p-ISSN : ?, e-ISSN : ? , available at : https://ejournal.unsrat.ac.id/index.php/informatika

Personal *Web* Hosting Design and *Reverse Proxy* Implementation

Rancang Bangun Personal *Web* Hosting Dan Implementasi *Reverse Proxy*

Faith Th. D. Posumah [1)], Yaulie D. Y. Rindengan [2)], Agustinus Jacobus [3)]
Jurusan Teknik Elektro, Universitas Sam Ratulangi Manado, Jl. Kampus Bahu, 95115, Indonesia
Email : 16021106083@student.unsrat.ac.id[1)], rindengan@unsrat.ac.id [2)], a.jacobus@unsrat.ac.id [3)]
Received: [date]; revised: [date]; accepted: [date]

***Abstract* — The era of internet technology advancement allows people to move, learn, work, and communicate using social networking media and internet-connected devices. Web Hosting is a solution for those who want to publish websites, mobile applications, and store files online. A hosting platform is essential for serving content quickly and efficiently, with various features such as Web Server, Database Server, DNS Server, FTP Server, and Email Server. Server security is also a concern in hosting platforms, to protect against hacker attacks and other illegal activities. Firewall protection is used to reduce the risk of attacks. In this context, the author wants to develop a complete and secure web hosting platform. The objective of this research is to design and implement a Reverse Proxy on a Personal Web Hosting service platform, with a focus on improving the security and performance of the platform.**

Keywords: Web Hosting, Reverse Proxy, Nginx, Apache.

***Abstrak* — Era kemajuan teknologi internet memungkinkan masyarakat untuk beraktivitas, belajar, bekerja, dan berkomunikasi menggunakan media jejaring sosial dan perangkat terhubung internet. *Web* Hosting menjadi solusi bagi mereka yang ingin mempublikasikan situs *web*, aplikasi mobile, dan menyimpan file secara online. Platform hosting penting untuk menyajikan konten dengan cepat dan efisien, dengan berbagai fitur seperti *Web Server*, Database *Server*, DNS *Server*, FTP *Server*, dan Email *Server*. Keamanan *server* juga menjadi perhatian dalam platform hosting, untuk melindungi dari serangan hacker dan kegiatan ilegal lainnya. Perlindungan firewall digunakan untuk mengurangi risiko serangan. Dalam konteks ini, penulis ingin mengembangkan platform *web* hosting yang lengkap dan aman. Tujuan dari penelitian ini adalah merancang dan mengimplementasikan *Reverse Proxy* pada platform layanan Personal *Web* Hosting, dengan fokus pada meningkatkan keamanan dan kinerja platform tersebut.**

Kata Kunci : Web Hosting, Reverse Proxy, Nginx, Apache.

I. PENDAHULUAN

Era kemajuan teknologi saat ini sangatlah pesat sehingga memudahkan masyarakat untuk bisa beraktivitas seperti belajar, bekerja, dan berkomunikasi antar lokasi yang berbeda dan menjalin interaksi sosial dengan menggunakan media jejaring sosial seperti Facebook, Twitter, Whatsapp, Messesnger, dan lainnya. Penggunaan telepon seluler, laptop, PC, hingga tablet yang terhubung dengan internet di masyarakat bukan lagi hal baru yang hanya dimiliki oleh segelintir orang, tetapi manfaat internet sendiri sudah dirasakan oleh semua kalangan. Dengan internet, masyarakat dapat dengan mudah mengakses semua jenis informasi. *Web* Hosting adalah salah satu teknologi yang dapat memudahkan orang untuk dapat mempublikasikan situs *web*, aplikasi mobile, maupun penyimpanan file secara online.

Saat membangun situs *web* ataupun system aplikasi, platform hosting menjadi salah satu faktor penentu dalam keberhasilan penyajian konten yang tersimpan. Platform hosting yang baik dituntut untuk dapat memproses setiap script dengan baik dan cepat sehingga pengguna yang mengakses situs *web* ataupun system aplikasi tersebut dapat dengan cepat menyelesaikan pekerjaan ataupun kebutuhan dari pengguna tersebut [2].

Platform hosting menggunakan kombinasi dari beberapa fitur yang pada umumnya dibutuhkan oleh pengembang situs *web* ataupun sistem aplikasi, seperti *Web Server*, Database *Server*, DNS *Server*, FTP *Server*, dan Email *server*. Dalam hal ini, *Web Server* bertugas untuk memproses *server*-side scripting, PHP (PHP: Hypertext Preprocessor). Database *server* bertugas untuk memproses basis data, dan menjadi tempat pengumpulan/gudang data, contohnya MySQL. DNS *Server* bertugas untuk mengkonversikan alamat yang berbentuk alfanumerik menjadi alamat IP address yang dimiliki oleh *server*, contohnya Bind9. FTP *Server* bertugas untuk mengatur masuk keluar file dari dalam *server*. Melalui FTP *Server* user dapat melakukan aktivitas upload dan download data langsung dari *server*, contohnya PRoFTP. Email *server* bertugas untuk mengatur masuk keluar email dari dalam sistem, melalui fitur Email *Server* user dapat mengirimkan dan menerima email langsung dari *server*, contohnya PostFix [3] .

Seiring dengan kebutuhan pengguna hostingan yang sudah semakin besar, aspek keamanan *server* juga menjadi hal wajib untuk diperhatikan dalam platform hosting ini. Aktivitas illegal dari hacker (peretas), botnet, dan serangna-serangan cyber lainnya dapat menggangu aktivitas dari pengguna, maka dari itu perlunya peningkatan keamanan dengan menggunakan proteksi firewall agar platform hosting tidak mudah diserang/diretas oleh hacker. Maka dari itu penulis ingin membangun platform *web* hosting yang lengkap dan minim dari serangan-serangan cyber.

