TUGAS III

ADVANCED NETWORK SECURITY

OLEH :

NAMA :HERMI NUR SAFITRI

NIM :105841116223

KELAS :5 JK A

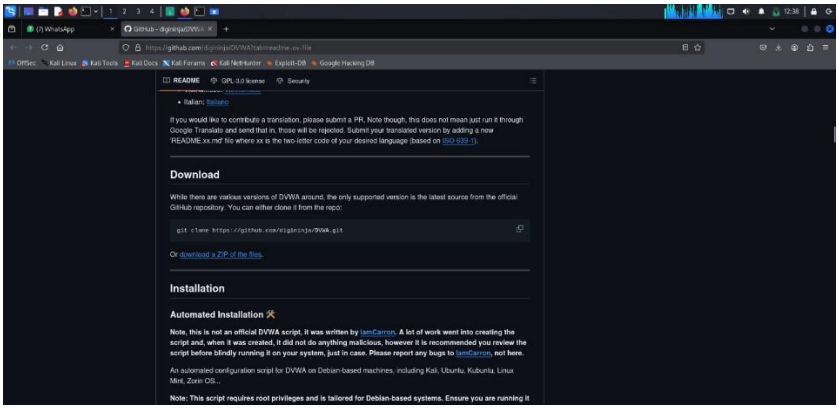
PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MUHAMMADIYAH MAKASSSAR

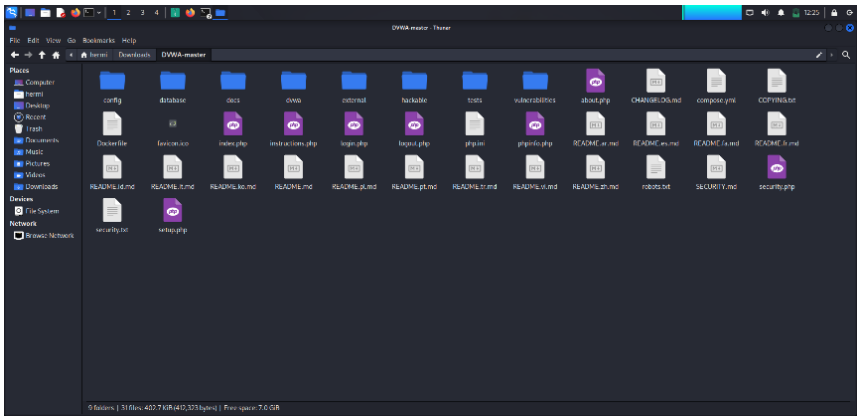
2025

1. Langkah Instalasi
    a. Proses instalasi



Install source code dari github

Pada langkah awal, dilakukan akses ke halaman repositori resmi DVWA di platform GitHub melalui peramban web. Gambar tersebut menunjukkan dokumentasi teknis yang menyediakan instruksi pengunduhan menggunakan perintah *git clone* maupun pengunduhan langsung dalam format ZIP. Terlihat pula informasi mengenai persyaratan sistem dan skrip instalasi otomatis yang dikhususkan untuk distribusi Linux berbasis Debian



Setelah file unduhan di extra

Setelah proses pengunduhan selesai, gambar kedua memperlihatkan isi dari direktori utama aplikasi yang telah diekstrak di dalam sistem operasi (menggunakan file manager Thunar di Linux). Struktur folder ini mencakup komponen-komponen vital seperti folder config yang digunakan untuk

pengaturan koneksi database, folder vulnerabilities yang berisi modul-modul celah keamanan, serta berbagai file skrip PHP yang membentuk antarmuka aplikasi. Keberadaan file-file ini di lingkungan lokal menandakan bahwa tahap persiapan data telah berhasil dilakukan dan aplikasi siap masuk ke tahap konfigurasi server web serta pengaturan basis data.

b. Pemindahan Direktori DVWA ke Web Server

Tahap Deployment ke Direktori Apache

Gambar ini mendokumentasikan proses Deployment, yaitu memindahkan aplikasi dari ruang kerja pribadi ke dalam direktori publik Web Server (dalam hal ini Apache). Penggunaan perintah mv dengan hak akses Superuser (root) dilakukan untuk menempatkan file di /var/www/html/. Direktori ini adalah Document Root standar pada sistem operasi Linux, di mana layanan web server akan mendengarkan permintaan HTTP dan menerjemahkan skrip PHP menjadi tampilan web yang dapat diakses oleh pengguna melalui protokol jaringan.

