Analisis Reliabilitas Multiserver Menggunakan Load Balancing Dengan Metode Denial Of Service

Mochamad Dimas Erlangga[1], Agus Prihanto[2]
[1,2] Jurusan Teknik Informatika Fakultas Teknik Universitas Negeri Surabaya
UNESA Kampus Ketintang Surabaya
[1]mochamad.17051204076@mhs.unesa.ac.id
[2]agusprihanto@unesa.ac.id

Abstrak — Pada saat ini jaringan komputer memiliki ancaman yang begitu besar, ancaman tersebut berupa Serangan Denial of Service. Dengan cara membebani server atau jaringan komputer serangan ini dilakukan dengan mengirim lalu lintas yang tidak berguna dan juga besar secara terus menerus pada satu server yang telah menjadi target serangan, sehingga server yang menjadi objek serangan harus menghabiskan begitu banyak waktu dan juga menambah beban kerjanya dalam menangani serangan lalu lintas. Web server merupakan salah satu sistem yang berpotensi terkena serangan, seperti serangan Denial Of Service berjenis Slowloris. Penelitian ini bertujuan untuk meminimalisir serangan DOS dengan menggunakan Load Balancing traffic kearah server menggunakan teknologi docker container. Dalam persiapan pengujian sistem, terdapat beberapa kebutuhan untuk analisis reliabilitas ini, diantaranya adalah Web Server berjalan pada OS Ubuntu dalam Virtual Machine, localhost http server sebagai alamat host website yang diserang, php dan python sebagai bahasa pemrograman, Slowloris sebagai tools penyerangan DoS, Docker sebagai sarana untuk mengaplikasikan Load Balancing, NginX sebagai Load Balancer, dan juga Wireshark sebagai alat monitoring anomali jaringan.

Hasil pengujian menunjukkan bahwa pada skenario jika tanpa menggunakan Load Balancing terdapat RTO yang telah dihitung rata-ratanya mencapai 1,333655 detik sedangkan dengan menggunakan Load Balancing terdapat RTO yang telah dihitung rata-ratanya mencapai waktu delay 0,83699 detik. Hal ini menunjukkan bahwa RTO berhasil diminimalisir dengan menggunakan Load Balancing jika dibandingkan dengan tanpa Load Balancing karena semakin tinggi RTO maka kualitas TCP kurang baik.

Kata Kunci - DOS, Docker Container, Load Balancing, RTO.

I. PENDAHULUAN

Pada Era Modernisasi ini masalah serangan Denial Of Service (DoS) pada suatu jaringan komputer terus menerus mengalami perkembangan di lingkungan masyarakat. Pada saat ini jaringan komputer memiliki ancaman yang begitu besar, ancaman tersebut berupa Serangan Denial of Service. Dengan cara membebani server atau jaringan komputer serangan ini dilakukan dengan mengirim lalu lintas yang tidak berguna dan juga besar secara terus menerus pada satu server yang telah menjadi target serangan, sehingga server yang menjadi objek serangan harus menghabiskan begitu banyak waktu dan juga menambah beban kerjanya dalam menangani serangan lalu lintas sedemikian rupa sehingga tidak dapat melakukan pekerjaan yang semestinya, karena permintaan yang ditempatkan pada beban kerja telah melebihi kapasitas beban kerja server tersebut [12]. Serangan DoS merupakan serangan terbesar yang dilakukan oleh penyusup dengan ditujukan pada jaringan alamat host website orang lain, biasanya dilakukan oleh oknum tertentu, dan efek serangan yang dilakukannya juga dapat menyebabkan klien yang ditargetkan kehabisan sumber daya sehingga tdiak dapat melayani request yang datang [9]. Sehingga untuk dapat menggali informasi sebagai bukti digital adanya serangan DoS pada alamat host website tersebut, maka diperlukan adanya suatu analisis tentang serangan DoS dengan tools Slowloris ini.

Reliabilitas merupakan suatu kemampuan sebuah sistem untuk memulihkan diri dari suatu gangguan, dengan cara mengalokasi sumber daya komputasi demi memenuhi permintaan dan memitigasi gangguan seperti kesalahan konfigurasi atau gangguan yang lain nya [14]. Serangan ini dilakukan oleh PC host OS terhadap PC rekayasa pada Virtual Machine. Virtual Machine merupakan suatu program perangkat lunak atau sebagai sistem operasi virtual yang bisa dijalankan pada sebuah perangkat keras secara bersamaan dengan sistem operasi asli dari perangkat tersebut. Nantinya pengguna dapat dengam mudah untuk menjalankan aplikasi, script maupun program yang terdapat pada perangkat lunak rekayasa secara virtual layaknya menggunakan perangkat yang berbeda, ini dapat terjadi karena adanya Virtual Machine saat ini. Virtual Machine juga menjadi aplikasi yang akan menaungi Ubuntu untuk menjadi system operasi yang menjalankan perangkat untuk diserang. Ubuntu merupakan suatu sistem operasi dan distribusi dari Linux yang bersifat open-source dan juga berbasis Debian. Dengan menggunakan infrastruktur Debian yang terdiri dari server, desktop, dan OS Linux, merupakan cara untuk menciptakan Ubuntu [17]. Website merupakan sebuah kumpulan halaman pada suatu domain di internet yang dibuat dengan tujuan tertentu dan saling berhubungan serta dapat diakses secara luas melalui halaman depan (home page) dengan menggunakan sebuah browser menggunakan URL website [3]. Alamat Website dapat diserang dengan DoS, karena serangan DoS (Denial of Service) merupakan serangan cyber dimana pelaku berupaya membuat mesin atau sumber daya jaringan yang tidak tersedia untuk pengguna yang dituju dengan mengganggu layanan sementara atau tanpa batas waktu yang terhubung ke internet. Serangan ini dapat dilakukan hanya dengan menggunakan satu mesin komputer, karena merupakan serangan yang dilakukan secara individu.

Metode DoS merupakan metode penyerangan yang dapat dilakukan jika PC penyerang lebih kuat daripada PC yang akan diserang, sehingga penyerang dapat membanjiri target dengan paket traffic besar yang dapat dikirimkan dengan sebanyak banyaknya. Penyerangan bertipe DoS pada saat ini akan menyebabkan korban sulit untuk melacak keberadaan penyerang yang sesungguhnya karena lebih variatif dan juga lebih terkoordinasi dengan baik. Penyerangan dengan DoS yang didistribusikan dengan menggunakan beberapa node (DoS menggunakan 1 node) akan berdampak lebih besar kepada target yang akan membanjiri target dengan mengirimkan banyak paket traffic secara serentak dari beberapa tempat [6]. Menyerang alamat Website dengan menggunakan serangan Denial of Service, yang intinya dapat membuat Website pada satu server akan mudah untuk terkena overload, dan juga permintaan server untuk request timeout akan berpotensi down akibat banyaknya pengunjung.