A. Penelitian Terkait

Terdapat beberapa penelitian sebelumnya yang terkait dengan rancang bangun personal *web* hosting beserta implementasi *reverse proxy* menggunakan CentOS, yaitu sebagai berikut :

1) Ananlisis dan Implementasi *Reverse Proxy* sebagai Media Komunikasi *Client Server* Menggunakan Apache oleh Ariyadi Dwi Utomo, Rr. Yuliana Rachmawati, dan Catur Iswahyudi. Penelitian ini membahas tentang penerapan *reverse proxy* sebagai solusi untuk meningkatkan komunikasi antara *client* dan *server* dalam lingkungan

Lab. Jaringan Komputer IST AKPRIND Yogyakarta. Hasil penelitian ini menunjukkan bahwa penggunaan *Reverse Proxy* dapat mempercepat akses *client* terhadap *server*, meningkatkan efisiensi komunikasi, dan mengurangi beban *server*.[4]

2) Analisis *Web Server* untuk Pengembangan Hosting *Server* Institusi: Pembandingan Kinerja *Web Server* Apache dengan *Nginx* oleh Abdul Aziz, dan Topan Tampati. Penelitian ini bertujuan untuk menganalisis dan membandingkan kinerja kedua *web server* tersebut dalam hal kecepatan, skalabilitas, dan efisiensi untuk mendukung kebutuhan hosting *server* institusi. Hasil analisis menunjukkan bahwa *Nginx* menunjukkan kinerja yang lebih baik dalam hal kecepatan dan skalabilitas, sementara Apache menonjol dalam hal kestabilan dan kemampuan mengelola berbagai jenis konten .[5]
3) Perbandingan Kualitas Komunikasi Penggunaan *Reverse Proxy* dan *Server* Block Pada *Web Server* Dalam Lingkup Virtual Machine oleh Bambang Prasetya Halim, dan Billy Susanto Panca. Penelitian ini bertujuan untuk menganalisis dan menbandingkan kinerja kedua metode tersebut dalam hal kecepatan efisiensi, dan kehandalan komunikasi antara *client* dan *server*. Hasil penelitian menunjukkan bahwa penggunaan *reverse proxy* dapat meningkatkan kulatias komunikasi dengan mengoptimalkan routing dan caching, serta memberikan fleksibilitas dalam manajemen trafik. Di sisi lain, *server* block juga memberikan performa yang baik dengan pendekatan yang lebih langsung dan efisien.[6]

B. Web Hosting

Web Hosting adalah layanan yang menyediakan tempat yang aman untuk menyimpan konten secara *online*. Kode, gambar, video, dan teks yang membentuk situs *web* yang dimana semuanya disimpan di komputer khusus yang disebut *server*. Ketika pengguna *internet* ingin melihat sebuah situs *web*, yang perlu dilakukan memasukkan alamat situs *web* atau *domain* ke *browser* pengguna. Komputer akan terhubung ke *server* situs *web* tersebut dan halaman *web* akan ditampilkan melalui *browser* pengguna. [7]

C. Firewall

Firewall adalah sistem keamanan jaringan yang memantau *traffic* masuk dan keluar untuk memblokir atau mengizinkan paket data sesuai aturan keamanan. Fungsinya adalah membentuk penghalang antara jaringan internal yang aman dan akses dari sumber eksternal untuk mencegah serangan dan *traffic* berbahaya.

Firewall juga menyaring konten tidak diinginkan dan membatasi penggunaan *bandwith*. Terdapat beberapa jenis *firewall*, seperti *next-generation firewall* dengan fitur pemeriksaan detail, *packet-filtering firewall* yang ringan, *proxy firewall* yang memproses isi paket, dan *stateful inspection firewall* yang memberikan keamanan dengan kemungkinan mengurangi performa sistem.[8]

D. Apache2

Apache adalah *server web* yang umum digunakan pada sistem operasi Linux. Fungsinya adalah melayani halaman *web* yang diminta oleh komputer klien melalui protokol HTTP. *Apache* juga mendukung protokol HTTPS dan FTP untuk mentransfer *file*

Apache sering digunakan dengan *database* MySQL dan bahasa *scripting* seperti PHP, Python, dan Pearl dalam platform LAMP. Konfigurasi *server Apache* dilakukan melalui *file* konfigurasi dan dapat dikontrol menggunakan modul. Secara default *Apache* akan mendengarkan alamat IP yang telah dikonfigurasi untuk menerima permintaan[9].

E. Nginx

Nginx adalah *software opensource* yang berfungsi sebagai *server* HTTP dan *Reverse Proxy* dengan kinerja yang baik. Dengan arsitektur modular, *Nginx* mendukung berbagai fitur seperti penyeimbang beban, *reverse proxy*, akses *cache* langsung, SSL, dan *streaming video flash*.

Nginx dipilih karena kinerjanya yang tinggi, pengaturan yang mudah, dan penggunaan sumber daya yang efisien. *Ngnix* telah digunakan oleh banyak layanan *web* besar seperti WordPress, SourceForge, Hulu, dan ComputerBase [10].

F. Reverse Proxy

Reverse Proxy merupakan jenis *proxy server* yang bertindak sebagai perantara antara klien dan *server* di belakangnya, menyembunyikan *server* tersebut dari klien. Keuntungan utama penggunaan *reverse proxy* adalah peningkatan kinerja aplikasi *web* dengan menyimpan konten yang sering diminta oleh klien di *cache*.

Reverse Proxy juga memperluas fungsionalitas aplikasi *web* dengan fitur seperti *load balancing* dan SSL *offloading*. Namun, konfigurasi kompleks dan *overhead* tambahan menjadi kelemahan penggunaan *reverse proxy*.

Meskipun demikian, *reverse proxy* tetap menjadi solusi populer untuk meningkatkan kinerja dan keamanan aplikasi *web* di berbagai jenis aplikasi. [11][12].

G. Control Web Panel (CWP)

Control Web Panel (CWP) sebelumnya dikenal dengan CentOS *Web* Panel adalah perangkat lunak administrasi *server* untuk sistem Linux. CWP memudahkan pengguna pemula dalam mengelola *server* melalui *web interface*, menghilangkan ketergantungan pada command-line interface (CLI).

CentOS *web* panel menawarkan panel kontrol *web hosting* gratis dengan banyak opsi dan fitur untuk menejemen *server*, termasuk instalasi otomatis tumpukan LAMP (Linux, Apache, MySQL, PHP) seperti Apache, MySQL, PHP, dan phpMyAdmin.[13].

