IMPLEMENTASI SISTEM KEAMANAN *WEB SERVER* MENGGUNAKAN PFSENSE

Molavi Arman[1], Nur Rachmat[2]
[1]Program Studi Manajemen Informatika, AMIK MDP, Palembang.
[2]Program Studi Teknik Komputer, AMIK MDP, Palembang.
e-mail: *[1]molavi.arman@mdp.ac.id, [2]nur.rachmat@mdp.ac.id

Abstrak

Seorang network administrator haruslah memiliki amanat dan rasa tanggung jawab terhadap network tempat dia bekerja dan mengamankan sumber daya perangkat device network sebagaimana mestinya. Kampus adalah salah satu tempat dosen, mahasiswa dan karyawan sebagai pengguna internet dan sistem informasi akademik kampus yang berbasis web. Pengguna atau user bisa menggunakan fasilitas dengan benar dan bisa pula menyalahgunakan ataupun tanpa sengaja mengganggu traffic network dan layanan sistem informasi pada kampus. Oleh karena itu, diperlukan suatu sistem dalam mengatasi dan menanggulangi penyalahgunaan kecerobohan sengaja atau tidak sengaja berupa ancaman terhadap web server. Sistem yang mendeteksi gangguan keamanan ini diimplementasikan dengan menggunakan aplikasi snort pada pfsense dan pfsense sebagai sistem operasi router yang diletakkan berhadapan dengan internet. Metode penelitian menggunakan PPDIOO sebagai metode pengembangan dalam implementasi. Berdasarkan hasil percobaan sistem snort mampu mengetahui sebagai alert dan menyimpannya di dalam log seperti serangan ping of death dan slowloris. Berdasarkan log snort pfsense mampu melakukan pemblokiran otomatis dalam durasi tertentu.
Kata kunci— web server, PPDIOO, Pfsense, snort

Abstract

A network administrator must be responsible for the network where he works and secure the resources of the network devices as they should be. Campus is a place where lecturers, students and employees are as users of internet and web-based campus academic information system. Users can use the facilities properly and they could also misuse or accidentally disrupt network traffic and information system services on campus. Therefore, a system to overcome the problem due to the intentional or unintentional carelessness that could be threats to the web server is needed. The system that detects security breaches is implemented by using the snort application on pfsense and pfsense as a router operating system that is placed facing the internet. The research method uses PPDIOO as a development method in implementation. The results of experiments showed the snort system is able to find out as alerts and store them in logs such as ping of death and slowloris attacks. Based on the log snort pfsense is able to do automatic blocking within a certain duration.
Keywords— web server, PPDIOO, Pfsense, snort

I. PENDAHULUAN

Jika Anda tidak khawatir tentang keamanan jaringan, Anda dapat merancang topologi jaringan terutama dengan mempertimbangkan kinerja. Nyatanya, dunia tidak seperti itu bagi kita, dan oleh karena itu kita perlu memikirkan keamanan tidak hanya dalam hal perangkat lunak dan perilaku manusia, tetapi juga dalam hal desain jaringan. Jika anda seorang network engineer profesional dan cukup beruntung di jagat raya ini adalah memiliki input untuk desain jaringan sebelum dibangun, jika anda masuk setelah jaringan ada dan berjalan, anda harus siap untuk berdebat tentang perubahan yang diperlukan untuk topologi yang diperlukan untuk mendukung langkah-langkah keamanan dan itu menyita waktu.

Seorang network engineer memiliki tanggung jawab terhadap keamanan sistem dan memastikan semua sistem berjalan dengan baik serta memonitoringnya. Institusi sekolah adalah lembaga dimana jaringan internet dan jaringan Local Area Network (LAN) digunakan dalam keseharian untuk kegiatan belajar mengajar ataupun aktifitas akademik. Dalam aktifitas tersebut bisa saja terjadi secara sengaja ataupun tidak sengaja melakukan aktifitas illegal. Hal ini mengakibatkan suatu sistem jaringan yang seharusnya digunakan sebagai pembelajaran akan tetapi disalahgunakan penggunaannya untuk aktifitas lain seperti mengakses jejaring sosial. Selain itu administrator harus mengetahui sesuatu log yang mengindentifikasi adanya serangan atau penyalahgunaan jaringan.