Cara mencegahnya supaya server tidak berpotensi mengalami down karena melayani request yang banyak, maka perlu menggunakan Multiserver yang ditangani oleh Load Balancing. Multiserver atau Multiple Server merupakan sistem yang mencakup lebih dari satu server, karena untuk memberikan layanan kepada pelanggan yang masuk ke antrian pelanggan merupakan tugas Multiserver. Model sistem Multiserver dapat dirancang dengan beberapa server serupa atau dengan berbagai jenis server. Serangkaian besar masalah praktis melibatkan studi sistem dengan banyak server. Banyak masalah dunia nyata dapat dimodelkan dengan skema ini. Model Multiserver paling sederhana termasuk antrian pelanggan tunggal Model dengan antrian yang terpisah untuk tiap-tiap server merupakan contoh lain dari model Multiserver [15]. Dengan adanya Multiserver dapat membantu kinerja suatu server, karena jika salah satu server menerima serangan paket traffic yang besar secara terus menerus, maka kinerjanya akan dimudahkan dengan cara mengalihkan beban traffic kepada server yang lainnya. Nantinya Multiserver yang dinaungi oleh Load Balancing akan bertugas untuk membagi traffic pada long server tersebut, dan balancer juga menjadi gateway utama buat front-end nya.

Load Balancing merupakan suatu proses pembagian beban traffic sebuah aplikasi atau server. Beban traffic tidak akan dibebankan kepada beberapa jalur koneksi saja dengan adanya Load Balancer [2]. Penerapan Load Balancing biasanya digunakan pada server virtual dengan virtualisasi, virtualisasi server merupakan teknologi untuk membagi sumber daya fisik secara virtual yang dapat melayani permintaan layaknya server. Pada penelitian Bansal & Kaur, (2015) yang berjudul "An Implementation of Servers using Lightweight Virtualization / Containerization", Server dapat di implementasikan pada mesin virtual menggunakan pendekatan hypervisor dari virtualisasi, tetapi karena kekurangan hypervisor, lebih baik menggunakan virtualisasi berbasis Sistem Operasi [1].

Konsep dari teknologi kontainer dan virtualisasi adalah untuk mendapatkan daya tarik pada lingkungan IT yang berbasis cloud [13]. Teknologi Docker merupakan teknologi virtualisasi yang populer pada saat ini. Docker merupakan suatu teknologi dengan arsitektur containerization, kontainer tersebut nantinya akan bertugas untuk memuat kumpulan gambar yang berisi data konfigurasi dan juga file pendukung yang lainnya. Sehingga, para tim developer seringkali menggunakan Docker sebagai suatu solusi dalam pengerjaan beberapa proyek pengembangan aplikasi di berbagai environtment yang ada [4]. Kontainer Docker akan membungkus perangkat lunak dalam file secara lengkap beserta sistem yang berisi semua data yang diperlukan untuk menjalankan: kode, runtime, alat sistem, pustaka sistem apa pun yang dapat diinstal pada server. Ini dapat menjamin bahwa perangkat lunak akan selalu berjalan secara bersamaan, terlepas dari environmentnya [7]. Dengan Docker, peneliti menerapkan dua server web pada satu host dengan overhead prosesor minimum. Server bekerja sama seperti ini ketika diterapkan pada masing-masing mesin. Dengan pendekatan ini, administrator jaringan dan administrator sistem dapat dengan mudah untuk mengelola beberapa server mereka pada satu host saja, sehingga dapat mengurangi biaya infrastruktur. Dalam menjalankan aplikasi ini diperlukan adanya suatu Web Server seperti NginX (dibaca Engine-X), NginX merupakan suatu software yang berjenis open source. NginX hanya memiliki fungsi sebagai HTTP web serving saja pada awalnya, tetapi NginX dapat berperan sebagai reverse proxy, HTTP Load Balancer, dan juga email proxy pada saat ini. Hingga sekarang, jumlah koneksi yang ditangani oleh web server terus bertambah. Karena itulah, NginX menawarkan suatu arsitektur yang berjenis event-driven dan asinkron. NginX sebagai salah satu server yang kecepatan dan skalabilitasnya dapat diandalkan dengan adanya arsitektur ini. NginX dengan kecepatan dan kemampuannya dalam menangani jumlah koneksi yang banyak, layanan NginX kerap digunakan oleh website yang memiliki traffic tinggi [16]. Dengan adanya NginX membuat server web tersebut menjadi lebih kuat, fleksibel, dan juga mampu dalam memberi keputusan pada server web mana yang akan diadopsi sepenuhnya tergantung pada kebutuhan pengguna [8]. NginX selain digunakan sebagai web server, NginX juga memiliki beberapa fitur untuk digunakan sebagai reverse proxy, HTTP cache, dan Load Balancer [5].

Mekanisme analisa Reliabilitas Multiserver ini memiliki acuan pada paket data yang berhasil dikirim ke Wireshark pada sebuah koneksi TCP akan diurutkan dengan sebuah antrian paket dan akan mengharapkan kiriman paket acknowledgement (ACK) dari penerima. Jika terdapat paket yang terlalu banyak dan berhasil dikirim maka akan menyebabkan paket ACK dari penerima tidak dapat mengirim sehingga memutuskan untuk meminta waktu penundaan sistem selama beberapa detik. Setelah melewati waktu penundaan sistem dari penerima dapat melakukan tugas yang semestinya lagi, peristiwa ini bisa juga disebut dengan RTO (Retransmission Timeout). Semakin banyak terdapat Retransmission, maka semakin buruk kualitas TCP nya. Dan jika tidak ada paket acknowledgement dari penerima, maka segmen TCP (protokol data unit dalam protokol) akan ditransmisikan ulang. Pada pihak penerima, segmen-segmen yang datang tidak sesuai dengan urutannya akan diletakkan di belakang untuk mengurutkan segmen-segmen TCP [18].

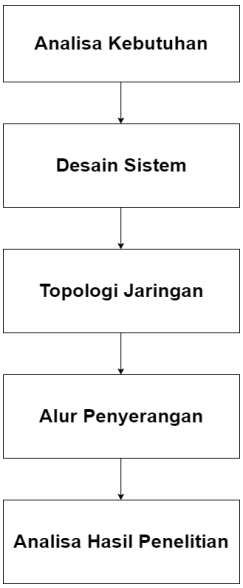
Pada penelitian ini attacker memakai sistem operasi utamanya yaitu Windows 10 Ultimate 64-bit dan software Slowloris sebagai tools yang digunakan dalam melakukan serangan DoS ini. Metode analisa yang digunakan pada penelitian ini yaitu dengan menggunakan tools dari Wireshark.