c. Perintah mv (Move) pada Terminal

Pemindahan Direktori Aplikasi ke Document Root Apache

Penggunaan perintah cd (change directory) yang berfungsi untuk memindahkan posisi kerja terminal ke dalam folder /var/www/html/dvwa. Langkah ini ibarat membuka folder tujuan di komputer agar kita bisa mengelola isinya secara langsung tanpa harus mengetik alamat lengkap folder tersebut secara berulang-ulang. Dengan masuk ke direktori operasional ini, kita dapat dengan mudah melakukan tahap berikutnya,

seperti mengubah pengaturan database di folder config atau mengatur izin akses file agar aplikasi DVWA dapat dijalankan dengan sempurna melalui browser. Proses ini sangat penting dalam administrasi sistem untuk memastikan bahwa setiap perintah konfigurasi yang dijalankan tepat sasaran pada file aplikasi yang sedang dipasang.

d. Pemberian Izin Akses Penuh (Recursive) pada Direktori DVWA

Eksekusi Perintah chmod (Change Mode)

Gambar ini menunjukkan pemberian izin akses penuh menggunakan perintah chmod -R 777 pada folder aplikasi. Kode 777 berarti semua pengguna (pemilik, grup, dan publik) memiliki hak untuk membaca, menulis, dan mengeksekusi file tersebut, sementara opsi -R memastikan aturan ini berlaku untuk seluruh folder dan file di dalamnya secara berjenjang. Langkah ini sering dilakukan di lingkungan pengujian agar server web tidak mengalami kendala saat harus melakukan operasi penulisan file atau pembuatan database pada aplikasi DVWA

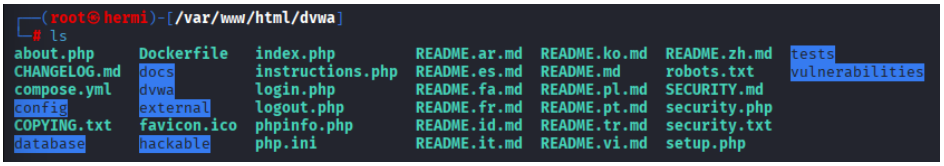
e. Duplikasi File Konfigurasi Default

Eksekusi Perintah cp (Copy)

Perintah cp config/config.inc.php.dist config/config.inc.php merupakan langkah krusial dalam inisialisasi aplikasi karena berfungsi untuk mengaktifkan berkas konfigurasi sistem. Secara teknis, aplikasi DVWA hanya menyediakan sebuah templat atau cetakan pengaturan bernama config.inc.php.dist yang tidak dapat dibaca langsung oleh server web sebagai instruksi operasional. Dengan melakukan penyalinan ini, kita membuat berkas konfigurasi aktif baru bernama config.inc.php yang nantinya akan menampung parameter penting seperti alamat host, nama

pengguna, dan kata sandi basis data. Prosedur ini sangat penting dilakukan agar aplikasi memiliki panduan untuk terhubung ke database MySQL/MariaDB, sekaligus memastikan bahwa berkas asli bertanda .dist tetap tersimpan sebagai cadangan atau referensi jika terjadi kesalahan saat proses pengeditan konfigurasi di kemudian hari

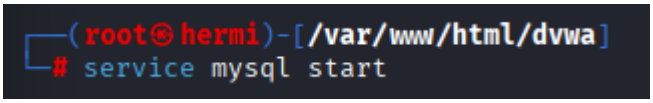
f. Pemeriksaan Struktur Direktori dan File DVWA



Eksekusi Perintah ls (List) di Terminal

penggunaan perintah ls yang berfungsi untuk menampilkan seluruh daftar file dan folder yang ada di dalam direktori kerja saat ini, yaitu /var/www/html/dvwa. Langkah ini bertujuan untuk memverifikasi secara visual bahwa semua komponen aplikasi—seperti folder config, database, vulnerabilities, serta file utama seperti index.php dan setup.php—telah berada di lokasi yang benar dan siap untuk digunakan. Verifikasi ini sangat penting dalam administrasi sistem untuk memastikan tidak ada data yang hilang selama proses pemindahan atau penyalinan sebelumnya, sehingga konfigurasi lanjutan melalui browser dapat berjalan tanpa kendala file tidak ditemukan (404 Not Found)