Pada penelitian [1] menjelaskan perbadingan firewall yang berbasis open source dengan komersial. Hal yang dibandingakan yaitu network level filtering, application level filtering dan kemampuan VPN. Penelitian ini menghasilkan misalnya firewall Linux memiliki kinerja laju transaksi yang unggul dan kemampuan penyaringan tingkat aplikasi. Firewall Cisco 10s secara fungsional unggul untuk pemfilteran tingkat jaringan, kemampuan VPN karena IPSEC, integrasi dengan lingkungan multi-protokol yang heterogen, dan dapat diskalakan dengan dukungan untuk PKI. Perlu dicatat bahwa beberapa proyek Linux dalam tahap percobaan dapat mengubah kesimpulan kami. Pada akhirnya, solusi firewall paling efektif mungkin kombinasi kedua tingkat aplikasi dan penyaringan paket tingkat jaringan.

Pada penelitian [2] menjelaskan perbandingan kinerja deteksi serangan antara dua sistem firewall terkenal, Endian dan pfSense, versi tidak berbayar dengan konfigurasi default telah diatur untuk diserang dari jaringan luar jaringan lokal. Skenario serangan termasuk pemindaian port, ping of death, flooding, dan password attack dalam kondisi yang berbeda. Hasil penelitian menunjukkan bahwa kinerja pfSense secara keseluruhan lebih baik. Pengaturan dasar Endian dapat mendeteksi lebih banyak kategori serangan tetapi lebih sedikit kejadian. Oleh karena itu, Endian bisa cocok untuk non-komersial, rumah, perusahaan kecil dengan jaringan dan data yang kurang penting. Di sisi lain, pfSense dapat digunakan dalam beragam organisasi, dengan beberapa modifikasi konfigurasi.

Pada penelitian [3] menjelaskan menyarankan sejumlah teknik optimasi keamanan jaringan yang akan berfungsi untuk meningkatkan kualitas pengalaman bagi pengguna internet, keamanan jaringan dan bagaimana menerapkan *Firewall* dan *Intrusion Detection System*. *Firewall* mengontrol masuk dan lalu lintas jaringan keluar. Mereka dapat mengizinkan paket tertentu untuk melewatinya atau menonaktifkan akses untuknya. Tetapi sebuah organisasi tidak dapat sepenuhnya bergantung pada *firewall* dan tidak ada sistem perlindungan yang dapat membuat

jaringan sepenuhnya aman terhadap serangan. Jika keamanan jaringan dilanggar, maka itu harus dilaporkan kepada administrator sehingga tindakan yang diperlukan dapat diambil. IDS / IPS membantu secara paralel dengan *firewall* untuk meningkatkan keamanan jaringan. pfSense adalah salah satu platform open source yang muncul yang menyediakan semua layanan ini. Instalasi dan konfigurasinya sederhana dan hemat biaya. Sangat direkomendasikan untuk usaha kecil dan menengah karena menyediakan daftar layanan yang luas, menjaga integritas dan keamanan jaringan di biaya sangat kurang.

