

TUGAS BESAR
ADVANCE NETWORK SECURITY
“Implementasi Honeypot Sebagai Pendekripsi Serangan Vps”



OLEH:

UMMUL MU'MININ

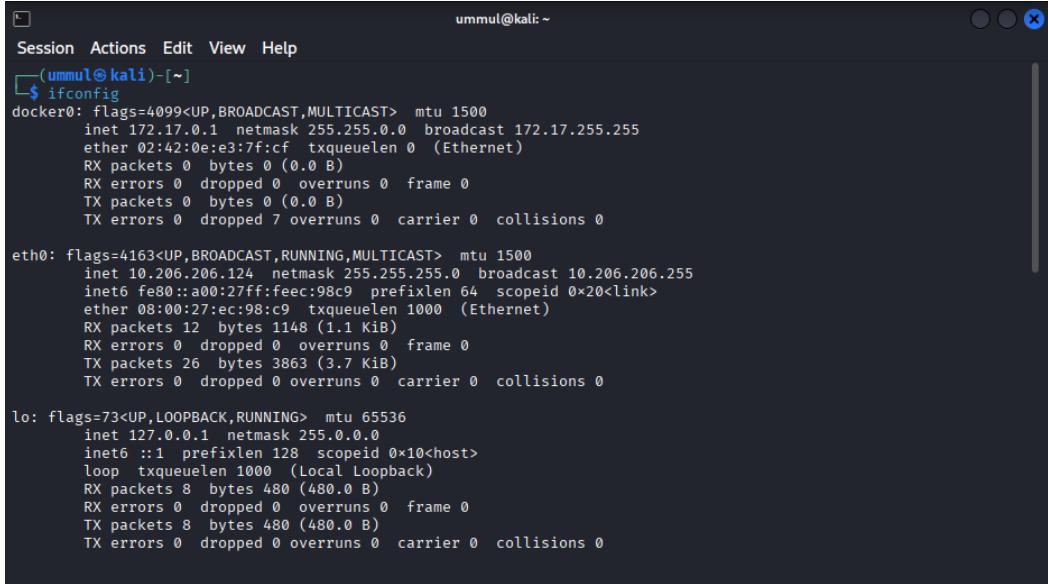
105841117323

RIRIN YULANDARI

105841117923

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH MAKASSAR
2026

1. Cek ip



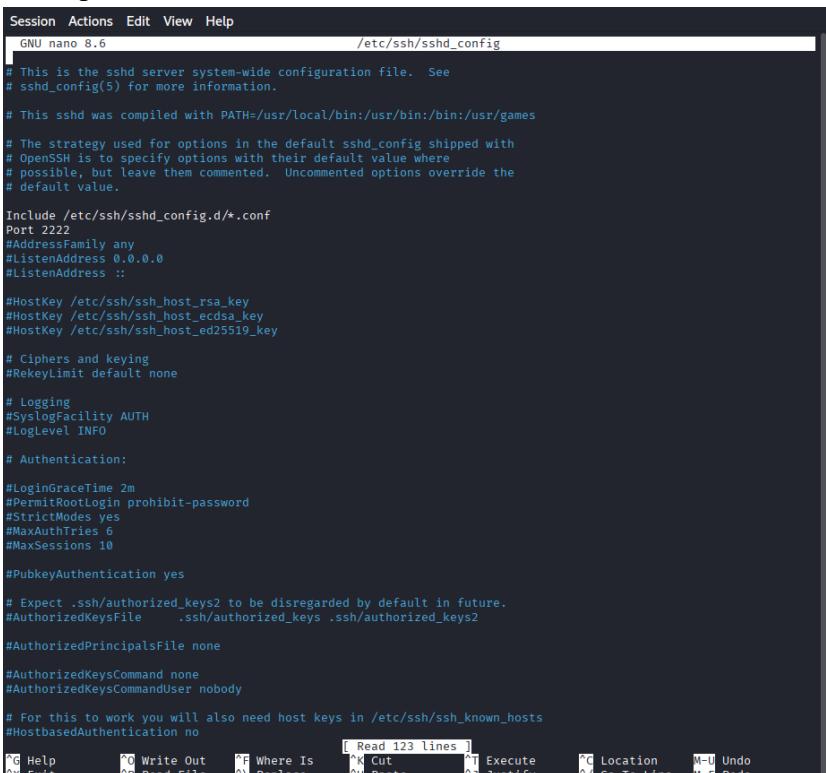
```
ummul@kali: ~
Session Actions Edit View Help
(ummul@kali)-[~]
$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
        ether 02:42:0e:e3:7f:cf txqueuelen 0 (Ethernet)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 0 bytes 0 (0.0 B)
        TX errors 0 dropped 7 overruns 0 carrier 0 collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.206.206.124 netmask 255.255.255.0 broadcast 10.206.206.255
        inet6 fe80::a00:27ff:feec:98c9 prefixlen 64 scopeid 0x20<link>
        ether 08:00:27:ec:98:c9 txqueuelen 1000 (Ethernet)
        RX packets 12 bytes 1148 (1.1 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 26 bytes 3863 (3.7 KiB)
        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 8 bytes 480 (480.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 8 bytes 480 (480.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Langkah awal yang paling krusial dalam prosedur pengujian ini adalah melakukan verifikasi alamat jaringan pada laptop target (ummul@kali) dengan mengeksekusi perintah ifconfig untuk memastikan target sudah terhubung secara benar ke dalam infrastruktur jaringan yang sama dengan penyerang. Berdasarkan hasil pembacaan pada antarmuka jaringan utama eth0, teridentifikasi bahwa target memiliki alamat IPv4 10.206.206.124, yang nantinya akan ditetapkan sebagai titik tuju tunggal bagi seluruh simulasi serangan.

2. Ubah port



```
Session Actions Edit View Help
GNU nano 8.6 /etc/ssh/sshd_config
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf
Port 2222
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile      .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no

