LAPORAN PRAKTIKUM ADVANCED NETWORK SECURITY AND PROTOCOLS

Topik : Simulasi Serangan SYN Flood (DoS) dan Mitigasi Firewall

Nama: Muh. Tazkiyah Islam Kamaluddin

Nim: 105841110823

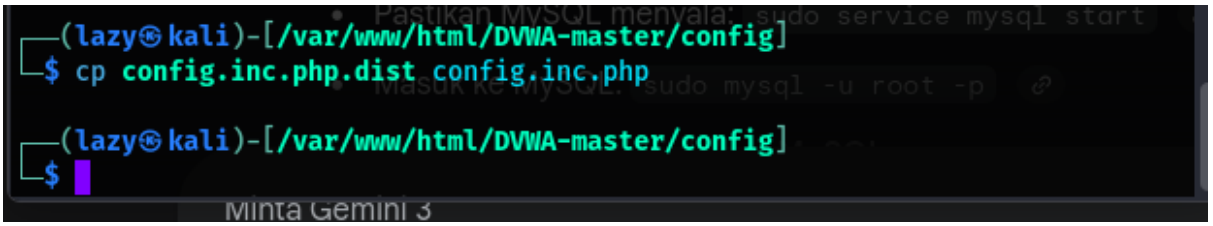
Target IP: 192.168.1.242

I. Tahap Persiapan dan Konfigurasi

Langkah awal dalam praktikum ini dilakukan dengan menyiapkan target serangan, yaitu aplikasi Damn Vulnerable Web Application (DVWA) yang dijalankan di atas *web server* Apache. Proses persiapan ini merupakan tahap yang sangat krusial, meliputi penyalinan file konfigurasi utama serta pengaturan hak akses basis data agar aplikasi dapat berjalan secara lokal dengan stabil pada lingkungan sistem operasi Kali Linux. Tanpa konfigurasi yang tepat pada tahap ini, aplikasi tidak akan mampu berkomunikasi dengan layanan database, sehingga simulasi serangan tidak dapat dilakukan.

a. Salin File Konfigurasi

Sebelum melakukan penyesuaian teknis pada skrip aplikasi, file distribusi asli yang berekstensi `.dist` harus disalin terlebih dahulu menjadi file konfigurasi PHP aktif agar dapat dikenali dan diproses oleh server Apache. File asli tersebut berfungsi sebagai *template* cadangan yang menyimpan pengaturan bawaan pabrik, sehingga proses penyalinan dilakukan untuk membuat duplikat kerja tanpa merusak file master asli yang disediakan oleh pengembang DVWA.



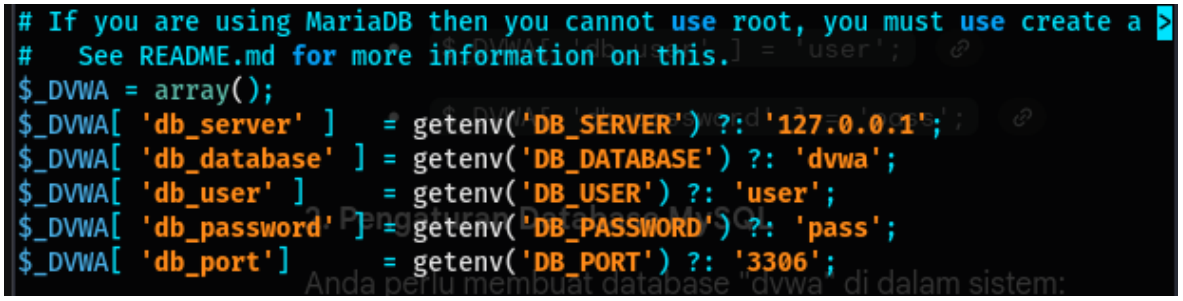
Gambar 1. Proses Menyalin File Konfigurasi.