Pada penelitian [4] tujuan serangan DoS adalah untuk membuat sumber daya komputer, yaitu *bandwidth* jaringan, waktu CPU, *website* pengguna dan lain-lain, tidak tersedia, overload dengan lalu lintas atau sangat lambat untuk pengguna. Pengguna cloud tidak akan dapat mengakses layanan. Di sisi lain, dalam serangan DoS, seorang penyerang mencoba untuk mempengaruhi browser dengan menyuntikkan kode berbahaya ke dalamnya. Browser yang terinfeksi akan mencoba untuk membuka banyak jendela sehingga pengguna yang sah tidak akan dapat mengakses layanan. Selain itu, penyerang mencoba membebani sistem cloud yang ditargetkan dengan permintaan layanan yang luar biasa. Akibatnya, sistem cloud yang ditargetkan berhenti merespons sama sekali untuk setiap permintaan dan sumber daya akan keluar dari layanan. Serangan DDoS memiliki serangan yang jauh lebih buruk daripada serangan DoS. Serangan DoS diluncurkan sebagian besar kasus dari satu mesin. Serangan DDoS diluncurkan dari banyak mesin. Mesin ini biasanya tidak dimiliki oleh penyerang, mereka semua dibajak oleh malware. Hasilnya akan menjadi jaringan dari kelompok mesin, yaitu botnet, yang akan digunakan oleh penyerang DDoS untuk melancarkan serangan. Sebagai serangan dapat didistribusikan melalui beberapa mesin, itu akan sangat sulit untuk membedakan pengguna otentik dari penyerang.

Pfsense adalah distribusi *firewall network* yang bebas, berdasarkan pada sistem operasi FreeBSD dengan kernel khusus dan termasuk paket perangkat lunak bebas pihak ketiga untuk fungsionalitas tambahan. Perangkat lunak pfSense, dengan bantuan sistem paket, mampu menyediakan fungsionalitas yang sama dengan *firewall* komersial. Penelitian ini menggunakan Pfsense sebagai penghubung koneksi dan pengamanan serta perlindungan terhadap *web server*.

Berdasarkan penelitian terdahulu dampak serangan DDoS dan DoS cukup membebani dan sangat menggangu kinerja sistem dan aktifitas pengguna. *firewall* atau router sebagai penghubung internet atau jaringan luar ke jaringan internal yang menuju sumber daya *server* harus melakukan filterisasi dan tindakan pencegahan agar serangan-serangan tersebut tidak berdampak besar pada sistem yang berjalan. Penelitian ini akan menggunakan firewall pfsense beserta komponennya guna menanggulangi serangan DDoS ataupun DoS.

II. TINJAUAN PUSTAKA

2.1 Serangan DDoS Pada Layer Aplikasi

Serangan DDoS yang diluncurkan pada lapisan aplikasi menimbulkan tantangan untuk dideteksi karena paket permintaan tampak mirip dengan paket permintaan normal [5] [6] [7] [8]. Konfigurasi dan fungsi yang terkait dengan aplikasi dapat menyebabkan serangan DDoS pada lapisan aplikasi [9], dan konsekuensi dari serangan ini dapat menghabiskan sumber daya, seperti *bandwidth* jaringan, pemrosesan CPU, dan memori [10]. [11] menjelaskan bahwa

serangan HTTP DDoS terjadi ketika permintaan HTTP yang sah dimulai dalam jumlah besar.

Serangan DDOS yang diluncurkan pada lapisan aplikasi membutuhkan bandwidth yang lebih rendah untuk mencegah pengguna yang sah menjelajahi server web, selain meniru lalu lintas yang dekat dengan lalu lintas asli [12]. Tiga faktor yang membuat deteksi DDOS sulit pada lapisan aplikasi adalah sebagai berikut [13] : (1) ketidakjelasan, protokol HTTP menggunakan koneksi *Transmission Control Protocol* (TCP) dan *User Datagram Protocol* (UDP) untuk menjalankan operasinya, karenanya kerumitan untuk membedakan lalu lintas yang sah dari yang tidak sah; (2) efisiensi, serangan HTTP DDOS hanya membutuhkan koneksi yang lebih sedikit untuk memulai serangan DDOS; dan (3) mematikan, kemampuan serangan untuk membanjiri server web dengan segera, sehingga mengakibatkan gangguan layanan terlepas dari jenis perangkat keras dan kinerjanya.