Dengan *CentOS Web Panel*, pengguna dapat mengelola beberapa *server* dengan mudah, baik *Dedicated* maupun VPS, tanpa perlu menggunakan SSH untuk setiap tugas kecil yang perlu diselesaikan.

II. METODE

A. Metode Pengumpulan Data

Metode pengumpulan data adalah cara yang dapat digunakan untuk mengumpulkan atau mendapatkan data dari fenomena empiris. Untuk mendapatkan data dan informasi yang lebih valid, teknik pengumpulan data yang digunakan ada dua yaitu

metode observasi dimana pengumpulan data melalui pengamatan bersifat langsung. Penulis melakukan observasi dengan menggunakan software berupa Pingtools, Slowloris, WinSCP, dan Google Chrome, kedua metode wawancara yaitu pengumpulan data dengan mengajukan pertanyaan pada responden, berdasarkan tujuan penelitian, guna untuk mendapatkan data mengenai objek penelitian.

B. Metode Perancangan Pembuatan Web Hosting

Untuk merancang pembuatan *web* hosting menggunakan Control *Web* Panel (CWP), ada beberapa hal yang perlu dipertimbangkan diantaranya: 1) Identifikasi kebutuhan. 2) Siapkan *server*. 3) Install sistem operasi CentOS versi 7. 4) Instalasi CWP. 5) Konfigurasi CWP. 6) Tambahkan domain. 7) Tambahkan akun pengguna. 8) Instalasi aplikasi berupa WordPress.

C. Prosedur Penelitian

Secara garis besar, penulis membagi menjadi 7 langkah utama dalam melakukan proses penelitian kali ini. 1) Melakukan analisa kebutuhan spesifikasi VPS yang akan digunakan, dengan mengambil nilai rata-rata pengetesan menjadi 10 akun hosting dengan kapasitas penyimpanan masing-masing 2-5GB per akun hosting. 2) Mendaftarkan domain utama, dalam hal ini penulis menggunakan domain faithposumah.my.id sebagai root domain untuk sistem *Web* Hosting 3) Melakukan beberapa diagnosis jaringan berupa ping ke root *server* dns (ns1.faithposumah.my.id), melakukan speedtest, dan melakukan test IP dari *client*. 4) Melakukan instalasi beberapa paket software yang akan digunakan seperti PHP (Versi 7.3.0), Apache2, MySQL/MariaDB, PHPMyAdmin, CSF (Config *Server* Firewall), ModSecurity, Mail *Server*. 5) Melakukan uji coba koneksi domain ke hosting. Domain yang dapat menggunakan ns1.faithposumah.my.id dan ns2.faithposumah.my.id sebagai pointer name *server*. 6) Melakukan assessment/pengujian fitur hosting antara lain mengakses fitur file manager, menginstall script CMS (Content Management System) lengkap beserta database. Dalam hal ini, penulis menggunakan Wordpress sebagai contoh. Mengakses fitur PHPmyadmin untuk mengatur database yang akan digunakan. Menggunakan dan mengkonfigurasi fitur CSF dan ModSecurity untuk pengamanan *server* dari serangan-serangan siber (DDoS, Bruteforce, dll). 7) Melakukan assessment pada firewall dengan simple DDoS Attack ke arah *server*, dalam hal ini penulis menggunakan Slowloris untuk menyerang *server*. Penulis menargetkan VPS *Server* tidak akan terganggu dari serangan DDoS Attack ini.

D. Skema Topologi Jaringan

Skema topologi jaringan yang melibatkan PC *client*, *internet*, dan *web server* dapat mengacu pada arsitektur jaringan yang umum digunakan untuk menyediakan akses internet dan hosting situs *web*. Dan juga skema ketika mengiplementasikan *Reverse Proxy Nginx* dan serangan *Distributed Denial-of-Service* (DDoS) yang dilakukan oleh penyerang (attacker).

1) *Web Server tanpa Filter*

Dalam skema topologi pada Gambar 1 ini, PC *client* terhubung ke internet melalui penyedia layanan internet (ISP) atau jaringan lokal seperti jaringan kantor atau sekolah. Koneksi internet memungkinkan PC *client* untuk mengirimkan permintaan ke *web server* yang terletak di tempat lain di internet. *Web server* menerima permintaan tersebut, memprosesnya, dan mengirimkan kembali data yang diminta ke PC *client* melalui internet. Dimana *web server* belum terfilterisasi.

2) *Attacker DDoS menyerang Web Server tanpa Filter*

Dalam skema Gambar 2, penyerang melancarkan serangan DDoS untuk mengganggu atau menghentikan layanan pada *web server* yang belum terfilterisasi. Penyerang menggunakan botnet, yaitu jaringan perangkat yang terinfeksi dan dikendalikan oleh penyerang, untuk mengirimkan traffic yang berlebihan ke *web server*. Dengan membanjiri *web server* dengan jumlah permintaan yang sangat tinggi, penyerang bertujuan untuk membuat *web server* menjadi tidak responsif, melambat, atau bahkan mengalami kegagalan.

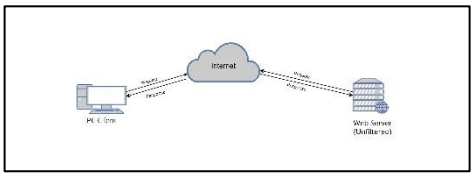
3) *Implementasi Reverse Proxy Nginx*

Dalam skema topologi Gambar 3, PC *client* mengirim permintaan ke *reverse proxy Nginx* melalui internet. *Reverse proxy Nginx* menerapkan aturan filter untuk memeriksa dan memodifikasi permintaan yang masuk sebelum diteruskan ke *web server*. Aturan filter ini dapat mencakup pembatasan akses, proteksi terhadap serangan, atau menyaring traffic berbahaya. Setelah permintaan difilter, *reverse proxy Nginx* meneruskan permintaan yang lolos ke *web server* yang sesuai. *Web server* kemudian memproses permintaan dan mengirimkan responsnya kembali ke *reverse proxy*, yang kemudian meneruskannya ke PC *client* melalui internet

