SIMULASI SERANGAN DOS (HPING3 & SLOWLORIS)

NAMA : **IRYAN TEGAR**

NIM : **105841113023**

MATA KULIAH : **ADVANCED NETWORK SECURITY AND PROTOCOL**

KELAS : **5 JK-A**

---

1. PENDAHULUAN

Perkembangan layanan berbasis jaringan yang semakin masif menjadikan ketersediaan (*availability*) sebagai salah satu aspek utama dalam keamanan informasi. Salah satu ancaman serius terhadap ketersediaan layanan adalah serangan Denial of Service (DoS), yang bertujuan untuk melumpuhkan layanan dengan membanjiri atau menghabiskan sumber daya sistem. Praktikum ini dilakukan untuk memahami cara kerja serangan DoS pada berbagai layer jaringan, khususnya SYN Flood menggunakan Hping3 (Network Layer) dan Slowloris (Application Layer), serta menganalisis dampaknya terhadap layanan web. Selain itu, praktikum ini juga bertujuan untuk menguji efektivitas mitigasi menggunakan firewall IPTables dalam memulihkan layanan dan membedakan akses antara attacker dan target setelah mitigasi diterapkan.

2. TUJUAN PRAKTIKUM

Praktikum ini bertujuan untuk menganalisis dampak serangan Denial of Service (DoS) terhadap ketersediaan layanan web menggunakan dua metode berbeda, yaitu SYN Flood dan Slowloris, serta menguji efektivitas mitigasi firewall dalam menghentikan serangan dan memulihkan layanan. Hasil pengujian diharapkan dapat memberikan pemahaman praktis mengenai ancaman DoS dan penerapan mekanisme pertahanan dasar pada sistem jaringan.

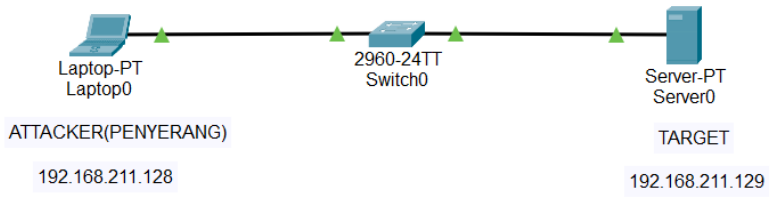
3. TOOLS YANG DIGUNAKAN

No	Tools / Software	Versi / OS	Fungsi
1	Kali Linux	Rolling / Latest	Sistem operasi attacker untuk menjalankan serangan DoS
2	Ubuntu Server	20.04 / 22.04	Server target (korban)
3	Apache2	Default Ubuntu Repo	Web server untuk menjalankan DVWA
4	MariaDB	Default Ubuntu Repo	Database server DVWA
5	PHP	7.x / 8.x	Backend aplikasi DVWA
6	DVWA	Latest GitHub	Aplikasi web target yang rentan
7	Hping3	Built-in Kali	Melakukan serangan SYN Flood
8	Slowloris	Built-in Kali	Melakukan serangan DoS layer aplikasi
9	IPTables	Linux Kernel	Mitigasi dan filtering trafik DoS
10	VMware Workstation	—	Virtualisasi lingkungan simulasi
11	Browser Firefox	Default	Pengujian akses layanan web

4. LATAR BELAKANG DAN SKENARIO PENGUJIAN

Praktikum ini bertujuan untuk membangun sebuah server target yang rentan, lalu melakukan uji coba serangan *Denial of Service* (DoS) untuk melihat dampaknya terhadap ketersediaan layanan web.

a. Topologi: Menggunakan mode NAT pada VMware agar Attacker dan Target berada dalam satu jaringan lokal terisolasi.



Gambar 1.1 Topologi Jaringan Simulasi DoS

Gambar di atas mengilustrasikan arsitektur jaringan yang digunakan dalam simulasi. Terdiri dari mesin Attacker (Kali Linux dengan IP 192.168.211.128) dan mesin Target (Ubuntu Server dengan IP 192.168.211.129) yang terhubung melalui Switch dalam satu segmen jaringan lokal (LAN) yang sama.

b. **Attacker:** Kali Linux (IP: 192.168.211.128)

