LAPORAN ADVANCED NETWORK SECURITY AND PROTOCOLS

“Implementasi Honeypot Sebagai Pendeteksi Serangan VPS”

DISUSUN OLEH :

ANDINI FEBRIANTI	**105841113223**
NUR QAMARIAH YUNUS	**105841104323**

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNIK

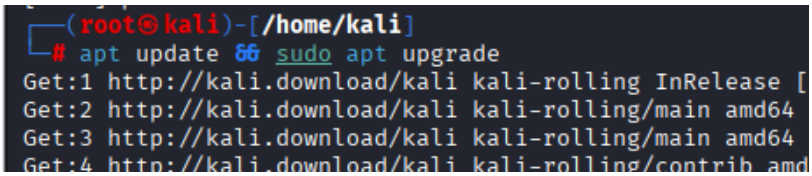
UNIVERSITAS MUHAMMADIYAH MAKASSAR

2026

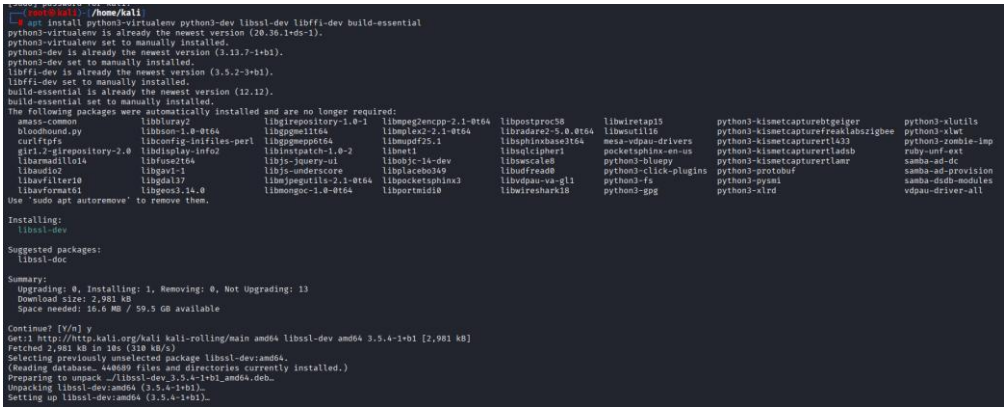
A. Implementasi honeypot

1. Install paket pendukung

Pada tahap awal implementasi, dilakukan pembaruan sistem operasi Kali Linux menggunakan perintah apt update dan apt upgrade. Proses ini bertujuan untuk memperbarui daftar repository serta menginstal pembaruan paket terbaru agar sistem dalam kondisi stabil dan aman sebelum melakukan instalasi honeypot. Pembaruan sistem penting dilakukan untuk menghindari error dependency serta memastikan seluruh library yang dibutuhkan tersedia dalam versi terbaru.



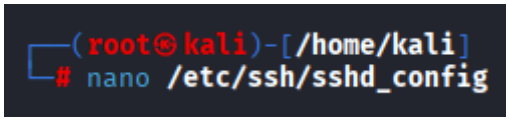
Gambar 1 Proses Pembaruan Repository dan Paket pada Kali Linux



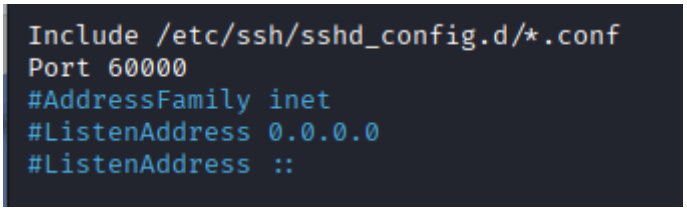
Gambar 2 Proses Instalasi Dependency Pendukung Honeypot

2. Ubah port SSH 22 ke port SSH 2222 untuk server, lalu restart.

Pada tahap ini dilakukan konfigurasi layanan SSH dengan mengakses file sshd_config menggunakan perintah nano /etc/ssh/sshd_config. File konfigurasi ini digunakan untuk mengatur pengaturan utama SSH server, termasuk perubahan port default dari 22 menjadi 2222. Perubahan ini dilakukan untuk mengamankan server asli dan mengalihkan port 22 agar dapat digunakan oleh honeypot sebagai umpan bagi attacker.

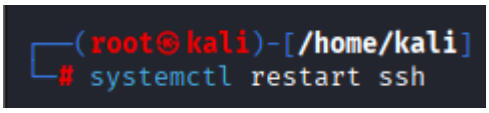


Gambar 3 Proses Akses File Konfigurasi SSH Server



Gambar 4 Perubahan Konfigurasi Port pada File sshd_config

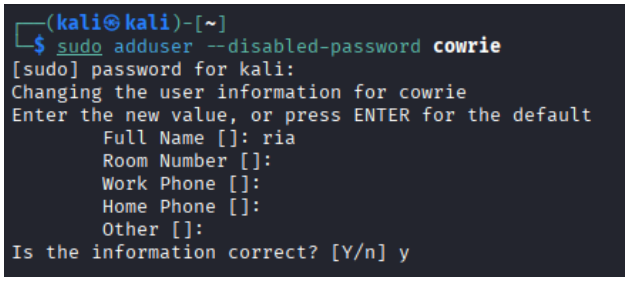
Setelah melakukan perubahan konfigurasi pada file sshd_config, langkah selanjutnya adalah me-restart layanan SSH menggunakan perintah systemctl restart ssh. Proses ini dilakukan agar konfigurasi baru, termasuk perubahan port, dapat diterapkan dan berjalan secara aktif pada sistem. Tanpa melakukan restart service, perubahan konfigurasi tidak akan berpengaruh terhadap layanan SSH yang sedang berjalan.



Gambar 5 Proses Restart Layanan SSH

3. Install honeypot cowrie
   - Buat user khusus, jangan jalankan sebagai root

     Pada tahap ini dilakukan pembuatan user khusus bernama **cowrie** menggunakan perintah sudo adduser --disabled-password cowrie. User ini dibuat tanpa password login langsung sebagai langkah keamanan agar honeypot tidak dijalankan menggunakan akun root. Penggunaan user khusus bertujuan untuk membatasi hak akses sistem sehingga jika terjadi eksploitasi pada honeypot, dampaknya tidak langsung mempengaruhi sistem utama.