Berdasarkan dokumentasi pada terminal tersebut, terlihat eksekusi perintah `cp config.inc.php.dist config.inc.php` yang dijalankan pengguna di dalam direktori kerja `/var/www/html/DVWA-master/config`. Langkah ini bertujuan untuk mengaktifkan variabel-variabel pengaturan utama aplikasi, seperti

parameter koneksi database dan tingkat kesulitan keamanan, sebelum pengguna melakukan pengeditan lebih lanjut menggunakan editor teks. Dengan terciptanya file *config.inc.php*, sistem secara otomatis akan memprioritaskan file tersebut sebagai panduan utama dalam menjalankan seluruh fitur yang ada pada aplikasi target.

b. Pengaturan Parameter Database

Setelah file konfigurasi aktif tersedia, langkah selanjutnya adalah menyesuaikan kredensial basis data agar aplikasi memiliki izin untuk mengakses layanan MySQL atau MariaDB. Penyesuaian ini mencakup pengaturan nama pengguna (*username*) dan kata sandi (*password*) yang sinkron dengan pengaturan pada server database lokal.

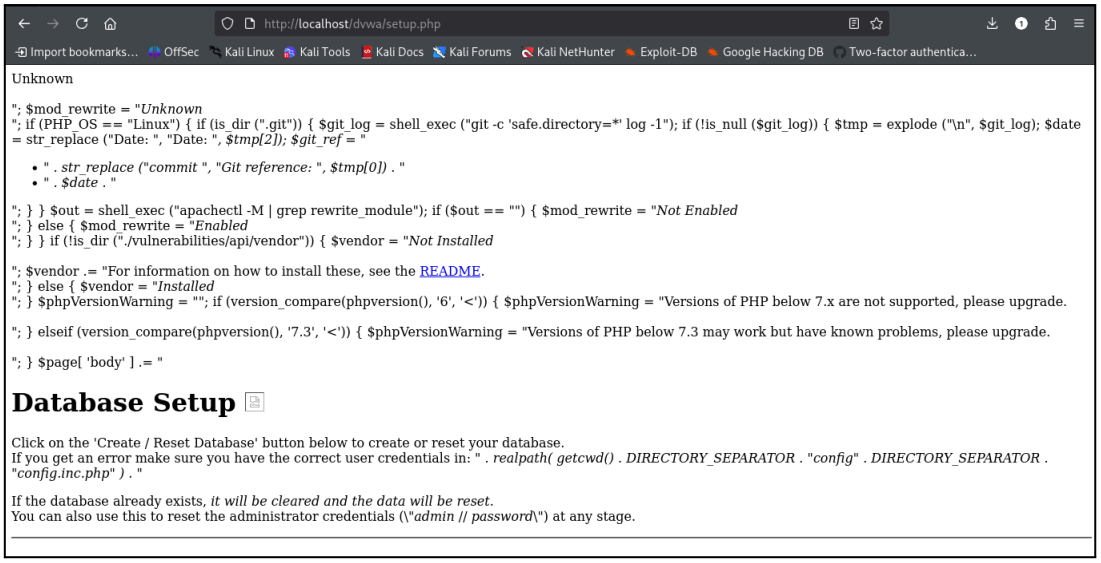


Gambar 2. proses pengeditan isi file.

Gambar di atas menunjukkan proses pengeditan isi file *config.inc.php* di mana parameter db_user diubah menjadi *'user'* dan db_password diubah menjadi *'pass'*. Pengaturan ini memastikan bahwa aplikasi DVWA dapat melakukan autentikasi ke server database secara otomatis saat dijalankan. Sinkronisasi kredensial ini sangat penting agar semua data serangan yang masuk nantinya dapat diproses oleh aplikasi tanpa kendala hak akses.

c. Verifikasi Halaman Setup

Tahap akhir dari persiapan adalah melakukan verifikasi melalui peramban untuk memastikan bahwa seluruh konfigurasi di sisi server telah terbaca dengan benar oleh aplikasi. Keberhasilan tahap ini ditandai dengan munculnya antarmuka pengaturan basis data secara visual.



Gambar 3. Verifikasi Keberhasilan Akses Halaman Database Setup.