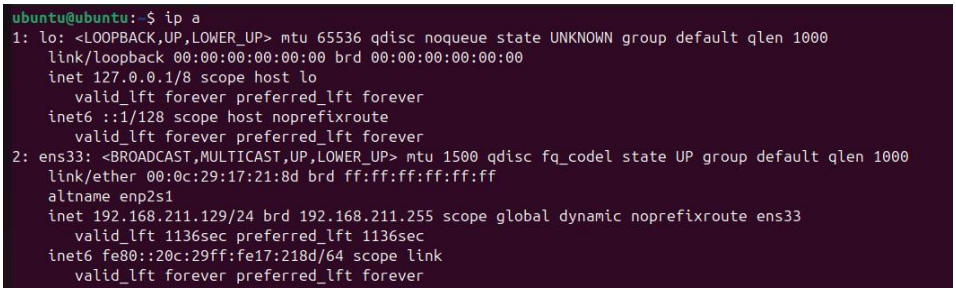
```
┌──(kali㉿iryantegar)-[~]
└─$ sudo su
[sudo] password for kali:
┌──(root㉿iryantegar)-[/home/kali]
└─# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:a4:39:e3 brd ff:ff:ff:ff:ff:ff
    inet 192.168.211.128/24 brd 192.168.211.255 scope global dynamic noprefixroute eth0
       valid_lft 1109sec preferred_lft 1109sec
    inet6 fe80::f239:dbd0:f5ef:97e0/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Gambar 2.1 Verifikasi IP Address pada Mesin Attacker

Gambar di atas menampilkan hasil eksekusi perintah ip a pada sistem operasi Kali Linux yang berperan sebagai mesin attacker. Perintah ini digunakan untuk memverifikasi konfigurasi antarmuka jaringan yang aktif sebelum pelaksanaan simulasi serangan. Berdasarkan output yang ditampilkan, antarmuka jaringan eth0 berada dalam kondisi aktif (*UP*) dan telah memperoleh alamat IP **192.168.211.128/24**, yang sesuai

dengan skema topologi jaringan yang dirancang. Keberhasilan verifikasi ini memastikan bahwa mesin attacker berada dalam satu segmen jaringan lokal dengan target, sehingga proses komunikasi dan pengujian serangan Denial of Service (DoS) dapat dilakukan tanpa kendala konektivitas jaringan.

- Target: Ubuntu Server (IP: 192.168.211.129)



Gambar 1.3 Verifikasi Alamat IP pada Mesin Target (Korban)

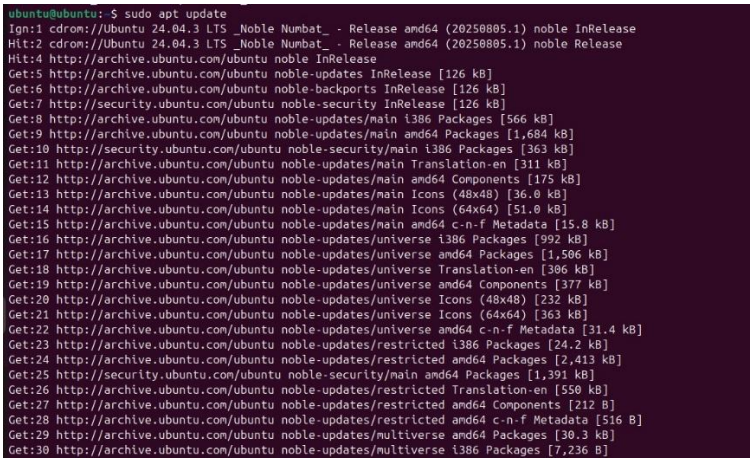
Gambar ini menunjukkan hasil pemeriksaan konfigurasi jaringan pada Ubuntu Server menggunakan perintah ip a. Dari informasi yang ditampilkan, antarmuka jaringan **ens33** terkonfirmasi dalam kondisi aktif dan memiliki alamat IP **192.168.211.129/24**. Konfigurasi tersebut menegaskan bahwa mesin target berada dalam satu segmen jaringan lokal (subnet) yang sama dengan mesin attacker, sehingga komunikasi jaringan dapat berlangsung secara langsung. Kondisi ini sangat penting untuk mendukung proses simulasi serangan Denial of Service (DoS), karena memastikan seluruh paket serangan dari attacker dapat diterima oleh server target tanpa melalui mekanisme routing tambahan.

5. PEMBANGUNAN TARGET (SERVER SETUP)

Langkah pertama adalah menyiapkan server Ubuntu menjadi Web Server fungsional dengan database, lalu menginstal aplikasi target DVWA.

a. Instalasi LAMP Stack & Git Mengupdate repository dan menginstal paket Apache, MariaDB, PHP, dan Git.

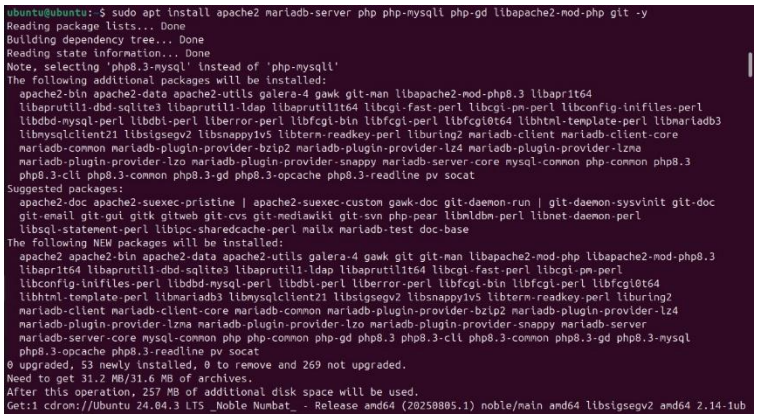
- sudo apt update



Gambar 2.1 Pembaruan Repository Sistem (System Update)

Pada terminal Ubuntu Server saat menjalankan perintah sudo apt update. Langkah ini dilakukan untuk menyegarkan daftar repositori lokal dan memastikan seluruh paket dependensi yang akan diinstal (seperti Apache dan MariaDB) berada dalam versi terbaru dan kompatibel.

- sudo apt install apache2 mariadb-server php php-mysqli php-gd libapache2-mod-php git -y



Gambar 2.2 Instalasi Dependensi Utama (LAMP Stack)

Proses instalasi paket-paket esensial untuk membangun *web server*. Perintah sudo apt install digunakan untuk memasang Apache2 (Web Server), MariaDB-Server (Database), PHP beserta modul pendukungnya, dan Git. Paket-paket ini merupakan fondasi agar aplikasi DVWA dapat berjalan dengan baik di atas server Ubuntu.

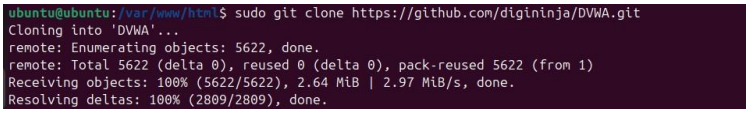
b. Instalasi DVWA Mengunduh *source code* DVWA dari GitHub ke direktori web server.

- cd /var/www/html

Gambar 2.3 Navigasi ke Direktori Root Web Server

Tangkapan layar terminal menunjukkan penggunaan perintah cd /var/www/html. Langkah ini bertujuan untuk memindahkan posisi direktori aktif (*current directory*) menuju folder *Document Root* milik Apache. Hal ini sangat penting dilakukan sebelum mengunduh aplikasi DVWA, agar file aplikasi langsung tersimpan di lokasi yang dapat diakses oleh *web server*.

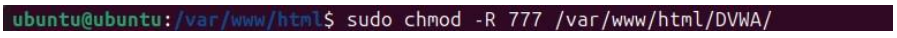
- sudo git clone https://github.com/digininja/DVWA.git\



Gambar 2.4 Proses Pengunduhan Aplikasi DVWA dari Repository GitHub

Gambar di atas menunjukkan proses pengunduhan source code Damn Vulnerable Web Application (DVWA) menggunakan perintah git clone pada sistem Ubuntu Server. Repository DVWA diambil langsung dari GitHub dan disimpan ke dalam direktori /var/www/html, yang merupakan document root Apache. Output terminal menampilkan proses cloning berjalan sukses, ditandai dengan pesan “Receiving objects: 100%” dan “done”. Keberhasilan tahap ini memastikan bahwa file aplikasi DVWA telah tersedia di server dan siap untuk dikonfigurasi serta dijalankan sebagai target pengujian keamanan.

- sudo chmod -R 777 /var/www/html/DVWA/



Gambar 2.5 Pengaturan Hak Akses Direktori DVWA

Gambar di atas memperlihatkan proses pengaturan hak akses direktori DVWA menggunakan perintah chmod -R 777 pada folder /var/www/html/DVWA. Pemberian

izin read, write, dan execute untuk seluruh user dilakukan agar web server Apache dapat mengakses, menulis, dan mengeksekusi seluruh file aplikasi tanpa kendala izin. Konfigurasi ini sengaja diterapkan pada lingkungan simulasi dan pembelajaran untuk memastikan DVWA dapat berjalan dengan lancar, meskipun secara praktik keamanan nyata pengaturan ini tergolong tidak aman.

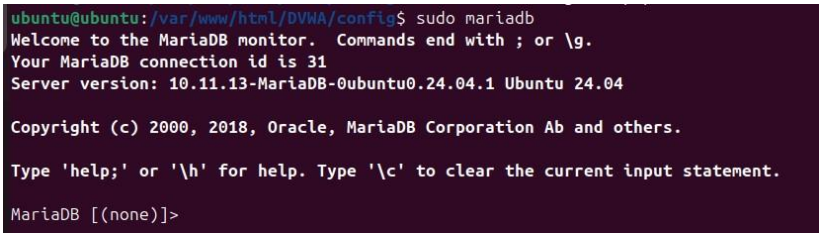
- ls /var/www/html

Gambar 2.6 Verifikasi Keberadaan Direktori DVWA pada Web Server

Gambar di atas menampilkan hasil eksekusi perintah ls /var/www/html pada Ubuntu Server untuk memverifikasi isi direktori document root Apache. Output menunjukkan bahwa folder DVWA telah berhasil dibuat dan berada pada lokasi yang benar, berdampingan dengan file index.html bawaan Apache. Hal ini membuktikan bahwa proses pengunduhan dan penempatan aplikasi DVWA telah berjalan dengan sukses, sehingga aplikasi siap diakses melalui browser dan dilanjutkan ke tahap konfigurasi serta pengujian keamanan.

c. Konfigurasi Database Membuat database khusus untuk DVWA agar aplikasi bisa berjalan.

- sudo MariaDB



Gambar 2.7 Akses MariaDB untuk Konfigurasi Database DVWA

Gambar di atas menunjukkan proses akses ke MariaDB Monitor menggunakan perintah sudo mariadb pada Ubuntu Server. Tampilan welcome message menandakan bahwa layanan MariaDB telah berjalan dengan normal dan pengguna berhasil masuk ke mode manajemen database dengan hak administratif. Tahap ini merupakan langkah awal dalam pembuatan database, user, serta pengaturan hak akses yang dibutuhkan agar aplikasi DVWA dapat terhubung dan beroperasi dengan database secara optimal.

- CREATE DATABASE dvwa;