[ Read 123 lines ]
^G Help      ^Q Write Out   ^F Where Is   ^K Cut          ^T Execute      ^C Location   M-U Undo
^X Exit      ^R Read File   ^\ Replace     ^M Paste       ^L Justify     ^Y Go To Line M-F Redo
```

Langkah selanjutnya dalam konfigurasi keamanan jaringan adalah melakukan perubahan pada port standar layanan SSH guna meningkatkan aspek keamanan melalui metode pengaburan (*security by obscurity*). Proses ini dilakukan dengan mengakses file konfigurasi utama melalui perintah nano /etc/ssh/sshd_config. Di dalam editor tersebut, baris parameter Port diubah nilainya dari port standar 22 menjadi port baru, yaitu 2222. Modifikasi ini bertujuan untuk meminimalisir risiko serangan *brute-force* otomatis yang umumnya menargetkan port *default* layanan komunikasi terenkripsi.

3. Install Docker

```
(ummul@kali)-[~]
$ sudo apt update && sudo apt install docker.io docker-compose -y
Hit:1 https://packages.microsoft.com/repos/code stable InRelease
Hit:2 http://http.kali.org/kali kali-rolling InRelease
1961 packages can be upgraded. Run 'apt list --upgradable' to see them.
Upgrading:
  libnl-3-200  libnl-genl-3-200  libnl-route-3-200  libperl5.40  perl  perl-base  perl-modules-5.40

Installing:
  docker-compose  docker.io

Installing dependencies:
  containerd  docker-cli  libintl-xs-perl  libproc-processtable-perl  python3-pycriu
  criu  libcompe11  libmodule-find-perl  libsort-naturally-perl  runc
  docker-buildx  libintl-perl  libnet9  needrestart  tini-static

Suggested packages:
  containernetworking-plugins  btrfs-progs  rinse  xfsprogs  | zfsutils-linux
  docker-doc  debootstrap  rootlesskit  zfs-fuse

Summary:
Upgrading: 7, Installing: 17, Removing: 0, Not Upgrading: 1954
Download size: 110 MB
Space needed: 432 MB / 5,671 MB available

Get:1 http://xsrv.moratelindo.io/kali kali-rolling/main amd64 libperl5.40 amd64 5.40.1-7 [4,317 kB]
Get:6 http://http.kali.org/kali kali-rolling/main amd64 containerd amd64 1.7.24-ds1-10 [33.6 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 perl amd64 5.40.1-7 [267 kB]
Get:4 http://kali.download/kali kali-rolling/main amd64 perl-modules-5.40 all 5.40.1-7 [3,012 kB]
Get:5 http://http.kali.org/kali kali-rolling/main amd64 runc amd64 1.3.3+ds1-2 [6,686 kB]
Get:7 http://kali.download/kali kali-rolling/main amd64 tini-static amd64 0.19.0-6 [277 kB]
Get:8 http://http.kali.org/kali kali-rolling/main amd64 docker.io amd64 27.5.1+dfsg4-1 [23.2 kB]
Get:3 http://xsrv.moratelindo.io/kali kali-rolling/main amd64 perl-base amd64 5.40.1-7 [1,679 kB]
Get:18 http://xsrv.moratelindo.io/kali kali-rolling/main amd64 libintl-perl all 1.35-1 [690 kB]
Get:24 http://xsrv.moratelindo.io/kali kali-rolling/main amd64 python3-pycriu all 4.2-1 [44.3 kB]
Get:9 http://http.kali.org/kali kali-rolling/main amd64 libnet9 amd64 1.3+dfsg-3 [51.3 kB]
Get:10 http://kali.download/kali kali-rolling/main amd64 libnl-genl-3-200 amd64 3.12.0-2 [18.8 kB]
Get:11 http://kali.download/kali kali-rolling/main amd64 libnl-route-3-200 amd64 3.12.0-2 [200 kB]
Get:12 http://kali.download/kali kali-rolling/main amd64 libnl-3-200 amd64 3.12.0-2 [62.2 kB]
Get:13 http://kali.download/kali kali-rolling/main amd64 libcompe11 amd64 4.2-1 [64.2 kB]
Get:15 http://http.kali.org/kali kali-rolling/main amd64 docker-buildx amd64 0.19.3+ds1-4 [13.9 kB]
Get:16 http://http.kali.org/kali kali-rolling/main amd64 docker-cli amd64 27.5.1+dfsg4-1 [7,650 kB]
Get:17 http://kali.download/kali kali-rolling/main amd64 docker-compose amd64 2.32.4-3 [13.5 kB]
Get:14 http://mirror.primealink.net.id/kali kali-rolling/main amd64 criu amd64 4.2-1 [557 kB]
Get:19 http://mirror.primealink.net.id/kali kali-rolling/main amd64 libintl-xs-perl amd64 1.35-1 [15.3 kB]
Get:20 http://kali.download/kali kali-rolling/main amd64 libmodule-find-perl all 0.17-1 [10.7 kB]
Get:21 http://http.kali.org/kali kali-rolling/main amd64 libproc-processtable-perl amd64 0.637-1+b1 [42.3 kB]
Get:22 http://kali.download/kali kali-rolling/main amd64 libsort-naturally-perl all 1.03-4 [13.1 kB]
Get:23 http://kali.download/kali kali-rolling/main amd64 needrestart all 3.11-1 [68.6 kB]
Fetched 110 MB in 33s (3.368 kB/s)
```

Setelah berhasil melakukan verifikasi alamat IP target, langkah krusial berikutnya dalam membangun infrastruktur pengujian adalah melakukan instalasi perangkat lunak *Docker* pada sistem operasi Kali Linux. Proses ini diawali dengan menjalankan perintah sinkronisasi repositori melalui sudo apt update, yang segera dilanjutkan dengan instalasi paket utama docker.io dan docker-compose untuk memungkinkan pengelolaan kontainer secara efisien. Selama proses berlangsung, sistem secara otomatis mengunduh berbagai dependensi teknis seperti containerd, runc, dan docker-buildx dengan total ukuran unduhan sekitar 110 MB guna memastikan seluruh fungsionalitas virtualisasi dapat berjalan optimal. Instalasi Docker ini merupakan pondasi utama

dalam skenario pengujian, karena nantinya layanan Honeypot Cowrie akan dijalankan di dalam lingkungan kontainer yang terisolasi dari sistem utama.

4. Aktifkan cowrie

