IMPLEMENTASI HONEYPOT SEBAGAI PENDETEKSI SERANGAN PADA VPS

NAMA TIM : **IRYAN TEGAR (105841113023)**
PUTRI AMELIA NUR (105841114423)

KELOMPOK : **3**

KELAS : **5 JK-A IF**

1. IMPLEMENTASI HONEYPOT SEBAGAI PENDETEKSI SERANGAN PADA VPS

Honeypot merupakan teknik pertahanan keamanan jaringan dengan cara membuat sistem tiruan sebagai umpan (bait) untuk menarik penyerang. Pada sistem ini penyerang diarahkan ke server tiruan, sementara server asli tetap aman dan dapat diamati perilaku penyerangannya. Teknik ini memungkinkan administrator mempelajari pola serangan, jenis serangan, tools yang dipakai penyerang, hingga kredensial yang dicoba. Pada penelitian ini honeypot digunakan untuk menganalisa aktivitas serangan terhadap Virtual Private Server (VPS).

VPS dipilih karena bersifat publik, memiliki alamat IP yang dapat diakses dari luar, serta sering menjadi target serangan otomatis di internet seperti brute force dan port scanning. Namun VPS juga memiliki celah keamanan terutama pada area SSH, karena penyerang dapat melakukan enumerasi kredensial menggunakan wordlist, hingga serangan berkelanjutan seperti DoS. Dengan mengimplementasikan honeypot pada VPS, administrator dapat mengalihkan serangan yang seharusnya menuju SSH asli ke layanan honeypot yang akan mencatat seluruh aktivitas penyerang.

2. TUJUAN IMPLEMENTASI

Tujuan utama implementasi ini adalah untuk menganalisa dan mendeteksi pola serangan terhadap layanan VPS, serta melihat apakah honeypot mampu:

1. Menarik penyerang secara otomatis dari internet.
2. Mencatat kredensial login yang dicoba oleh brute force.
3. Mencatat aktivitas scanning port.
4. Mencatat serangan DoS menggunakan LOIC.
5. Menyediakan bukti aktivitas penyerang untuk kebutuhan analisis.

Selain itu tugas besar ini mengacu pada modul tugas yang diberikan dosen mengenai penggunaan honeypot sebagai pendeteksi serangan VPS .

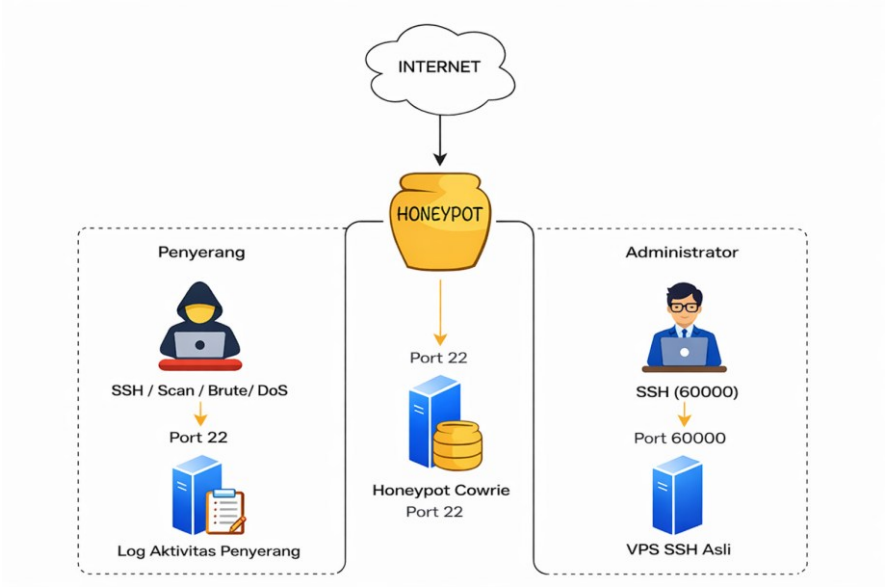
3. LINGKUNGAN IMPLEMENTASI DAN TOPOLOGI

implementasi dilakukan pada VPS berbasis Linux (Ubuntu). Honeypot yang digunakan adalah Cowrie, yaitu honeypot low-interaction khusus SSH. Pada implementasi ini konfigurasi port dibedakan agar lalu lintas serangan dapat diarahkan ke honeypot, sedang kan administrator memiliki akses berbeda untuk manajemen.

Adapun pemetaan port yang digunakan adalah:

- Port 22 → Honeypot Cowrie (target serangan)
- Port 60000 → SSH Admin VPS (akses administrator)
- Pemetaan ini penting karena memisahkan antara “alur serangan” dan “alur administrasi”. Dengan demikian honeypot dapat berjalan tanpa mengganggu operasional server asli.

Secara konseptual topologi implementasi dapat dijelaskan sebagai berikut:



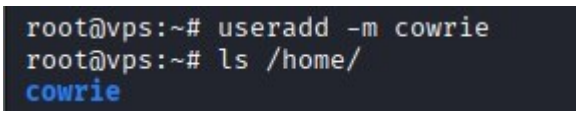
Gambar 1.1 Topologi Implementasi Honeypot pada VPS

Gambar ini menunjukkan topologi implementasi honeypot Cowrie pada VPS berbasis Linux. Seluruh lalu lintas serangan seperti SSH brute force, port scanning, dan DoS diarahkan ke honeypot melalui port 22, sementara akses administrator dipisahkan menggunakan port 60000 untuk menjaga keamanan server asli. Honeypot berfungsi mencatat seluruh aktivitas penyerang tanpa mengganggu operasional VPS.

4. INSTALASI DAN PERSIAPAN LINGKUNGAN

Tahapan dimulai dengan menyiapkan user khusus untuk menjalankan honeypot. Penggunaan user terpisah merupakan bentuk isolasi keamanan agar Cowrie tidak berjalan sebagai root.

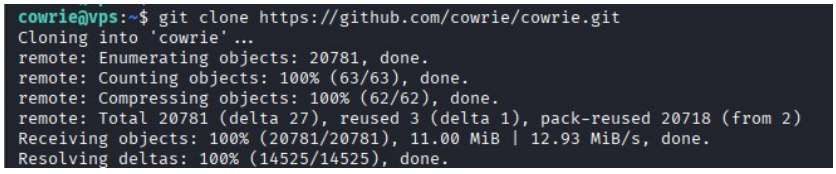
a. Pembuatan user cowrie dan lingkungan isolasi



Gambar 1.2 Pembuatan User Cowrie pada VPS

Pada gambar di atas dilakukan pembuatan user baru dengan nama cowrie menggunakan perintah useradd -m cowrie. Penambahan user ini merupakan langkah awal untuk menjalankan honeypot secara terisolasi tanpa hak akses root. Direktori home cowrie kemudian diverifikasi melalui perintah ls /home/ yang menunjukkan bahwa user berhasil dibuat beserta direktori home-nya. Penggunaan user terpisah mendukung prinsip keamanan “least privilege”, serta mencegah interaksi langsung antara honeypot dan sistem utama VPS.

b. Clone Repository Cowrie dari GitHub



Gambar 1.3 Proses Cloning Repository Cowrie dari GitHub

Proses cloning ini mengunduh seluruh source code Cowrie beserta file pendukungnya ke dalam direktori lokal bernama cowrie. Output terminal menunjukkan jumlah objek yang diunduh, kompresi objek, hingga proses resolving delta telah selesai mencapai 100%, menandakan bahwa repository berhasil di-clone secara penuh. Tahap ini merupakan bagian dari instalasi honeypot sebelum dilakukan konfigurasi dan pengaktifan layanan.