```
MariaDB [(none)]> CREATE DATABASE dvwa;
Query OK, 1 row affected (0.001 sec)
```

Gambar 2.9 Pembuatan Database DVWA pada MariaDB

Gambar di atas memperlihatkan proses pembuatan database baru bernama **dvwa** menggunakan perintah CREATE DATABASE dvwa; pada MariaDB. Pesan "Query OK, 1 row affected" menandakan bahwa database berhasil dibuat tanpa error. Database ini akan digunakan sebagai media penyimpanan data aplikasi DVWA, termasuk konfigurasi dan informasi pengguna, sehingga memungkinkan aplikasi web berjalan secara dinamis selama proses pengujian keamanan berlangsung.

- CREATE USER 'bxryyan'@'localhost' IDENTIFIED BY '12345';

```
MariaDB [(none)]> CREATE USER 'bxryyan'@'localhost' IDENTIFIED BY '12345';
Query OK, 0 rows affected (0.001 sec)
```

Gambar 2.10 Pembuatan User Database untuk Aplikasi DVWA

Gambar di atas menunjukkan proses pembuatan user database baru pada MariaDB menggunakan perintahCREATE USER 'bxryyan'@'localhost' IDENTIFIED BY '12345';. User ini dibuat khusus untuk mengelola akses aplikasi DVWA ke database, sehingga tidak menggunakan akun root secara langsung. Konfigurasi ini bertujuan agar koneksi database DVWA dapat berjalan sesuai dengan prinsip pemisahan hak akses, meskipun pada lingkungan simulasi masih menggunakan kredensial sederhana untuk memudahkan proses praktikum.

- GRANT ALL PRIVILEGES ON dvwa.* TO 'user_dvwa'@'localhost';

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON dvwa.* TO 'bxryyan'@'localhost';
Query OK, 0 rows affected (0.001 sec)
```

Gambar 2.11 Pemberian Hak Akses User ke Database DVWA

Gambar di atas menampilkan perintah GRANT ALL PRIVILEGES ON dvwa.* TO 'user_dvwa'@'localhost'; yang digunakan untuk memberikan seluruh hak akses database dvwa kepada user yang telah dibuat sebelumnya. Hak akses ini mencakup kemampuan membaca, menulis, dan memodifikasi tabel di dalam database. Pengaturan ini diperlukan agar aplikasi DVWA dapat berinteraksi penuh dengan database selama proses pengujian, meskipun secara praktik keamanan nyata pemberian hak akses penuh tidak direkomendasikan.

- FLUSH PRIVILEGES;

```
MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 sec)
```

Gambar 2.12 Penerapan Perubahan Hak Akses Database (Flush Privileges)

Gambar di atas menunjukkan penggunaan perintah FLUSH PRIVILEGES; pada MariaDB untuk menerapkan seluruh perubahan hak akses yang telah dikonfigurasi sebelumnya. Perintah ini berfungsi untuk memuat ulang tabel hak akses sehingga user database yang baru dibuat dapat langsung digunakan tanpa perlu me-restart layanan database. Tahap ini menandai bahwa konfigurasi database DVWA telah selesai dan siap digunakan oleh aplikasi web.

- EXIT;

```
MariaDB [(none)]> EXIT;
Bye
ubuntu@ubuntu:/var/www/html/DVWA/config$
```

Gambar 2.13 Keluar dari MariaDB Setelah Konfigurasi Database DVWA

Gambar di atas menampilkan penggunaan perintah EXIT; untuk keluar dari MariaDB Monitor setelah seluruh proses pembuatan database, user, serta pengaturan hak akses selesai dilakukan. Langkah ini menandakan bahwa konfigurasi database untuk aplikasi DVWA telah berhasil diselesaikan, sehingga sistem siap dilanjutkan ke tahap pengaturan file konfigurasi aplikasi dan pengujian akses melalui web browser.

d. Konfigurasi Koneksi Menyalin file konfigurasi dan menyesuaikannya dengan database yang baru dibuat.

- cd /var/www/html/DVWA/config

```
ubuntu@ubuntu:/var/www/html$ cd DVWA/config
```

Gam*bar 2.14 Navigasi ke Direktori Konfigurasi Aplikasi DVWA*

Gambar di atas menunjukkan proses perpindahan direktori ke folder /var/www/html/DVWA/config menggunakan perintah cd. Direktori ini berisi file konfigurasi utama aplikasi DVWA, termasuk pengaturan koneksi database. Akses ke folder ini diperlukan sebelum melakukan penyalinan dan pengeditan file config.inc.php, agar aplikasi dapat terhubung dengan database yang telah dikonfigurasi sebelumnya.

- sudo cp config.inc.php.dist config.inc.php

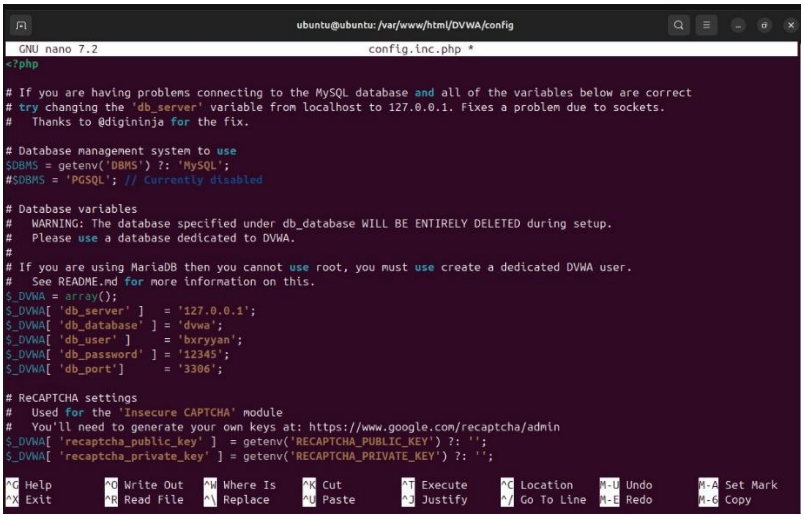
Gambar 2.15 Penyalinan File Konfigurasi Default DVWA

Gambar di atas memperlihatkan proses penyalinan file konfigurasi default **DVWA** menggunakan perintah sudo cp config.inc.php.dist config.inc.php. Langkah ini dilakukan untuk membuat file konfigurasi aktif (config.inc.php) dari template bawaan DVWA. File inilah yang nantinya akan diedit untuk menyesuaikan pengaturan koneksi database, sehingga aplikasi DVWA dapat terhubung dengan database MariaDB yang telah disiapkan sebelumnya.

- sudo nano config.inc.php

Gambar 2.15 Proses Pengeditan File Konfigurasi config.inc.php pada DVWA

Gambar di atas menampilkan proses pengeditan file konfigurasi utama DVWA, yaitu config.inc.php, menggunakan text editor nano dengan perintah sudo nano config.inc.php. Pada file ini dilakukan penyesuaian parameter koneksi database, meliputi nama database, username, password, dan host, agar sesuai dengan database MariaDB yang telah dibuat. Konfigurasi ini menjadi tahap krusial karena menentukan keberhasilan aplikasi DVWA dalam mengakses database dan berjalan secara normal pada web server.