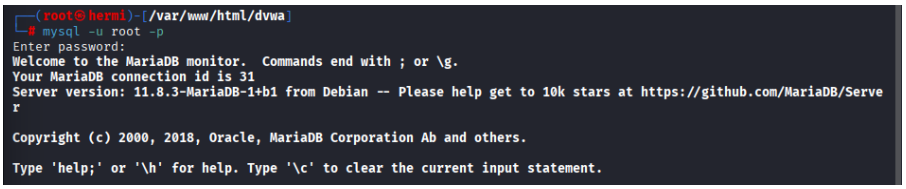
g. Memulai Layanan Database MySQL



Eksekusi Perintah service

Proses mengaktifkan layanan *database* menggunakan perintah service mysql start. Karena DVWA memerlukan basis data untuk menyimpan data pengguna dan skenario serangan, layanan MySQL harus dalam kondisi aktif agar aplikasi dapat terhubung dan berfungsi dengan normal

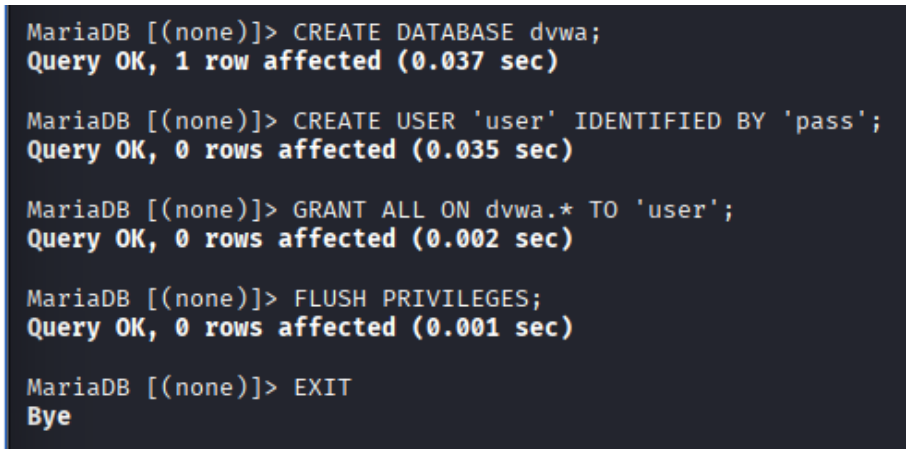
h. Verifikasi dan Akses Manajemen Basis Data



Login ke MariaDB Monitor

Gambar tersebut mendokumentasikan langkah masuk ke sistem manajemen basis data menggunakan perintah mysql -u root -p. Melalui perintah ini, pengguna masuk sebagai administrator (*root*) untuk melakukan konfigurasi langsung pada mesin MariaDB. Tampilan "Welcome to the MariaDB monitor" menunjukkan bahwa koneksi telah berhasil dilakukan, sehingga administrator dapat mulai membuat database baru, mengatur hak akses pengguna khusus untuk DVWA, atau melakukan pengecekan struktur tabel yang diperlukan sebelum aplikasi siap digunakan sepenuhnya

i. Konfigurasi Database Manual (SQL)

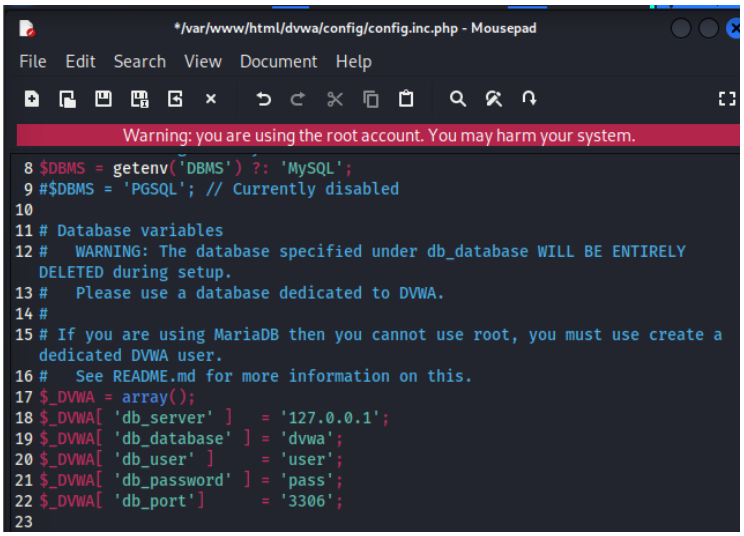


- Membuat Database (CREATE DATABASE dvwa;)
  Perintah ini membuat sebuah basis data kosong baru bernama **dvwa**. Ini adalah tempat di mana semua tabel, data pengguna, dan log aktivitas aplikasi akan disimpan nantinya.
- Membuat Pengguna Baru (CREATE USER 'user' IDENTIFIED BY 'pass';)

Demi alasan keamanan, aplikasi sebaiknya tidak menggunakan akun 'root' sistem. Perintah ini membuat identitas pengguna baru bernama **'user'** dengan kata sandi **'pass'** yang nantinya akan didaftarkan pada file config.inc.php.