```
(ummul㉿kali)-[~]
└─$ sudo docker run -p 22:2222 cowrie/cowrie
/cowrie/cowrie-env/lib/python3.11/site-packages/twisted/conch/ssh/transport.py:110: CryptographyDeprecationWarning
g: TripleDES has been moved to cryptography.hazmat.deprecate.ciphers.algorithms.TripleDES and will be removed from
cryptography.hazmat.primitives.ciphers.algorithms in 48.0.0.
b"3des-cbc": (algorithms.TripleDES, 24, modes.CBC),
/cowrie/cowrie-env/lib/python3.11/site-packages/twisted/conch/ssh/transport.py:117: CryptographyDeprecationWarning
g: TripleDES has been moved to cryptography.hazmat.deprecate.ciphers.algorithms.TripleDES and will be removed from
cryptography.hazmat.primitives.ciphers.algorithms in 48.0.0.
b"3des-ctr": (algorithms.TripleDES, 24, modes.CTR),
2026-01-27T15:18:59+0000 [-] Reading configuration from ['/cowrie/cowrie-git/etc/cowrie.cfg.dist']
2026-01-27T15:19:00+0000 [-] Python Version 3.11.2 (main, Apr 28 2025, 14:11:48) [GCC 12.2.0]
2026-01-27T15:19:00+0000 [-] Twisted Version 25.5.0
2026-01-27T15:19:00+0000 [-] Cowrie Version 2.9.9.dev1+g7d81de406
2026-01-27T15:19:00+0000 [-] Sensor UUID: 80ea96c2-fab7-11f0-bb6a-ee532cd24139
2026-01-27T15:19:00+0000 [-] Loaded output engine: jsonlog
2026-01-27T15:19:00+0000 [-] twisted.scripts._twistd_unix.UnixAppLogger#info] twistd 25.5.0 (/cowrie/cowrie-env/bin/
python3 3.11.2) starting up.
2026-01-27T15:19:00+0000 [-] twisted.scripts._twistd_unix.UnixAppLogger#info] reactor class: twisted.internet.epollr
eactor.EPollReactor.
2026-01-27T15:19:00+0000 [-] CowrieSSHFactory starting on 2222
2026-01-27T15:19:00+0000 [cowrie.ssh.factory.CowrieSSHFactory#info] Starting factory <cowrie.ssh.factory.CowrieSS
HFactory object at 0x7fed20e591d0>
2026-01-27T15:19:00+0000 [-] Ready to accept SSH connections
```

Setelah berhasil menginstal Docker, langkah krusial berikutnya adalah mengaktifkan layanan *Honeypot Cowrie* menggunakan perintah sudo docker run untuk membuat "pintu jebakan" SSH yang aktif. Dalam proses ini, sistem melakukan pemetaan port (*port mapping*) di mana port fisik 22 pada laptop target dihubungkan langsung ke port internal 2222 milik kontainer Cowrie. Begitu perintah dijalankan, terminal akan menampilkan inisialisasi mesin Cowrie, mulai dari pembacaan konfigurasi cowrie.cfg.dist hingga pemuatan mesin log berbasis JSON. Indikator keberhasilan dari tahap ini terlihat pada baris log terakhir yang menyatakan "*Ready to accept SSH connections*", yang berarti SSH jebakan tersebut kini sudah aktif sepenuhnya dan siap merekam setiap interaksi ilegal dari penyerang. Dengan aktifnya Cowrie di dalam lingkungan Docker, laptop target kini memiliki pertahanan berlapis yang mampu mensimulasikan layanan SSH palsu tanpa mengekspos keamanan sistem operasi yang sebenarnya.

5. Port Scanning

a. Penyerang