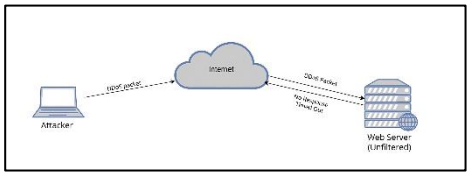
4) *Attacker DDoS menyerang Web Server yang sudah terfilter*

Dalam skema Gambar 4 ini, PC *client* mengirimkan permintaan ke *reverse proxy Nginx* melalui internet. *Reverse proxy Nginx* kemudian meneruskan permintaan ke *web server* yang sudah terfilterisasi. *Web server* yang sudah terfilterisasi memproses permintaan dan mengirimkan respons kembali melalui *reverse proxy Nginx* kepada PC *client*.

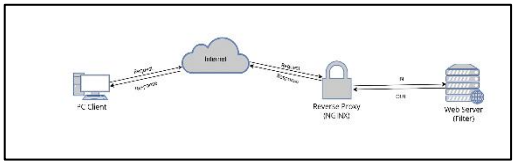
Namun, dengan adanya serangan DDoS dengan kecepatan 10.000 req/s, *web server* dan *reverse proxy Nginx* mungkin menghadapi beban lalu lintas yang sangat tinggi dan tidak dapat ditangani. Oleh karena itu *Nginx* juga dikonfigurasi dengan kebijakan yang menentukan bahwa jika jumlah permintaan melebihi 20 req/s, *Nginx* akan menolak atau menghapus permintaan tersebut (request drop). Ini bertujuan untuk melindungi *web server* dari serangan DDoS dengan membatasi jumlah permintaan yang diterima. Dengan menolak permintaan yang berlebihan, *Nginx* dapat memprioritaskan dan menjaga kelancaran traffic masuk yang di request oleh pengguna.



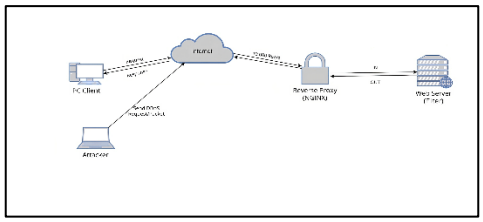
Gambar 1. *Web server tanpa filter*



Gambar 2. *Attacker DDoS menyerang Web Server tanpa Filter*



Gambar 3. *Implementasi Reverse Proxy Nginx*



Gambar 4. *Attacker DDoS menyerang Web Server yang sudah terfilter*

III. HASIL DAN PEMBAHASAN

A. *Konfigurasi CWP*

Pertama CentOS *Web* Panel harus terinstal, setelah berhasil dilakukan, langkah selanjutnya adalah mempelajari konfigurasi CWP. Namun, sebelum dapat melakukannya, perlu masuk ke dashboard CWP menggunakan username dan password yang telah dibuat pada saat proses instalasi. Setelah proses instalasi selesai, buka *web* browser dan akses halaman *web* panel dengan mengetikkan IP VPS diikuti dengan port 2031 untuk melakukan konfigurasi. https:// 202.73.25.12:2031. (Lihat Gambar 5).

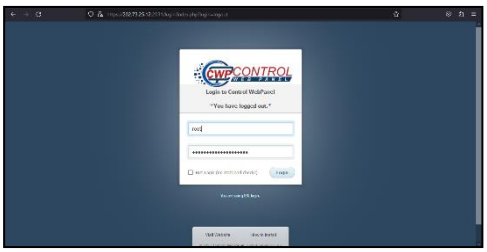
Setelah berhasil login, langkah pertama yang perlu dilakukan adalah menambahkan akun user utama yang akan di kofigurasikan. Dengan melengkapi beberapa informasi seperti nama domain, username, password, admin email seperti yang telah tertera pada Gambar 6.

Selanjutnya memperbarui alamat email pada halaman admin CentOS *Web* Panel (Lihat Gambar 7). Terdapat pada menu di sebelah kiri CWP Settings dan pilh Edit Settings, masukkan alamat email aktif pada kolom Admin email. Email yang digunakan adalah faithposumah@gmail.com.

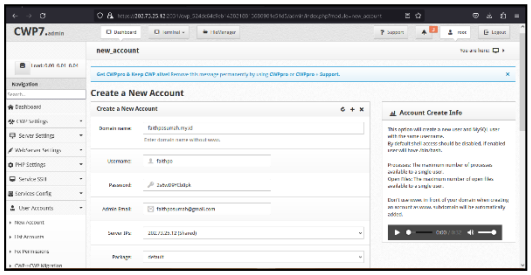
Untuk melakukan pembaruan pada name*server*, tahap berikutnya adalah masuk ke menu DNS Function dan pilih Edit Name*server* IPs. Sesuaikan Name *Server* dengan Name *Server* yang digunakan pada domain *server* (Lihat Gambar 8). Name *Server* 1 diubah menjadi ns1.faithposumah.my.id dengan IP address 202.73.25.12, dan Name *Server* 2 diubah menjadi ns2.faithposumah.my.id dengan IP address 202.73.25.12.

Selanjutnya, melakukan pengaturan pada *web server* sesuai dengan kebutuhan. Dengan menggunakan *Nginx* & Apache seperti yang tertera pada Gambar 9 dan Gambar 10 sebagai back-end *web server* dan sebagai main *web server* untuk penelitian ini.

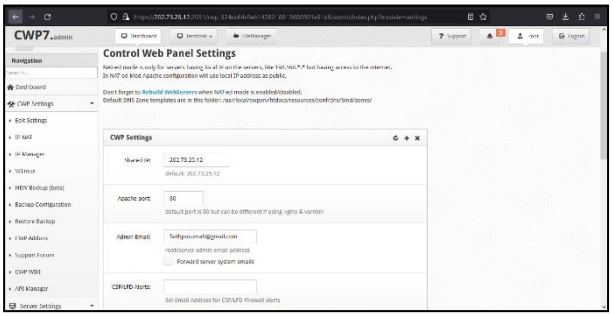
Tahap selanjutnya melakukan melakukan pengaturan pada PHP yang akan digunakan pada *server*. PHP version yang akan di gunakan dalam penelitian ini adalah versi 7.3.0 (lihat Gambar 11). Maka setelah proses pengaturan PHP selesai, akan terlihat pada Gambar 12 bahwa domain yang telah didaftarkan sudah tersedia.