Gambar 6 Proses Pembuatan User Khusus Cowrie

   - Pindah ke user cowrie

     Setelah user *cowrie* berhasil dibuat, langkah berikutnya adalah berpindah ke user tersebut menggunakan perintah sudo su - cowrie. Proses ini dilakukan agar seluruh instalasi dan konfigurasi honeypot dijalankan menggunakan akun non-root. Hal ini bertujuan untuk meningkatkan keamanan sistem dengan menerapkan prinsip *least privilege*, sehingga honeypot tidak memiliki hak akses penuh terhadap sistem utama.

Gambar 7 Proses Berpindah ke User Cowrie

- **Clone source code:** git clone http://github.com/cowrie/cowrie

  Pada tahap ini dilakukan proses pengambilan source code honeypot Cowrie dari repository resmi GitHub menggunakan perintah git clone http://github.com/cowrie/cowrie. Proses cloning ini bertujuan untuk mengunduh seluruh file dan struktur program Cowrie ke dalam sistem lokal. Setelah proses selesai, folder cowrie akan terbentuk dan berisi seluruh file konfigurasi serta script yang dibutuhkan untuk instalasi dan menjalankan honeypot

```
(cowrie㉿kali)-[~]
$ git clone http://github.com/cowrie/cowrie
Cloning into 'cowrie' ...
warning: redirecting to https://github.com/cowrie/cowrie/
remote: Enumerating objects: 20802, done.
remote: Counting objects: 100% (65/65), done.
remote: Compressing objects: 100% (49/49), done.
remote: Total 20802 (delta 40), reused 18 (delta 16), pack-reused 20737 (from 2)
Receiving objects: 100% (20802/20802), 11.02 MiB | 4.73 MiB/s, done.
Resolving deltas: 100% (14551/14551), done.
```

Gambar 8 Proses Cloning Repository Cowrie dari GitHub

- Setup virtual environment

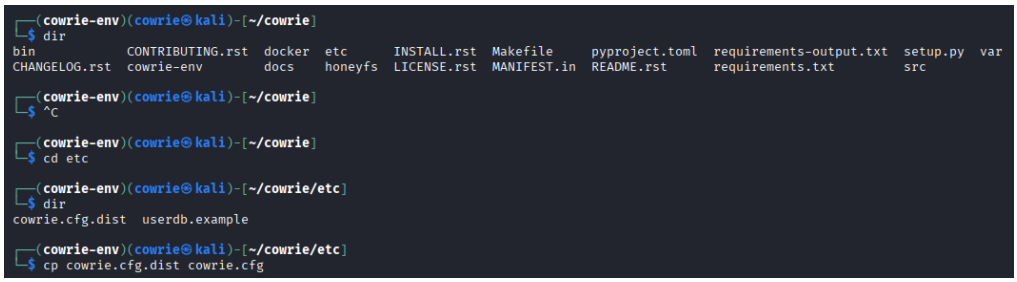
  Pada tahap ini dilakukan pembuatan dan aktivasi virtual environment untuk menjalankan honeypot Cowrie. Virtual environment dibuat menggunakan perintah virtualenv --python=python3 cowrie-env, kemudian diaktifkan dengan source cowrie-env/bin/activate. Setelah environment aktif, dilakukan pembaruan pip dan instalasi seluruh dependency yang tercantum dalam file requirements.txt.

```
(cowrie㉿kali)-[~]
$ cd cowrie

(cowrie㉿kali)-[~/cowrie]
$ virtualenv --python=python3 cowrie-env
created virtual environment CPython3.13.11.final.0-64 in 532ms
  creator CPython3Posix(dest=/home/cowrie/cowrie/cowrie-env, clear=False, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, via=copy, app_data_dir=/home/cowrie/.local/share/virtualenv)
    added seed packages: pip==25.3
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator

(cowrie㉿kali)-[~/cowrie]
$ source cowrie-env/bin/activate

(cowrie-env)(cowrie㉿kali)-[~/cowrie]
$ pip install --upgrade pip
Requirement already satisfied: pip in ./cowrie-env/lib/python3.13/site-packages (25.3)

(cowrie-env)(cowrie㉿kali)-[~/cowrie]
$ pip install -r requirements.txt
Collecting attrs==25.4.0 (from -r requirements.txt (line 1))
  Downloading attrs-25.4.0-py3-none-any.whl.metadata (10 kB)
Collecting bcrypt==5.0.0 (from -r requirements.txt (line 2))
  Downloading bcrypt-5.0.0-cp39-abi3-manylinux_2_34_x86_64.whl.metadata (10 kB)
```

Gambar 9 Pembuatan dan Instalasi Dependency Virtual Environment Cowrie

- Salin file onfigurasi dalam folder cowrie/etc

  Pada tahap ini dilakukan konfigurasi awal honeypot dengan masuk ke direktori etc di dalam folder Cowrie. File konfigurasi default cowrie.cfg.dist kemudian disalin menjadi cowrie.cfg menggunakan perintah cp cowrie.cfg.dist cowrie.cfg.



Gambar 10 Proses Penyalinan File Konfigurasi Cowrie

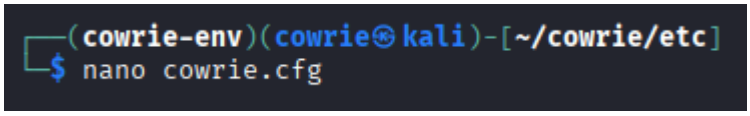
- Install cowrie

  Pada tahap ini dilakukan proses instalasi Cowrie secara lokal menggunakan perintah pip install -e .. Perintah tersebut menginstal source code Cowrie dalam mode *editable*, sehingga setiap perubahan konfigurasi atau pengembangan dapat langsung diterapkan tanpa perlu instalasi ulang.