```
(root@rixin)-[/home/ririn]
└─# nmap -Pn -sV -p 22 10.206.206.124
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-27 23:50 WITA
Nmap scan report for 10.206.206.124
Host is up (0.021s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
MAC Address: 50:BB:B5:34:BC:26 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.73 seconds
```

Setelah infrastruktur pertahanan aktif, tahap selanjutnya dalam simulasi ini adalah melakukan *Port Scanning* dari sisi penyerang (root@rixin) sebagai bagian dari fase

pengintaian aktif (*active reconnaissance*). Proses ini dilakukan menggunakan *tool* Nmap dengan perintah nmap -Pn -sV -p 22 10.206.206.124 yang bertujuan untuk mengidentifikasi status port serta versi layanan yang berjalan pada target. Hasil pemindaian menunjukkan bahwa port 22/tcp dalam status *open* dan berhasil mengelabui penyerang dengan menampilkan identitas layanan palsu berupa OpenSSH 9.2p1 Debian. Identitas ini sebenarnya merupakan hasil simulasi dari Honeypot Cowrie, yang dirancang untuk terlihat seperti layanan SSH asli guna memancing penyerang melakukan interaksi lebih jauh. Keberhasilan deteksi port ini menjadi jembatan bagi penyerang untuk melanjutkan ke tahap serangan berikutnya, yaitu *Bruteforce* dan *DDoS*, karena penyerang kini meyakini bahwa terdapat celah masuk yang valid pada server target.

b. Target

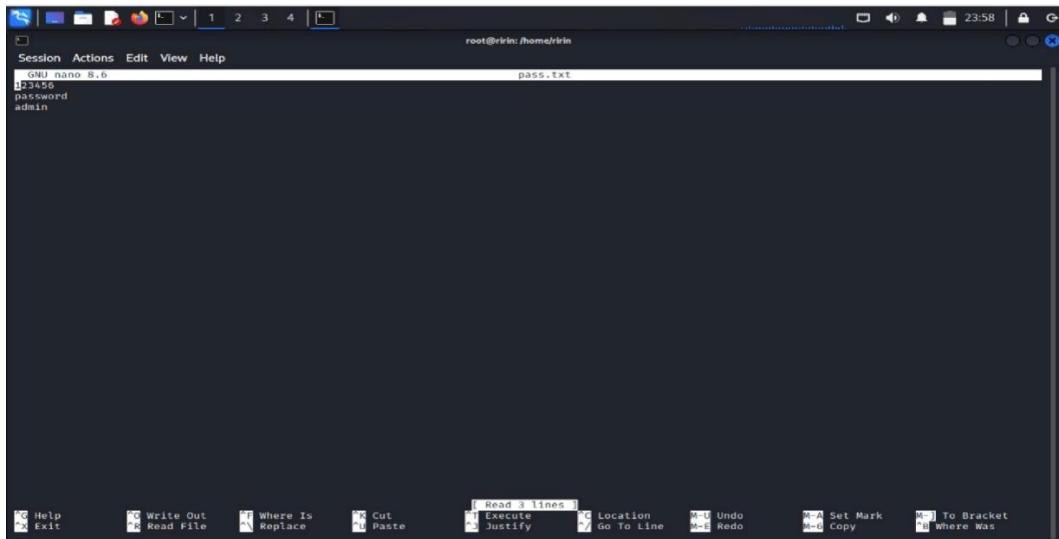
```
(ummul㉿kali)-[~]
└─$ sudo docker run -p 22:2222 cowrie/cowrie
[sudo] password for ummul:
[cowrie/cowrie-env/lib/python3.11/site-packages/twisted/conch/ssh/transport.py:110: CryptographyDeprecationWarning: TripleDES has been moved to cryptography.hazmat.decrepit.ciphers.algorithms.TripleDES and will be removed from cryptography.hazmat.primitives.ciphers.algorithms in 48.0.0.
b"3des-cbc": (algorithms.TripleDES, 24, modes.CBC),
[cowrie/cowrie-env/lib/python3.11/site-packages/twisted/conch/ssh/transport.py:117: CryptographyDeprecationWarning: TripleDES has been moved to cryptography.hazmat.decrepit.ciphers.algorithms.TripleDES and will be removed from cryptography.hazmat.primitives.ciphers.algorithms in 48.0.0.
b"3des-ctr": (algorithms.TripleDES, 24, modes.CTR),
2026-01-27T15:36:25+0000 [-] Reading configuration from ['/cowrie/cowrie-git/etc/cowrie.cfg.dist']
2026-01-27T15:36:26+0000 [-] Python Version 3.11.2 (main, Apr 28 2025, 14:11:48) [GCC 12.2.0]
2026-01-27T15:36:26+0000 [-] Twisted Version 25.5.0
2026-01-27T15:36:26+0000 [-] Cowrie Version 2.9.9.dev1+g7d81de406
2026-01-27T15:36:26+0000 [-] Sensor UUID: 80ea96c2-fab7-11f0-bb6a-ee532cd24139
2026-01-27T15:36:26+0000 [-] Loaded output engine: jsonlog
2026-01-27T15:36:26+0000 [twisted.scripts._twistd_unix.UnixAppLogger#info] twistd 25.5.0 (/cowrie/cowrie-env/bin/python3 3.11.2) starting up.
2026-01-27T15:36:26+0000 [twisted.scripts._twistd_unix.UnixAppLogger#info] reactor class: twisted.internet.epollreactor.EPollReactor.
2026-01-27T15:36:26+0000 [-] CowrieSSHFactory starting on 2222
2026-01-27T15:36:26+0000 [cowrie.ssh.factory.CowrieSSHFactory#info] Starting factory <cowrie.ssh.factory.CowrieSSHFactory object at 0x7f5ebd5716d0>
2026-01-27T15:36:27+0000 [-] Ready to accept SSH connections
2026-01-27T15:50:20+0000 [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha1
2026-01-27T15:50:20+0000 [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha256
2026-01-27T15:50:20+0000 [cowrie.ssh.factory.CowrieSSHFactory] New connection: 10.206.206.39:59612 (172.17.0.2:22) [session: d533ac968675]
2026-01-27T15:50:20+0000 [HoneyPotSSHTransport,0,10.206.206.39] Remote SSH version:
2026-01-27T15:50:20+0000 [HoneyPotSSHTransport,0,10.206.206.39] Bad protocol version identification: b''
2026-01-27T15:50:20+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2026-01-27T15:50:20+0000 [HoneyPotSSHTransport,0,10.206.206.39] Connection lost after 0.0 seconds
```

Pada sisi target (ummul@kali), aktivitas pengintaian yang dilakukan oleh penyerang melalui *port scanning* terekam secara mendetail di dalam log kontainer Cowrie. Segera setelah Nmap mengeksekusi perintah pemindaian, terminal target menampilkan baris log New connection: 10.206.206.39 yang mengonfirmasi bahwa Honeypot telah berhasil mencegat upaya koneksi dari alamat IP penyerang pada port 22. Sistem kemudian mencatat upaya identifikasi versi protokol dengan keterangan Bad protocol version identification, yang merupakan karakteristik umum dari pemindaian otomatis Nmap saat mencoba menentukan versi layanan tanpa melakukan proses *handshake* penuh. Rangkaian log ini diakhiri dengan pesan connection lost setelah Nmap selesai mengambil data yang diperlukan,

membuktikan bahwa Cowrie secara *real-time* mampu mendeteksi dan mengidentifikasi setiap jejak aktivitas mencurigakan sejak fase pengintaian awal dilakukan.

6. Bruteforce Attack

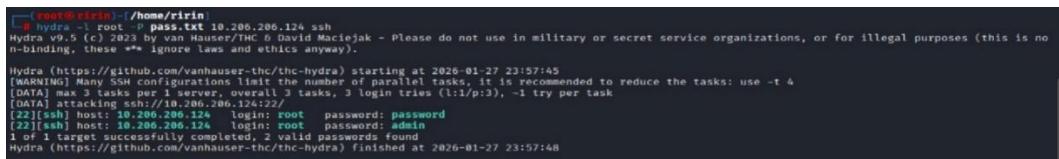
a. Membuat Password (Penyerang)