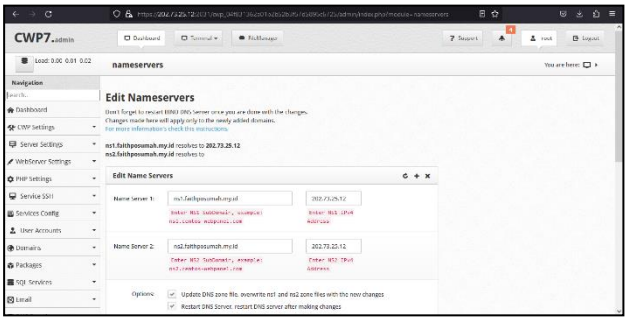
Gambar 5. Mengakses CWP



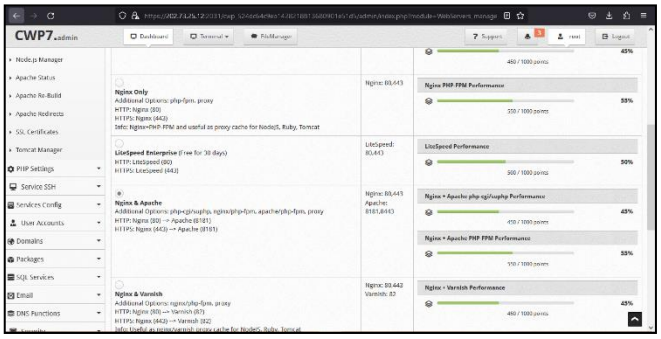
Gambar 6. Menambahkan akun user baru di CWP



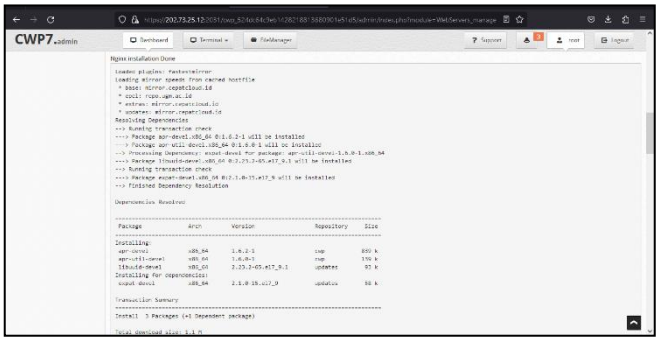
Gambar 7. Memperbaharui alamat email sebagai root/admin *server*



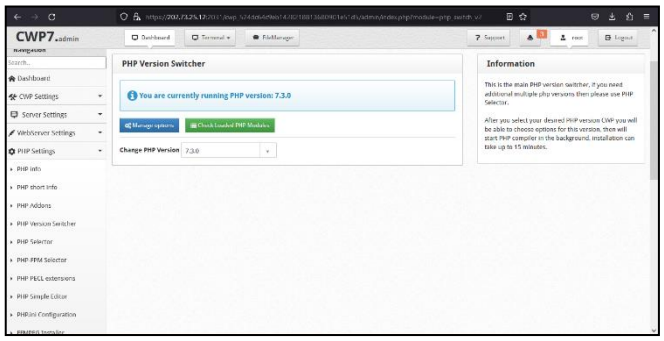
Gambar 8. Mengubah Name*server* 1 dan 2



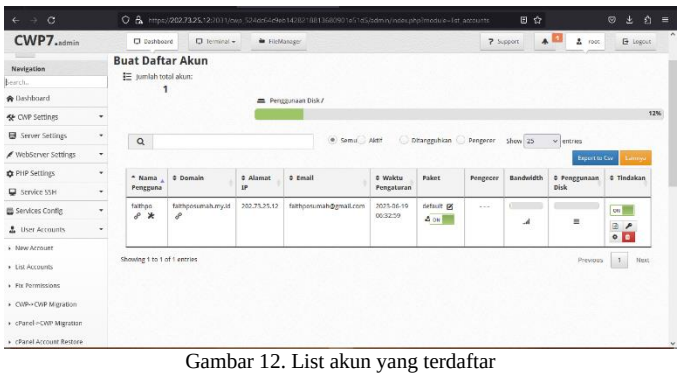
Gambar 9. Memilih *web server Nginx* & Apache



Gambar 10. *Nginx* & Apache berhasil terinstall



Gambar 11. Menggunakan PHP version 7.3.0



Gambar 12. List akun yang terdaftar

B. *Pengujian Menggunakan Serangan DoS Slowloris*

Untuk menguji tipe serangan slowloris, gunakan opsi '-H' dengan 10000 koneksi '-c'. Statistik akan ditampilkan di file 'output.csv' dan 'output.html', yang dapat ditentukan dengan opsi '-o'. Interval antara setiap header adalah 10 detik '-i', dan kecepatan koneksi adalah 200 koneksi per detik '-r'. '-t' digunakan untuk menentukan kata kerja yang digunakan dalam permintaan, sedangkan '-x' menentukan panjang maksimum data yang akan dikirim dalam tindak lanjut. '-p' mengatur interval waktu untuk menunggu respons HTTP pada koneksi probe sebelum menandai *server* sebagai DoSed. (Lihat Gambar13)

Pada Gambar 14 dapat diketahui bahwa ketika penyerangan berhasil dilakukan tanpa perlindungan dari firewall apapun, maka *web server* yang telah di serang akan mengalami gangguan atau *server* down. Dalam hal ini *web server* di serang dan mengalami gangguan hanya dalam kurun waktu 50sec.