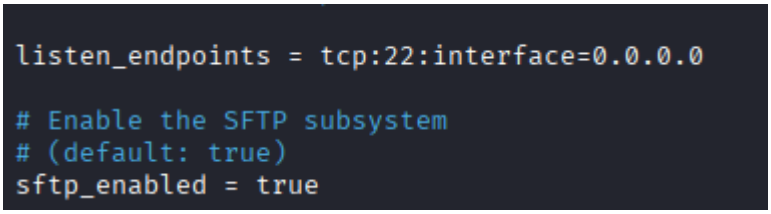
Gambar 11 Proses Instalasi Lokal Honeypot Cowrie

- Masuk ke server cowrie dan modifikasi file cowrie.cfg (ubah port nya ke 22)

  Pada tahap ini dilakukan pengeditan file konfigurasi utama honeypot menggunakan perintah nano cowrie.cfg. File cowrie.cfg berisi pengaturan penting seperti konfigurasi port, sistem logging, serta parameter operasional honeypot.



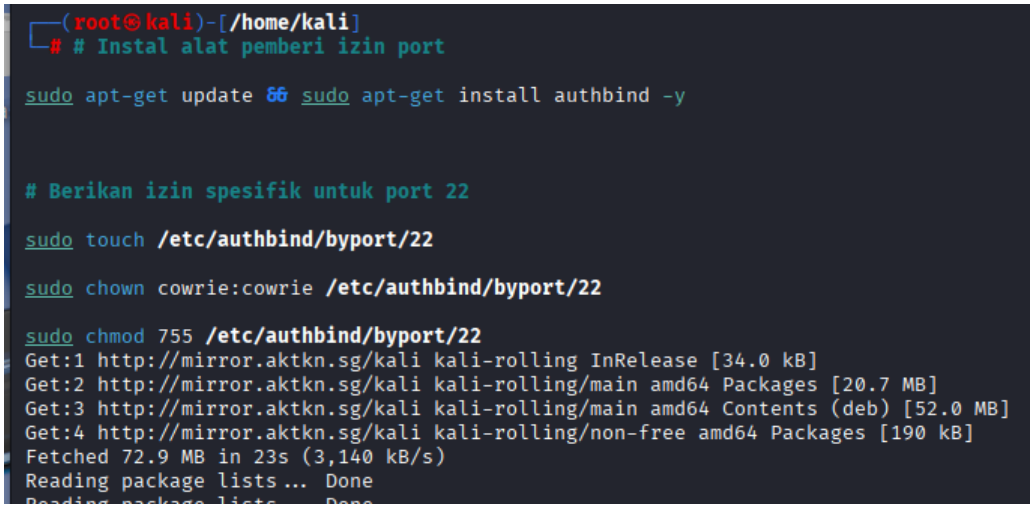
Gambar 12 Proses Pengeditan File Konfigurasi Cowrie



Gambar 13 Konfigurasi Port Honeypot pada File cowrie.cfg

- Izinkan Port 22 untuk User Non-Root (Authbind)

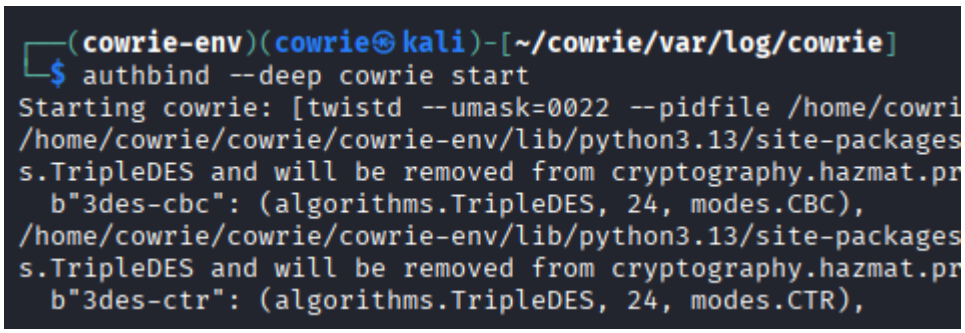
  Pada tahap ini dilakukan instalasi dan konfigurasi **authbind** untuk mengizinkan user non-root menjalankan layanan pada port di bawah 1024, khususnya port 22. Secara default, port 22 hanya dapat dijalankan oleh user root. Oleh karena itu, authbind digunakan agar honeypot Cowrie yang berjalan dengan user *cowrie* tetap dapat menggunakan port 22 tanpa harus dijalankan sebagai root.



Gambar 14 Konfigurasi Authbind untuk Izin Port 22

- Masuk ke cowrie dan jalankan dengan authbind

  Pada tahap ini honeypot Cowrie dijalankan menggunakan perintah authbind --deep bin/cowrie start. Penggunaan opsi --deep memastikan bahwa seluruh proses turunan (child process) juga tetap memiliki izin menggunakan port 22.



Gambar 15 Proses Menjalankan Honeypot Cowrie dengan Authbin

- Bersihkan iptables dan izinkan port 22 dan 60000 untuk admin

  Pada tahap ini dilakukan konfigurasi firewall menggunakan iptables untuk memastikan lalu lintas jaringan berjalan sesuai kebutuhan implementasi honeypot. Perintah iptables -F dan iptables -t nat -F digunakan untuk membersihkan aturan firewall sebelumnya agar tidak terjadi konflik konfigurasi.

```
┌──(root㉿kali)-[/home/kali]
└─# sudo iptables -F
sudo iptables -t nat -F
# Izinkan port 22 dan port 60000 (untuk Admin)
sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
sudo iptables -A INPUT -p tcp --dport 60000 -j ACCEPT
```

Gambar 16 Konfigurasi Firewall untuk Mengizinkan Port Honeypot dan Admin

B. Tools yang digunakan

1. Kali linux

   Kali Linux digunakan sebagai sistem operasi utama dalam proses implementasi dan pengujian. Sistem ini digunakan untuk melakukan konfigurasi server, instalasi honeypot, serta menjalankan berbagai tools pengujian keamanan seperti Nmap, Hydra, dan LOIC.

2. Cowrie Honeypot

   Cowrie merupakan honeypot berbasis SSH yang digunakan untuk mensimulasikan layanan SSH palsu pada port 22. Tools ini berfungsi untuk:

   - Merekam aktivitas koneksi masuk
   - Mencatat percobaan login
   - Menyimpan informasi IP penyerang
   - Mendokumentasikan detail sesi dan protokol

   Cowrie dijalankan menggunakan virtual environment dan dikonfigurasi agar berjalan pada port 22 menggunakan authbind.