Untuk mengidentifikasi kemungkinan terjadinya suatu intrusi, maka harus dilakukan analisis dengan membandingkan record dari tiap-tiap paketnya. Analisis paket traffic penelitian ini menggunakan Wireshark karena Wireshark merupakan suatu perangkat lunak dengan berbagai macam alat untuk monitoring anomali dan analisa jaringan yang digunakan untuk menganalisis paket lalu lintas jaringan [10]. Jika IDS (Intrusion Detecting System) mendeteksi ancaman yang telah diketahui maka metode ini akan efektif, tetapi jika terdapat ancaman baru yang tidak diketahui oleh IDS maka metode ini tidak akan efektif. IDS adalah sistem keamanan jaringan yang berfungsi untuk mendeteksi interferensi pada jaringan komputer berdasarkan pola anomali yang disebabkan olehnya [10]. Server lokal yang telah dibuat yang terdapat Load Balancing didalamnya akan menjadi objek penyerangan dengan menggunakan tools Slowloris. Dengan jenis serangan DoS seperti Slowloris akan memakan sumber daya terutama pada layanan target secara signifikan yang dapat mempengaruhi sumber daya penyerang, karena Slowloris merupakan serangan tingkat rendah [11]. Setelah itu traffic pada alamat host tersebut akan dianalisa dengan menggunakan aplikasi Wireshark.

Berdasarkan paparan hasil dari penelitian terdahulu diatas terkait serangan yang terjadi pada jaringan komputer, dapat diketahui bahwa masih terdapat juga celah kelemahan pada penelitian diatas yaitu perlu untuk mempertimbangkan lagi serangan Denial Of Service pada suatu server. Sehingga penulis dapat berasumsi bahwa untuk meminimalisir serangan Denial Of Service diperlukan suatu pendistribusian traffic pada server yang mengalami overload, terhadap server yang lainnya.

II. METODOLOGI PENELITIAN

Jenis penelitian ini merupakan penelitian eksperimen, yaitu metode penelitian yang memiliki tujuan untuk meneliti pengaruh dari Load Balancing pada Multiserver terhadap serangan Denial Of Service yang berjenis Slowloris. Multiserver yang dinaungi oleh Load Balancing nantinya akan bertugas untuk melakukan pendistribusian traffic. Metode merupakan suatu tata cara yang dirancang dan juga dipakai untuk mencapai suatu tujuan tertentu. Beberapa tahap yang telah dilakukan dalam analisis serangan DoS pada web server tersebut, yaitu:



Gbr 1. Skenario Penelitian

A. Analisa Kebutuhan

Dalam analisis reliabilitas ini, terdapat beberapa kebutuhan yang perlu dianalisa. Kebutuhan tersebut akan digunakan sebagai bahan untuk membantu analisis ini. Analisa kebutuhan dibagi menjadi beberapa bagian, yaitu:

1. Kebutuhan Data

Data yang diperlukan untuk penelitian ini diambil dari beberapa referensi. Untuk pendistribusian serangan oleh Load Balancing diambil dari local host dan untuk Analisis reliabilitas Multiserver merujuk pada dokumentasi resmi docker. Pengumpulan data pada penelitian ini terbagi menjadi dua jenis, yaitu studi literatur dan juga observasi.

a. Studi literatur

Pada penelitian ini mengambil beberapa referensi dari berbagai macam literatur yang relevan dengan perancangan Load Balancing dan juga sistem serangan DoS yang berjenis Slowloris untuk mendalami pengetahuan. Literatur yang digunakan diantaranya berupa laporan, jurnal nasional dan internasional, video pada youtube, makalah, situs resmi dan juga beberapa sumber dari internet.

b. Observasi

Penelitian ini juga melakukan observasi dengan mengunjungi beberapa situs referensi Load Balancing dan juga serangan DoS yang berjenis Slowloris serta situs dokumentasi docker.

2. Kebutuhan Alat

Spesifikasi perangkat yang diperlukan untuk mendukung penelitian dalam melakukan analisis reliabilitas Multiserver adalah :

a. Processor AMD Ryzen 5
b. RAM 8 GB
c. Harddisk 500 GB
d. Sistem Operasi Windows 10 64-bit

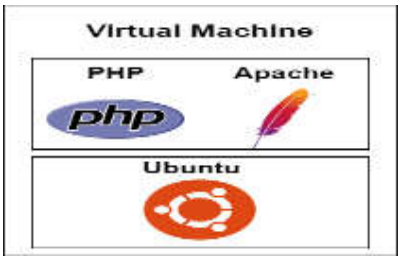
Sedangkan perangkat lunak yang diperlukan dalam penelitian ini adalah sebagai berikut :

a. Docker Desktop sebagai media dalam membangun aplikasi dalam container.
b. WSL2 (Windows Subsystem for Linux) sebagai media komunikasi backend linux dengan host OS pada windows.
c. Virtual Machine sebagai media untuk menjalankan PC rekayasa dengan OS Ubuntu.
d. Wireshark berperan sebagai media untuk menganalisis paket lalu lintas jaringan yang memiliki berbagai macam alat untuk monitoring dan analisa jaringan.
e. Notepad ++ teks editor sebagai media untuk menulis teks dan source code yang berjalan di system operasi windows.
f. Pyloris software open source Slowloris sebagai tools penyerangan pada web server.

B. Desain Sistem

Serangan yang akan di analisis dalam penelitian ini ialah serangan Denial of Service. Load Balancing memberikan dampak pada proses serangan, lebih tepatnya untuk mengatasi serangan tersebut. Didalam desain sistem tahap-tahap yang dilakukan adalah sebagai berikut :

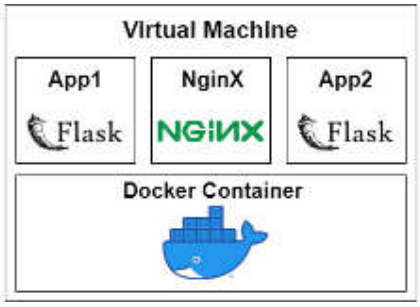
1. Arsitektur Server Web Tanpa Load Balancing



Gbr 2. Arsitektur Tanpa Load Balancing

Dalam serangan tanpa menggunakan Load Balancing, Web Server menggunakan Apache dengan script PHP pada root folder Apache 'var/www/html' untuk menampilkan tulisan "dimas" yang dijalankan pada OS Ubuntu dalam Virtual Machine.