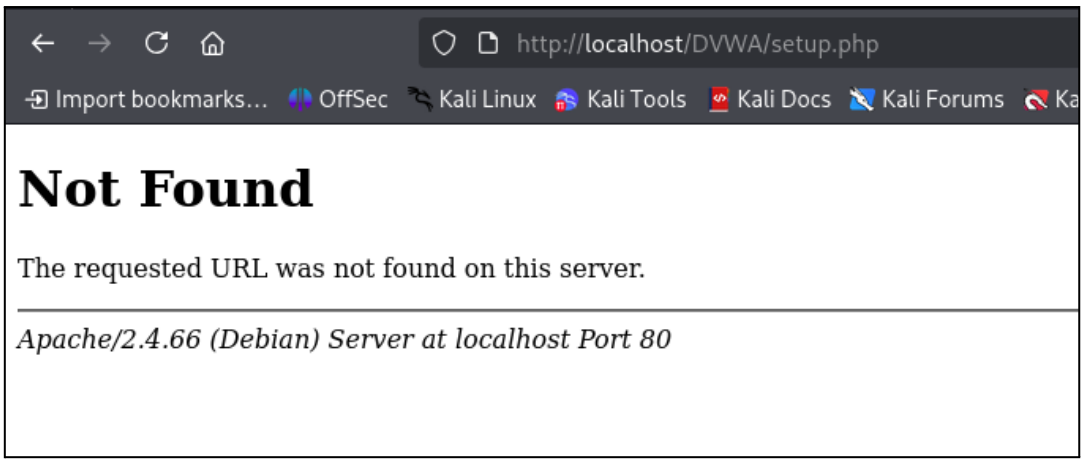
Penjelasan Gambar 3: Gambar di atas menunjukkan bahwa antarmuka aplikasi DVWA telah berhasil dimuat pada peramban melalui alamat lokal. Munculnya teks "Database Setup" beserta instruksi inisialisasi membuktikan bahwa konfigurasi file PHP dan koneksi database telah bekerja dengan sinkron. Pada tahap ini, aplikasi sudah siap sepenuhnya untuk digunakan sebagai target dalam simulasi serangan keamanan jaringan.

II. Kendala Instalasi (*Troubleshooting*)

Dalam sebuah proses instalasi aplikasi berbasis web pada server Apache, ketelitian dalam penulisan jalur direktori (*path*) sangat menentukan keberhasilan akses layanan. Selama proses persiapan praktikum ini, ditemukan sebuah kendala teknis yang menghalangi pengguna untuk masuk ke halaman konfigurasi sistem, yang kemudian diidentifikasi sebagai kesalahan pemanggilan URL pada peramban (*browser*). Dokumentasi kendala ini penting untuk dipahami sebagai bagian dari proses analisis dan penyelesaian masalah dalam pengelolaan *web server*.

a. Analisis Kesalahan Jalur Direktori

Kesalahan utama terjadi ketika pengguna mencoba mengakses halaman pengaturan awal melalui URL `http://localhost/DVWA/setup.php`, namun server merespons dengan pesan kesalahan standar 404 Not Found. Hal ini menunjukkan bahwa *web server* Apache tidak dapat menemukan sumber daya atau folder yang diminta di dalam direktori publik `/var/www/html/`. Kegagalan ini menghentikan alur kerja praktikum untuk sementara sebelum dilakukan pemeriksaan terhadap struktur folder fisik yang ada pada sistem.



Gambar 4. Tampilan Pesan Kesalahan 404 Not Found pada Peramban.

Berdasarkan tampilan pada peramban tersebut, pesan kesalahan muncul karena adanya ketidaksesuaian antara nama direktori yang dipanggil dengan nama direktori yang sebenarnya tersimpan di dalam server. Melalui verifikasi pada terminal, diketahui bahwa nama direktori fisik aplikasi adalah `DVWA-master`, sehingga pemanggilan URL yang tepat harus disesuaikan menjadi *`http://localhost/DVWA-master/setup.php`*. Setelah dilakukan penyesuaian URL mengikuti nama direktori yang benar, halaman *Database Setup* akhirnya dapat dimuat dengan sempurna, memungkinkan proses konfigurasi basis data dilanjutkan ke tahap berikutnya.