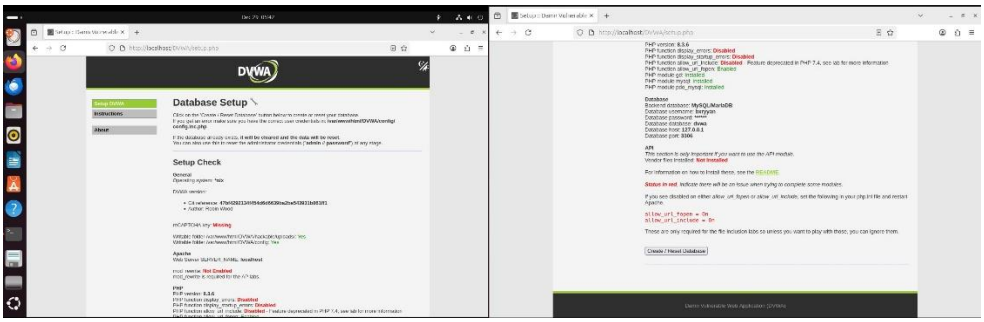


Gambar 2.16 Konfigurasi Koneksi Database pada File config.inc.php DVWA

Gambar di atas menampilkan isi file konfigurasi config.inc.php pada aplikasi DVWA setelah dilakukan pengeditan. Parameter koneksi database disesuaikan dengan database yang telah dibuat sebelumnya, meliputi alamat server (127.0.0.1), nama database (dvwa), username (bryan), password, serta port yang digunakan. Pengaturan ini memungkinkan aplikasi DVWA terhubung secara langsung ke MariaDB, sehingga proses instalasi dan pengujian keamanan dapat berjalan dengan normal pada web server Ubuntu.

e. Verifikasi Awal Konfigurasi DVWA

- Verifikasi dan Inisialisasi Aplikasi DVWA

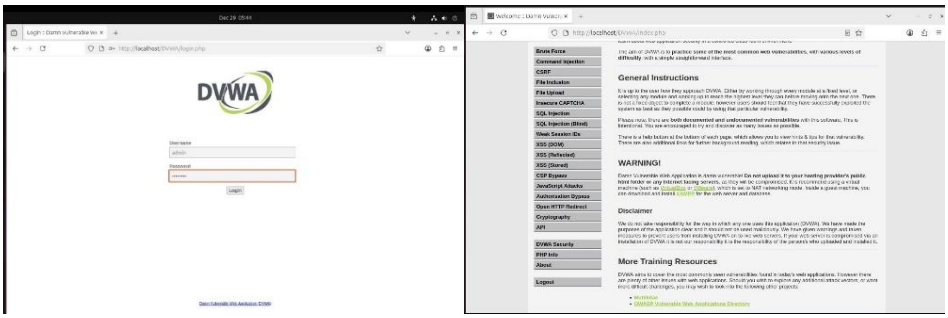


Gambar 2.17 Halaman Status Konfigurasi Awal DVWA

Gambar di atas menampilkan halaman Database Setup pada aplikasi Damn Vulnerable Web Application (DVWA) yang diakses melalui web browser pada Ubuntu Server. Halaman ini digunakan untuk memverifikasi kesiapan konfigurasi server, termasuk koneksi database, modul PHP, serta pengaturan sistem yang dibutuhkan agar DVWA dapat berjalan dengan baik.

Pada tampilan tersebut terlihat status konfigurasi PHP dan database MySQL/MariaDB, serta tombol Create / Reset Database yang digunakan untuk melakukan inisialisasi database DVWA. Proses ini akan membuat tabel-tabel yang diperlukan sesuai dengan konfigurasi pada file config.inc.php. Keberhasilan tahap ini menandakan bahwa web server Apache, database MariaDB, dan aplikasi DVWA telah terintegrasi dengan benar, sehingga server siap digunakan sebagai target simulasi serangan Denial of Service (DoS) pada tahap pengujian selanjutny

- Tampilan Halaman Login Damn Vulnerable Web Application (DVWA)



Gambar 2.18 Tampilan Halaman Login Damn Vulnerable Web Application (DVWA)

Gambar di atas menunjukkan halaman login aplikasi Damn Vulnerable Web Application (DVWA) yang diakses melalui web browser pada Ubuntu Server menggunakan alamat http://localhost/DVWA/login.php. Halaman ini menampilkan form autentikasi berupa kolom username dan password yang digunakan untuk mengakses sistem DVWA. Keberhasilan pemuatan halaman login ini membuktikan bahwa web server Apache dan aplikasi DVWA telah berjalan dengan baik, sehingga server target siap digunakan sebagai media simulasi dan pengujian serangan Denial of Service (DoS).

6. PENGONDISIAN KEAMANAN (VULNERABILITY SETUP)

Agar simulasi serangan DoS dapat dianalisis dampaknya tanpa diblokir oleh pertahanan otomatis Ubuntu, dilakukan pelemahan sistem keamanan secara sengaja.

a. Mematikan Firewall Mematikan UFW agar port 80 terbuka lebar tanpa filter.

- sudo ufw disable

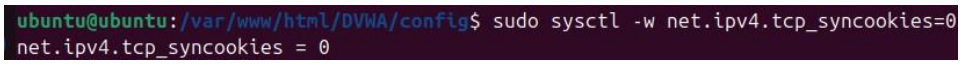
```
ubuntu@ubuntu:/var/www/html/DVWA/config$ sudo ufw disable
Firewall stopped and disabled on system startup
```

Gambar 3.1 Penonaktifan Firewall UFW pada Ubuntu Server

Gambar di atas menunjukkan penggunaan perintah sudo ufw disable untuk menonaktifkan Uncomplicated Firewall (UFW) pada Ubuntu Server. Penonaktifan firewall ini dilakukan agar seluruh port layanan, khususnya port 80 (HTTP), dapat diakses tanpa adanya pemfilteran paket. Langkah ini sengaja diterapkan pada lingkungan simulasi untuk memastikan serangan Denial of Service (DoS) dapat berjalan dan dampaknya terhadap ketersediaan layanan web dapat diamati secara jelas.

b. Mematikan Anti-DoS (SYN Cookies) Menonaktifkan fitur kernel yang berfungsi menolak paket banjir (SYN Flood). Jika ini aktif, serangan hping3 tidak akan efektif.

- sudo sysctl -w net.ipv4.tcp_syncookies=0

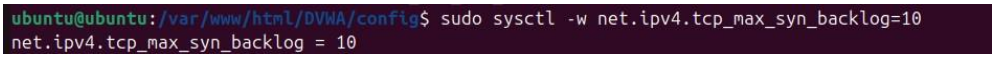


Gambar 3.2 Penonaktifan Fitur TCP SYN Cookies pada Kernel ubuntu

Gambar di atas menampilkan eksekusi perintah sudo sysctl -w net.ipv4.tcp_syncookies=0 pada Ubuntu Server. Perintah ini digunakan untuk menonaktifkan mekanisme **TCP SYN Cookies**, yaitu fitur keamanan pada kernel Linux yang berfungsi melindungi sistem dari serangan **SYN Flood**. Penonaktifan fitur ini dilakukan secara sengaja pada lingkungan simulasi agar server menjadi lebih rentan, sehingga dampak serangan **DoS menggunakan hping3** dapat diamati secara nyata terhadap kinerja dan ketersediaan layanan web.