Perintah "nano /etc/*nginx*/*nginx*.conf" pada Gambar 15 digunakan untuk membuka berkas konfigurasi utama *Nginx* yang terletak di direktori "/etc/*nginx*" dengan menggunakan editor teks Nano pada sistem operasi Linux atau Unix.

File konfigurasi /etc/*nginx*/*nginx*.conf memungkinkan konfigurasi dan pengaturan yang diperlukan untuk mengelola *server Nginx*. Dengan membuka berkas ini menggunakan editor teks Nano, mudah untuk mengedit dan memodifikasi aturan-aturan ini sesuai kebutuhan.

worker_rlimit_nofile 102400 (Lihat Gambar 16) menentukan jumlah maksimum file descriptor yang dapat dipegang oleh setiap proses pekerja. File descriptor adalah sebuah bilangan bulat yang mengidentifikasi suatu file yang sedang dibuka oleh sistem operasi. Ketika *server Nginx* menangani permintaan HTTP atau HTTPS, setiap proses pekerja akan membuka file descriptor untuk melayani permintaan tersebut. Dengan mengatur worker_rlimit_nofile, kita dapat menentukan jumlah maksimum file descriptor yang dapat dipegang oleh setiap proses pekerja. Dalam kasus ini, batasan jumlah file descriptor adalah 102400.

worker_connections 100000 menentukan jumlah maksimum koneksi klien yang dapat ditangani oleh setiap proses pekerja. Dalam kasus ini, batasan jumlah koneksi adalah 100000. Jumlah maksimum koneksi klien yang dapat ditangani oleh *server Nginx* bergantung pada sumber daya sistem seperti memori, CPU, dan bandwidth. Dengan mengatur worker_connections, sehingga dapat membatasi jumlah koneksi yang ditangani oleh *server Nginx* agar tidak melebihi sumber daya yang tersedia.

Perintah "nano /etc/sysctl.conf" pada Gambar 17 digunakan untuk membuka file konfigurasi sysctl.conf menggunakan editor teks nano. Sysctl.conf adalah sebuah file konfigurasi yang berisi daftar parameter sistem yang dapat diubah untuk meningkatkan kinerja atau mengoptimalkan penggunaan sistem.

Setiap perubahan yang dilakukan pada file konfigurasi sysctl.conf akan berdampak pada berbagai aspek kinerja sistem seperti manajemen memori, manajemen proses, keamanan jaringan, manajemen file sistem, dan lain sebagainya (lihat Gambar 18).

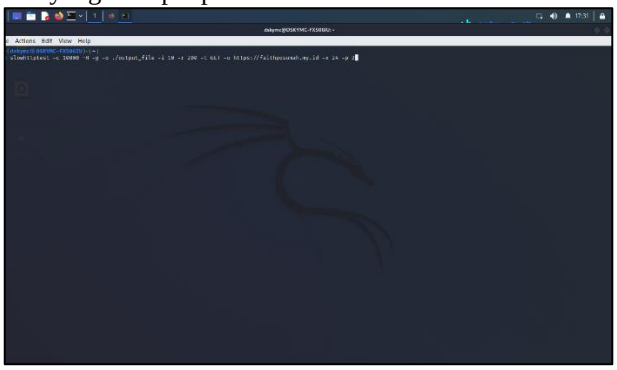
Ketika perubahan dilakukan pada parameter sistem di dalam file konfigurasi “sysctl.conf” (Lihat Gambar 19), perubahan tersebut tidak diterapkan secara langsung pada sistem. Perintah “sysctl -p” perlu dijalankan untuk memuat ulang file konfigurasi “sysctl.conf” dan menerapkan perubahan tersebut pada sistem.

Dengan menggunakan perintah "cat /proc/sys/fs/file-max" yang terdapat pada Gambar 20, jumlah maksimum file descriptor yang dapat dibuka pada sistem Linux dapat ditampilkan. File descriptor merupakan nomor yang digunakan oleh sistem operasi untuk mengidentifikasi sebuah file yang sedang dibuka oleh suatu proses. Ketika proses membuka

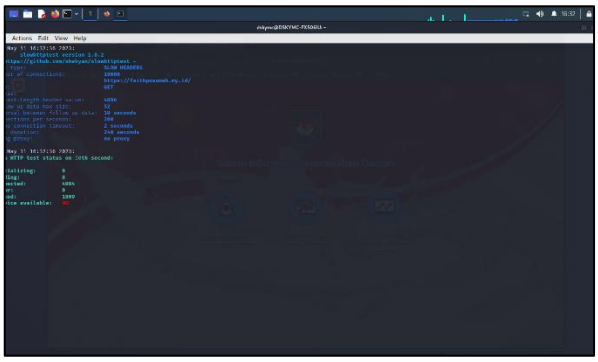
sebuah file, nomor file descriptor diberikan oleh sistem operasi kepada proses tersebut. Parameter file-max, yang terletak pada direktori "/proc/sys/fs/", menentukan jumlah maksimum file descriptor yang dapat dibuka pada sistem Linux.

Kemudian ketika semua konfigurasi berhasil terpasang, maka dilakukan penyerangan yang sama sekali lagi untuk mencoba apakah *web server* "https://faithposumah.my.id" berhasil terlindungi seperti yang terdapat dalam Gambar 22

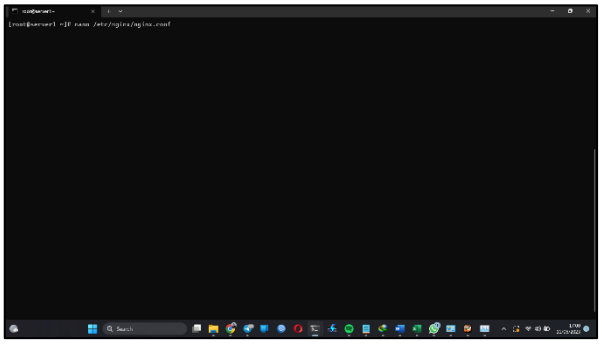
Setelah melakukan konfigurasi dan pengaturan yang diperlukan untuk mengelola *server Nginx*, dan berhasil terpasang, maka *web server* akan terproteksi dari serangan DoS, sehingga tidak akan mengalami gangguan ataupun *server* down seperti yang terdapat pada Gambar 23.