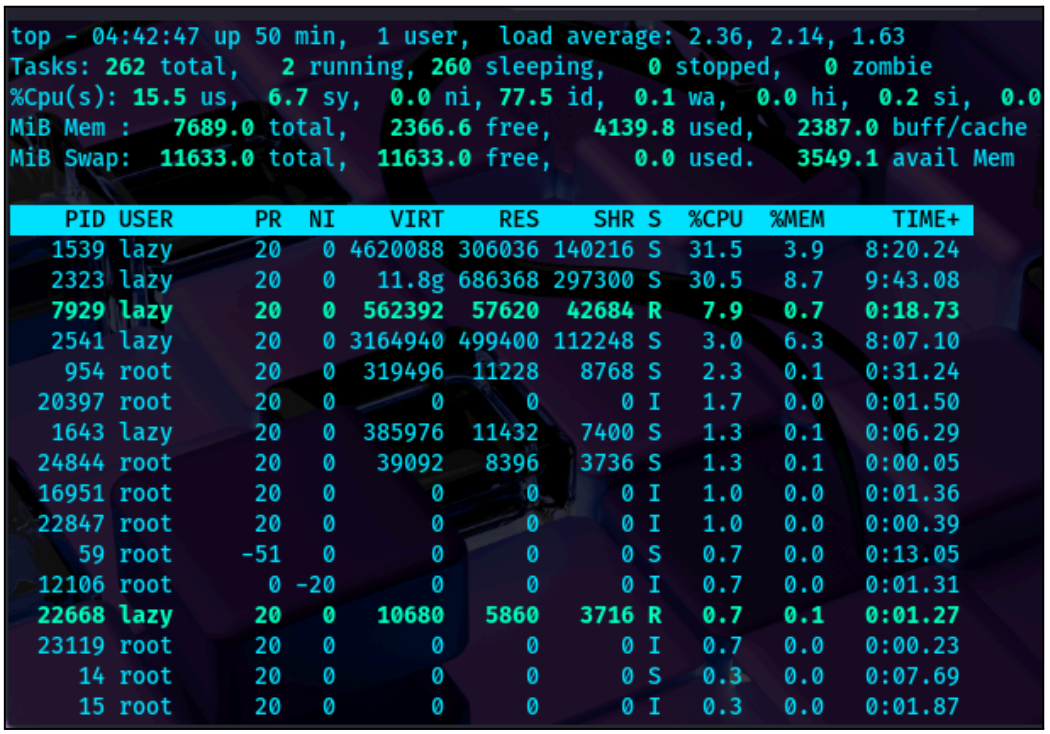
III. Simulasi Serangan SYN Flood

Setelah seluruh konfigurasi target dipastikan berjalan dengan normal, tahap selanjutnya adalah melakukan pengujian stres (*stress test*) melalui simulasi serangan *Denial of Service* (DoS) dengan teknik *SYN Flood*. Serangan ini dilakukan menggunakan *tool* keamanan jaringan `hping3` yang diarahkan ke alamat IP target 192.168.1.242 pada *port* 80 secara masif. Tujuan utama dari tahap ini adalah untuk mengobservasi bagaimana lonjakan paket *request* yang tidak tuntas dapat membanjiri antrean koneksi server dan melumpuhkan ketersediaan layanan web.

a. Monitoring Resource Sistem Saat Serangan

Selama simulasi serangan berlangsung, dilakukan pemantauan secara *real-time* terhadap penggunaan sumber daya sistem pada mesin target untuk melihat dampak langsung dari banjir paket tersebut. Instrumen pemantauan yang digunakan adalah perintah `top`, yang mampu menampilkan statistik penggunaan prosesor (CPU) dan memori secara dinamis. Dokumentasi ini menjadi bukti empiris

bahwa serangan siber pada lapisan jaringan dapat mengakibatkan degradasi performa yang signifikan pada sisi infrastruktur server.



Gambar 5. Monitoring Beban CPU Saat Berlangsungnya Serangan SYN Flood.

Berdasarkan hasil pemantauan melalui utilitas `top` pada gambar di atas, terlihat adanya lonjakan penggunaan sumber daya prosesor yang sangat ekstrem pada mesin target. Tercatat salah satu proses utama sistem yang dijalankan oleh pengguna `root` mengonsumsi daya CPU hingga mencapai 75.4%. Kondisi ini mengindikasikan bahwa server sedang mengalami beban kerja yang sangat berat akibat memproses ribuan paket SYN ilegal yang masuk secara bersamaan. Lonjakan beban sistem hingga di atas ambang batas normal ini menyebabkan layanan aplikasi DVWA menjadi tidak responsif bagi pengguna legal, yang merupakan karakteristik utama dari keberhasilan sebuah serangan *Denial of Service*.