c. Konfigurasi Kernel untuk Simulasi Kerentanan Serangan DoS

- sudo sysctl -w net.ipv4.tcp_max_syn_backlog=10



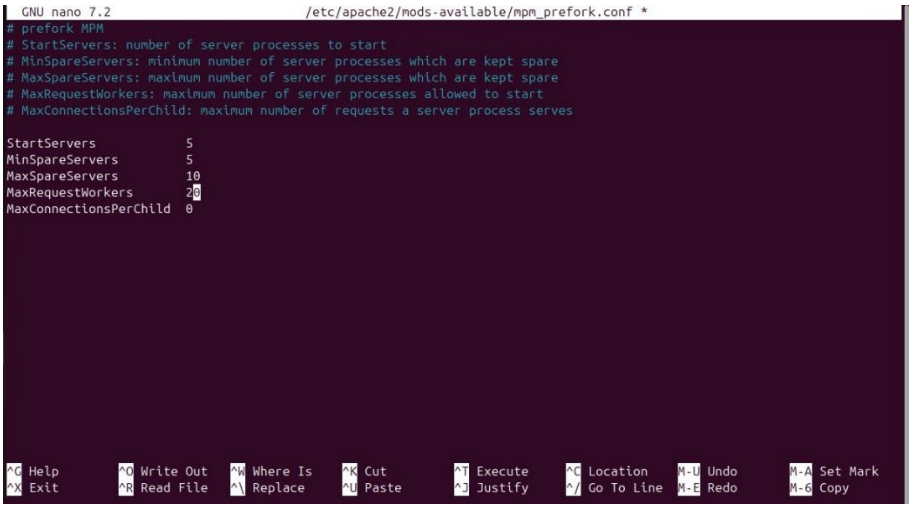
Gambar 3.3 Penyesuaian Nilai TCP SYN Backlog untuk Meningkatkan Kerentanan DoS

Gambar di atas menunjukkan proses pengaturan parameter kernel Linux menggunakan perintah sudo sysctl -w net.ipv4.tcp_max_syn_backlog=10. Parameter tcp_max_syn_backlog mengatur jumlah maksimum antrean koneksi TCP yang belum selesai (half-open connection). Dengan menurunkan nilai backlog menjadi sangat kecil, server menjadi lebih rentan terhadap serangan SYN Flood, karena antrean koneksi cepat penuh saat menerima trafik dalam jumlah besar. Pengaturan ini dilakukan secara sengaja pada lingkungan simulasi untuk memperjelas dampak serangan DoS menggunakan hping3 terhadap ketersediaan layanan web.

d. Konfigurasi Apache untuk Simulasi Kerentanan Serangan Slowloris

- sudo nano /etc/apache2/mods-available/mpm_prefork.conf

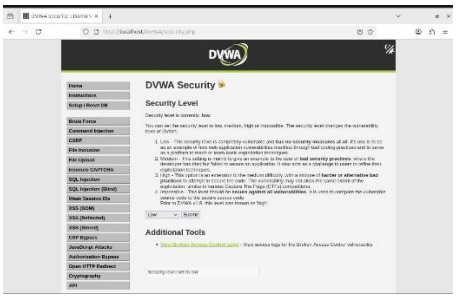
Gambar 3.4 Pengeditan Konfigurasi Apache MPM Prefork



Gambar 3.5 Penyesuaian Parameter MPM Prefork Apache untuk Simulasi Slowloris

Gambar di atas menampilkan isi file konfigurasi mpm_prefork.conf setelah dilakukan penyesuaian parameter. Nilai MaxRequestWorkers dibatasi menjadi 20, sehingga jumlah koneksi simultan yang dapat ditangani Apache menjadi sangat terbatas. Dengan konfigurasi ini, serangan Slowloris yang membuka banyak koneksi dan menahannya dalam kondisi half-open dapat dengan cepat menghabiskan slot koneksi Apache, menyebabkan layanan web menjadi tidak responsif meskipun server masih aktif secara sistem.

e. Pengeturan security level DVWA

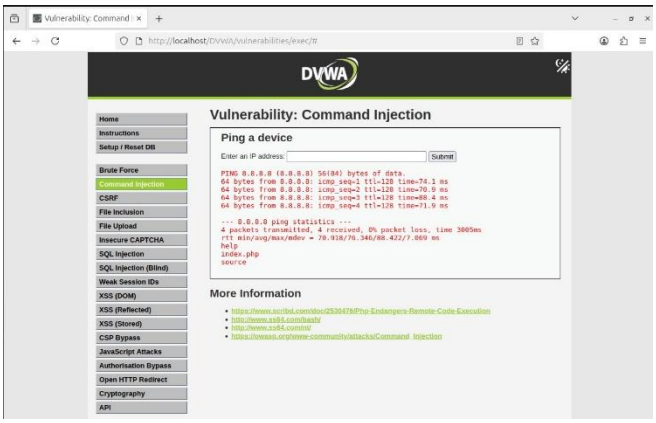


Gambar 3.6 Tampilan Pengaturan Security Level pada DVWA

Gambar di atas menampilkan halaman DVWA Security yang digunakan untuk mengatur tingkat keamanan aplikasi Damn Vulnerable Web Application (DVWA). Pada halaman ini, tingkat keamanan dapat diatur ke level Low, Medium, High, atau Impossible, yang menentukan seberapa besar mekanisme perlindungan yang diterapkan pada aplikasi. Dalam simulasi ini, pengaturan keamanan disesuaikan untuk menciptakan kondisi server

yang lebih rentan, sehingga memudahkan pengamatan dampak serangan Denial of Service (DoS) terhadap ketersediaan layanan web pada tahap pengujian selanjutnya.

f. Pengujian Kerentanan Command Injection pada DVWA



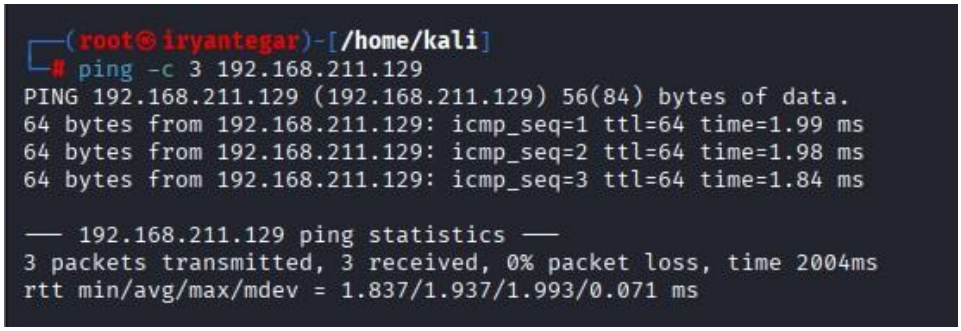
Gambar 3.2 Tampilan Modul Vulnerability: Command Injection pada DVWA

Gambar di atas menunjukkan halaman Vulnerability: Command Injection pada aplikasi Damn Vulnerable Web Application (DVWA), di mana pengguna dapat memasukkan alamat IP untuk diuji menggunakan perintah ping. Output yang ditampilkan pada halaman tersebut memperlihatkan hasil eksekusi perintah sistem secara langsung oleh server, yang mengindikasikan adanya celah keamanan pada mekanisme input validation. Modul ini digunakan untuk mensimulasikan kondisi aplikasi web yang rentan, sehingga membantu memahami bagaimana kelemahan konfigurasi keamanan dapat dimanfaatkan oleh penyerang dalam skenario pengujian keamanan, termasuk sebagai bagian dari analisis kesiapan server sebelum dilakukan simulasi serangan Denial of Service (DoS).

7. UJI KONEKSI AWAL

Sebelum serangan dimulai, dilakukan pengecekan untuk memastikan target bisa diakses dalam keadaan normal.

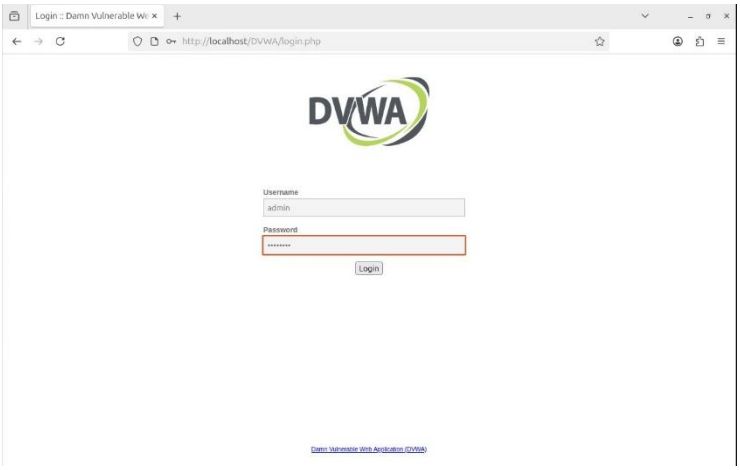
1. Ping Test: Dilakukan dari Kali Linux ke IP Ubuntu.