```
root@ririn: /home/ririn
GNU nano 8.6
123456
password
admin
```

Tahapan selanjutnya dalam skenario penyerangan adalah mempersiapkan serangan *Bruteforce* SSH pada sisi penyerang (root@ririn) dengan membuat daftar kata sandi (*wordlist*) yang akan digunakan untuk menguji kredibilitas keamanan target. Proses ini dilakukan menggunakan editor teks nano untuk membuat sebuah file bernama pass.txt, yang di dalamnya berisi daftar kata sandi umum seperti 123456, password, dan admin. Pembuatan daftar kata sandi ini merupakan langkah krusial sebelum menjalankan *tool* Hydra, karena efektivitas serangan *Bruteforce* sangat bergantung pada kualitas dan relevansi daftar kata yang digunakan untuk menebak autentikasi pada port 22 target. Dengan tersedianya file pass.txt ini, penyerang telah memiliki basis data untuk melakukan percobaan login secara otomatis dan masif terhadap alamat IP target 10.206.206.124 yang sebelumnya telah teridentifikasi memiliki port SSH yang terbuka.

b. Menyerang



```
root@ririn: /home/ririn
# hydra -l root -P pass.txt 10.206.206.124 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is no n-binding, these ** ignore laws and ethics anyway).
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2026-01-27 23:57:45
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 3 tasks per 1 server, overall 3 tasks, 3 login tries (l:1/p:3), -1 try per task
[DATA] attacking ssh://10.206.206.124:22/
[DATA] [ssh] host: 10.206.206.124 login: root password: password
[DATA] [ssh] host: 10.206.206.124 login: root password: admin
1 of 1 target successfully completed, 2 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2026-01-27 23:57:48
```

Setelah daftar kata sandi dipersiapkan, tahap serangan dimulai dengan mengeksekusi *Bruteforce Attack* menggunakan *tool* Hydra dari laptop penyerang

(root@ririn). Penyerang menjalankan perintah hydra -l root -P pass.txt 10.206.206.124 ssh, yang mengarahkan serangan secara otomatis ke layanan SSH pada IP target menggunakan daftar kata sandi yang telah dibuat sebelumnya. Dalam hitungan detik, Hydra melaporkan hasil percobaan login dan secara mengejutkan menunjukkan bahwa ditemukan dua kata sandi yang dianggap valid, yaitu password dan admin, untuk pengguna root. Hasil ini sebenarnya merupakan bagian dari keberhasilan simulasi Honeypot Cowrie, yang sengaja menerima percobaan login tersebut guna memancing penyerang masuk lebih dalam ke dalam sistem jebakan untuk memantau aktivitas mereka. Eksekusi serangan ini membuktikan bahwa kerentanan autentikasi dapat dieksplorasi dengan sangat cepat oleh penyerang jika tidak dilindungi oleh sistem keamanan yang memadai.

c. Target

