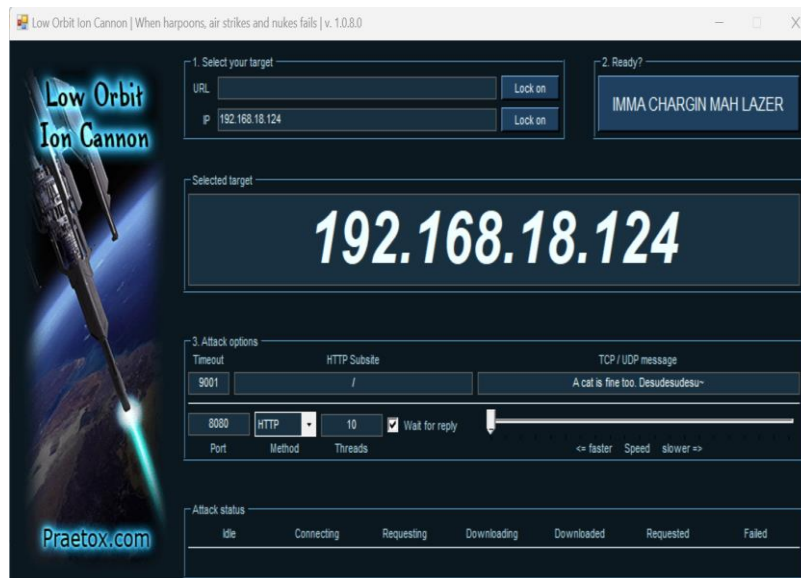
### 3.5.2.2 Skenario Kedua

Skenario kedua yaitu serangan DoS (*Daniel of Service*) merupakan serangan yang membanjiri trafik atau mengirim paket-paket dalam jumlah banyak pada jaringan sehingga akses untuk masuk ke jaringan tersebut tidak bisa terjangkau dan *server* akan mengalami *hank*. Sebelum melakukan serangan DoS penyerang harus terlebih dahulu melakukan *scanning* pada jaringan yang akan diserang dan melihat *port* mana saja yang terbuka sehingga bisa dilakukan penyerangan, biasanya *port* yang dipilih untuk melancarkan serangan DoS memilih *port* 80. Pada penelitian ini penulis menggunakan LOIC untuk melancarkan serangan DoS.



**Gambar 3. 20 Serangan DoS**

Pada gambar 3.20 merupakan alur saat *attacker* melancarkan serangan DoS kepada *server target* yang sudah dipasangkan *honeypot*.



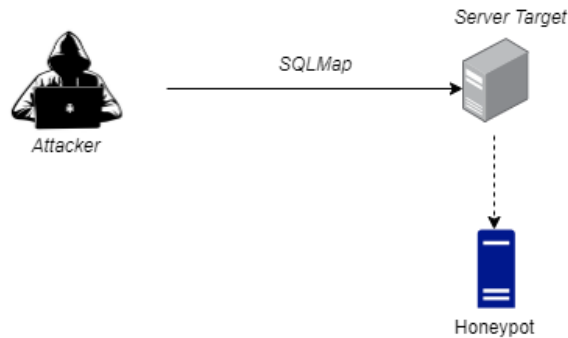
**Gambar 3. 21 LOIC**

Pada gambar 3.21 untuk melancarkan serangan loic kita hanya perlu mengetahui ip yang akan diserang, contoh ip yang diserang yaitu 192.168.18.124, dan *port* yang dituju yaitu 8080 dengan metode http dan untuk setiap kali penyerangan loic akan mengirimkan 10 permintaan per detik ke *server* yang dituju. Apabila kita sudah siap melancarkan serangan kita hanya perlu mengklik IMMA CHARGIN MAH LAZER.

### 3.5.2.3 Skenario Ketiga

Pada skenario terakhir penulis melancarkan serangan SQL bertujuan untuk mengeksploitasi kelemahan dalam aplikasi web yang menggunakan

SQL untuk menginteraksi dengan *database* atau memanfaatkan kerentanan dalam aplikasi yang menggunakan input pengguna dalam kueri SQL tanpa melakukan sanitasi atau validasi yang memadai. Pada penelitian ini penulis menggunakan serangan SQLMap untuk melancarkan serangan.



**Gambar 3. 22 Serangan SQLMap**

Pada gambar 3.22 merupakan alur dari *attacker* saat ingin melancarkan serangan SQLMap kepada *server target* yang sudah dipasangkan *honeypot*.