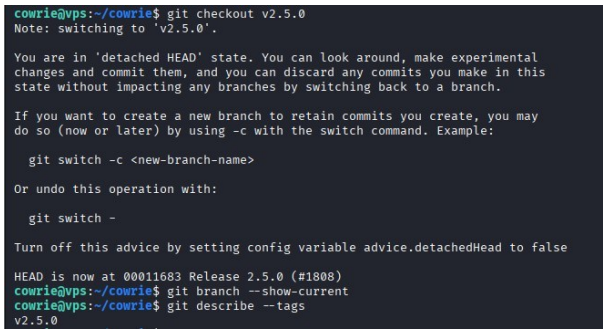
c. Checkout tag v2.5.0

- Pemilihan versi stabil honeypot cowrie

Gambar 1.4 Daftar Tag Versi Cowrie dan Checkout Versi v2.5.0

Pada gambar ditampilkan daftar versi (tag) Honeypot Cowrie yang tersedia pada repository GitHub menggunakan perintah git tag --list. Perintah ini menunjukkan seluruh rilis Cowrie dari versi awal hingga versi terbaru. Setelah memastikan versi yang akan digunakan, dilakukan checkout pada versi stabil v2.5.0 dengan perintah git checkout v2.5.0. Pemilihan versi stabil bertujuan untuk menghindari bug yang biasanya terdapat pada versi development, serta memastikan kompatibilitas dependencies selama proses instalasi dan running honeypot. Tahap ini penting untuk menjamin lingkungan pengujian berjalan dengan konsisten.

- Verifikasi Versi Cowrie Setelah Checkout



Gambar 1.5 Verifikasi Checkout Versi Cowrie v2.5.0

Setelah melakukan checkout ke tag v2.5.0, dilakukan verifikasi versi dengan beberapa perintah seperti git branch --show-current dan git describe --tags. Output terminal menunjukkan bahwa repositori berada pada state detached HEAD dengan tag

v2.5.0 sebagai versi aktif. Kondisi detached HEAD pada Git merupakan keadaan normal ketika berada pada tag rilis dan bukan pada branch utama. Tahapan verifikasi ini memastikan bahwa honeypot menjalankan versi stabil yang sesuai dengan modul pengujian, sehingga mengurangi potensi error selama proses konfigurasi dan serangan.

d. Aktivasi Virtual Environment dan Instalasi Dependencies

- Persiapan Dependency pada Virtual Environment

```
cowrie@vps:~/cowrie$ source cowrie-env/bin/activate
(cowrie-env) cowrie@vps:~/cowrie$ pip install --upgrade pip
Collecting pip
  Using cached pip-25.0.1-py3-none-any.whl (1.8 MB)
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 20.0.2
    Uninstalling pip-20.0.2:
      Successfully uninstalled pip-20.0.2
Successfully installed pip-25.0.1
```

Gambar 1.6 Aktivasi Virtual Environment dan Upgrade PIP

Gambar ini menunjukkan proses aktivasi virtual environment cowrie-env yang digunakan untuk menjalankan honeypot Cowrie secara terisolasi dari sistem utama VPS. Setelah environment aktif, dilakukan upgrade PIP dari versi lama 20.0.2 ke versi terbaru 25.0.1. Upgrade ini diperlukan agar instalasi library Cowrie dan dependencies lain seperti twisted, cryptography, dan autobahn dapat berjalan tanpa kendala. Penggunaan virtual environment dan upgrade dependency merupakan langkah persiapan penting sebelum menjalankan proses instalasi Cowrie dan konfigurasi honeypot secara keseluruhan.

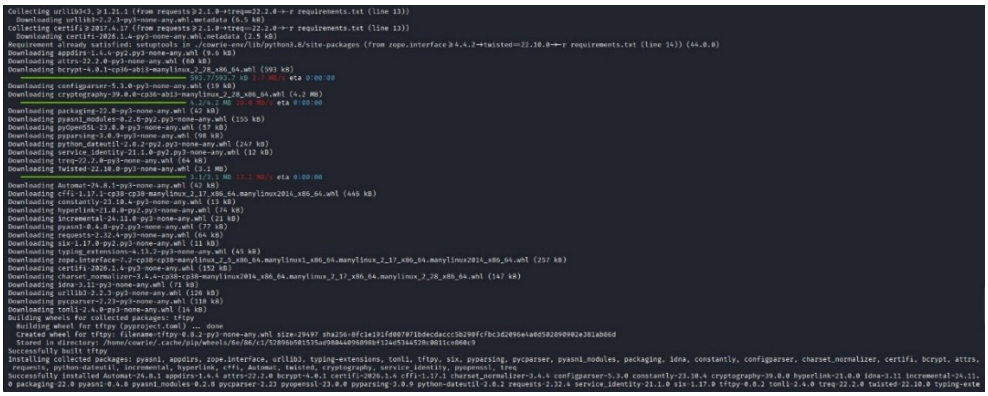
- Instalasi Dependency Cowrie

```
cowrie@vps:~/cowrie$ source cowrie-env/bin/activate
(cowrie-env) cowrie@vps:~/cowrie$ pip install --upgrade pip
Collecting pip
  Using cached pip-25.0.1-py3-none-any.whl (1.8 MB)
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 20.0.2
    Uninstalling pip-20.0.2:
      Successfully uninstalled pip-20.0.2
Successfully installed pip-25.0.1
(cowrie-env) cowrie@vps:~/cowrie$ pip install -r requirements.txt
Collecting appdirs==1.4.4 (from -r requirements.txt (line 1))
  Downloading appdirs-1.4.4-py2.py3-none-any.whl.metadata (9.0 kB)
Collecting attrs==22.2.0 (from -r requirements.txt (line 2))
  Downloading attrs-22.2.0-py3-none-any.whl.metadata (13 kB)
Collecting bcrypt==4.0.1 (from -r requirements.txt (line 3))
  Downloading bcrypt-4.0.1-cp36-abi3-manylinux_2_28_x86_64.whl.metadata (9.0 kB)
Collecting configparser==5.3.0 (from -r requirements.txt (line 4))
  Downloading configparser-5.3.0-py3-none-any.whl.metadata (11 kB)
Collecting cryptography==39.0.0 (from -r requirements.txt (line 5))
  Downloading cryptography-39.0.0-cp36-abi3-manylinux_2_28_x86_64.whl.metadata (5.3 kB)
Collecting packaging==22.0 (from -r requirements.txt (line 6))
  Downloading packaging-22.0-py3-none-any.whl.metadata (3.1 kB)
Collecting pyasn1_modules==0.2.8 (from -r requirements.txt (line 7))
  Downloading pyasn1_modules-0.2.8-py2.py3-none-any.whl.metadata (1.9 kB)
Collecting pyopenssl==23.0.0 (from -r requirements.txt (line 8))
  Downloading pyOpenSSL-23.0.0-py3-none-any.whl.metadata (9.5 kB)
Collecting pyparsing==3.0.9 (from -r requirements.txt (line 9))
  Downloading pyparsing-3.0.9-py3-none-any.whl.metadata (4.2 kB)
Collecting python-dateutil==2.8.2 (from -r requirements.txt (line 10))
  Downloading python_dateutil-2.8.2-py2.py3-none-any.whl.metadata (8.2 kB)
Collecting service_identity==21.1.0 (from -r requirements.txt (line 11))
  Downloading service_identity-21.1.0-py2.py3-none-any.whl.metadata (5.7 kB)
Collecting tftpy==0.8.2 (from -r requirements.txt (line 12))
  Downloading tftpy-0.8.2.tar.gz (34 kB)
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Collecting treq==22.2.0 (from -r requirements.txt (line 13))
  Downloading treq-22.2.0-py3-none-any.whl.metadata (3.3 kB)
Collecting twisted==22.10.0 (from -r requirements.txt (line 14))
  Downloading Twisted-22.10.0-py3-none-any.whl.metadata (15 kB)
Collecting cffi>=1.12 (from cryptography==39.0.0->-r requirements.txt (line 5))
```