3. Virtualenv (Phyton Virtual Environment)

   Virtualenv digunakan untuk membuat lingkungan Python terisolasi khusus untuk menjalankan Cowrie. Hal ini bertujuan agar dependency yang digunakan tidak mengganggu sistem utama.

4. Authbind

   Authbind digunakan untuk mengizinkan user non-root (user cowrie) menjalankan layanan pada port 22. Secara default, port di bawah 1024 hanya dapat dijalankan

oleh root, sehingga authbind diperlukan untuk menerapkan prinsip keamanan least privilege.

5. IPTables

   IPTables digunakan untuk mengatur firewall pada server. Dalam implementasi ini digunakan untuk:

   - Membersihkan rule sebelumnya
   - Mengizinkan akses ke port 22 (honeypot)
   - Mengizinkan akses ke port 60000 (SSH admin)

6. Nmap

   Nmap digunakan untuk melakukan pengujian port scanning terhadap server target. Fungsinya:

   - Mendeteksi port terbuka
   - Mengidentifikasi layanan aktif
   - Mengetahui versi service dengan opsi -sV

     Nmap digunakan pada pengujian Individual, Double, dan Multiple Attack.

7. Hydra

   Hydra digunakan untuk melakukan pengujian brute force terhadap layanan SSH. Fungsinya:

   - Mencoba berbagai kombinasi password
   - Menguji keamanan kredensial login
   - Mensimulasikan serangan login berulang

8. Hydra low Orbit Ion Cannon (LOIC)

   LOIC digunakan untuk mensimulasikan serangan DDoS dengan metode TCP flooding ke port 22.

   Fungsinya:

   - Menghasilkan lonjakan koneksi dalam jumlah besar
   - Menguji kemampuan honeypot dalam mendeteksi trafik abnormal

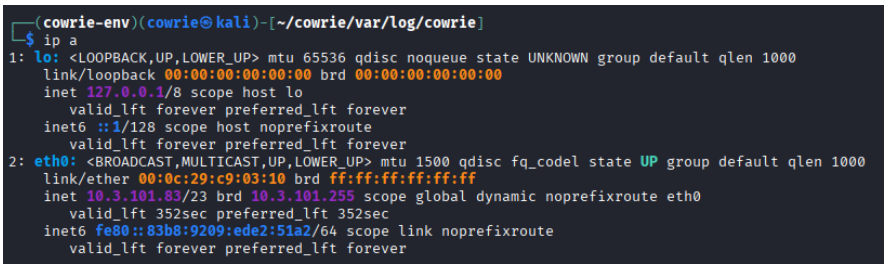
9. Nano Text Editor

   Nano digunakan untuk mengedit file konfigurasi seperti:

   - /etc/ssh/sshd_config
   - cowrie.cfg

C. Pengujian

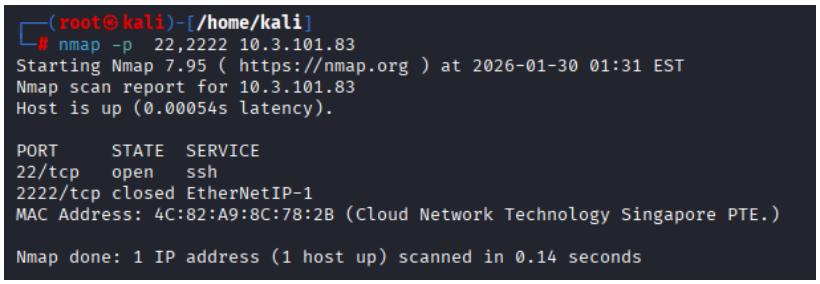
Sebelum melakukan pengujian serangan, langkah awal yang dilakukan adalah mengetahui alamat IP server menggunakan perintah ip a. Informasi ini diperlukan agar proses pengujian seperti port scanning, brute force, dan DDoS dapat diarahkan ke target yang benar.



Gambar 17 Identifikasi Alamat IP Server Menggunakan Perintah ip a

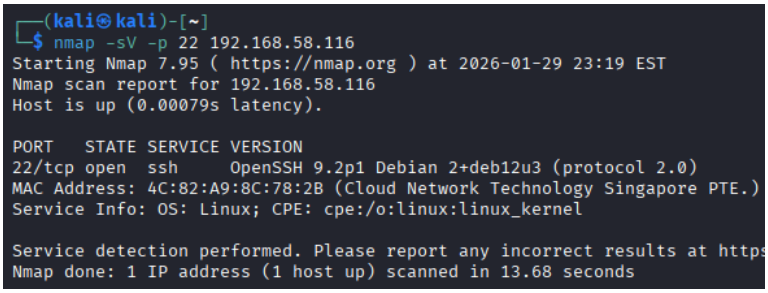
1. Individual
   - Port Scanning
     a. Lakukan nmap pada ip server

Pada tahap pengujian individual, dilakukan serangan **port scanning** menggunakan tools Nmap dengan perintah nmap -p 22,2222 10.3.101.83. Pengujian ini bertujuan untuk mengetahui status port yang terbuka pada server target.



Gambar 18 Hasil Pengujian Port Scanning Menggunakan Nmap

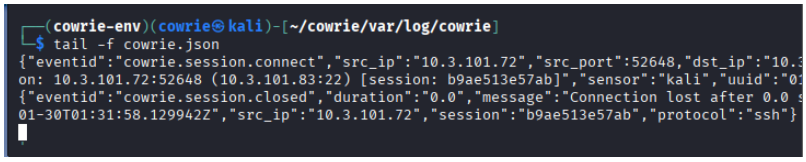
Pada pengujian ini dilakukan port scanning menggunakan perintah nmap -sV -p 22 192.168.58.116 untuk mendeteksi versi layanan yang berjalan pada port 22. Opsi -sV digunakan untuk mengidentifikasi detail service dan versi sistem yang aktif pada target.