- Memberikan Hak Akses (GRANT ALL ON dvwa.* TO 'user';)
  Perintah ini memberikan izin penuh kepada pengguna **'user'** untuk mengelola semua tabel di dalam database **dvwa**. Tanpa hak akses ini, aplikasi tidak akan bisa membuat tabel atau menyimpan data meskipun koneksi berhasil.
- Menyegarkan Izin & Keluar (FLUSH PRIVILEGES; lalu EXIT)
  FLUSH PRIVILEGES memerintahkan server untuk segera menerapkan semua perubahan hak akses yang baru saja dibuat. Setelah itu, perintah EXIT (atau Bye) digunakan untuk memutus koneksi terminal dari monitor MariaDB dan kembali ke prompt terminal biasa.

j. Sinkronisasi Kredensial Database pada Aplikasi



Pengeditan Berkas config.inc.php menggunakan Mousepad

Setelah membuat database dan pengguna melalui terminal, langkah selanjutnya adalah memasukkan informasi tersebut ke dalam file konfigurasi agar aplikasi DVWA dapat terhubung ke server database. Dalam gambar tersebut, file config.inc.php dibuka menggunakan editor teks

Mousepad dengan hak akses root. Fokus utama pengeditan ini terletak pada bagian **Database variables**, di mana nilai-nilai di dalam array $_DVWA disesuaikan dengan perintah SQL yang telah dijalankan sebelumnya. Penjelasannya sebagai berikut:

- **db_server**: Diatur ke 127.0.0.1 yang berarti server database berada di komputer yang sama (localhost).
- **db_database**: Diisi dengan dvwa, sesuai dengan nama database yang dibuat dengan perintah CREATE DATABASE.
- **db_user**: Diisi dengan user, sesuai dengan nama pengguna yang dibuat dengan perintah CREATE USER.
- **db_password**: Diisi dengan pass, sesuai dengan kata sandi yang telah ditetapkan sebelumnya.
- **db_port**: Diatur ke 3306, yang merupakan port standar untuk layanan MySQL/MariaDB.

k. konfigurasi utama PHP (versi 8.4)

```
┌──(root㉿hermi)-[/var/www/html/dvwa]
└─# mousepad /etc/php/8.4/apache2/php.ini
```

membuka **file konfigurasi utama PHP** (versi 8.4) yang digunakan oleh web server Apache menggunakan editor teks grafis bernama **Mousepad**. File php.ini ini adalah "otak" dari pengaturan bahasa pemrograman PHP di server Anda

```
861 ; https://php.net/allow-url-fopen
862 allow_url_fopen = On
```

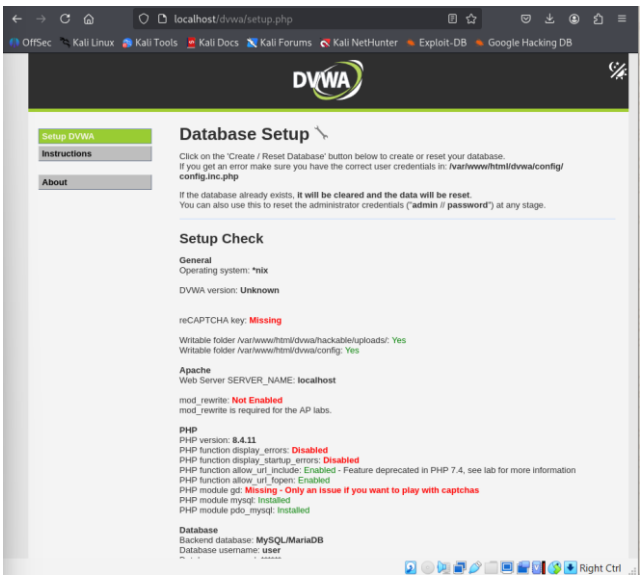
Diaktifkan agar simulasi serangan *Remote File Inclusion* (RFI) bisa berjalan.

```
865 ; https://php.net/allow-url-include
866 allow_url_include = on
867
```

Mengizinkan PHP mengambil data dari URL luar.