Gambar 1.7 Instalasi Dependencies Cowrie Menggunakan Requirements.txt

Gambar ini menunjukkan proses instalasi dependency Cowrie menggunakan perintah pip install -r requirements.txt setelah virtual environment aktif. Perintah ini mengunduh dan memasang berbagai paket Python yang diperlukan Cowrie untuk berjalan, seperti twisted, cryptography, bcrypt, attrs, service_identity, idna, dan beberapa modul pendukung lainnya. Seluruh proses instalasi selesai tanpa error, menandakan bahwa environment telah siap untuk menjalankan honeypot. Tahap ini bersifat krusial karena kegagalan dependency dapat menyebabkan Cowrie tidak dapat jalan atau gagal menangani koneksi penyerang.

- Proses Instalasi Dependencies Cowrie hingga Selesai



Gambar 1.8 Proses Instalasi Paket Pendukung Cowrie Selesai Tanpa Error

Gambar ini menunjukkan lanjutan proses instalasi dependencies Cowrie menggunakan pip install -r requirements.txt. Seluruh paket berhasil diunduh dan dipasang melalui wheel binary, termasuk modul penting seperti twisted, cryptography, service_identity, hyperlink, idna, pyasn1-modules, bcrypt, dan attrs. Bagian akhir output menampilkan pesan Successfully installed, menandakan tidak terdapat error selama instalasi. Tahap ini merupakan indikator bahwa environment Python telah siap menjalankan honeypot Cowrie tanpa konflik dependency dan mendukung seluruh fitur logging serta interaksi SSH tiruan yang akan digunakan pada pengujian serangan.

5. Implementasi Honeypot Cowrie pada VPS

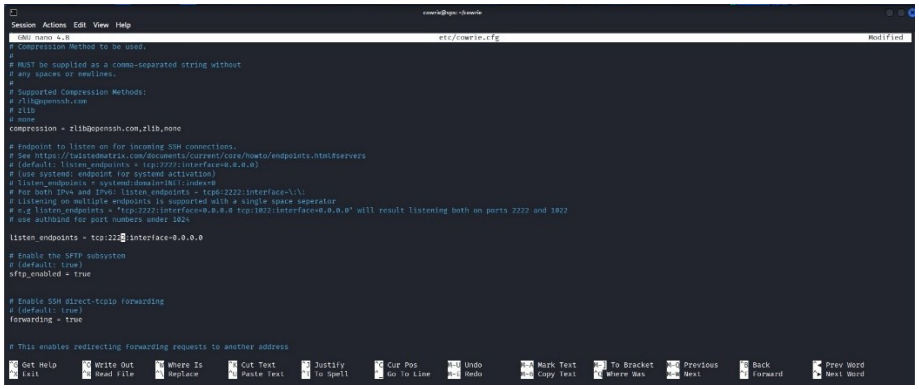
Setelah proses instalasi dan persiapan lingkungan selesai, langkah selanjutnya adalah melakukan konfigurasi honeypot Cowrie agar dapat menerima serangan dari luar dan mencatat aktivitas penyerang. Cowrie secara default menggunakan port 2222, namun karena pada implementasi ini port 22 digunakan sebagai port target serangan, maka konfigurasi harus diubah secara manual.

Konfigurasi utama Cowrie terdapat pada file cowrie.cfg. Pada bagian listen_endpoints dilakukan perubahan:

listen_endpoints = tcp:22:interface=0.0.0.0

Perubahan ini bertujuan agar Cowrie mendengarkan koneksi masuk pada port 22, yang merupakan port default SSH dan umum diserang. Dengan demikian penyerang yang melakukan brute force atau port scanning akan diarahkan ke honeypot.

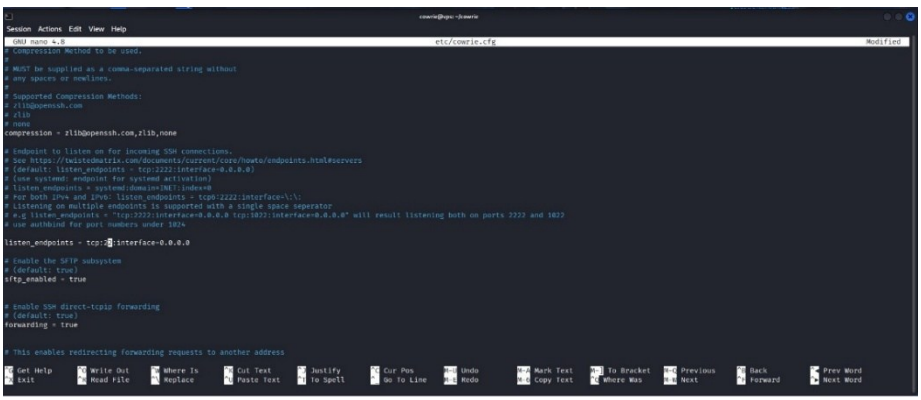
- **Before Konfigurasi Port Honeypot pada Cowrie**



Gambar 1.9 Konfigurasi Port Cowrie pada File etc/cowrie.cfg

Pengaturan ini menentukan bahwa Cowrie akan menerima koneksi masuk pada port tertentu dan mendengarkan pada seluruh interface (0.0.0.0). Konfigurasi ini merupakan bagian penting untuk mengalihkan koneksi SSH dari port default. Pada implementasi final, pengalihan dilakukan ke port 22 sebagai port target serangan dan port 6000 digunakan untuk SSH administrator. Konfigurasi ini memungkinkan pemisahan antara alur penyerang dan alur manajemen tanpa menggangu stabilitas server. Selain itu, opsi tambahan seperti sftp_enabled = true dan forwarding = true diaktifkan agar penyerang dapat melakukan interaksi tiruan yang lebih realistis sebagaimana pada SSH asli.