Gambar 19 Hasil Port Scanning dengan Deteksi Versi Service (-sV)

b. Hasil log file sesudah dilakukan nmap

Setelah dilakukan pengujian port scanning menggunakan Nmap, dilakukan pengecekan log honeypot dengan perintah tail -f cowrie.json. Hasil log menunjukkan adanya event "cowrie.session.connect" yang mencatat alamat IP sumber (src_ip) penyerang serta port tujuan (dst_port: 22).

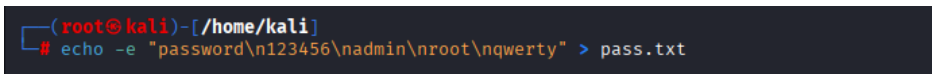


Gambar 20 Log Deteksi Serangan Port Scanning pada Honeypot

- Brute force

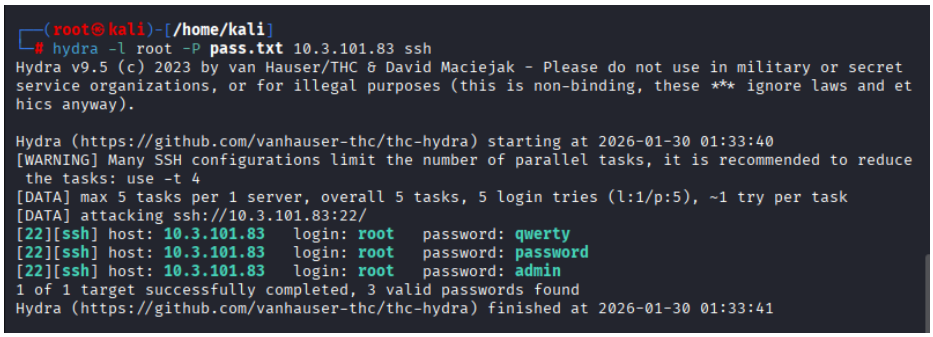
a. buat file yang berisi daftar password lalu lakukan serangan menggunakan hydra.

Pada tahap ini dibuat file daftar password yang akan digunakan dalam pengujian brute force menggunakan Hydra. Perintah echo -e "password\n123456\nadmin\nroot\nqwerty" > pass.txt digunakan untuk membuat file bernama pass.txt yang berisi beberapa kombinasi password umum.



Gambar 21 Pembuatan File Daftar Password untuk Pengujian Brute Force

Pada tahap ini dilakukan pengujian serangan brute force menggunakan tools Hydra dengan perintah: hydra -l root -P pass.txt 10.3.101.83 ssh Serangan ini bertujuan untuk mencoba berbagai kombinasi password terhadap akun root pada layanan SSH port 22. Hasil pengujian menunjukkan bahwa Hydra berhasil menemukan beberapa password yang dianggap valid, seperti qwerty, password, dan admin.



Gambar 22 Hasil Serangan Brute Force Menggunakan Hydra

b. Hasil brute force meggunakan hydra

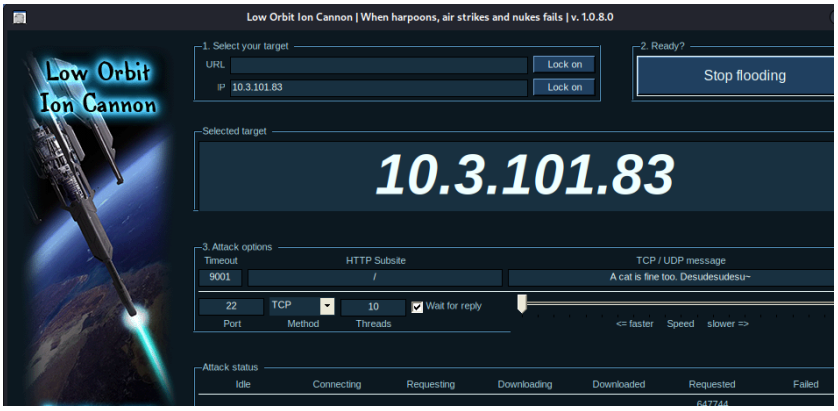
Gambar 23 Log Deteksi Serangan Brute Force pada Honeypot

Informasi src_ip (10.3.101.72) dan port tujuan (22) tercatat dengan jelas. Hal ini membuktikan bahwa honeypot tidak hanya mendeteksi percobaan login, tetapi juga merekam detail teknis koneksi seperti algoritma enkripsi dan versi protokol SSH yang digunakan.

- DDoS attack

a. Lakukan serangan DDoS menggunakan loic

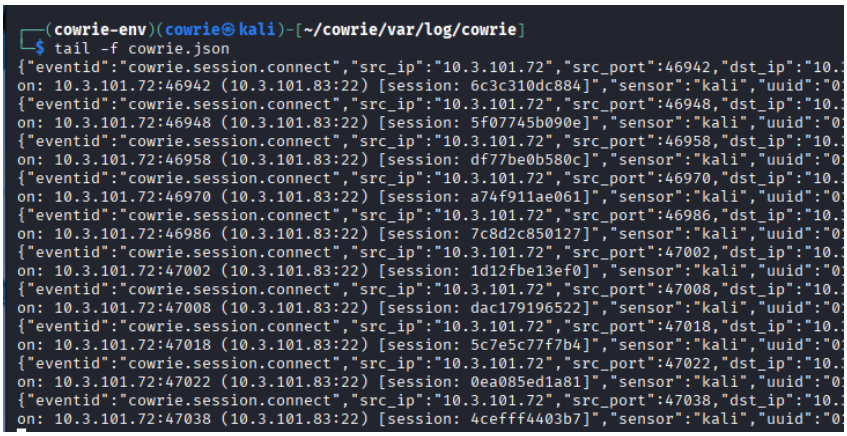
Pada pengujian ini dilakukan simulasi serangan DDoS (Distributed Denial of Service) menggunakan aplikasi Low Orbit Ion Cannon (LOIC) dengan target IP 10.3.101.83 pada port 22 (layanan honeypot). Metode yang digunakan adalah protokol TCP dengan sejumlah thread untuk menghasilkan trafik koneksi dalam jumlah besar secara berulang.