Gambar 4.1 Hasil Pengujian Konektivitas Jaringan Menggunakan Perintah Ping

Gambar di atas menampilkan hasil pengujian konektivitas jaringan menggunakan perintah ping dari mesin Attacker (Kali Linux) ke mesin Target (Ubuntu Server) dengan alamat IP 192.168.211.129. Hasil pengujian menunjukkan bahwa seluruh paket ICMP yang dikirim berhasil diterima kembali tanpa adanya packet loss (0%), dengan waktu respon rata-rata sekitar 1,9 ms. Hal ini membuktikan bahwa koneksi jaringan antara attacker dan target berada dalam kondisi normal dan stabil, sehingga simulasi serangan Denial of Service (DoS) dapat dilanjutkan pada tahap berikutnya

2. Akses Web: Membuka browser dan mengakses alamat DVWA.



Gambar 4.2 Pengujian Akses Web ke Aplikasi DVWA

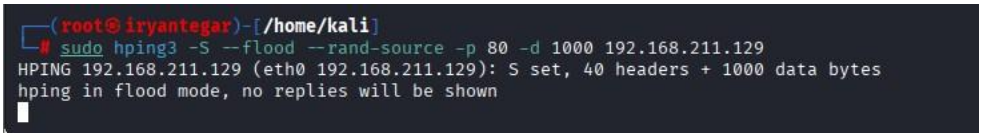
Gambar di atas menunjukkan proses pengujian akses web dengan membuka browser dan mengakses aplikasi Damn Vulnerable Web Application (DVWA) melalui alamat IP server target. Halaman DVWA dapat dimuat dengan normal tanpa kendala koneksi atau waktu tunggu yang berlebihan, yang menandakan bahwa layanan web Apache berjalan dengan baik sebelum dilakukan serangan. Kondisi ini menjadi acuan awal (baseline) untuk membandingkan dampak yang terjadi pada layanan web setelah simulasi serangan Denial of Service (DoS) dijalankan.

8. EKSEKUSI SERANGAN 1: HPING3 (NETWORK LAYER)

Serangan ini membanjiri target dengan paket TCP SYN dalam jumlah masif untuk menghabiskan bandwidth dan resource CPU.

a. Perintah Serangan (di Kali Linux)

sudo hping3 -S --flood --rand-source -p 80 -d 1000 192.168.211.129

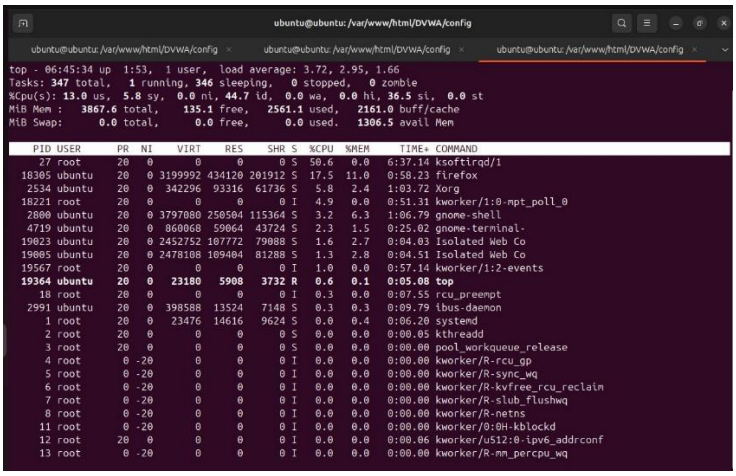


Gambar 5.1 Eksekusi Serangan SYN Flood Menggunakan Hping3 dengan Random Source

Gambar di atas menampilkan eksekusi serangan Denial of Service (DoS) jenis SYN Flood menggunakan tool Hping3 pada sistem operasi Kali Linux dengan perintah sudo hping3 -S --flood --rand-source -p 80 -d 1000 192.168.211.129. Perintah ini mengirimkan paket TCP SYN secara terus-menerus ke port 80 server target dengan alamat IP sumber acak dan ukuran payload 1000 byte, sehingga mempercepat konsumsi sumber daya jaringan dan antrean koneksi server. Serangan ini bertujuan untuk membanjiri server dengan permintaan koneksi palsu, yang berdampak pada menurunnya ketersediaan layanan web bagi pengguna normal.

b. Monitoring Dampak (di Ubuntu)

Top

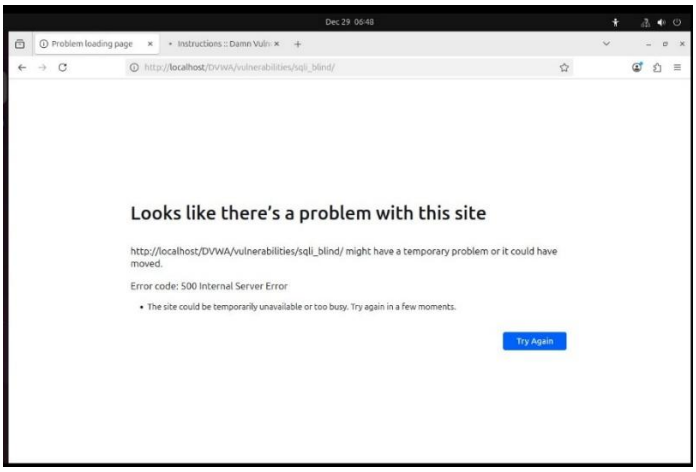


Gambar 5.2 Monitoring Penggunaan Sumber Daya Server saat Serangan SYN Flood

Gambar di atas menampilkan hasil pemantauan sumber daya sistem pada Ubuntu Server menggunakan perintah top saat serangan SYN Flood dari Hping3 sedang berlangsung. Terlihat adanya peningkatan beban sistem yang signifikan, ditunjukkan oleh nilai load average yang meningkat serta tingginya penggunaan CPU oleh proses kernel seperti ksoftirqd, yang menangani interrupt jaringan. Kondisi ini mengindikasikan bahwa server menerima lonjakan trafik jaringan yang tidak normal, sehingga sumber daya CPU dan memori digunakan secara intensif untuk memproses paket masuk, yang berdampak pada menurunnya performa dan responsivitas layanan web.

c. Dampak Layanan Browser menjadi tidak responsif dan mengalami *Time Out*.

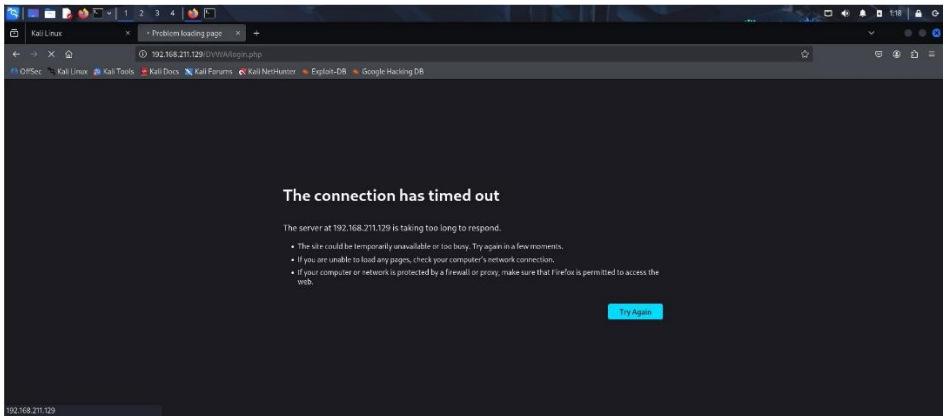
- Browser ubuntu (target)



Gambar 5.3 Tampilan Aplikasi DVWA Tidak Responsif pada Browser Ubuntu (Target)

Gambar di atas menunjukkan kondisi layanan web DVWA yang diakses melalui browser pada mesin Ubuntu Server (target) saat serangan SYN Flood menggunakan Hping3 sedang berlangsung. Halaman DVWA terlihat mengalami waktu muat yang sangat lama (loading) dan berujung pada kegagalan akses (time out), yang menandakan bahwa server tidak mampu merespons permintaan HTTP secara normal. Kondisi ini disebabkan oleh sumber daya server yang telah tersita untuk menangani banjir paket SYN, sehingga layanan web menjadi tidak responsif dan tidak dapat diakses oleh pengguna sah selama serangan berlangsung.