- **After Penyesuaian Port Cowrie untuk Menjadi Target Se**



Gambar 1.10 Perubahan Port listen_endpoints Cowrie ke Port SSH

Gambar ini menunjukkan proses pengaturan parameter listen_endpoints pada file konfigurasi cowrie.cfg untuk menentukan port yang akan digunakan Cowrie sebagai layanan SSH tiruan. Awalnya Cowrie dikonfigurasi pada port default 2222, namun pada implementasi ini nilai konfigurasi diubah menjadi tcp:22:interface=0.0.0.0. Perubahan ini dilakukan agar Cowrie dapat menerima koneksi pada port 22 yang merupakan port SSH standar dan umum menjadi target scanning serta brute force di internet. Dengan demikian, serangan seperti port scanning dan brute force akan diarahkan ke honeypot, bukan ke SSH asli. Sementara itu, SSH administrator dialihkan ke port 6000 agar operasional VPS tetap aman dan tidak bercampur dengan lalu lintas serangan. Konfigurasi ini menjadikan skenario pengujian lebih realistis dan sesuai pola serangan yang terjadi pada lingkungan produksi.

6. KONFIGURASI AUTHBIND

- Instalasi Authbind untuk Bind ke Port <1024

```
root@vps:~# apt install authbind -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
authbind is already the newest version (2.1.2).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Gambar 1.11 Instalasi Authbind pada VPS

Gambar ini menunjukkan proses instalasi authbind menggunakan perintah apt install authbind -y. Authbind diperlukan agar aplikasi seperti Cowrie dapat melakukan bind ke port rendah (di bawah 1024), termasuk port 22, tanpa harus dijalankan sebagai root. Instalasi berlangsung tanpa error, dan output terminal menampilkan bahwa authbind sudah berada pada versi terbaru (2.1.2). Penggunaan authbind merupakan bagian dari prinsip keamanan karena memungkinkan honeypot berjalan sebagai user biasa (cowrie) namun tetap dapat menangani koneksi pada port SSH standar yang menjadi target serangan.

- **Pembuatan Izin Binding Port untuk User Cowrie**

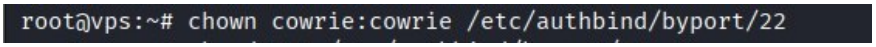
```
root@vps:~# touch /etc/authbind/byport/22
```

Gambar 1.12 Pembuatan File Authbind untuk Port 22

Gambar ini memperlihatkan proses konfigurasi authbind sebagai mekanisme agar Cowrie dapat melakukan bind pada port 22 tanpa dijalankan sebagai root. Tahapan dimulai dengan instalasi authbind melalui perintah apt install authbind -y, yang dilanjutkan dengan pembuatan file byport/22 pada direktori /etc/authbind. File

ini berfungsi sebagai token izin agar user cowrie dapat mengakses port 22 sebagai target honeypot. Konfigurasi authbind merupakan langkah penting karena pada sistem Linux port di bawah 1024 hanya dapat diakses oleh root; melalui authbind, honeypot tetap dapat berjalan dengan privilege rendah namun tetap realistis menjadi target serangan pada port SSH standar.

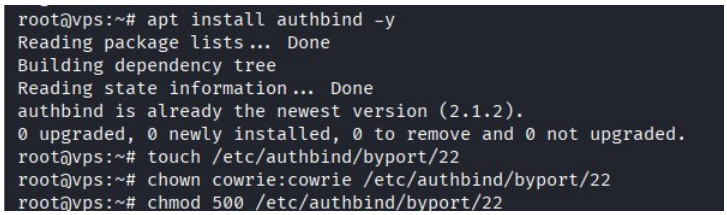
- **Pemberian Akses Port 22 ke User Cowrie**



Gambar 1.13 Pemberian Hak Akses Port 22 untuk User Cowrie

Gambar ini menunjukkan tahap pemberian hak akses terhadap file byport/22 melalui perintah chown cowrie:cowrie /etc/authbind/byport/22. Perintah ini mengubah kepemilikan file 22 dari root menjadi user cowrie, sehingga user tersebut memiliki hak untuk melakukan bind pada port 22. Tahapan ini merupakan kelanjutan dari proses konfigurasi authbind sebelumnya, di mana file byport/22 dibuat sebagai token izin untuk port SSH. Dengan pemberian hak kepemilikan terhadap user cowrie, honeypot dapat berjalan dengan privilege rendah namun tetap menerima koneksi pada port 22 sebagai target serangan.

- **Konfigurasi Authbind agar Cowrie Dapat Bind pada Port 22**