2.2 Jenis Serangan DDOS di Lapisan Aplikasi

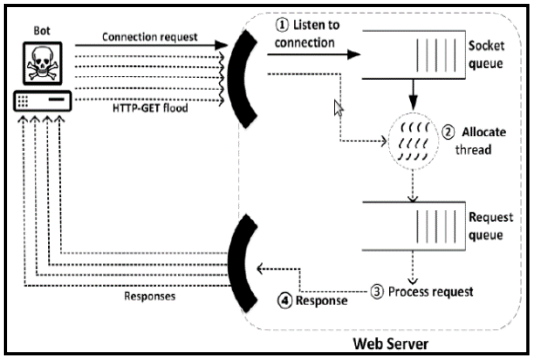
Serangan DDOS pada lapisan aplikasi berdasarkan kategori berikut [5] [14] :

1. *Session Flooding Attack*, sumber daya *server* menjadi habis ketika tingkat permintaan sesi lebih tinggi dari pengguna yang valid. Aktivitas jahat ini dapat mengakibatkan serangan *flooding* DDOS, misalnya serangan *flooding* HTTP GET / *POST*. Dalam menjalankan serangan ini, penyerang membutuhkan permintaan HTTP otentik yang besar, dan biasanya, botnet digunakan sebagai kemampuannya untuk menghasilkan permintaan yang valid, biasanya melebihi 10 permintaan dalam satu detik. Serangan ini hanya membutuhkan botnet untuk berhasil memulai serangan.
2. *Request Flooding Attack*, serangan ini terjadi ketika seorang penyerang memulai sejumlah permintaan yang besar dalam satu sesi. Permintaan ini lebih besar dari permintaan pengguna yang valid. Sesi HTTP *GET* / *POST* adalah turunan dari serangan dalam kategori ini yang memanfaatkan fitur HTTP 1.1, yang memungkinkan lebih dari satu permintaan dalam satu sesi HTTP. Struktur HTTP 1.1 memungkinkan penyerang untuk membatasi tingkat sesi HTTP. Penggunaan HTTP 1.1 juga menyebabkan penyerang memotong mekanisme pertahanan tingkat sesi sejumlah sistem keamanan. [15] mengklaim bahwa botnet digunakan untuk serangan ini. Botnet dirancang untuk memiliki struktur perintah dan kontrol yang memungkinkan pengganggu *cyber* untuk mengeluarkan perintah ke mesin botnet. Mesin yang terkena dampak dikenal sebagai *botnet* terdaftar di *server* perintah dan kontrol saat mereka memberikan instruksi kepada *botnet* untuk mengirim *flood* HTTP *GET*. Serangan ini dapat menghabiskan sumber daya *server* karena botnet mengirimkan banyak permintaan flood HTTP *GET* ke *server*.
3. *Asymmetric attack*, penyerang mengirim sesi yang berisi jumlah permintaan beban kerja tinggi yang lebih besar. Tujuan utama penyerang adalah melahap sumber daya seperti CPU, memori server dan menurunkannya.
4. *Slow* HTTP *attack,* serangan *slow* HTTP mengeksploitasi metode kerja protokol HTTP, yang mengharuskan setiap permintaan dari klien diterima sepenuhnya oleh *server* sebelum diproses. Jika permintaan HTTP tidak lengkap, atau jika kecepatan transfer

sangat rendah, *server* tetap sibuk menunggu sisa data. Jika *server* menyimpan terlalu banyak sumber daya sibuk, maka ini membuat penolakan layanan [16].