2. Arsitektur Server Web Dengan Menggunakan Load Balancing

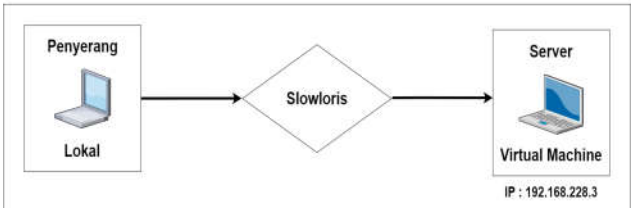


Gbr 3. Arsitektur Menggunakan Load Balancing

Dalam serangan Slowloris pada Web Server yang menggunakan Load Balancing, Virtual Machine sebagai system operasi rekayasa untuk menaungi Web Server, balancer dan juga kontainer. Python Web Apps menggunakan framework Flask yang akan menjadi objek serangan dari DoS dengan tools Slowloris. NginX bertujuan untuk menjadi Load Balancer, kemudian Wireshark sebagai alat untuk monitoring dan analisa jaringan.

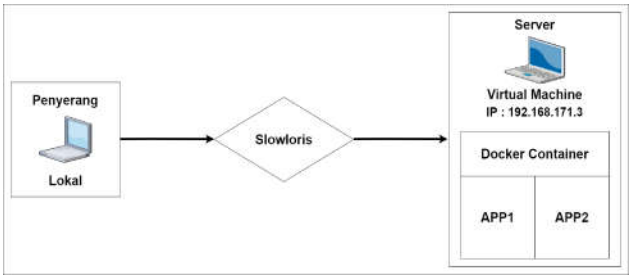
C. Topologi Jaringan

Desain Topologi digunakan untuk perancangan server yang tanpa menggunakan Load Balancing dan juga menggunakan Load Balancing sebagai objek penyerangan Slowloris.



Gbr 4. Pola Serangan Tanpa Load Balancing

Pada gambar 4. PC penyerang berada pada lokal PC, kemudian melakukan serangan Slowloris terhadap Web Server yang dijalankan pada Virtual Machine dengan IP yang ditentukan yaitu 192.168.228.3.

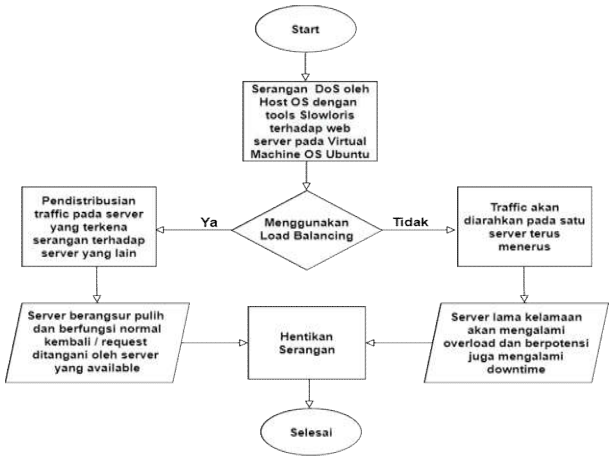


Gbr 5. Pola Serangan Dengan Load Balancing

Pada gambar 5. PC penyerang berada pada lokal PC, kemudian melakukan serangan Slowloris terhadap 2 Web Server (APP1 dan APP2) yang dinaungi Load Balancing terhadap Docker Kontainer dan dijalankan pada Virtual Machine dengan IP yang ditentukan yaitu 192.168.171.3.

D. Alur Penyerangan

Untuk memudahkan dalam penyerangan web server, dan juga untuk mempermudah pengguna dalam memahami alur serangan, maka dibuat suatu perancangan alur penyerangan seperti gambar dibawah.

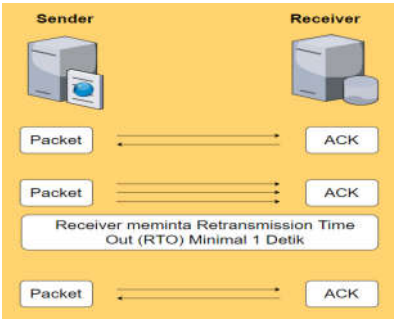


Gbr 6. Alur Penyerangan

E. Analisa Hasil Penelitian

Penarikan kesimpulan dari hasil pengujian mengacu pada data yang dimonitoring pada Wireshark, untuk menentukan reliabilitas server terdapat dua faktor diantaranya, yaitu:

1. RTO (Retransmission Time Out)



Gbr 7. RTO

RTO dapat terjadi ketika sender mengirim paket yang terlalu banyak sehingga receiver tidak bisa menerima semuanya, sehingga meminta waktu minimal 1 detik untuk memulihkan diri. Sampai sender mengirim paket yang sewajarnya lagi sehingga receiver dapat melakukan tugas yang semestinya.

2. RTT (Round-Trip Time)

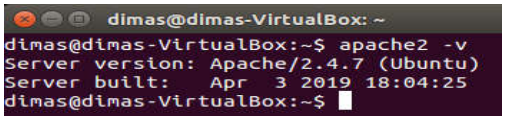
RTT adalah durasi dalam milidetik (ms) yang diperlukan untuk permintaan jaringan untuk pergi dari titik awal ke tujuan dan kembali lagi ke titik awal. RTT adalah metrik penting dalam menentukan kesehatan koneksi di jaringan lokal atau Internet yang lebih besar, dan biasanya digunakan oleh administrator jaringan untuk mendiagnosis kecepatan dan keandalan koneksi jaringan.

III. HASIL DAN PEMBAHASAN

Setelah merancang Topologi Jaringan dan juga alur serangan peneliti melakukan penerapan.

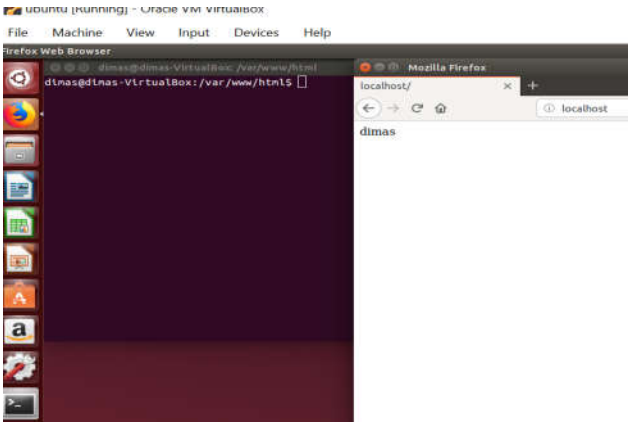
A. Serangan tanpa menggunakan Load Balancing

Langkah pertama buka Ubuntu pada VM untuk menyiapkan server pengembangan apache. Peneliti menggunakan apache versi 2.4.7 Ubuntu seperti gambar dibawah ini.