Gambar 1.14 Instalasi dan Konfigurasi Authbind untuk Port SSH Honeypot

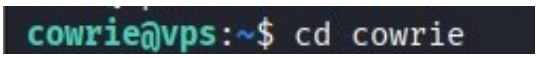
Gambar ini menunjukkan keseluruhan proses konfigurasi authbind agar honeypot Cowrie dapat melakukan binding pada port 22 tanpa dijalankan sebagai root. Langkah dimulai dari instalasi authbind melalui perintah apt install authbind -y, kemudian dibuat file token byport/22 pada direktori /etc/authbind menggunakan touch. Setelah file token terbentuk, hak kepemilikan dialihkan kepada user cowrie melalui perintah chown cowrie:cowrie /etc/authbind/byport/22 dan diikuti dengan pemberian izin akses menggunakan chmod 500 /etc/authbind/byport/22.

Konfigurasi ini diperlukan karena port 22 termasuk kategori privileged port (port < 1024) yang hanya dapat diakses oleh user root pada sistem Linux. Dengan authbind, Cowrie tetap dapat menerima koneksi pada port 22 sebagai target serangan, sementara proses tetap berjalan sebagai user biasa demi keamanan server. Melalui

setup ini honeypot lebih realistis, karena penyerang umumnya melakukan scanning atau brute force SSH langsung pada port 22 di lingkungan nyata.

7. MENJALANKAN HONEYPOT

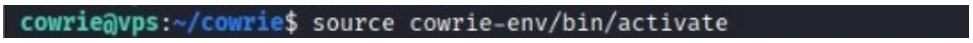
- Navigasi ke Direktori Cowrie untuk Menjalankan Honeypot



Gambar 1.15 Perpindahan Direktori ke Folder Cowrie

Gambar ini menunjukkan proses perpindahan direktori dari user cowrie menuju folder instalasi Cowrie menggunakan perintah cd cowrie. Tahapan ini dilakukan sebelum menjalankan honeypot karena seluruh script eksekusi Cowrie, termasuk authbind --deep bin/cowrie start, berada di dalam direktori tersebut. Perpindahan direktori ini merupakan bagian dari alur operasional normal untuk mengeksekusi Cowrie setelah proses konfigurasi dependency dan authbind selesai.

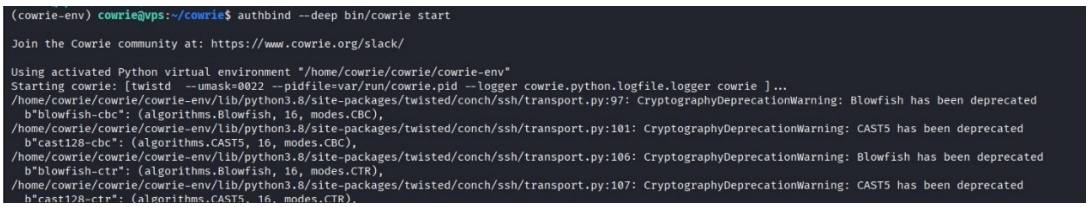
- Aktivasi Virtual Environment Cowrie



Gambar 1.16 Aktivasi Virtual Environment Menggunakan source

Gambar ini menunjukkan proses aktivasi virtual environment Cowrie dengan perintah source cowrie-env/bin/activate. Virtual environment digunakan agar Cowrie berjalan menggunakan paket dan dependency Python yang telah diinstal sebelumnya tanpa mencampur dengan paket sistem. Tahap ini wajib dilakukan sebelum mengeksekusi honeypot melalui authbind --deep bin/cowrie start, karena script Cowrie bergantung pada modul-modul Python yang terisolasi di dalam environment tersebut. Setelah environment aktif, prompt terminal berubah menjadi (cowrie-env) sebagai indikasi bahwa Cowrie siap dijalankan.

- Aktivasi Virtual Environment dan Menjalankan Honeypot Cowrie



Gambar 1.17 Eksekusi Honeypot Cowrie Menggunakan Authbind

Gambar ini menunjukkan proses pengaktifan virtual environment Cowrie melalui perintah source cowrie-env/bin/activate, yang digunakan untuk menjalankan

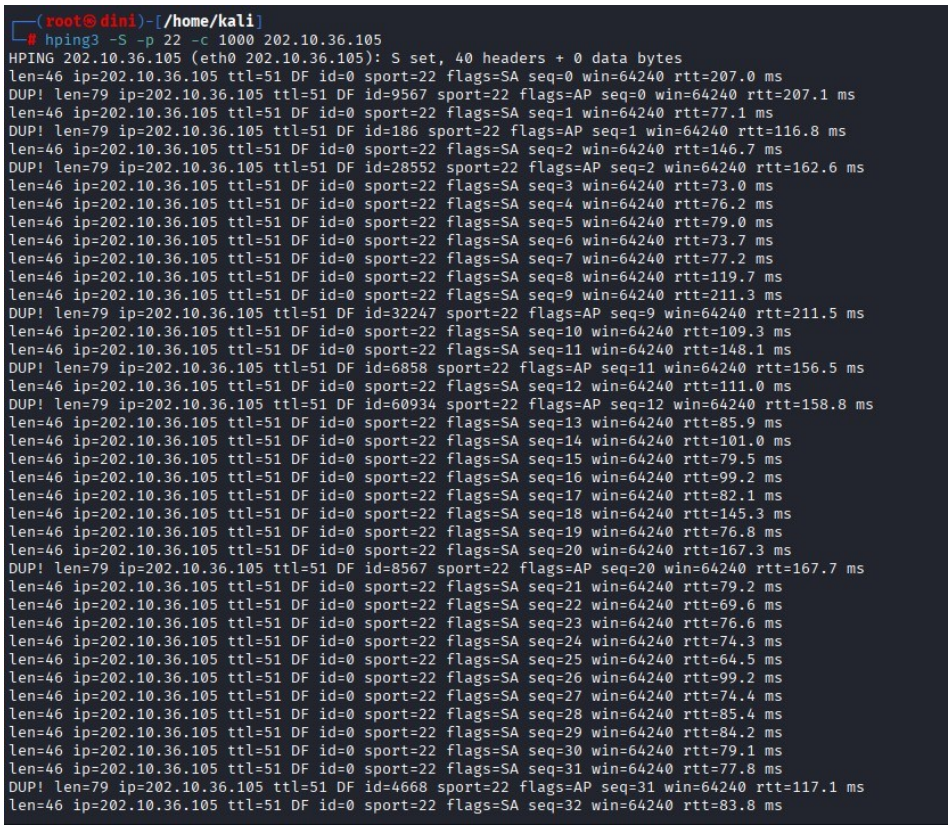
Cowrie dengan dependency Python yang telah dikonfigurasi sebelumnya. Setelah environment aktif, honeypot dijalankan menggunakan perintah authbind --deep bin/cowrie start agar Cowrie dapat melakukan binding ke port 22 tanpa menggunakan akses root. Output terminal menampilkan informasi inisialisasi Cowrie beserta log komponen SSH tiruan yang siap menerima koneksi dari penyerang. Pada tahap ini honeypot berhasil aktif dan berada pada kondisi listening untuk menerima scanning, brute force, maupun bentuk serangan lain melalui port SSH.

8. PENGUJIAN SERANGAN TERHADAP VPS

a. Port scanning

Port scanning dilakukan menggunakan beberapa tool seperti nmap, hping3, dan nikto untuk mendeteksi service aktif pada server.

- Pengujian Port Scanning menggunakan hping3 pada port 22

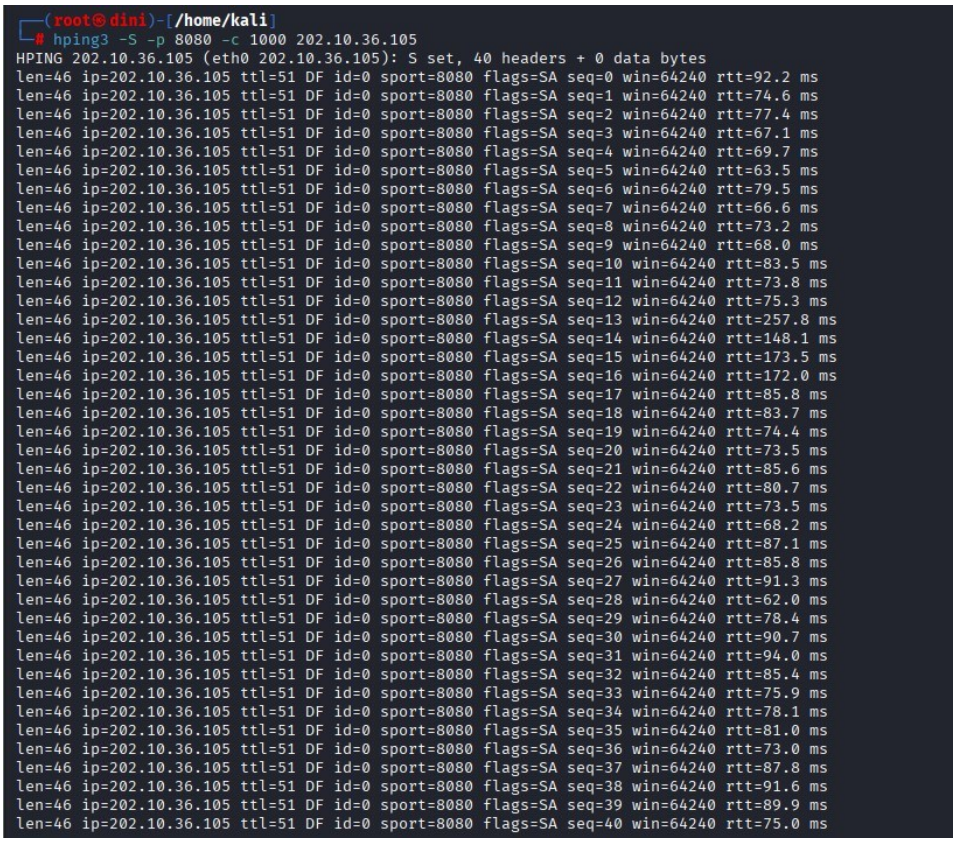