2.3 Arsitektur *web server* yang Rentan

Desain operasional yang melekat pada server membuat rentan terhadap *Get flood attack*. Tipe Arsitektur *web server* di ilustrasikan pada gambar 1. Seorang client memulai proses koneksi dengan mengirimkan permintaan ke *server*. Antrian socket menampung semua permintaan koneksi hingga *threads* khusus dialokasikan untuk menangani permintaan tersebut. Client mengirimkan layanan (GET request) ke server hanya setelah koneksi TCP dibuat. Semua permintaan layanan ini diakumulasikan dalam antrian permintaan tempat penjadwal selanjutnya memproses dan merespon permintaan individu. Seperti halnya pengguna yang sah lainnya, *bot* pada awalnya juga membuat koneksi otentik untuk berkomunikasi dengan server. *Bot* kemudian mengirimkan sejumlah besar permintaan *HTTP GET* ke *server* dan menunggu tanggapan seperti pengguna yang sah. Selama *HTTP-GET flood attack*, permintaan dari *bot* dengan cepat terakumulasi dalam antrian permintaan, yang mengarah untuk menjatuhkan permintaan masuk berikutnya yang dikirim oleh client yang sah. Tantangan utama untuk server disini adalah mengklasifikasikan pencetus permintaan ini sebagai bot atau manusia. Untuk memperburuk situasi, bot bahkan mencoba untuk meniru perilaku akses pengguna yang jika berhasil menentang logika dibalik berbagai teknik deteksi serangan yang modern. Server terus memproses dan menanggapi permintaan yang diterima dari bot karena dianggap sah. Penyerang membuat *web server* terlibat dalam pemrosesan terus menerus dengan menghasilkan permintaan pada tingkat diatas normal, sehingga menurunkan kualitas layanan keseluruhan yang dikirim ke pengguna yang sah [9].



Gambar 1. HTTP-*GET flood DDoS attack* terhadap arsitektur *web server* [9]

2.4 Strategi Serangan DDOS di Lapisan Aplikasi

Serangan HTTP DDoS dikelompokkan menjadi beberapa subkelas [9], sebagai berikut :

1. *Server Load*, penyerang menggunakan *botnet* untuk secara terus menerus mengirimkan permintaan berbahaya ke *web server* secara agresif, sehingga menyebabkan server untuk membatalkan permintaan yang sah ketika sumber daya *web server* habis.
2. *Icreasing*, Penyerang menggunakan nilai permintaan rendah untuk memulai serangan dan perlahan-lahan meningkatkan nilai. Perilaku serangan ini sulit dideteksi karena lalu lintas HTTP y.ang jahat tidak mengirim permintaa secara agresif ke server selama seragan.
3. *Constant*, *cyber intruders* harus menentukan sejumlah tingkat permintaan yang akan dikirim ke *web server* HTTP korban. Nomor permintaan disebut konstan karena memiliki nomor yang sama seperti ketika botnet mengirimkan permintaan jahat ke *web server*.
4. *Target Web Page*, serangan HTTP DDoS terjadi pada satu dan beberapa

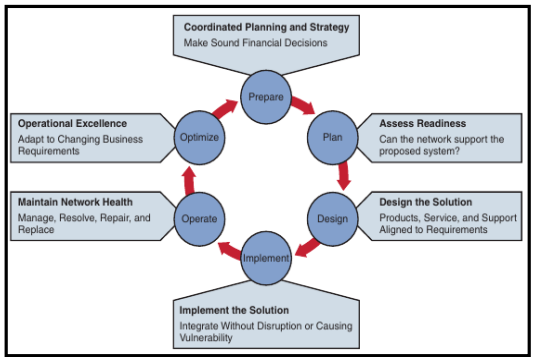
halaman *web* , dimana penyerang meniru pola seseorang pengguna mengakses untuk menipu deteksi serangan. Pada halaman itu diakses oleh *botnet* yang meniru manusia dengan pola akses.