Terdapat empat perintah yang digunakan untuk melancarkan serangan SQLMap yaitu:

```
#sqlmap -u <url> --cookie <cookie website> --dbs
```

```
(root@redi)-[~]
# sqlmap -u 'http://192.168.0.185/DVWA-1.9/vulnerabilities/sqli/?id=1&Submit=Submit#' --cookie="PHPSESSID=82gek1e22lmi7jg4v7tatkc7d0;security=low" --dbs
```

**Gambar 3. 23 Perintah Serangan Sqlmap database**

Pada gambar 3.23 merupakan potongan kode untuk uji penetrasi pada kerentanan SQL Injection pada URL target `http://192.168.0.185/DVWA-1.9/`. Sqlmap merupakan alat uji penetrasi atau nama perangkat lunak yang dijalankan untuk mendeteksi dan mengeksploitasi kerentanan SQL Injection. Perintah `-u` digunakan untuk menentukan URL target yang ingin diuji. Perintah `--cookie` digunakan untuk menyertakan informasi kuki dalam permintaan HTTP. Perintah `-dbs` digunakan untuk menginstruksikan SQLMap untuk mencari daftar basis data yang ada di *server* yang ditargetkan.

```
#sqlmap -u <url> -- cookie<cookie website> -D <database> --tables
```

```
(root@redi)-[~]
# sqlmap -u 'http://192.168.0.185/DVWA-1.9/vulnerabilities/sqli/?id=16Submit=Submit#' --cookie="PHPSESSID=82gekie22lmi7jg4v7atkc7d0;security=low" -D dvwa --tables
```

**Gambar 3. 24 Perintah Membuka Database dvwa**

Pada gambar 3.24 merupakan potongan kode untuk uji penetrasi pada kerentanan SQL Injection pada URL target <http://192.168.0.185/DVWA-1.9/>. Sqlmap merupakan alat uji penetrasi atau nama perangkat lunak yang dijalankan untuk mendeteksi dan mengeksploitasi kerentanan SQL Injection. Perintah -u digunakan untuk menentukan URL target yang ingin diuji. Perintah --cookie digunakan untuk menyertakan informasi kuki dalam permintaan HTTP. Perintah -D digunakan untuk menentukan nama basis data yang ingin dijelajah. Perintah --tables merupakan instruksi sqlmap untuk mengambil daftar table dalam basis data yang ditentukan sebelumnya.

```
#sqlmap -u <url> -- cookie<cookie website> -D <database> --columns -D <database> -T <tabel pada database>
```

```
(root@redi)-[~]
# sqlmap -u 'http://192.168.0.185/DVWA-1.9/vulnerabilities/sqli/?id=16Submit=Submit#' --cookie="PHPSESSID=82gekie22lmi7jg4v7atkc7d0;security=low" --columns -D dvwa -T users
```

**Gambar 3. 25 Perintah Membuka Database dvwa Table Users**

Pada gambar 3.25 merupakan potongan kode untuk uji penetrasi pada kerentanan SQL Injection pada URL target <http://192.168.0.185/DVWA-1.9/>. Sqlmap merupakan alat uji penetrasi atau nama perangkat lunak yang dijalankan untuk mendeteksi dan mengeksploitasi kerentanan SQL Injection. Perintah -u digunakan untuk menentukan URL target yang ingin di uji. Perintah --cookie digunakan untuk menyertakan informasi kuki dalam permintaan HTTP. Perintah -D digunakan untuk menentukan nama basis data yang ingin dijelajah. Perintah --columns digunakan untuk menginstruksikan sqlmap untuk mencari kolom dalam tabel yang ditentukan nanti. Perintah -T digunakan untuk menentukan nama tabel yang akan dijelajahi.

```
#sqlmap -u <url> -- cookie<cookie website> -D <database> --columns -T <tabel pada database> -C <kolom-kolom yang ingin diekstrak dari tabel> --dump
```

```
(redi@redi)-[~]
# sqlmap -u "http://192.168.18.124/DVWA-1.9/vulnerabilities/sqli/?id=16Submit=Submit#" --cookie="PHPSESSID=tmmespojbt19kf6sfeks3gid6;security=low" -D dvwa -T users -C user,password,first_name,last_name --dump
```

**Gambar 3. 26 Perintah Membuka Database dvwa user dan password**

Pada gambar 3.26 merupakan potongan kode untuk uji penetrasi pada kerentanan SQL Injection pada URL target `http://192.168.0.185/DVWA-1.9/`. `Sqlmap` merupakan alat uji penetrasi atau nama perangkat lunak yang dijalankan untuk mendeteksi dan mengeksploitasi kerentanan SQL Injection. Perintah `-u` digunakan untuk menentukan URL target yang ingin diuji. Perintah `--cookie` digunakan untuk menyertakan informasi kuki dalam permintaan HTTP. Perintah `-D` digunakan untuk menentukan nama basis data yang ingin dijelajah. Perintah `-T` digunakan untuk menentukan nama tabel yang akan dijelajahi. Perintah `-C` digunakan untuk menentukan kolom-kolom yang ingin diekstrak dari tabel "users". Perintah `-dump` digunakan untuk mengambil data dari kolom yang ditentukan dalam bentuk hasil.