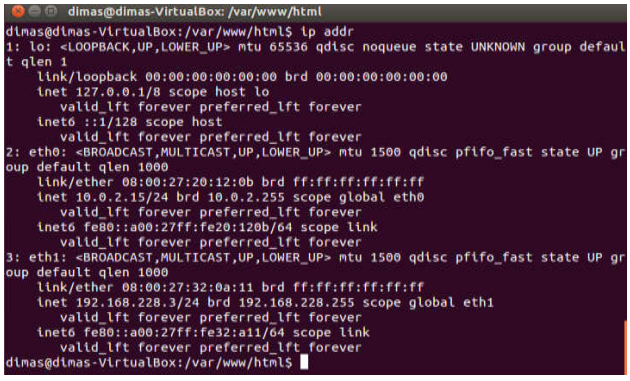
Gbr 8. Versi apache pada Ubuntu

Apache digunakan sebagai objek web server yang akan digunakan dalam penyerangan. Selanjutnya cek server yang akan diserang pada website yang terdapat di Ubuntu.



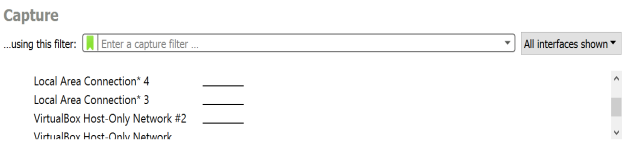
Gbr 9. Tampilan server yang akan diserang pada Ubuntu

Pada gambar 9 merupakan tampilan utama, peneliti telah menambahkan index.php untuk menampilkan tulisan “dimas” pada root folder apache. Kemudian masukkan perintah ke terminal pada ubuntu dan lakukan pengecekan alamat ip yang akan diserang.



Gbr 10. Ip address yang akan diserang

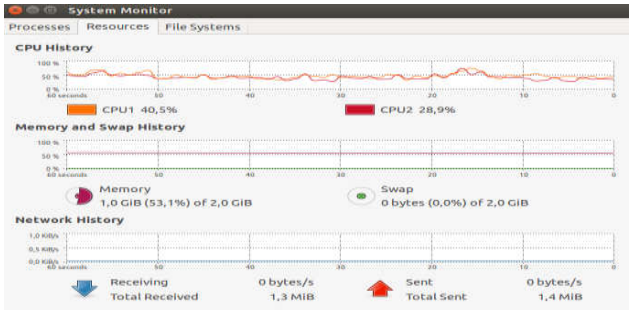
Pada gambar diatas telah diketahui bahwa ip address 192.168.228.3. Lalu untuk menganalisa serangan ini peneliti membuka Wireshark.



Gbr 11. Menu capture pada Wireshark

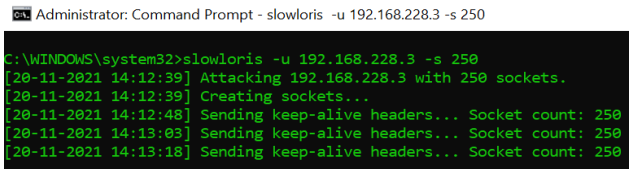
Setelah terbuka lalu pilih VirtualBox Host-Only Network#2 seperti pada gambar diatas untuk menganalisa

serangan pada jaringan VM. Selanjutnya pilih tools System Monitor pada Ubuntu, lalu pilih tampilan resource.



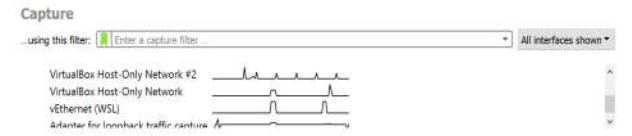
Gbr 12. Resource pada Ubuntu sebelum serangan

Gambar diatas merupakan tampilan resource pada Ubuntu sebelum dilakukan serangan Slowloris. Selanjutnya peneliti melakukan serangan Slowloris dengan menuliskan kode di Command Prompt pada Ip yang telah didefinisikan pada jaringan VirtualBox Host-Only Network#2.



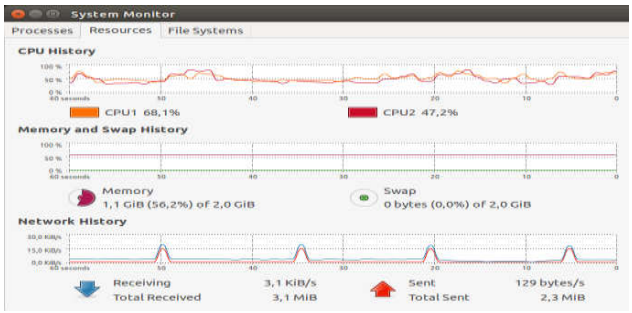
Gbr 13. Melakukan serangan Slowloris

Pada gambar 13 merupakan tahapan serangan Slowloris pada Ip yang ditargetkan dengan mengirim 250 paket serangan traffic. Lalu buka Kembali Wireshark untuk melihat anomali traffic yang terdeteksi.



Gbr 14. Tampilan pada Wireshark setelah dilakukan penyerangan

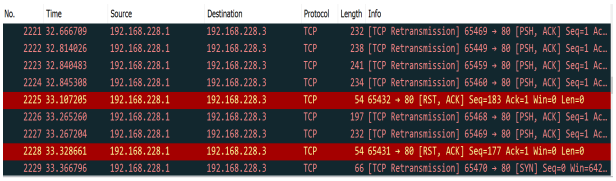
Berbeda dengan tampilan Wireshark yang belum diserang diagramnya hanya garis lurus, tetapi setelah diserang terdapat grafik naik turun. Selanjutnya peneliti membuka kembali tools system monitor pada Ubuntu untuk melihat dampak setelah dilakukan serangan.



Gbr 15. Resource pada Ubuntu setelah serangan

Berbeda dengan tampilan sebelum diserang CPU1 40,5% dan CPU2 28,9% tetapi setelah diserang mengalami peningkatan menjadi 68,1% dan 47,2%.

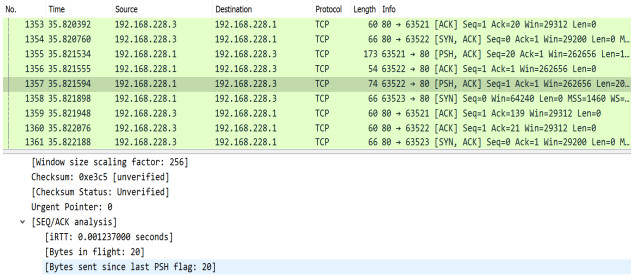
Lalu peneliti membuka Wireshark untuk menganalisa jaringan pada pilihan VirtualBox Host-Only Network#2.



Gbr 16. Highlight paket serangan pada Wireshark

Gambar diatas merupakan data paket serangan yang telah dikirim dan sukses dianalisa oleh Wireshark.