Gambar 24 Simulasi Serangan DDoS Menggunakan LOIC

Setelah dilakukan simulasi serangan DDoS menggunakan LOIC, dilakukan pemantauan log honeypot melalui perintah tail -f cowrie.json. Hasil log menunjukkan banyaknya event "cowrie.session.connect" yang muncul secara berulang dalam waktu singkat dari IP sumber yang sama (10.3.101.72) dengan port sumber yang berbeda-beda.

b. Hasil serangan di file log

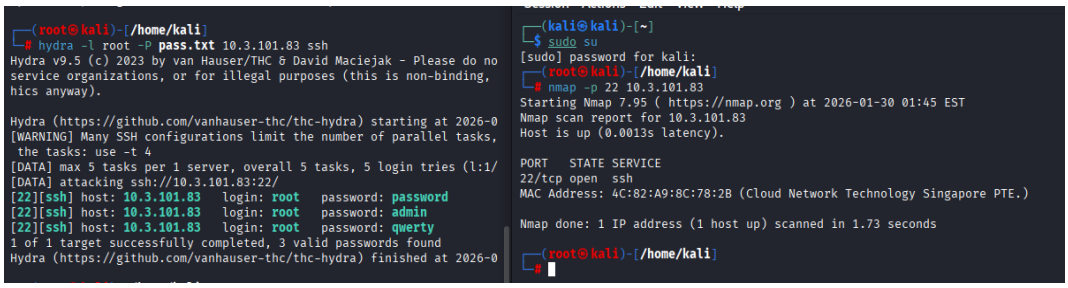


Gambar 25 Log Deteksi Serangan DDoS pada Honeypot

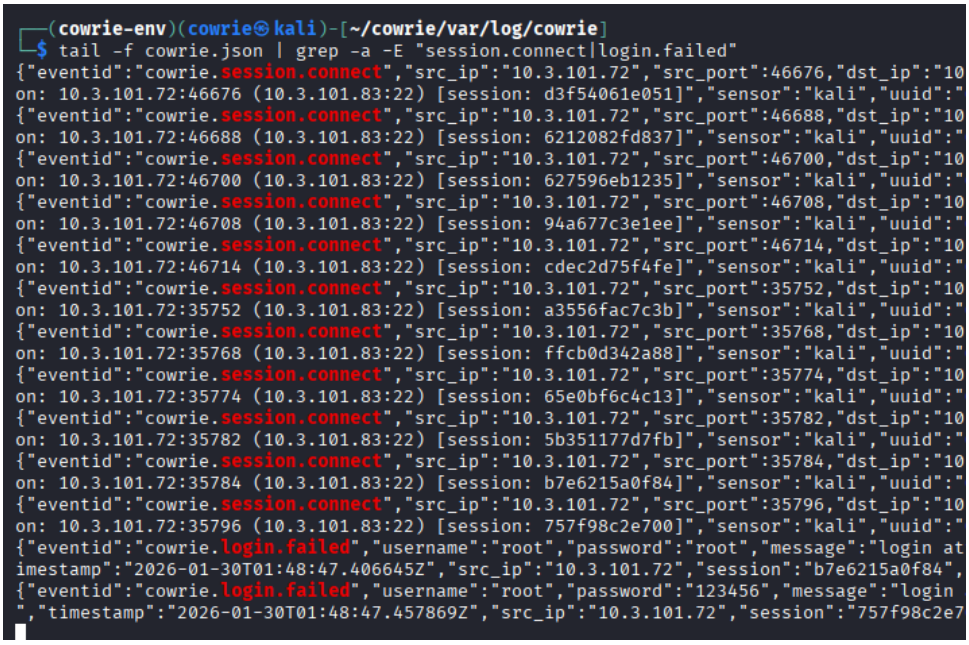
2. Double

- Port Scanning & Brute Force Attack

Pertama, dilakukan scanning menggunakan Nmap untuk memastikan port 22 dalam kondisi terbuka dan layanan SSH aktif. Hasil menunjukkan port 22 berstatus open (ssh), sehingga target dapat diakses untuk tahap berikutnya. Selanjutnya, dilakukan serangan brute force menggunakan Hydra dengan file `pass.txt` sebagai daftar password. Hasil menunjukkan beberapa password terdeteksi sebagai valid, seperti `password`, `admin`, dan `qwerty`. Namun, layanan yang diakses merupakan honeypot Cowrie, sehingga login tersebut merupakan simulasi untuk menjebak penyerang. Kombinasi kedua serangan ini berhasil diarahkan ke honeypot dan seluruh aktivitasnya tercatat pada file log. Dengan demikian, pengujian Double Attack (Port Scanning & Brute Force) dinyatakan berhasil terdeteksi 100%.



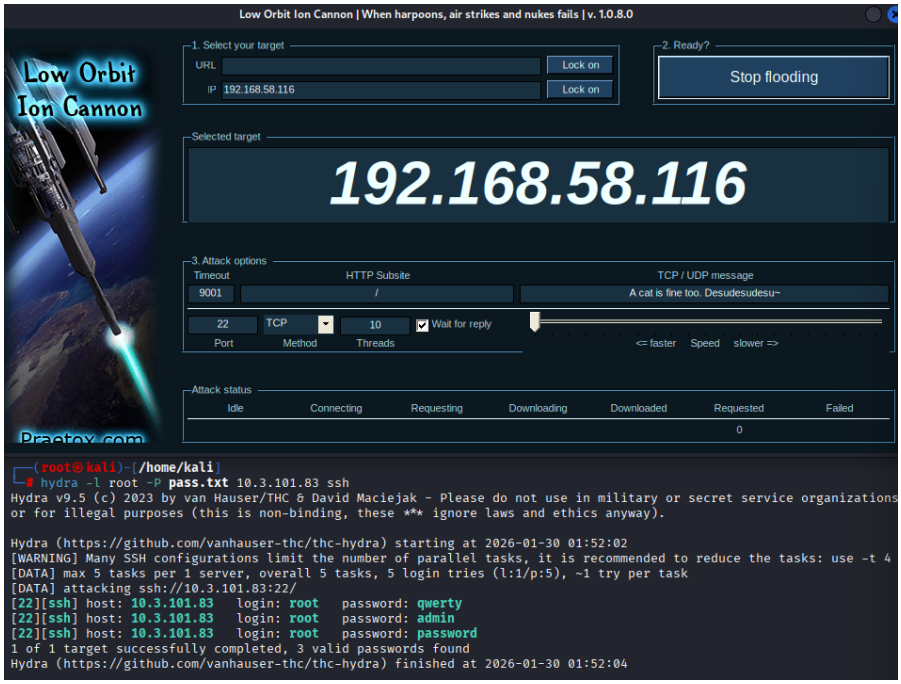
Gambar 26 Pengujian Double Attack: Port Scanning dan Brute Force