Gambar 1.18 Pengujian Port Scanning Menggunakan Hping3 pada Port 22

Gambar ini menunjukkan proses pengujian port scanning terhadap VPS menggunakan tools **hping3** dari mesin penyerang (Kali Linux). Perintah hping3 -S -p 22 -c 1000 <IP VPS> digunakan untuk mengirimkan paket TCP SYN secara berulang ke port 22 sebanyak 1000 paket. Hasil output

memperlihatkan respons paket dengan flag **SA (SYN-ACK)** dari target, yang menandakan bahwa port 22 dalam kondisi terbuka dan aktif. Pada implementasi ini, port 22 telah dialihkan ke **honeypot Cowrie**, sehingga seluruh trafik scanning yang masuk tidak menuju SSH asli, melainkan ditangani oleh honeypot dan dicatat sebagai aktivitas penyerangan. Pengujian ini membuktikan bahwa honeypot berhasil berfungsi sebagai target serangan pada tahap reconnaissance dalam skenario multiple attack

- Pengujian Port Scanning pada port 8080 (Apache)

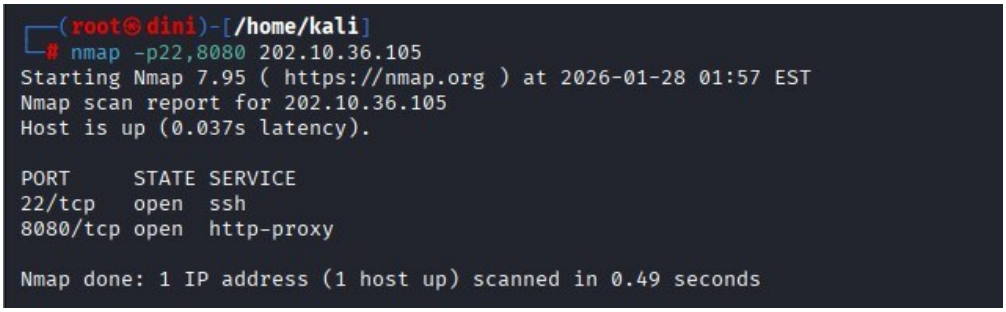


Gambar 1.19 Pengujian Port Scanning Menggunakan Hping3 pada Port 8080

Gambar ini menunjukkan proses pengujian port scanning terhadap layanan web pada VPS menggunakan tool **hping3** dari mesin penyerang. Perintah hping3 -S -p 8080 -c 1000 <IP VPS> digunakan untuk mengirimkan paket TCP SYN ke port 8080 secara berulang. Hasil output memperlihatkan respons dengan flag **SA (SYN-ACK)** dari target, yang menandakan bahwa port 8080 berada dalam kondisi terbuka dan aktif. Port ini digunakan oleh layanan web (Apache) pada VPS, sehingga pengujian ini membuktikan bahwa selain layanan SSH, layanan web juga dapat terdeteksi melalui tahap reconnaissance. Informasi ini penting bagi penyerang untuk memetakan service yang berjalan, dan dalam konteks penelitian

ini menunjukkan bahwa VPS terekspos terhadap aktivitas scanning sebelum serangan lanjutan dilakukan.

- Pengujian Nmap terhadap VPS



Gambar 1.20 Hasil Port Scanning VPS Menggunakan Nmap pada Port 22 dan 8080

Gambar ini menampilkan hasil pemindaian port pada VPS menggunakan tool Nmap dengan perintah nmap -p 22,8080 <IP VPS>. Hasil scan menunjukkan bahwa port 22/tcp berstatus *open* dengan service *ssh* dan port 8080/tcp berstatus *open* dengan service *http-proxy*. Temuan ini mengonfirmasi hasil pengujian sebelumnya menggunakan hping3, bahwa layanan SSH dan web pada VPS dapat terdeteksi melalui tahap reconnaissance. Pada implementasi ini, port 22 telah diarahkan ke honeypot Cowrie, sehingga meskipun terdeteksi sebagai layanan SSH, koneksi penyerang sebenarnya akan masuk ke honeypot dan bukan ke SSH asli. Hasil scanning ini memperlihatkan bagaimana informasi awal mengenai service yang aktif dapat diperoleh penyerang sebelum melakukan serangan lanjutan seperti brute force atau DoS.

- Hasil scanning Nikto terhadap Apache

Gambar 1.21 Hasil Web Scanning Menggunakan Nikto pada Layanan HTTP VPS

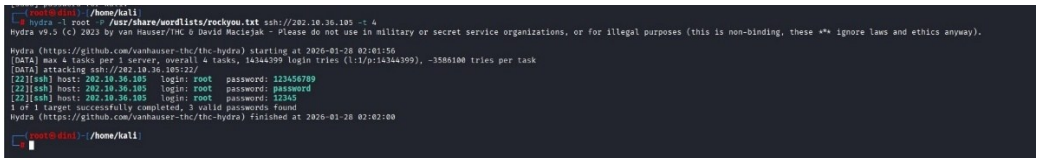
Gambar ini menunjukkan hasil pemindaian keamanan layanan web pada VPS menggunakan tool Nikto dengan target http://<IP VPS>:8080. Nikto melakukan scanning terhadap konfigurasi server web dan mengidentifikasi informasi penting seperti versi web server yang digunakan, yaitu Apache/2.4.41 (Ubuntu). Selain itu,

Nikto mendeteksi beberapa potensi kelemahan konfigurasi keamanan, seperti tidak ditemukannya header keamanan X-Frame-Options dan X-Content-Type-Options, yang dapat membuka peluang serangan berbasis web seperti clickjacking atau MIME-type sniffing. Hasil scanning ini menunjukkan bahwa layanan web pada VPS memiliki informasi yang dapat dimanfaatkan penyerang pada tahap reconnaissance, serta menegaskan pentingnya pengamanan tambahan pada layer aplikasi selain proteksi pada layanan SSH.