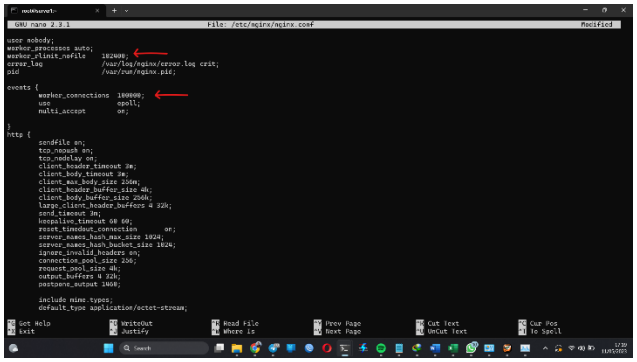
Gambar 13. Memasukan *syntax* untuk menyerang *server*



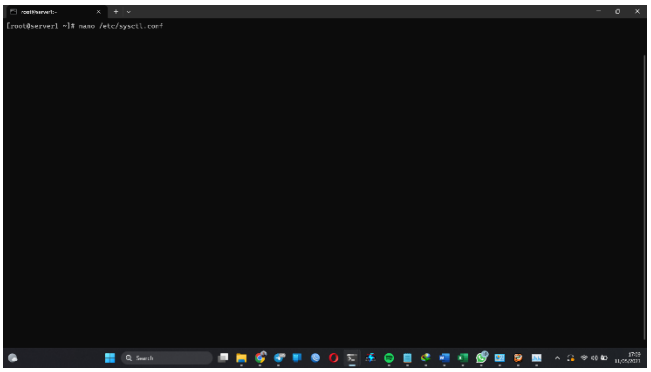
Gambar 14. Hasil penyerangan sebelum di proteksi



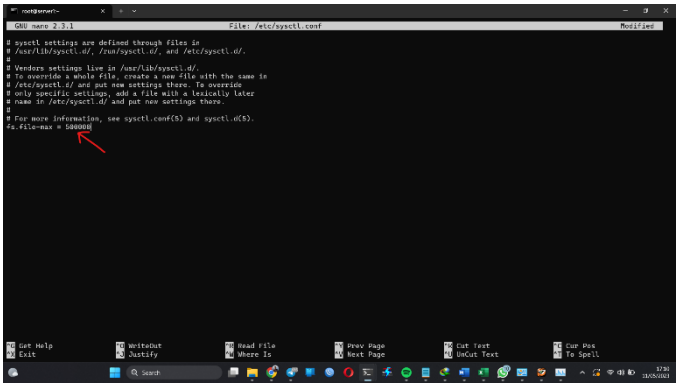
Gambar 15. Mengkonfigurasikan *Nginx*



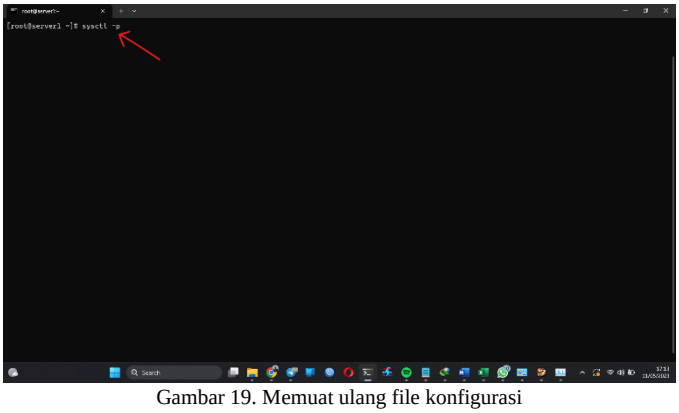
Gambar 16. Mengatur batasan jumlah file descriptor dan koneksi klien



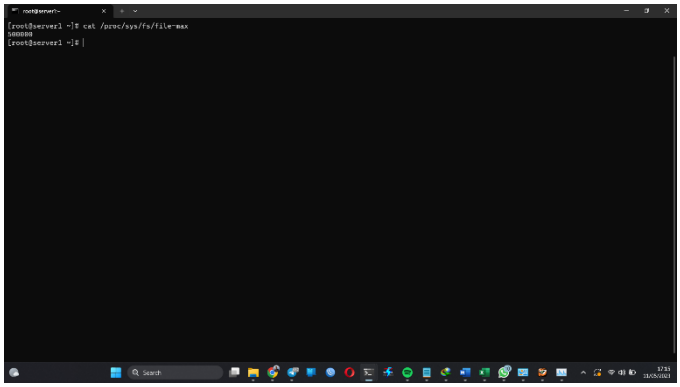
Gambar 17. Membuka file konfigurasi *sysctl.conf*



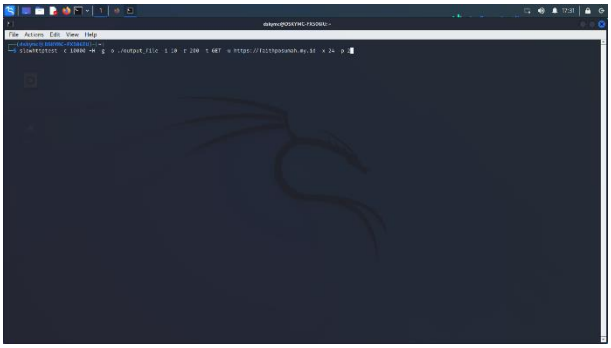
Gambar18. File-Max 500000



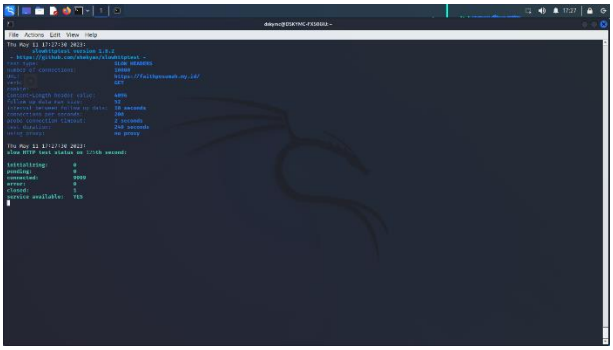
Gambar 19. Memuat ulang file konfigurasi