5. *Single Web Page Attack*, penyerang menggunakan satu halaman *web* milik situs *web*. *Botnet* terus mengirim permintaan HTTP berbahaya ke sweb server
6. *Main Page Attack*, cyber intruders secara khusus focus halaman utama situs web untuk menolak akses di antara meniru pengguna . Botnet traffic digunakan untuk mengirim berulang kali permintaan jahat ke server HTTP. Dampaknya ini serangan hanya terjadi pada halaman utama situs web, sedangkan subhalaman situs web tidak terpengaruh.
7. *Dominant Page Attack*, cyber intruder menemukan halaman web yang dicara oleh pengguna yang sah untuk akses. Penyerang kemudian berfokus pada halaman itu untuk mengeksekusi serangan HTTP DDoS untuk mencegah pengguna sah mengakses minat yang lebih besar dari halaman web. Serangan ini hanyak mempengaruhi halaman web dengan minat lebih besar bagi pengguna untuk menjelajah.
8. *Multiple Page Attack*, seorang penyerang mulai menyerang beberapa halaman web dari suatu situs web. Teknik ini menghindari deteksi karena permintaan HTTP jahat meniru pola akses manusia. Misalnya, seorang manusia akan membuka lebih dari satu halaman web untuk mencari informasi saat menjelajahi situs web. Selama serangan ini, halaman yang ada disitus web tidak akan dapat diakses karena penyerang menargetkan beberapa halaman web.
9. *Reply Flood Attack*, perintah *botnet* oleh penyerang mengirimkan lalu lintas HTTP pada tingkat yang meningkat untuk mendapatkan sumber daya *server web* untuk mencegah *server web* untuk menjelajahi permintaan HTTP yang sah. Serangan bekerja dengan mendapatkan pola akses manusia untuk melarang sistem deteksi dari memblokir permintaan jahat.
10. *Rare Change Page Attack*, Struktur umum sistem web akan mengelompokkan halaman ke dalam grup tertentu untuk membuat konten halaman lebih terstruktur dan mudah digunakan. Karena susunan halaman *web* dikelompokkan, penyerang dapat kompromi halaman grup dengan memerintah botnetthat halaman web. Karena halaman web dirancang dalam grup, halaman menjadi grup yang paling bertarget. Serangan ini mencegah pengguna membuka halaman web milik ke grup tertentu..
11. *Frequent Change Attack*, penyerang melakukan serangan ke halaman *web* yang termasuk dalam kategori yang berbeda. Serangan ini akan memutar permintaan berbahaya dikirim ke kategori halaman web yang berbeda. Serangan ini hanya mempengaruhi kategori tertentu dari situs *web*, karena halaman *web* lainnya masih dapat diakses selama serangan.
12. *Hot Pages Attack*, setiap sistem berbasis *web* akan memiliki halaman terbuka .Oleh karena itu, pengganggu cyber menyerang halaman yang paling banyak dikunjungi untuk mencegah pengguna yang sah mengakses, karena tujuan utama serangan DDoS adalah untuk menghindari pengguna membuka halaman.
13. *Web Proxy Attack*, penyerang menggunakan *server proxy* sebagai mediator untuk menghasilkan *traffic*

serangan. Pengguna *proxy server* untuk menghasilkan lalu lintas serangan kesulitan dalam mengenali mendeteksi serangan sumber. Beberapa *server proxy* digunakan untuk menghasilkan banyak permintaan HTTP untuk membanjiri *web server*.

III. METODOLOGI PENELITIAN

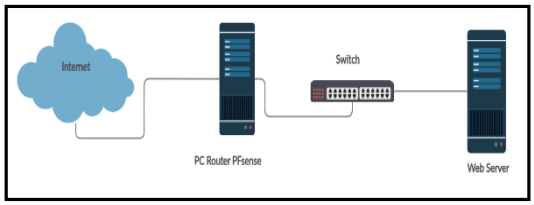
3.1 Metodologi Pendekatan PPDIOO *Network Life Cycle*



Gambar 2. PPDIOO *Network Life Cycle* [17]

PPDIOO *network life cycle* adalah metodologi pengembangan yang dibuat oleh CISCO [17]. Pendekatan siklus hidup jaringan PPDIOO mencerminkan fase *lifecycle* dari jaringan standar. Fase PPDIOO adalah sebagai berikut :