Eksekusi perintah ini dilakukan dengan hak akses **root** (terlihat dari tanda # dan teks merah pada gambar sebelumnya) karena file tersebut berada di direktori sistem /etc/ yang terproteksi. Tanpa melakukan penyesuaian pada file ini, beberapa modul serangan di DVWA akan berstatus "Disabled" atau tidak berfungsi saat Anda mengaksesnya lewat browser nanti.

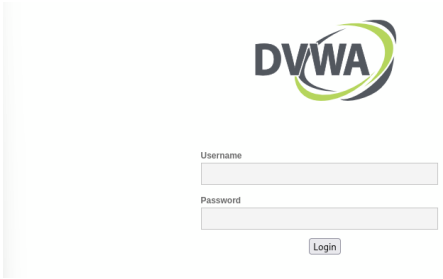
l. Halaman Setup DVWA



Tampilan Web Setup Check

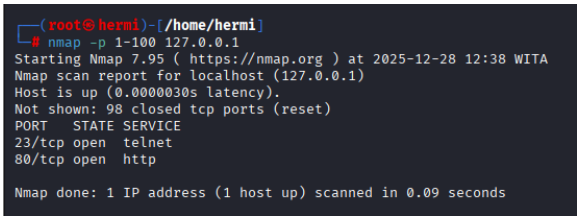
Halaman ini menunjukkan status kesiapan sistem. Terlihat folder konfigurasi sudah "Writable" (Hijau), allow_url_include sudah "Enabled", dan database sudah menggunakan user. Langkah terakhir tinggal menekan tombol "Create / Reset Database" di bagian bawah halaman tersebut

m. Halaman Login



Disini adalah halaman login untuk masuk dan username nya adalah admin sedangkan password nya adalah password

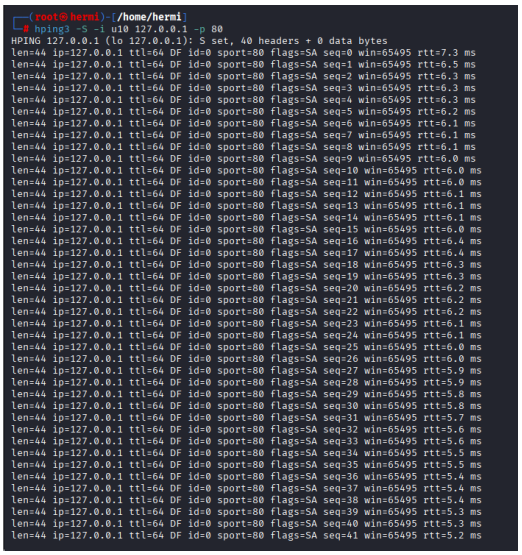
2. Proses Penyerangan
   a. Pemindaian port (scanning)



Proses network reconnaissance (pengintaian jaringan) menggunakan teknik *TCP Connect Scan* atau *SYN Scan* melalui *utility* Nmap. Proses ini bekerja dengan mengirimkan paket data ke alamat IP *loopback* 127.0.0.1 untuk menguji respons *three-way handshake* pada lapisan transport dalam model OSI. Berdasarkan output tersebut, sistem mengonfirmasi keberadaan *socket* yang aktif pada Port 23 (Telnet) dan Port 80 (HTTP), yang mengindikasikan bahwa *daemon* layanan terkait sedang berada dalam status *listening* dan siap menerima koneksi masuk.

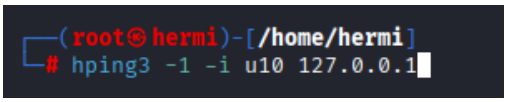
Eksistensi Port 80 yang berstatus *open* secara teknis merupakan jalur komunikasi protokol Hypertext Transfer Protocol yang digunakan untuk menyajikan antarmuka grafis aplikasi web, seperti halaman login DVWA yang Anda gunakan. Kehadiran port ini memungkinkan transfer data antara server web dan klien melalui mekanisme *request-response*. Sebaliknya, 98 port lainnya merespons dengan paket RST (Reset)