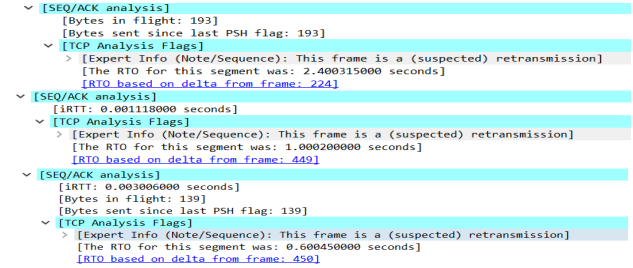
Pilih salah satu paket serangan dengan informasi yang tidak terdapat Retransmission untuk mencari adanya RTT. Letak RTT terdapat pada section SEQ/ACK.



Gbr 17. Pencarian RTT

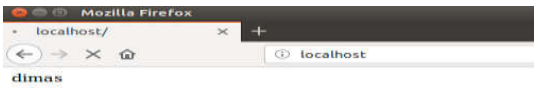
Pada gambar 17 merupakan letak RTT dengan waktu 0,001237000 detik.

Setelah itu peneliti mencoba mencari adanya RTO dengan memilih salah satu paket serangan dengan informasi yang tidak terdapat Retransmission untuk mencari adanya RTT. Letak RTT terdapat pada section SEQ/ACK.



Gbr 18. Pencarian RTO

Pada gambar diatas telah ditemukan adanya beberapa RTO dengan waktu delay yang beragam, menandakan serangan telah sukses dilakukan sehingga server yang diserang meminta waktu delay. Dan jika dihitung rata-ratanya mencapai 1,333655 detik. Lalu membuka lagi website yang diserang pada Ubuntu.

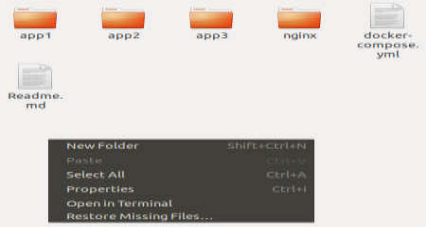


Gbr 19. Tampilan Website yang diserang pada Ubuntu

Selanjutnya peneliti memuat ulang halaman, ketika halaman dimuat ulang terjadi perlambatan request. Hal tersebut mengindikasikan bahwa web server telah mengalami penyerangan Slowloris.

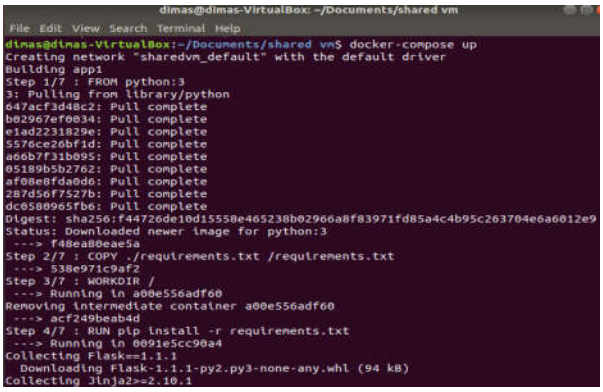
B. Serangan dengan menggunakan Load Balancing

Langkah pertama buka Ubuntu pada VM untuk masuk pada folder load balancing.



Gbr 20. Tampilan Web pada folder load balancing

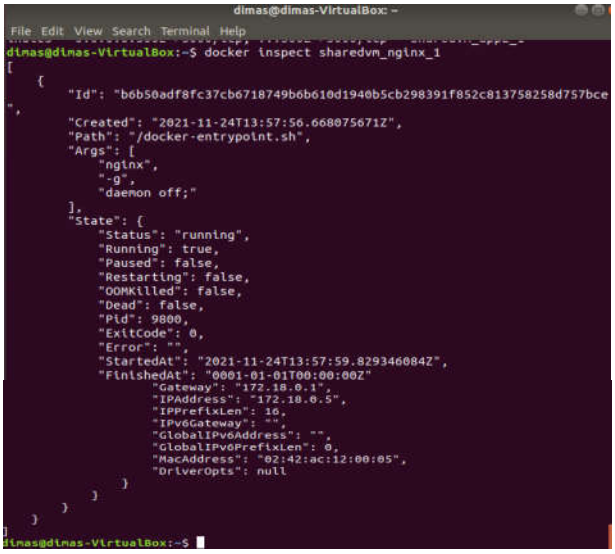
Lalu penulis membukanya pada terminal seperti pada gambar diatas. Kemudian masukkan perintah "docker-compose up" pada terminal.



Gbr 21. Perintah docker-compose up

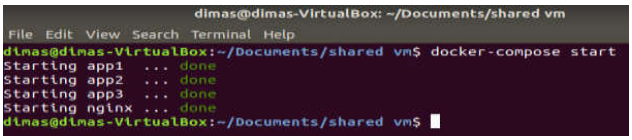
Perintah docker-compose up digunakan untuk mengaktifkan semua image yang telah dibangun.

Lalu peneliti melakukan perintah pada terminal "docker inspect sharedvm_nginx_1".



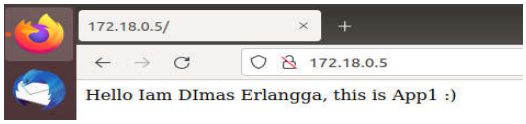
Gbr 22. Perintah docker inspect sharedvm_nginx_1

Perintah docker inspect untuk melihat secara detail properti entitas docker pada container sharedvm_nginx_1. Selanjutnya peneliti melakukan perintah "docker-compose start".



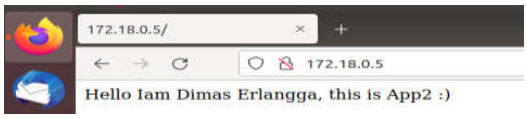
Gbr 23. Perintah docker-compose start

Lalu peneliti mengecek kembali website dengan IP lokal :172.18.0.5 untuk melihat server App1 dan App2.



Gbr 24. Tampilan web pada server pertama

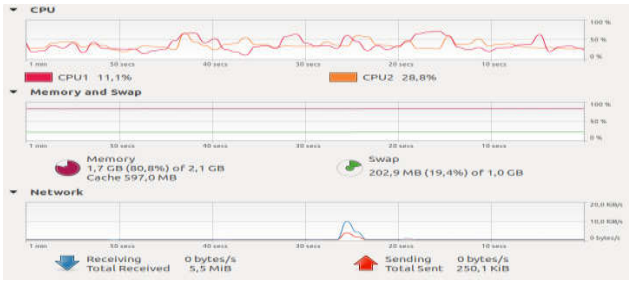
Gambar diatas merupakan tampilan pada salah satu web server yang akan coba diserang, disini diberi nama App1. Setelah itu lakukan refresh website untuk mengganti server yang lainnya.