1. *Prepare* : Tahap *Prepare* melibatkan penetapan persyaratan, mengembangkan strategi jaringan, mengusulkan arsitektur konseptual tingkat tinggi, dan mengidentifikasi teknologi yang dapat mendukung arsitektur secara terbaik. Fase persiapan dapat menetapkan justifikasi keuangan untuk strategi jaringan dengan menilai kasus bisnis untuk arsitektur yang diusulkan. Pada tingkatan ini mempersiapkan *hardware* seperti 2 buah *personal computer* (PC), salah satu pc dijadikan sebagai *router* pfsense menggunakan 2 buah ethernet (kartu jaringan) sebagai jalur internet dan menghubungkan jalur koneksi LAN menuju *web server*. Mempersiapkan *switch* 8 *port* dan kabel jaringan dengan tipe kategori 7. Pada perangkat lunak disiapkan sistem operasi *router* yaitu pfsense dan *web server* menggunakan system operasi *Linux* Debian 10.
2. *plan* : tahap *plan* melibatkan identifikasi persyaratan jaringan awal berdasarkan tujuan, fasilitas, kebutuhan pengguna, dan sebagainya. Fase rencana melibatkan karakterisasi lokasi, penilaian jaringan yang ada, dan melakukan analisis kesenjangan untuk menentukan apakah infrastruktur sistem, lokasi, dan lingkungan operasional yang ada dapat mendukung sistem yang diusulkan. Sebuah rencana proyek memfasilitasi pengelolaan tugas, tanggung jawab, tonggak penting, dan sumber daya yang diperlukan untuk mengimplementasikan perubahan pada jaringan. Rencana proyek harus sejajar dengan ruang lingkup, biaya, dan parameter sumber daya yang ditetapkan dalam persyaratan bisnis asli. Pada tahapan *plan* ini akan menggunakan koneksi internet yaitu kabel *fiber optic* menggunakan jasa salah satu ISP, kemudian lokasi infrastruktur *hardware* akan diletakkan dalam ruangan 2.5m x 2.5m dengan AC 2 buah masing-masing 2pk dan 1 buah UPS.
3. *Design* : tahap *design* merupakan persyaratan awal yang diturunkan dalam fase rencana mendorong aktivitas spesialis desain jaringan. Spesifikasi desain jaringan adalah desain komprehensif dan terperinci yang memenuhi persyaratan bisnis dan teknis saat ini dan menggabungkan spesifikasi untuk mendukung ketersediaan, keandalan, keamanan, skalabilitas, dan kinerja. Spesifikasi desain adalah dasar untuk kegiatan implementasi. Pada tahan *design* ini digambarkan dalam sebuah topologi pada gambar 3.



Gambar 3. Topologi Jaringan

4. *Implement* : tahap *implement* yang dilakukan membangun topologi yang sudah di *design* , instalasi pfsense pada PC *router* dan melakukan konfigurasi yaitu mengaktifkan fitur *snort* dan autoblocking terhadap jenis-jenis serangan berdasarkan kategori yang dikenali *snort*. Implement ini dilakukan berulang-ulang penguijian serangan untuk melihat bahwa serangan tersebut tidak menggangu sumber daya pada *web server*. Konfigurasi dilakukan terus menerus sehingga ditemukan konfigurasi yang ideal untuk menambah akselerasi kecepatan akses pada *network* yang digunakan.
5. *Operate* : tahan *operate* ini melibatkan pemeliharaan *network* dan sistem pada pfsense. Pada pfsense dalam direktori */var* dikarenakan ada *update* pada *snort* setiap harinya maka direktori tersebut akan penuh , maka harus diperhatikan jangan sampai persentase mendekati 100%, untuk itu sistem pfsense harus di-*restart* secara berkala. Memonitor sistem kerja pfsense terlihat grafik *cpu*, memori , hdd dan total *traffic* pada network yang sedang berjalan.
6. *Optimize* : tahap *optimize* harus melibatkan manajemen yang proaktif. Tujuan dari manajemen proaktif adalah untuk mengidentifikasi dan menyelesaikan masalah sebelum mereka memberikan informasi pada organisasi.. Dalam PPDIOO proses, fase optimisasi dapat mendorong perancangan ulang jaringan , jika terlalu banyak masalah dan kesalahan jaringan muncul, jika kinerja tidak memenuhi harapan, atau jika aplikasi baru diidentifikasi untuk mendukung persyaratan organisasi dan teknis.