- Browser kali linux (attacker)



Gambar 5.4 Kegagalan Akses DVWA pada Browser Kali Linux akibat Time Out

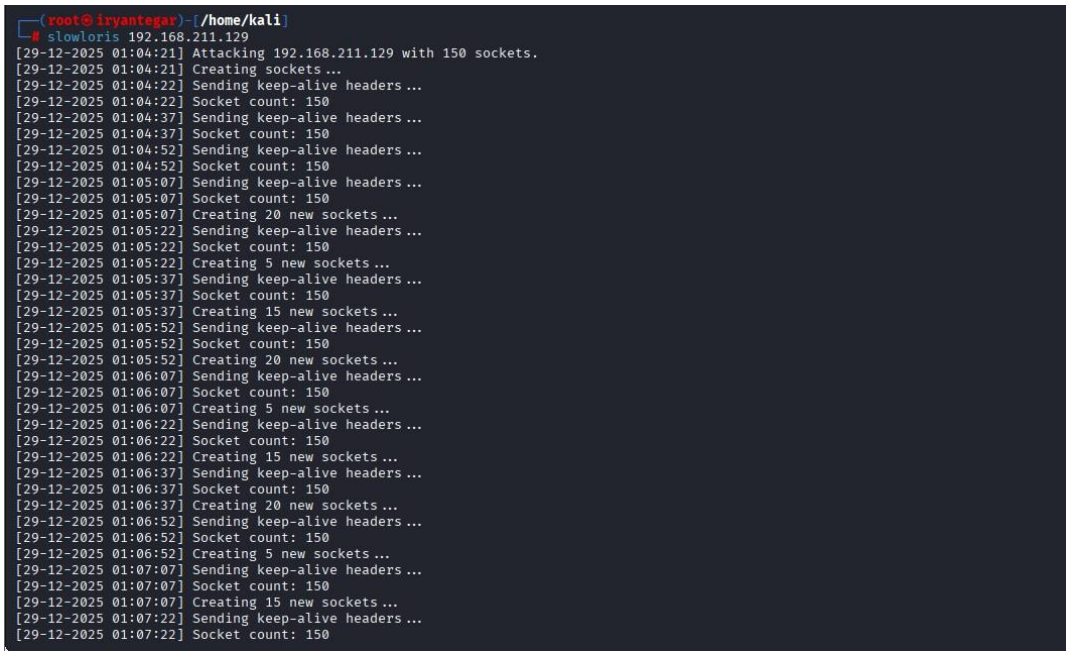
Gambar di atas menunjukkan kondisi akses aplikasi DVWA melalui browser pada mesin Kali Linux (attacker) saat serangan SYN Flood menggunakan Hping3 sedang berlangsung. Browser menampilkan pesan kegagalan koneksi (connection timed out), yang menandakan bahwa server target tidak mampu memberikan respons terhadap permintaan HTTP. Hal ini membuktikan bahwa serangan Denial of Service (DoS) tidak hanya berdampak pada pengguna dari sisi target, tetapi juga menyebabkan layanan web sepenuhnya tidak dapat diakses dari jaringan mana pun selama serangan berlangsung.

9. EKSEKUSI SERANGAN 2: SLOWLORIS (APPLICATION LAYER)

Serangan ini bekerja dengan cara membuka banyak koneksi ke server namun menahannya agar tidak pernah selesai (menggantung), sehingga slot koneksi Apache habis.

a. Perintah Serangan (di Kali Linux)

slowloris 192.168.2111.129

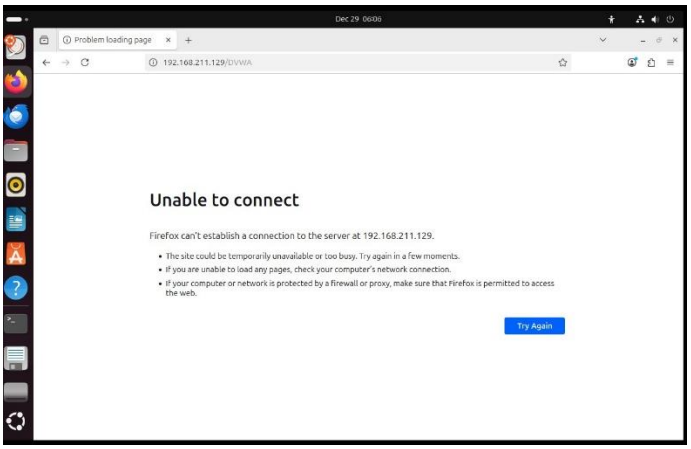


Gambar 6.1 Eksekusi Serangan Slowloris pada Layanan Web Apache

Gambar di atas menampilkan eksekusi perintah slowloris 192.168.211.129 pada sistem operasi Kali Linux untuk melakukan serangan Denial of Service (DoS) jenis Slowloris terhadap server target. Serangan ini bekerja dengan membuka banyak koneksi HTTP ke server Apache dan mempertahankannya dalam kondisi tidak selesai (half-open connection), sehingga menghabiskan slot koneksi yang tersedia. Akibatnya, server tidak mampu menerima koneksi baru dari pengguna sah, menyebabkan layanan web menjadi sangat lambat atau tidak dapat diakses selama serangan berlangsung.

b. Dampak Layanan

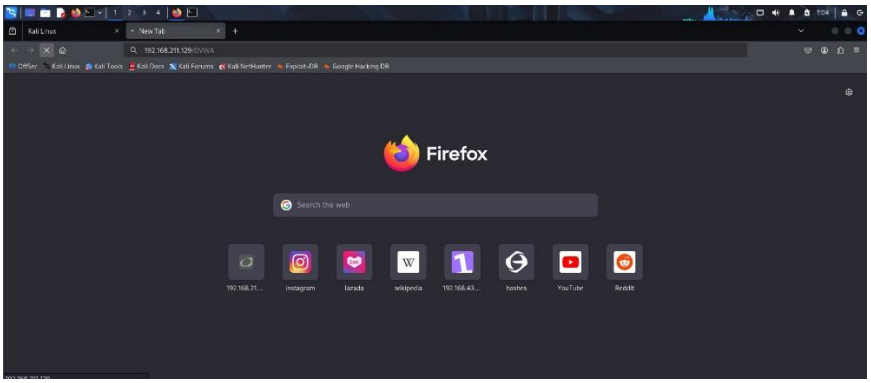
- Browser ubuntu



Gambar 6.2 Kegagalan Akses Layanan Web pada Browser Ubuntu akibat Serangan Slowloris

Gambar di atas menunjukkan kondisi layanan web DVWA yang diakses melalui browser pada mesin Ubuntu Server (target) saat serangan Slowloris sedang berlangsung. Browser menampilkan pesan unable to connect atau gagal terhubung ke server, yang menandakan bahwa Apache tidak mampu menerima permintaan koneksi baru. Hal ini terjadi karena seluruh slot koneksi telah habis digunakan oleh koneksi palsu yang dipertahankan oleh Slowloris, sehingga layanan web menjadi tidak tersedia bagi pengguna sah meskipun server masih aktif secara sistem.