Gambar 27 Log Deteksi Double Attack (Port Scanning dan Brute Force)

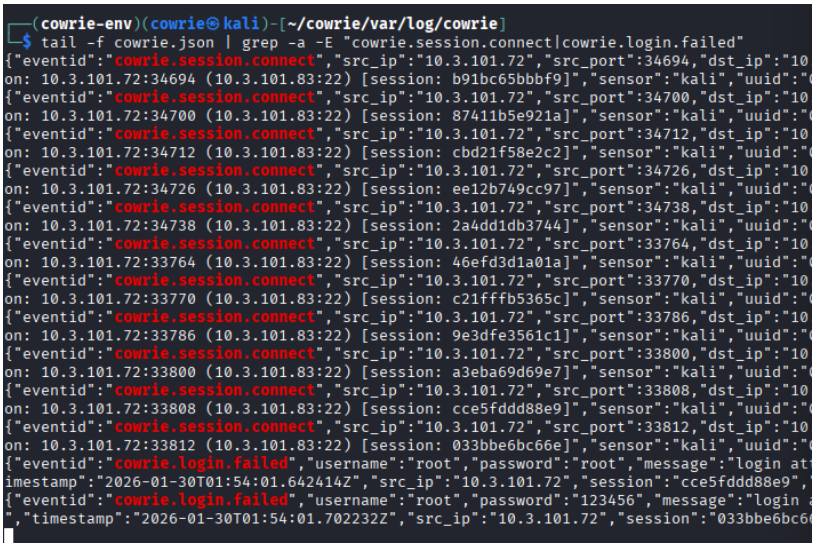
Pola ini membuktikan bahwa setelah port berhasil terdeteksi terbuka, attacker langsung melakukan eksploitasi melalui brute force. Honeypot Cowrie berhasil merekam kedua aktivitas tersebut secara bersamaan dalam satu rangkaian log, sehingga pengujian Double Attack (Port Scanning & Brute Force) dinyatakan terdeteksi 100%.

- Brute Force Attack & DDoS Attack



Gambar 29 Pengujian Double Attack: Brute Force dan DDoS

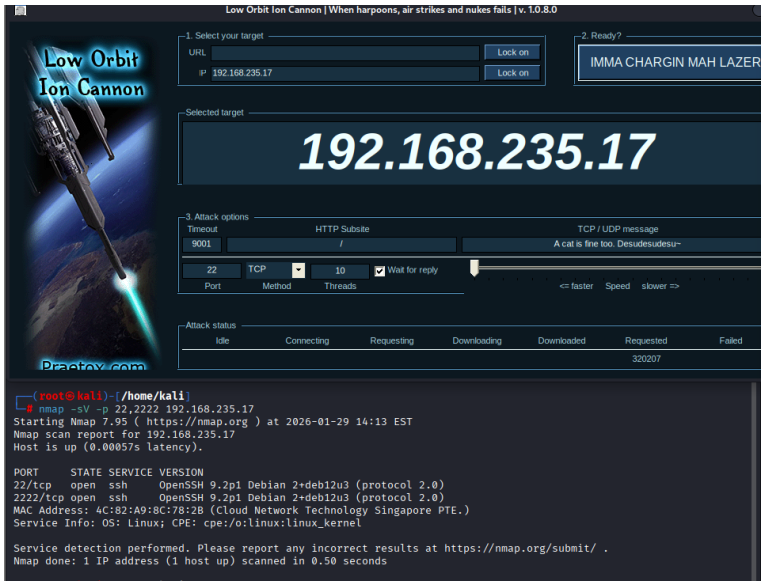
Kombinasi kedua serangan ini mensimulasikan kondisi serangan nyata di mana attacker tidak hanya mencoba menebak kredensial, tetapi juga melakukan flooding untuk melemahkan layanan. Dalam implementasi ini, seluruh aktivitas diarahkan ke honeypot Cowrie dan tidak ke server asli. Honeypot berhasil mencatat percobaan login serta lonjakan koneksi secara bersamaan, sehingga pengujian dinyatakan terdeteksi 100%.



Gambar 30 Log Deteksi Multiple Attack pada Honeypot

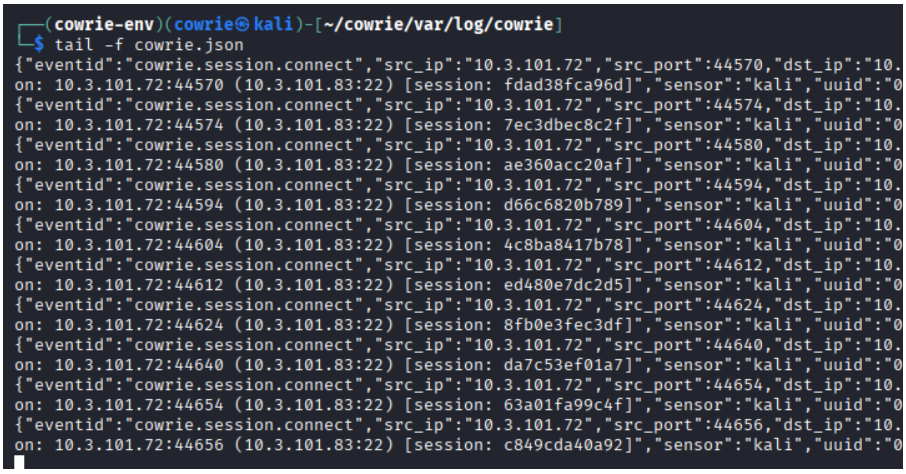
- DDoS Attack & Port Scanning

Kombinasi ini mensimulasikan kondisi di mana attacker melakukan pemetaan layanan (reconnaissance) sekaligus mencoba membanjiri server dengan trafik berlebihan. Dalam implementasi ini, seluruh trafik diarahkan ke honeypot, sehingga server asli tetap aman. Honeypot tetap aktif dan mampu mencatat aktivitas koneksi meskipun terjadi lonjakan trafik.