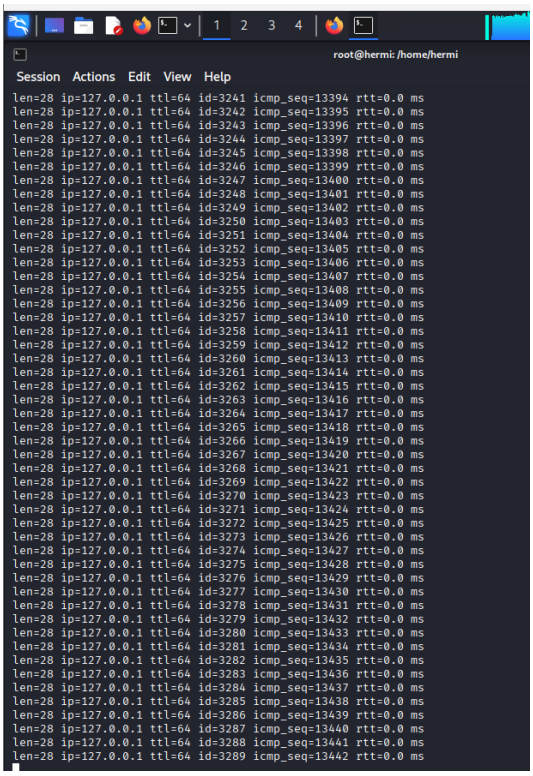
b. SYN_flood



Perintah yang dijalankan, yaitu hping3 -S -i u10 127.0.0.1 -p 80, menginstruksikan sistem untuk mengirimkan paket TCP SYN secara terus-menerus ke Port 80 (layanan HTTP) pada alamat lokal dengan interval sangat singkat, yaitu setiap 10 mikrodetik (u10). Penggunaan bendera -S (SYN) mensimulasikan tahap pertama dari proses *three-way handshake* TCP. Dalam output terminal, terlihat setiap baris melaporkan bendera flags=SA (SYN-ACK), yang secara ilmiah membuktikan bahwa Port 80 pada target memberikan respons aktif terhadap setiap permintaan koneksi yang masuk. Hal ini sinkron dengan hasil pemindaian Nmap sebelumnya yang menyatakan bahwa Port 80 berada dalam status *open*

c. Ping_flood





Pengujian dilakukan dengan mengeksekusi perintah hping3 -1 -i u10 127.0.0.1 melalui terminal dengan hak akses *root*. Penggunaan parameter -1 menetapkan mode protokol pada ICMP (Internet Control Message Protocol), yang secara fungsional serupa dengan perintah ping standar namun dengan kendali yang lebih presisi. Parameter -i u10 mengonstruksi pengiriman paket dengan interval yang sangat agresif, yakni setiap 10 mikrosekon, yang bertujuan untuk mensimulasikan trafik data berkepadatan tinggi terhadap alamat *loopback* (127.0.0.1).

Berdasarkan hasil pemantauan pada *output* terminal, sistem menunjukkan respon yang stabil terhadap beban trafik tersebut. Hal ini ditandai dengan nilai RTT (Round Trip Time) sebesar 0.0 ms, yang mengindikasikan bahwa pemrosesan paket terjadi hampir instan di dalam *internal stack* jaringan tanpa adanya latensi fisik. Meskipun nomor urut paket (icmp_seq) meningkat dengan sangat cepat hingga melampaui angka 13.000 dalam durasi singkat, ukuran paket tetap konsisten pada 28 byte. Aktivitas ini secara teknis dikategorikan sebagai simulasi ICMP Flood, yang umumnya digunakan dalam skenario uji beban (*stress testing*) untuk

mengevaluasi ketahanan sistem atau konfigurasi *firewall* dalam menangani potensi serangan Denial of Service (DoS

d. Serangan Smurf

Perintah hping3 -a 127.0.0.1 127.255.255.255 -1 --fast yang dieksekusi, pengujian ini melakukan simulasi serangan berbasis IP Spoofing dan Broadcast Amplification. Dengan menggunakan parameter -a, alamat IP asal dipalsukan menjadi 127.0.0.1, sementara paket diarahkan ke alamat *broadcast* 127.255.255.255 menggunakan protokol ICMP (mode -1). Penggunaan opsi --fast memastikan paket dikirimkan secara beruntun dengan kecepatan 10 paket per detik. Output sistem mengonfirmasi bahwa mode ICMP telah aktif dengan ukuran *header* 28 byte yang dikirimkan melalui *interface* loopback (lo). Secara teknis, skenario ini serupa dengan mekanisme Smurf Attack, di mana tujuannya adalah memicu respons masif dari seluruh perangkat dalam jaringan broadcast untuk membanjiri target yang alamat IP-nya telah dipalsukan