Gbr 25. Tampilan web pada server terakhir

Setelah dicek tenyata sudah berganti, Ini merupakan tampilan pada salah satu web server yang akan coba diserang, disini diberi nama App2. Untuk mengganti lagi pada server lainnya lakukan refresh lagi.

Perintah docker-compose start digunakan untuk memulai folder image docker, yaitu image load-balancing-master. Lalu buka system monitor dalam Ubuntu untuk menganalisis resource.



Gbr 26. Tampilan resource sebelum dilakukan serangan

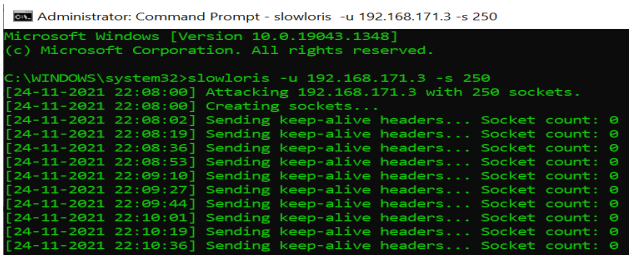
Pada gambar 26 merupakan tampilan resource pada Ubuntu sebelum dilakukan serangan. CPU1 11,1% dan CPU2 28,8%.

Selanjutnya peneliti juga membuka Wireshark untuk melakukan analisis paket traffic jaringan, lalu klik VirtualBox Host-Only Network#3 pada Wireshark untuk monitoring anomali jaringan.

Gbr 27. Tampilan VB Host-Only Network 3 sebelum adanya serangan

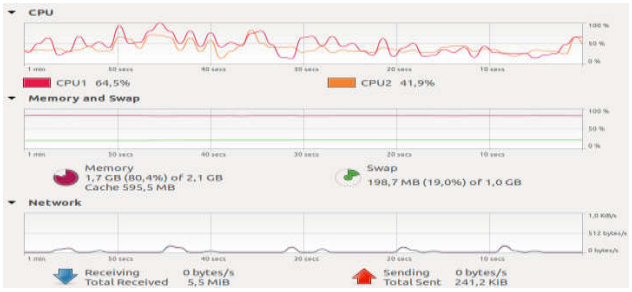
VirtualBox Host-Only Network #3 merupakan jaringan yang dikoneksikan antara PC lokal dan mesin OS dengan Web Server Load Balancing. Kemudian memilih jaringan tersebut

untuk melakukan analisa traffic. Selanjutnya peneliti melakukan serangan Slowloris dengan menuliskan kode di Command Prompt pada Ip yang telah didefinisikan pada jaringan VirtualBox Host-Only Network#3.



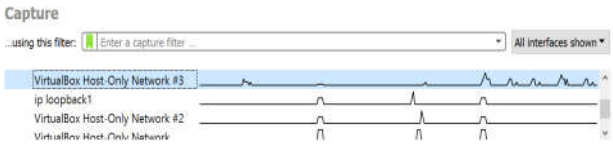
Gbr 28. Proses serangan Slowloris

Pada gambar 28 merupakan tahapan serangan Slowloris, dengan pengalamatan IP:192.168.171.3 dan mengirim 250 paket serangan pada website yang akan diserang. Lalu peneliti membuka kembali system monitor dalam Ubuntu untuk menganalisis resource.



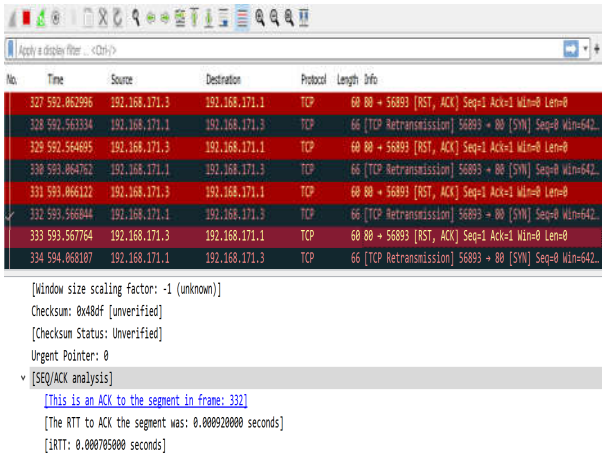
Gbr 29. Tampilan resource setelah dilakukan serangan

Pada gambar 29 merupakan tampilan resource pada Ubuntu setelah dilakukan serangan. Terlihat berbeda dengan tampilan sebelum diserang CPU1 11,1% dan CPU2 28,8% tetapi setelah diserang mengalami peningkatan menjadi 64,5% dan 41,9%. Selanjutnya peneliti membuka Wireshark untuk monitoring anomaly traffic yang terdeteksi.



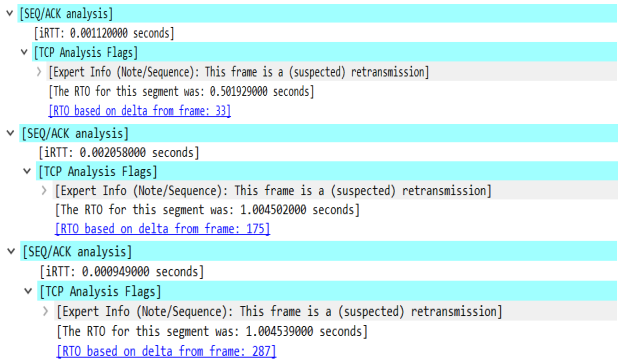
Gbr 30. Tampilan VB Host-Only Network 3 setelah adanya serangan

Berbeda dengan tampilan Wireshark yang belum diserang diagramnya hanya garis lurus, tetapi setelah diserang terdapat grafik naik turun. Setelah itu peneliti mencari apakah terdapat RTT. Pilih salah satu paket serangan dengan informasi yang tidak terdapat Retransmission untuk mencari adanya RTT. Letak RTT terdapat pada section SEQ/ACK.



Gbr 31. Mencari letak RTT

Pada gambar 31 merupakan letak RTT dengan waktu 0,000920000 detik. Setelah itu peneliti mencoba mencari adanya RTO dengan memilih salah satu paket serangan dengan informasi yang tidak terdapat Retransmission untuk mencari adanya RTT. Letak RTT terdapat pada section SEQ/ACK.



Gbr 32. Mencari letak RTO

Pada gambar diatas telah ditemukan adanya beberapa RTO dengan waktu delay yang beragam, menandakan serangan telah sukses dilakukan sehingga server yang diserang meminta waktu delay dan juga jika dihitung rata-rata waktu delaynya 0,83699 detik waktu delay. Lalu membuka lagi website yang diserang pada Ubuntu.