- Browser kali linux



Gambar 6.3 Tampilan Browser Kali Linux Tidak Responsif saat Serangan Slowloris

Gambar di atas menunjukkan kondisi akses layanan web DVWA melalui browser pada mesin Kali Linux (attacker) saat serangan Slowloris sedang berlangsung. Browser Firefox terlihat berhenti pada tampilan awal (stuck/loading) tanpa berhasil memuat

halaman web, yang menandakan bahwa server target tidak mampu memberikan respons HTTP. Kondisi ini membuktikan bahwa serangan Slowloris efektif melumpuhkan layanan web secara menyeluruh, sehingga baik pengguna dari sisi target maupun dari jaringan lain tidak dapat mengakses layanan selama serangan berlangsung.

10. PENGUJIAN MITIGASI SERANGAN DENIAL OF SERVICE (DoS)

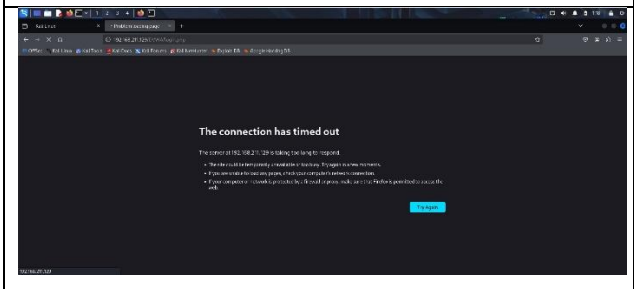
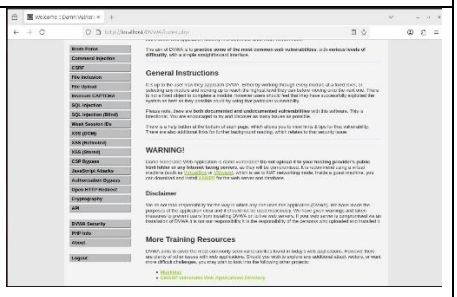
a. Pengujian Mitigasi Serangan SYN Flood (Hping3)

- Sudo iptables -A INPUT -s 192.168.211.128 -j DROP

Gambar 7.1 Penerapan Firewall IPTables untuk Mitigasi Serangan SYN Flood

Gambar di atas menunjukkan penerapan aturan firewall menggunakan perintah sudo iptables -A INPUT -s 192.168.211.128 -j DROP pada Ubuntu Server untuk memblokir seluruh trafik masuk yang berasal dari alamat IP attacker. Aturan ini bertujuan untuk menghentikan paket TCP SYN yang dikirim secara masif oleh Hping3, sehingga serangan SYN Flood tidak lagi dapat mencapai server. Setelah aturan firewall diterapkan, layanan web DVWA dapat kembali diakses secara normal, yang membuktikan bahwa penggunaan IPTables efektif dalam memitigasi serangan Denial of Service (DoS) pada layer jaringan.

- Perbedaan akses layananan web antara attacker dan target setelah mitigasi firewal

Attacker	Target
	

Gambar 7.2 Perbedaan Akses Layanan Web antara Attacker dan Target setelah Mitigasi

Gambar di atas menunjukkan perbedaan kondisi akses layanan web DVWA setelah penerapan mitigasi serangan Denial of Service (DoS) menggunakan firewall. Pada sisi

attacker (Kali Linux), browser tidak dapat mengakses halaman DVWA karena koneksi diblokir oleh aturan IPTables, sehingga serangan tidak lagi mencapai server. Sebaliknya, pada sisi target (Ubuntu Server), layanan web DVWA dapat diakses kembali dengan normal tanpa mengalami *time out*. Perbedaan ini membuktikan bahwa mitigasi yang diterapkan berhasil memutus trafik berbahaya dari attacker sekaligus mempertahankan ketersediaan layanan bagi pengguna yang sah.

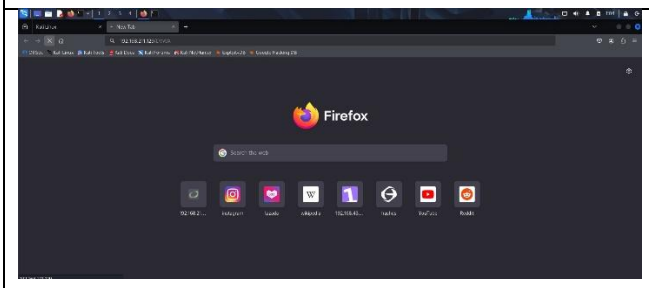
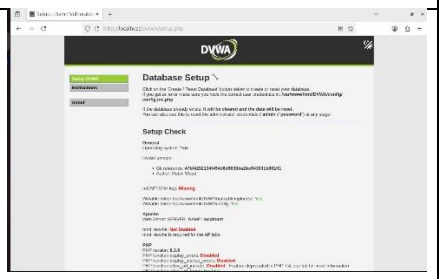
b. Pengujian Mitigasi Serangan Slowloris

- Sudo iptables -A INPUT -p tcp –syn -m limit –limit 1/s -j ACCEPT

Gambar 7.3 Penerapan Pembatasan Koneksi TCP untuk Mitigasi Serangan Slowloris

Gambar di atas menunjukkan penerapan aturan firewall menggunakan perintah sudo iptables -A INPUT -p tcp --syn -m limit --limit 1/s -j ACCEPT pada Ubuntu Server untuk membatasi laju koneksi TCP yang masuk ke layanan web. Aturan ini bertujuan untuk mencegah pembukaan koneksi secara berlebihan seperti yang dilakukan oleh serangan Slowloris. Setelah mitigasi diterapkan, akses ke layanan web DVWA dari sisi attacker menjadi terhambat atau gagal terhubung, sementara dari sisi target layanan web dapat diakses kembali dengan normal. Perbedaan kondisi ini membuktikan bahwa pembatasan koneksi melalui firewall efektif dalam mengurangi dampak serangan Slowloris dan menjaga ketersediaan layanan web.

- Perbedaan akses layananan web antara attacker dan target setelah mitigasi firewall

Attacker	target
	

Gambar 7.4 Perbedaan Akses Layanan Web antara Attacker dan Target setelah Mitigasi Firewall

Gambar di atas menunjukkan perbedaan kondisi akses layanan web DVWA setelah mitigasi serangan Denial of Service (DoS) diterapkan menggunakan firewall. Pada sisi attacker (Kali Linux), browser tidak dapat mengakses layanan web dan menampilkan kegagalan koneksi karena trafik dari alamat IP penyerang telah dibatasi atau diblokir oleh aturan firewall. Sebaliknya, pada sisi target (Ubuntu Server), aplikasi DVWA dapat diakses kembali secara normal tanpa mengalami time out. Perbedaan ini membuktikan bahwa mitigasi firewall berhasil menghentikan trafik berbahaya dari attacker sekaligus menjaga ketersediaan layanan web bagi pengguna yang sah.

11. KESIMPULAN

Berdasarkan simulasi yang dilakukan:

a. Perintah netstat berhasil membuktikan adanya lonjakan trafik tidak wajar saat serangan Hping3 terjadi.
b. Mematikan tcp_syncookies adalah faktor kunci yang membuat serangan SYN Flood berhasil masuk ke server Ubuntu.
c. Kedua metode serangan (Hping3 dan Slowloris) sukses melumpuhkan layanan web DVWA, mengakibatkan pengguna tidak bisa mengakses halaman login.