```
2026-01-27T15:57:41+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#debug] incoming: b'aes256-ctr' b'hmac-sha2-256' b'none'
2026-01-27T15:57:41+0000 [HoneyPotSSHTransport,2,10.206.206.39] SSH client hash fingerprint: 742b4fd5532ca4f243aae081017fe8c5
2026-01-27T15:57:41+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#debug] kex alg=b'curve25519-sha256' key alg=b'ssh-ed25519'
2026-01-27T15:57:41+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#debug] outgoing: b'aes256-ctr' b'hmac-sha2-256' b'none'
2026-01-27T15:57:41+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#debug] incoming: b'aes256-ctr' b'hmac-sha2-256' b'none'
2026-01-27T15:57:41+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#debug] NEW KEYS
2026-01-27T15:57:41+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#debug] NEW KEYS
2026-01-27T15:57:41+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#debug] NEW KEYS
2026-01-27T15:57:41+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#debug] starting service b'ssh-userauth'
2026-01-27T15:57:41+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#debug] starting service b'ssh-userauth'
2026-01-27T15:57:41+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#debug] starting service b'ssh-userauth'
2026-01-27T15:57:41+0000 [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'root' trying auth b'none'
2026-01-27T15:57:41+0000 [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'root' trying auth b'none'
2026-01-27T15:57:41+0000 [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'root' trying auth b'none'
2026-01-27T15:57:41+0000 [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'root' trying auth b'password'
2026-01-27T15:57:41+0000 [HoneyPotSSHTransport,2,10.206.206.39] Could not read etc/userdb.txt, default database activated
2026-01-27T15:57:41+0000 [HoneyPotSSHTransport,2,10.206.206.39] login attempt [b'root'/b'123456'] failed
2026-01-27T15:57:41+0000 [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'root' trying auth b'password'
2026-01-27T15:57:41+0000 [HoneyPotSSHTransport,4,10.206.206.39] Could not read etc/userdb.txt, default database activated
2026-01-27T15:57:41+0000 [HoneyPotSSHTransport,4,10.206.206.39] login attempt [b'root'/b'admin'] succeeded
2026-01-27T15:57:41+0000 [HoneyPotSSHTransport,4,10.206.206.39] Initialized emulated server as architecture: linux-x64-lsb
2026-01-27T15:57:41+0000 [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'root' authenticated with b'password'
2026-01-27T15:57:41+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#debug] starting service b'ssh-connection'
2026-01-27T15:57:41+0000 [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'root' trying auth b'password'
2026-01-27T15:57:41+0000 [HoneyPotSSHTransport,3,10.206.206.39] Could not read etc/userdb.txt, default database activated
2026-01-27T15:57:41+0000 [HoneyPotSSHTransport,3,10.206.206.39] login attempt [b'root'/b'password'] succeeded
2026-01-27T15:57:41+0000 [HoneyPotSSHTransport,3,10.206.206.39] Initialized emulated server as architecture: linux-x64-lsb
2026-01-27T15:57:41+0000 [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'root' authenticated with b'password'
2026-01-27T15:57:41+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#debug] starting service b'ssh-connection'
2026-01-27T15:57:41+0000 [HoneyPotSSHTransport,3,10.206.206.39] avatar root logging out
2026-01-27T15:57:41+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2026-01-27T15:57:41+0000 [HoneyPotSSHTransport,3,10.206.206.39] Connection lost after 0.4 seconds
2026-01-27T15:57:41+0000 [HoneyPotSSHTransport,4,10.206.206.39] avatar root logging out
2026-01-27T15:57:41+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2026-01-27T15:57:41+0000 [HoneyPotSSHTransport,4,10.206.206.39] Connection lost after 0.4 seconds