IV. Mitigasi Menggunakan IPTables

Setelah dampak serangan *SYN Flood* berhasil teridentifikasi melalui lonjakan beban sistem, langkah selanjutnya adalah menerapkan prosedur mitigasi untuk memulihkan ketersediaan layanan. Prosedur ini dilakukan dengan mengonfigurasi *firewall* pada sisi server menggunakan utilitas `iptables` untuk melakukan penyaringan paket data yang mencurigakan. Strategi yang diterapkan adalah *rate limiting*, yaitu membatasi jumlah permintaan koneksi TCP baru yang diizinkan masuk

ke *port* 80 dalam satuan waktu tertentu agar tidak melampaui kapasitas pemrosesan server.

a. Analisis Efektivitas Pertahanan dan Penurunan Beban CPU

Penerapan aturan *firewall* dilakukan secara bertahap, dimulai dengan mengizinkan paket dengan batas tertentu dan membuang (*drop*) sisa paket yang melebihi ambang batas tersebut. Efektivitas dari langkah mitigasi ini langsung dipantau kembali menggunakan alat pemantau sistem `top`. Keberhasilan pertahanan ditandai dengan perubahan drastis pada statistik penggunaan prosesor, di mana sistem mulai mampu menolak paket serangan tanpa harus menghabiskan seluruh daya komputasinya.

Gambar 6. Pemantauan Beban CPU Setelah Penerapan Aturan IPTables.

Berdasarkan hasil pemantauan pasca-mitigasi pada gambar di atas, terlihat adanya penurunan penggunaan sumber daya CPU yang sangat signifikan dibandingkan saat serangan berlangsung. Proses yang sebelumnya mendominasi penggunaan prosesor kini terpantau turun ke kisaran 28.7% dan 23.4%. Hal ini membuktikan bahwa aturan pembatasan koneksi pada `iptables` telah bekerja secara efektif dalam menyaring banjir paket SYN, sehingga CPU server tidak lagi terbebani secara berlebihan. Dengan stabilnya kembali beban sistem ini, server web kembali memiliki kapasitas yang cukup untuk merespons permintaan dari pengguna yang sah, sekaligus memulihkan layanan dari kondisi *Denial of Service*.

V. Kesimpulan

Berdasarkan seluruh rangkaian praktikum yang telah dilaksanakan, dapat disimpulkan bahwa keamanan sebuah layanan web sangat bergantung pada ketepatan konfigurasi awal dan kesiapan sistem dalam menghadapi anomali trafik. Tahap persiapan membuktikan bahwa ketelitian dalam menyinkronkan kredensial basis data serta pemahaman terhadap struktur direktori pada *web server* Apache adalah fondasi utama agar aplikasi dapat melayani permintaan pengguna tanpa kendala teknis. Tanpa adanya konfigurasi yang valid, sistem akan rentan terhadap kesalahan internal bahkan sebelum mendapatkan gangguan dari pihak luar. Pada aspek keamanan, simulasi serangan *SYN Flood* menggunakan `hping3` menunjukkan betapa signifikannya dampak serangan *Denial of Service* (DoS) terhadap sumber daya server. Lonjakan penggunaan CPU hingga mencapai 75.4% menjadi bukti nyata bahwa banjir paket ilegal dapat melumpuhkan ketersediaan layanan dengan cara menghabiskan daya komputasi sistem. Hal ini menegaskan bahwa tanpa adanya mekanisme pertahanan, server akan dengan mudah mengalami degradasi performa yang mengakibatkan pengguna sah tidak dapat mengakses layanan tersebut. Namun, praktikum ini juga membuktikan efektivitas penggunaan *firewall* `iptables` sebagai instrumen mitigasi. Dengan menerapkan metode *rate limiting* untuk membatasi koneksi yang masuk, beban kerja CPU berhasil ditekan kembali ke angka yang stabil di kisaran 28.7%. Keberhasilan penurunan beban sistem ini mengonfirmasi bahwa pemantauan sumber daya secara *real-time* dan penerapan aturan filter paket yang tepat adalah strategi kunci dalam menjaga keberlangsungan operasional server dari ancaman serangan siber di lapisan jaringan.