Gambar 31 Pengujian Double Attack: DDoS dan Port Scanning

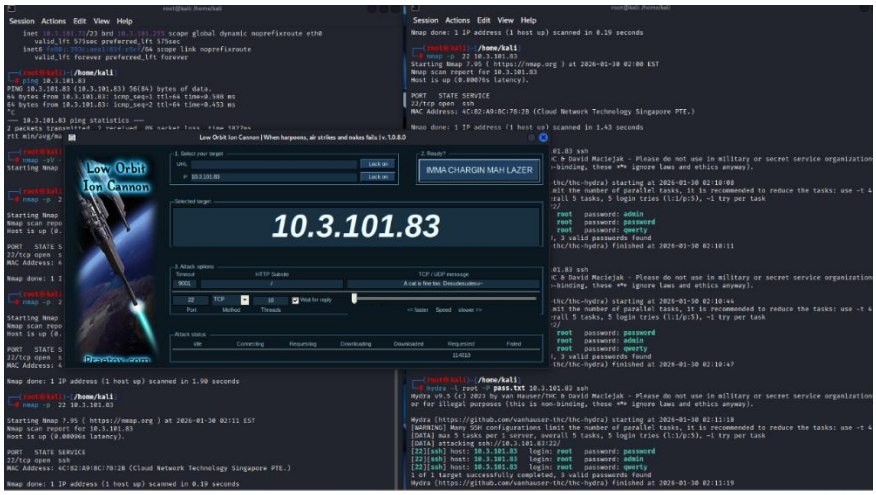
Honeypot Cowrie berhasil mencatat seluruh aktivitas koneksi tersebut secara real-time, termasuk IP penyerang, port sumber, dan session ID. Dengan demikian, kombinasi serangan DDoS dan Port Scanning dinyatakan berhasil terdeteksi 100%.



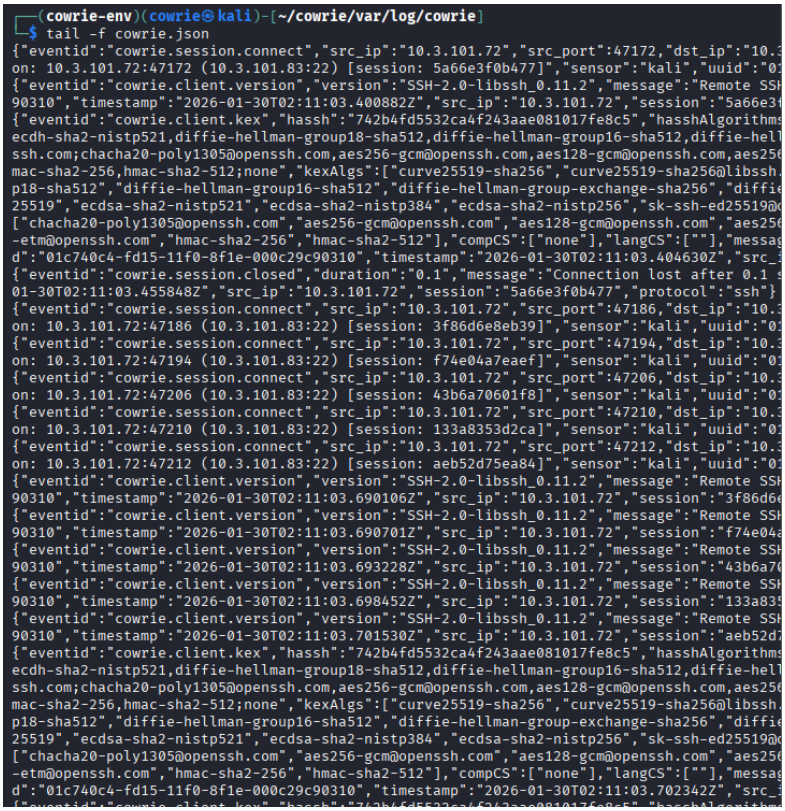
Gambar 32 Log Deteksi Double Attack (DDoS dan Port Scanning)

3. Multiple

Kombinasi ketiga serangan ini mensimulasikan kondisi serangan nyata yang kompleks, di mana attacker melakukan reconnaissance, eksploitasi kredensial, dan flooding trafik dalam satu waktu. Dalam implementasi ini, seluruh aktivitas diarahkan ke honeypot sehingga server asli tetap aman. Honeypot berhasil mencatat seluruh koneksi, percobaan login, serta lonjakan trafik tanpa mengalami gangguan layanan.



Gambar33 Pengujian Multiple Attack (Port Scanning, Brute Force, dan DDoS)



Gambar 34 Log Deteksi Multiple Attack pada Honeypot Cowrie

Berdasarkan hasil pemantauan file cowrie.json, terlihat berbagai jenis event yang tercatat secara berurutan dalam waktu sangat singkat. Event "cowrie.session.connect" muncul berulang kali dari IP 10.3.101.72 dengan port sumber yang berbeda-beda, menunjukkan adanya lonjakan koneksi (indikasi DDoS dan aktivitas scanning).

D. Hasil pengujian

No	Pola Serangan	Nama Pengujian	Kondisi Sebelum (%)	Kondisi Sesudah (%)	Hasil (Terdeteksi / Tidak)
1	Individual	Port Scanning	0%	100%	Terdeteksi
2	Individual	Bruteforce Attack	0%	100%	Terdeteksi
3	Individual	DDoS Attack	0%	100%	Terdeteksi
4	Double	Port Scanning & Bruteforce Attack	0%	100%	Terdeteksi
5	Double	Bruteforce Attack & DDoS Attack	0%	100%	Terdeteksi

6	Double	DDoS Attack & Port Scanning	0%	100%	Terdeteksi
7	Multiple	Port Scanning, Bruteforce Attack, & DDoS Attack	0%	100%	Terdeteksi