2026-01-27T15:57:42+0000 [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'root' failed auth b'password'
2026-01-27T15:57:42+0000 [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] unauthorized login: ()
2026-01-27T15:58:12+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2026-01-27T15:58:12+0000 [HoneyPotSSHTransport,2,10.206.206.39] Connection lost after 31.4 seconds
```

Setelah penyerang melancarkan serangan *Bruteforce* menggunakan Hydra, seluruh aktivitas tersebut terekam secara komprehensif pada log Honeypot Cowrie di sisi target (ummul@kali). Log menunjukkan bahwa sistem mendeteksi percobaan login berulang untuk pengguna root dari alamat IP penyerang 10.206.206.39. Cowrie

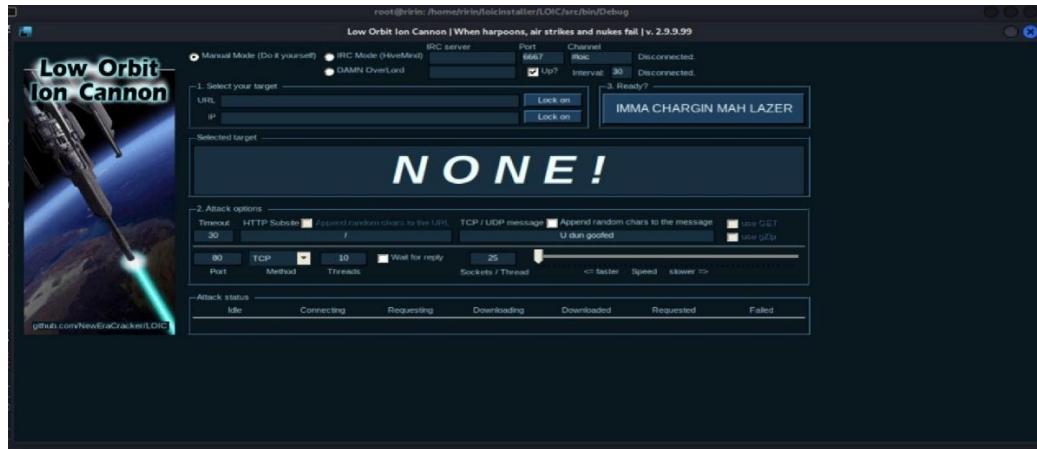
mencatat setiap detail upaya autentikasi, mulai dari kegagalan login untuk kata sandi 123456 hingga keberhasilan login palsu menggunakan kata sandi admin dan password. Setelah penyerang berhasil "masuk", Honeypot segera menginisialisasi server emulasi dengan arsitektur linux-x64-lsb untuk memantau aktivitas penyerang lebih lanjut di dalam sistem jebakan tersebut. Rangkaian log ini membuktikan kemampuan Cowrie dalam membedakan serta merekam setiap kombinasi *username* dan *password* yang dicoba oleh penyerang, sekaligus memberikan gambaran nyata mengenai interaksi penyerang dengan layanan SSH yang sedang disimulasikan.

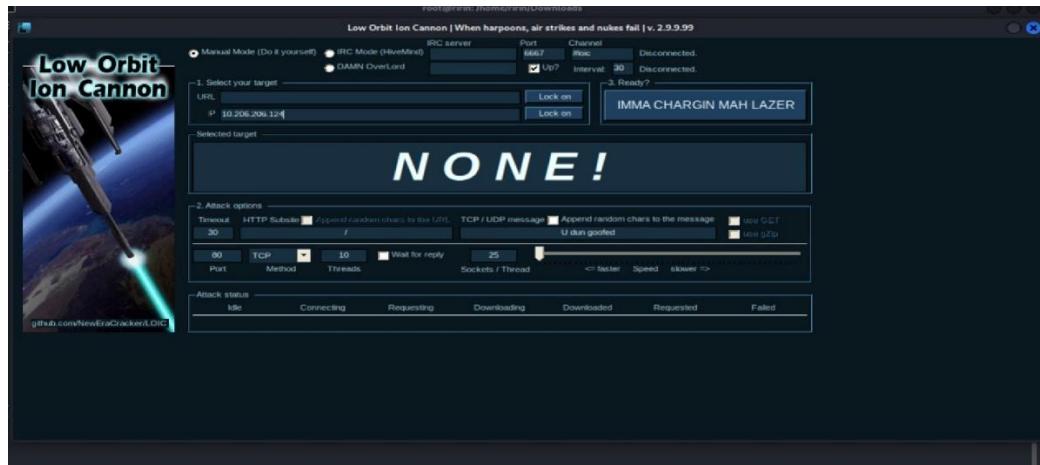
7. Ddos

a. Masuk ke LOIC (Penyerang)

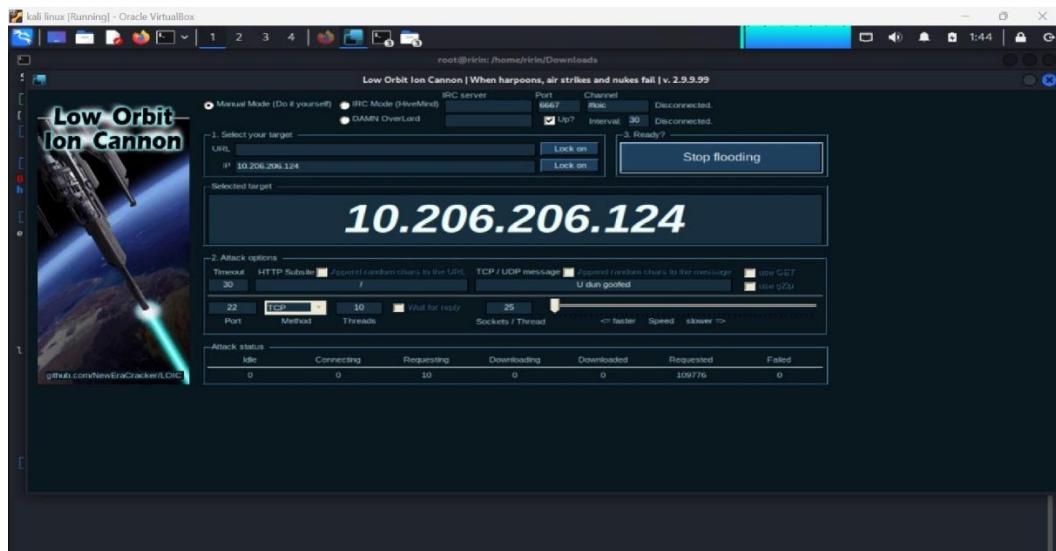
```
[root@ririn:~/home/ririn/loic/installer/LOIC/src/bin/Debug]
# mono LOIC.exe
Gtk not found (missing LD_LIBRARY_PATH to libgtk-x11-2.0.so.0?), using built-in colorscheme
```

Setelah berhasil melakukan simulasi serangan *Bruteforce*, tahap berikutnya dalam skenario serangan ganda (*Double Attack*) adalah meluncurkan serangan DDoS (Distributed Denial of Service) menggunakan perangkat lunak LOIC (Low Orbit Ion Cannon) dari sisi penyerang (root@ririn). Penyerang menavigasi terminal ke direktori binari aplikasi di /home/.../LOIC/src/bin/Debug dan mengeksekusi perintah mono LOIC.exe untuk menjalankan aplikasi berbasis .





Setelah perintah mono LOIC.exe dieksekusi, jendela aplikasi Low Orbit Ion Cannon (LOIC) akan terbuka, memungkinkan penyerang untuk melakukan konfigurasi target serangan secara grafis. Pada antarmuka utama, langkah pertama yang dilakukan adalah memasukkan alamat IP target, yaitu 10.206.206.124, ke dalam kolom "IP" yang tersedia di bagian *Select your target*. Setelah IP dimasukkan, penyerang menekan tombol "Lock on" untuk mengunci target sehingga sistem secara otomatis mengenali alamat tersebut sebagai titik akhir yang akan dibanjiri oleh paket data.



Setelah target berhasil dikunci pada alamat IP 10.206.206.124, langkah selanjutnya dalam antarmuka LOIC adalah melakukan konfigurasi parameter serangan pada bagian *Attack options*. Penyerang secara spesifik mengubah nilai "Port" menjadi 22 agar serangan banjir trafik tepat mengarah pada layanan SSH yang sedang disimulasikan oleh Honeypot. Selain itu, metode serangan ditetapkan pada protokol TCP dengan pengaturan *Threads* sebanyak 10 untuk memastikan volume paket data yang dikirimkan cukup masif untuk membebani target. Setelah seluruh parameter sesuai, penyerang menekan tombol "IMMA CHARGIN MAH LAZER", yang kemudian berubah status menjadi "Stop flooding", menandakan bahwa serangan

DDoS sedang berlangsung secara aktif. Indikator keberhasilan serangan ini terlihat pada bagian *Attack status*, di mana kolom *Requested* menunjukkan angka yang terus meningkat pesat hingga mencapai lebih dari 100.000 permintaan, membuktikan adanya pengiriman paket data dalam skala besar ke arah target.

b. Target