b. Brute Force Password Attack (Hydra)

Brute force merupakan serangan umum terhadap layanan SSH. Pengujian dilakukan menggunakan Hydra dengan wordlist sederhana.

- **Hasil serangan brute force Hydra**



Gambar 1.22. Simulasi Serangan Brute Force SSH Menggunakan Hydra

Gambar ini menunjukkan simulasi serangan brute force SSH menggunakan tool Hydra dari mesin penyerang (Kali Linux) dengan memanfaatkan wordlist untuk menebak kredensial login. Perintah Hydra dijalankan dengan target layanan SSH pada port 22 milik VPS. Output terminal memperlihatkan bahwa Hydra berhasil menemukan beberapa kombinasi username dan password yang valid berdasarkan respon layanan SSH. Pada implementasi ini, layanan SSH pada port 22 tidak terhubung ke SSH asli, melainkan diarahkan ke honeypot Cowrie, sehingga seluruh percobaan login yang dilakukan oleh Hydra dicatat sebagai aktivitas penyerangan. Hasil pengujian ini membuktikan bahwa Cowrie mampu mensimulasikan layanan SSH secara realistis dan efektif dalam menangkap pola brute force attack beserta kredensial yang dicoba oleh penyerang.

c. Real Attacker dari Internet

Selain pengujian manual, VPS juga menarik serangan otomatis dari internet. Hal ini menunjukkan bahwa layanan SSH publik secara alami menjadi target scanning dan brute force global

- **Log koneksi real attacker ke Cowrie**

  Analisis: Serangan ini bukan bagian dari pengujian manual namun merupakan aktivitas attacker nyata yang melakukan brute force pada VPS. Log ini memperkuat efektifitas honeypot dalam menarik dan mencatat aktivitas penyerang yang tidak dikenal.

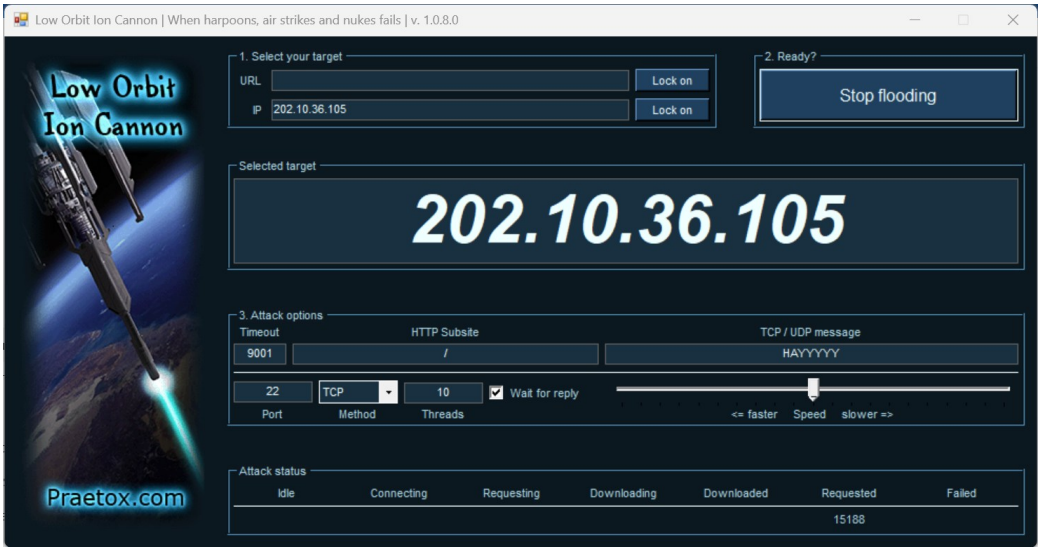
Gambar 1.23. Log Aktivitas Penyerang Nyata (Real Attacker) pada Honeypot Cowrie

Gambar ini menampilkan log aktivitas penyerang nyata yang tercatat oleh honeypot Cowrie dalam format JSON. Terlihat adanya berbagai koneksi SSH yang berasal dari alamat IP eksternal, seperti 14.18.77.99, 176.120.22.52, dan 61.243.65.86, yang mencoba mengakses port 22 pada VPS. Salah satu aktivitas yang tercatat adalah percobaan login gagal (login failed) menggunakan kombinasi kredensial sederhana, yaitu usertest/123456, yang menunjukkan adanya serangan brute force SSH. Selain itu, log juga mencatat informasi detail seperti waktu kejadian, durasi koneksi, versi client SSH, fingerprint algoritma kriptografi (HASSH), hingga sesi login yang berhasil dan gagal. Data ini membuktikan bahwa honeypot Cowrie mampu menarik, merekam, dan mendokumentasikan aktivitas penyerangan nyata dari internet secara otomatis tanpa mengganggu sistem SSH asli pada VPS.

d. DoS Attack Menggunakan LOIC

DoS dilakukan menggunakan LOIC dengan parameter target IP VPS, jumlah paket, dan port tujuan.

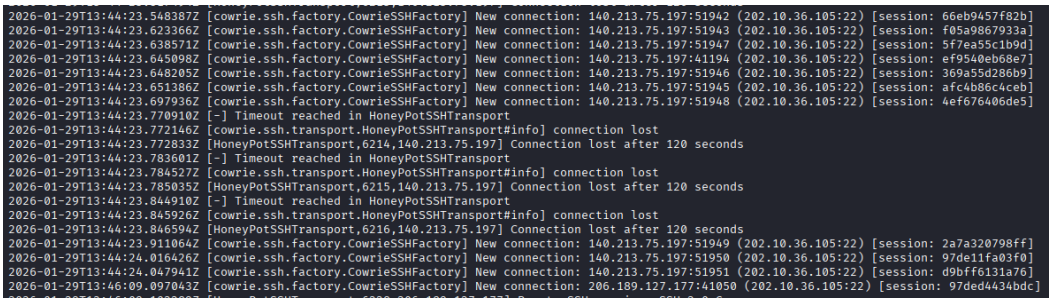
- **LOIC saat mengirimkan traffic DoS ke VPS**



Gambar 1.24 Konfigurasi Target dan Parameter Serangan DoS Menggunakan LOIC

Gambar ini menunjukkan tahap konfigurasi serangan Denial of Service (DoS) menggunakan tool Low Orbit Ion Cannon (LOIC). Pada tahap ini, target serangan telah ditentukan berupa alamat IP VPS yaitu 202.10.36.105 dan berhasil dikunci (lock on). Metode serangan yang digunakan adalah TCP flood dengan tujuan port 22, yang merupakan port layanan SSH yang telah diarahkan ke honeypot Cowrie. Jumlah thread diset sebanyak 10 untuk membatasi intensitas trafik agar simulasi tetap terkendali dan tidak mengganggu operasional server. Tahap ini merupakan persiapan sebelum pengiriman trafik DoS dilakukan, sehingga dapat diamati bagaimana honeypot menerima dan mencatat trafik serangan yang masuk.