Gambar 20. Jumlah maksimum file descriptor yang dapat ditampilkan



Gambar 21. Memasukan kembali perintah test untuk menyerang *server*



Gambar 22. *Server* berhasil di proteksi

IV. KESIMPULAN DAN SARAN

A. *Kesimpulan*

Perancangan dan implementasi "Implementasi *Reverse Proxy* Pada Platform Layanan Personal *Web* Hosting" sebagai sistem *web* hosting yang mengimplementasikan *reverse proxy* agar dapat mengatasi permasalahan yang diakibatkan oleh serangan DDoS (Distributed Denial-of-Service). Serangan DDoS menggunakan metode pengiriman request/packet terus menerus dan mengakibatkan *web server* mengalami crash karena beban yang berlebihan. Implementasi *Reverse Proxy* pada *web server* berfungsi untuk mengidentifikasi paket/request yang berlebihan, dan melimitasi packet/request dari satu sumber IP yang sama menjadi packet/request yang normal (bukan DDoS). Melalui implementasi *reverse proxy*, *web server* dikategorikan menjadi lebih stabil dan aman karena *web server* akan terus berjalan dan tidak akan mengalami downtime karena serangan DDoS.

B. *Saran*

Untuk melindungi web server dari serangan DDoS, perlu dilakukan langkah-langkah mitigasi yang tepat. Ini meliputi penggunaan solusi anti-DDoS yang canggih untuk mendeteksi dan menghalangi serangan, penerapan firewall yang kuat, penggunaan layanan mitigasi DDoS yang disediakan oleh penyedia layanan jaringan, atau penggunaan teknik load balancing untuk mendistribusikan lalu lintas dengan lebih efisien.

V. KUTIPAN

[1] Rollin G. Thomas, *Our Modern Banking and Monetary System*. Prentice Hall, 1957.

[2] S. A. Setiawan and N. Puspitasari, “Preferensi Struktur Organisasi Bagi Generasi Millenial,” *J. Borneo Adm.*, vol. 14, no. 2, pp. 101–118, 2018, doi: 10.24258/jba.v14i2.336.

[3] G. Costin, “Millennial Spending Habits and Why They Buy,” *Forbes*, 2019. https://www.forbes.com/sites/forbesbooksauthors/2019/05/01/millennial-spending-habits-and-why-they-buy/?sh=521deef6740b (accessed Jan. 05, 2019).

[4] M. Nastiti and A. Sunyoto, “BERBASIS ANDROID Keywords : Analisis dan Perancangan,” *J. Dasi*, vol. 13, no. 2, pp. 38–43, 2012.

[5] A. Susanto, A. Noertjahyana, and A. Setiawan, “Aplikasi Pengelola Keuangan Pribadi Berbasis Android,” *J. Infra*, no. 031, pp. 2–5, 2016.

[6] B. A. Syarwan, K. R. Purba, and A. Setiawan, “Pembuatan Aplikasi Management Keuangan Pribadi Berbasis Android,” *J. Infra Petra*, pp. 3–6, 2018.

[7] N. Yulianti and M. Silvy, “Sikap pengelola keuangan dan perilaku perencanaan investasi keluarga di Surabaya,” *J. Bus. Bank.*, vol. 3, no. 1, pp. 57–68, 2013.

[8] N. Al Kholilah and R. Iramani, “Studi Financial Management Behavior Pada Masyarakat Surabaya,” *J. Bus. Bank.*, vol. 3, no. 1, p. 69, 2013, doi: 10.14414/jbb.v3i1.255.

[9] A. Zen, *Kakeibo : Seni Cerdas Finansial Ala Jepang Agar Uang Anda Tak Habis Terbuang*. Klaten: Caesar Media Pustaka, 2020.

[10] Eric Whiteside, “What Is the 50/20/30 Budget Rule?,” *Investopedia*, 2020. https://www.investopedia.com/ask/answers/022916/what-502030-budget-rule.asp (accessed Dec. 12, 2020).

[11] Y. Yudhanto and A. Wijayanto, *Mudah Membuat dan Berbisnis Aplikasi Android Dengan Android Studio*. Jakarta: PT Elex Media Komputindo, 2019.

[12] M. Alow, A. Jacobus, and S. Paturusi, “Sistem Informasi Geografis Rest Area Di Provinsi Sulawesi Utara Berbasis Mobile,” *Jurnal. Teknik. Informatika.*, vol. 14, pp. 395–402, 2019.

[13] Y. S. Dwanoko, “Implementasi Software Development Life Cycle (Sdlc) Dalam Penerapan Pembangunan Aplikasi Perangkat,” *J. Teknol. Inf.*, vol. 7, no. 2, pp. 83–94, 2016.

[14] P. Sulistyorini, “Pemodelan Visual dengan Menggunakan UML dan Rational Rose,” *J. Teknol. Inf. Din. Vol.*, vol. XIV, no. 1, pp. 23–29, 2009.

TENTANG PENULIS

Penulis bernama lengkap Faith Theadorin Davila Posumah, lahir di tondano pada tanggal 3 Juli 1998. Penulis telah menyelesaikan studi di Sekolah Dasar Negeri 4 Tondano pada tahun 2010, setelah itu melanjutkan studi di Sekolah Menengah Pertama Negeri 1 Tondano dan lulus pada tahun 2013 dan pada tahun yang sama melanjutkan studi di Sekolah Menengah Atas Kristen 1 Tomohon dan lulus pada tahun 2016. Melanjutkan pendidikan strata satu (S1) di Fakultas Teknik Jurusan Teknik Elektro Program Studi Informatika Universitas Sam Ratulangi Manado yang dimulai pada bulan Juli 2016 melalui jalur Tumou Tou (T2) pada tahun 2016. Aktif dalam organisasi Himpunan Mahasiswa Elektro dan beberapa kegiatan Unit Pelayanan Kerohanian Kristen Fakultas Teknik.