```
x-x64-lsb
2026-01-27T15:57:41+0000 [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'root' authenticated with b'password'
2026-01-27T15:57:41+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#debug] starting service b'ssh-connection'
2026-01-27T15:57:41+0000 [HoneyPotSSHTransport,3,10.206.206.39] avatar root logging out
2026-01-27T15:57:41+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2026-01-27T15:57:41+0000 [HoneyPotSSHTransport,3,10.206.206.39] Connection lost after 0.4 seconds
2026-01-27T15:57:41+0000 [HoneyPotSSHTransport,4,10.206.206.39] avatar root logging out
2026-01-27T15:57:41+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2026-01-27T15:57:41+0000 [HoneyPotSSHTransport,4,10.206.206.39] Connection lost after 0.4 seconds
2026-01-27T15:57:42+0000 [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'root' failed auth b'password'
2026-01-27T15:57:42+0000 [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] unauthorized login: ()
2026-01-27T15:58:12+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2026-01-27T15:58:12+0000 [HoneyPotSSHTransport,2,10.206.206.39] Connection lost after 31.4 seconds
2026-01-27T17:43:58+0000 [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha1
2026-01-27T17:43:59+0000 [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha256
2026-01-27T17:43:59+0000 [cowrie.ssh.factory.CowrieSSHFactory] New connection: 10.206.206.39:41772 (172.17.0.2:22
22) [session: eb50b0491e13]
2026-01-27T17:43:59+0000 [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha1
2026-01-27T17:43:59+0000 [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha256
2026-01-27T17:43:59+0000 [cowrie.ssh.factory.CowrieSSHFactory] New connection: 10.206.206.39:41784 (172.17.0.2:22
22) [session: 55c308b47324]
2026-01-27T17:43:59+0000 [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha1
2026-01-27T17:43:59+0000 [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha256
2026-01-27T17:43:59+0000 [cowrie.ssh.factory.CowrieSSHFactory] New connection: 10.206.206.39:41794 (172.17.0.2:22
22) [session: c00d7eabb9a7]
2026-01-27T17:43:59+0000 [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha1
2026-01-27T17:43:59+0000 [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha256
2026-01-27T17:43:59+0000 [cowrie.ssh.factory.CowrieSSHFactory] New connection: 10.206.206.39:41802 (172.17.0.2:22
22) [session: 85bd2aec1770]
2026-01-27T17:44:00+0000 [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha1
2026-01-27T17:44:00+0000 [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha256
2026-01-27T17:44:00+0000 [cowrie.ssh.factory.CowrieSSHFactory] New connection: 10.206.206.39:41806 (172.17.0.2:22
22) [session: 3c8b1cd52ccf]
2026-01-27T17:44:00+0000 [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha1
2026-01-27T17:44:00+0000 [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha256
2026-01-27T17:44:00+0000 [cowrie.ssh.factory.CowrieSSHFactory] New connection: 10.206.206.39:41814 (172.17.0.2:22
22) [session: 7e79811f924]
2026-01-27T17:44:01+0000 [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha1
2026-01-27T17:44:01+0000 [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha256
2026-01-27T17:44:01+0000 [cowrie.ssh.factory.CowrieSSHFactory] New connection: 10.206.206.39:41828 (172.17.0.2:22
22) [session: 3a650affa17c]
2026-01-27T17:44:01+0000 [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha1
2026-01-27T17:44:01+0000 [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha256
2026-01-27T17:44:01+0000 [cowrie.ssh.factory.CowrieSSHFactory] New connection: 10.206.206.39:41842 (172.17.0.2:22
22) [session: cee57ea5fd94]
2026-01-27T17:44:02+0000 [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha1
2026-01-27T17:44:02+0000 [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha256
2026-01-27T17:44:02+0000 [cowrie.ssh.factory.CowrieSSHFactory] New connection: 10.206.206.39:41858 (172.17.0.2:22
22) [session: c64409536d0d]
2026-01-27T17:44:02+0000 [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha1
2026-01-27T17:44:02+0000 [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha256
22) [session: /4b58daard26]
2026-01-27T17:45:27+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2026-01-27T17:45:27+0000 [HoneyPotSSHTransport,14,10.206.206.39] Connection lost after 84.7 seconds
2026-01-27T17:45:27+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2026-01-27T17:45:27+0000 [HoneyPotSSHTransport,5,10.206.206.39] Connection lost after 88.2 seconds
2026-01-27T17:45:27+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2026-01-27T17:45:27+0000 [HoneyPotSSHTransport,9,10.206.206.39] Connection lost after 87.2 seconds
2026-01-27T17:45:27+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2026-01-27T17:45:27+0000 [HoneyPotSSHTransport,10,10.206.206.39] Connection lost after 86.7 seconds
2026-01-27T17:45:27+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2026-01-27T17:45:27+0000 [HoneyPotSSHTransport,7,10.206.206.39] Connection lost after 88.2 seconds
2026-01-27T17:45:27+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2026-01-27T17:45:27+0000 [HoneyPotSSHTransport,13,10.206.206.39] Connection lost after 85.3 seconds
2026-01-27T17:45:27+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2026-01-27T17:45:27+0000 [HoneyPotSSHTransport,8,10.206.206.39] Connection lost after 87.8 seconds
2026-01-27T17:45:27+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2026-01-27T17:45:27+0000 [HoneyPotSSHTransport,6,10.206.206.39] Connection lost after 88.2 seconds
2026-01-27T17:45:27+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2026-01-27T17:45:27+0000 [HoneyPotSSHTransport,11,10.206.206.39] Connection lost after 86.3 seconds
2026-01-27T17:45:27+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2026-01-27T17:45:27+0000 [HoneyPotSSHTransport,12,10.206.206.39] Connection lost after 85.8 seconds
```

Setelah serangan DDoS dilancarkan menggunakan LOIC, log pada sisi target (ummul@kali) menunjukkan lonjakan aktivitas koneksi yang sangat masif dan terjadi secara simultan. Honeypot Cowrie merekam ribuan baris log baru bertuliskan New connection: 10.206.206.39 dalam interval waktu yang sangat singkat, yang mengonfirmasi bahwa target sedang dibanjiri permintaan dari alamat

IP penyerang. Karena volume permintaan yang sangat tinggi dan sifat serangan TCP yang berulang, sistem mulai mencatat kegagalan protokol seperti ketiadaan modul diffie-hellman-group-exchange untuk setiap sesi baru yang dibuat. Selanjutnya, log menunjukkan rentetan pesan connection lost yang berurutan, menandakan bahwa koneksi tersebut segera diputus setelah membebani sumber daya Honeypot. Fenomena ini membuktikan bahwa serangan DDoS berhasil menciptakan beban kerja yang signifikan pada layanan SSH palsu tersebut, sekaligus menunjukkan kemampuan Cowrie dalam mendokumentasikan setiap paket banjir data sebagai bukti forensik serangan *denial of service*.

8. Hasil Penyerangan

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

9. Kesimpulan

Berdasarkan seluruh rangkaian simulasi yang telah dilakukan, dapat disimpulkan bahwa implementasi Honeypot Cowrie di dalam lingkungan Docker pada sistem target (10.206.206.124) telah berhasil berfungsi sebagai sistem pertahanan aktif yang sangat efektif. Honeypot ini terbukti mampu mendeteksi dan merekam setiap tahapan serangan secara *real-time*, mulai dari fase pengintaian menggunakan Nmap, upaya masuk paksa (*Bruteforce*) melalui Hydra, hingga serangan banjir trafik (DDoS) menggunakan LOIC. Kemampuan Cowrie dalam mensimulasikan layanan SSH palsu berhasil mengelabui penyerang sehingga setiap perintah, percobaan kata sandi, dan volume paket data yang dikirimkan terdokumentasi secara akurat di dalam log sistem. Dengan demikian, penggunaan Honeypot tidak hanya berperan sebagai alat deteksi dini, tetapi juga sebagai sumber data forensik yang sangat berharga untuk menganalisis teknik serangan tanpa membahayakan integritas sistem operasi yang sebenarnya.