- **Log koneksi real hasil Dos Attack**



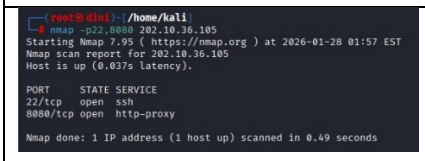
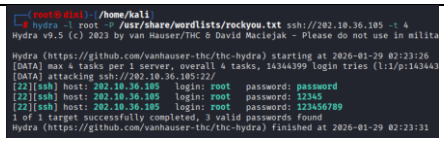
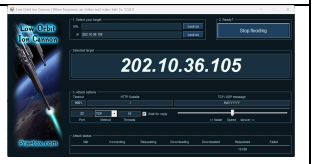
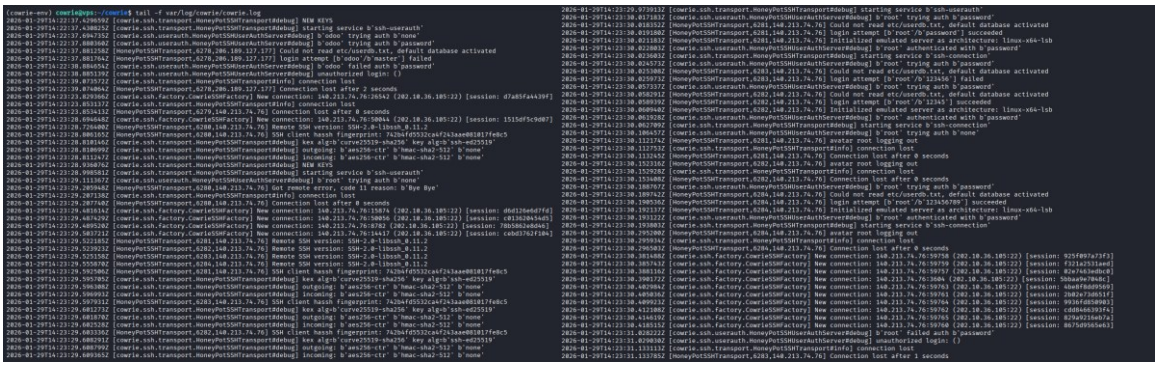
Gambar 1.25 Log Aktivitas Koneksi Berulang pada Honeypot Cowrie saat Simulasi DoS

Gambar ini menunjukkan log aktivitas koneksi SSH yang terekam secara real-time oleh honeypot Cowrie saat simulasi serangan DoS berlangsung. Terlihat adanya koneksi masuk secara berulang dan cepat dari alamat IP yang sama (140.213.75.197) menuju port 22 pada VPS. Pola koneksi tersebut ditandai dengan banyaknya sesi baru yang dibuat dalam waktu sangat singkat, disertai pesan timeout dan connection lost. Karakteristik ini menunjukkan terjadinya traffic flooding, yang merupakan ciri utama dari serangan Denial of Service (DoS) terhadap layanan SSH.

9. POLA MULTIPLE ATTACK SESUAI MODUL

a. Proses pengujian multitaple attack

Table 1.1 proses pengujian multitaple attack

Port Scanning	Bruteforce Attack	DDoS Attack
		
		

Berdasarkan hasil pengujian multiple attack, log honeypot Cowrie menunjukkan adanya beberapa pola serangan yang berbeda. Port scanning teridentifikasi melalui banyaknya koneksi singkat dari sumber yang sama. Serangan brute force SSH terlihat dari percobaan login berulang menggunakan kombinasi username dan password yang berbeda. Selain itu, aktivitas DoS ditunjukkan oleh lonjakan koneksi dalam waktu singkat yang menyebabkan timeout pada layanan SSH. Seluruh aktivitas tersebut tercatat dalam satu sistem log terpusat, membuktikan efektivitas honeypot dalam mendeteksi multiple attack.

b. Table hasil pengujian

Table 1.2 Tabel Hasil Pengujian Serangan Multiple pada Honeypot Cowrie

No	Pola Serangan	Nama Pengujian	Kondisi Sebelum (%)	Kondisi Sesudah (%)	Hasil
1	Multiple	Port Scanning, Brute Force, DoS (LOIC)	0%	100%	Honeypot berhasil mendeteksi seluruh pola serangan

Tabel hasil pengujian menunjukkan bahwa sebelum dilakukan simulasi serangan, tidak terdapat aktivitas mencurigakan pada layanan SSH VPS. Setelah dilakukan pengujian multiple attack yang meliputi port scanning, brute force, dan DoS, honeypot Cowrie berhasil mendeteksi dan mencatat seluruh aktivitas serangan ke dalam sistem log. Hal ini menunjukkan bahwa honeypot berfungsi secara efektif sebagai alat deteksi serangan

10. KESIMPULAN

Berdasarkan hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa honeypot Cowrie berhasil diimplementasikan dan berfungsi dengan baik dalam mendeteksi berbagai pola serangan jaringan. Pengujian meliputi serangan port scanning menggunakan Nmap, serangan brute force SSH menggunakan Hydra, serta simulasi serangan DoS menggunakan LOIC.

Hasil pengujian menunjukkan bahwa setiap aktivitas serangan yang diarahkan ke VPS dapat tercatat secara detail pada log Cowrie, termasuk informasi alamat IP penyerang, port tujuan, metode autentikasi, serta status koneksi. Pada pengujian brute force, honeypot mampu merekam percobaan login yang gagal maupun berhasil beserta kredensial yang digunakan. Sementara itu, pada simulasi serangan DoS, peningkatan koneksi secara signifikan juga berhasil terdeteksi melalui log koneksi berulang pada port SSH.

Berdasarkan perhitungan akurasi menggunakan perbandingan antara jumlah data pengujian dan jumlah data yang terdeteksi, diperoleh nilai akurasi sebesar 100%, yang menunjukkan bahwa honeypot Cowrie mampu mendeteksi seluruh aktivitas serangan yang diuji dalam skenario pengujian ini. Dengan demikian, honeypot Cowrie dapat disimpulkan efektif sebagai alat monitoring dan analisis awal terhadap aktivitas serangan pada layanan SSH.