Gbr 33. Tampilan web server APP1 dan APP2 setelah diserang

Selanjutnya peneliti memuat ulang halaman, ketika halaman dimuat ulang tidak terjadi perlambatan request, berbeda dengan serangan tanpa menggunakan Load Balancing yang mengalami perlambatan request. Hal tersebut mengindikasikan bahwa web server yang mengalami penyerangan Slowloris, serangannya telah berhasil diminimalisir oleh Load Balancing.

IV. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, peneliti berhasil melakukan penyerangan pada jaringan komputer dengan metode Denial Of Service beserta cara meminimalisir serangan tersebut. Pada serangan baik tanpa menggunakan Load Balancing dan yang menggunakan Load Balancing keduanya menggunakan acuan ukuran Retransmission Timeout (RTO) dan Resource pada Ubuntu. Jika tanpa menggunakan Load Balancing terdapat RTO yang telah dihitung rata-ratanya mencapai 1,333655 detik sedangkan dengan menggunakan Load Balancing terdapat RTO yang telah dihitung rata-ratanya mencapai waktu delay 0,83699 detik. Hal ini menunjukkan bahwa RTO berhasil diminimalisir dengan menggunakan Load Balancing jika dibandingkan dengan tanpa Load Balancing, karena semakin banyak RTO maka kualitas TCP kurang baik.

V. SARAN

Berdasarkan hasil yang telah diperoleh, peneliti memberikan saran ke depannya untuk meminimalisir serangan Denial Of Service berjenis Slowloris pada web server level production dapat menggunakan Load Balancing untuk mendistribusikan traffic ke beberapa server sehingga dapat mengurangi waktu delay pada RTO. Mekanisme pengantisipasian ini masih berada pada layer aplikasi kemudian juga dapat memberikan proteksi tambahan pada layer network.

UCAPAN TERIMA KASIH

Penulis senantiasa mengucap rasa syukur yang sangat besar kepada Tuhan YME atas segala berkah, rahmat dan juga pertolonganNya, sehingga penulis mampu menyelesaikan proyek dan artikel ilmiah ini dengan baik, Terimakasih penulis ucapkan juga kepada kedua Orang tua yang selalu memberi semangat dan juga dukungan, Dosen Pembimbing Skripsi yang selalu memberikan masukan dan saran yang membangun kepada penulis, sahabat dan teman yang selalu memberikan dorongan dan dukungan dalam melakukan penelitian. Terimakasih kepada diri sendiri karena dapat berkompromi dan juga menjaga komitmen untuk menggapai tujuan yang ingin dicapai.

REFERENSI

[1] S. E. Prasetyo, "Design and Implementation of Lightweight Virtualization Using Docker Container in Distributing Web Application with Experimental," *JITE (Journal of Informatics and Telecomunication Engineering),* pp. 270-276, 23 December 2021.

[2] A. P. Safira, "goldenfast," 23 December 2020. [Online]. Available: https://www.goldenfast.net/blog/pengertian-load-balancing/. [Accessed 28 April 2021].

[3] Waryanto, "niagahoster," 22 January 2021. [Online]. Available: https://www.niagahoster.co.id/blog/pengertian-website/. [Accessed 28 April 2021].

[4] M. R. Adani, "sekawanmedia," 3 February 2021. [Online]. Available: https://www.sekawanmedia.co.id/belajar-docker/. [Accessed 29 April 2021].

[5] A. Y. Chandra, "Analisis Performansi Antara Apache & Nginx Web Server dalam Menangani Client Request," *JURNAL SISTEM DAN INFORMATIKA (JSI),* vol. 14, pp. 2460-3732, 28 November 2019.

[6] A. F, "Konsep DoS dan DDoS (Distributed Denial of Service), serta Mekanisme Serangan DoS dan DDoS dan cara penanggulangannya," *STMIK Dipanegara Makassar,* 2015.

[7] T. Chakraborty, "Docker adn Google Kubernetics," *International Journal of Research Studies in Computer Science and Engineering (IJRSCSE),* vol. 5, no. 4, pp. 24-35, 2018.

[8] Douglas Kunda, Sipiwe Chihana, Sinyida Muwanei, "Web Server Performance of Apache and Nginx: A Systematic Literature Review," *School of Science Engineering and Technology, Mulungushi University, PO box 80415 kabwe, Zambia,* vol. l8, pp. 2222-2863, 2017.

[9] A. Prakash, M.Satish, T.Sri Sai Bhargav, Dr. N. Bhalaji, "Detection and Mitigation of Denial of Sevice Attack Using," *4th International Conference on Recent Trends in Computer Science & Engineering,* pp. 275-280, 2016.

[10] Finandhito Adhana, I Ketut Gede Suhartana, "Detection of Denial of Service on Website With Wireshark," *Jurnal Elektronik Ilmu Komputer Udayana,* vol. 8, pp. 2301-5373, 2020.

[11] Vinicius da Silva Faria, Jéssica Alcântara Gonçalves, Camilla Alves Mariano da Silva, Gabriele de Brito Vieira dan Dalbert Matos Mascarenhas, "SDToW: A Slowloris Detecting Tool WMNs," *Centro Federal de Educação Tecnológica Celso Suckow da Foseca–CEFET/RJ, Petrópolis 25600-000, Brasil;,* p. 544, 2020.

[12] Ankur Rana, Jaspreet Srivastava, Mayur Srivastava, "DENIAL OF SERVICE ATTACKS AND COUNTER MEASURES," *Asstt. Prof, CSE, Quantum School of Technology, Roorkee-India Research Scholar, Uttrakhand Technical University, Dehradun,* vol. 4, no. 10, pp. 2393-8374, 2017.

[13] Jeeva Chelladhurai, Pethuru Raj Chelliah, Sathish Alampalayam Kumar, "Securing Docker Containers from Denial of Service," *Conference Paper,* 2016.

[14] D. Indonesia, "Cara Validasi Keandalan Arsitektur Cloud: Teknik AWS," 29 May 2021. [Online]. Available: https://www.dicoding.com/blog/validasi-keandalan-arsitektur-cloud/. [Accessed 01 10 2021].

[15] J. M. Garrido, "Models of Multi-Server Systems," *Object Oriented Simulation,* pp. 281-295, 2009.

[16] A. C., "Hostinger Tutorial," Glosarium, 03 September 2021. [Online]. Available: https://www.hostinger.co.id/tutorial/apa-itu-nginx. [Accessed 27 August 2021].

[17] Dewawebteam, "Dewaweb," 28 August 2021. [Online]. Available: https://www.dewaweb.com/blog/berkenalan-dengan-ubuntu/. [Accessed 10 10 2021].

[18] M. Nathalia, "Analisis Unjuk Kerja TCP Reno Di Jaringan Single Hop Wireless Link," 2016.