Metodologi penelitian merupakan uraian tentang metode metode yang digunakan dalam penelitian contoh nya metode pengumpulan data, metode pengembangan sistem, metode pengujian sistem, metode analisis dan desin sistem

IV. HASIL DAN PEMBAHASAN

4.1 Pembahasan

1. Pengujian Awal

Pada pengujian awal dilakukan dengan tidak menggunakan pfsense terhadap web server dan mencoba penyerangan web server tersebut. Hasil yang didapat adalah yaitu:

1. Ketika pengujian serangan, komputer korban tidak mengetahui ada penyerang karena tidak tidak terpasang *snort* pada pfsense.
2. Pada komputer korban tidak diketahui log atau bukti yang dialami adalah computer tersebut kehabisan sumber daya sehingga terbebani dan tidak mampu diakses lagi.

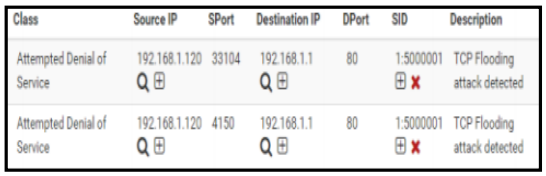
2. Pengujian Akhir Jaringan

Pada pengujian akhir komputer akan terpasang sistem pfsense dan aplikasi *snort*. Sistem pfsense dan *snort* akan dicoba dengan percobaan serangan dan akan menampilkan *alert* sesuai ancamannya dan menampilkan penyalahgunaan jaringan yang terdeteksi.

Tahapan pengujian :

1) LOIC **(***Low Ordbit Ion Cannon*)

PC *router* pfsense dan *snort* akan dicoba dengan metode penyerangan LOIC dari komputer penyerang. Hasil yang terjadi pada saat komputer diserang adalah sebagai berikut:

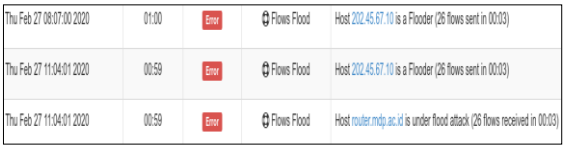


Gambar 4. Serangan LOIC terdeteksi

Gambar 4 menunjukkan IP 192.168.0.120 (komputer penyerang) melakukan *ping of death* terhadap IP 192.168.1.1 (komputer yang terpasang *snort* pfsense) dengan *description* "TCP *flooding attack detected*".

2) *SlowLoris Attack*

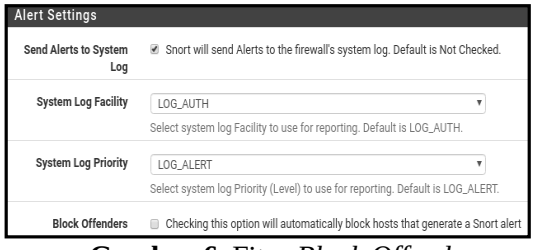
PC *router* pfsense dan *snort* dicoba dengan penyerangan *SlowLoris attack*. Hasil yang didapat adalah terlihat pada log adalah sebagai berikut :



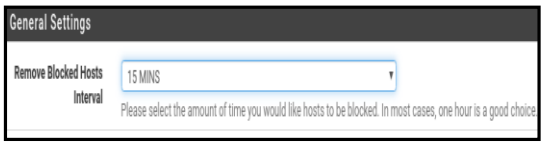
Gambar 5. Log penyerang pada protokol HTTP yang terdeteksi

Pada gambar 5 terdeteksi serangan dari ip address penyerang yang terdeteksi *snort* atas log inilah pfsense akan melakukan blokir selama beberapa waktu yang telah ditentukan pada konfigurasi.

3) Pada pfsense diperlukan konfigurasi pada menu settings interface *snort* yang harus diaktifkan adalah *check list* pada *Block Offenders.*



Gambar 6. Fitur *Block Offenders*

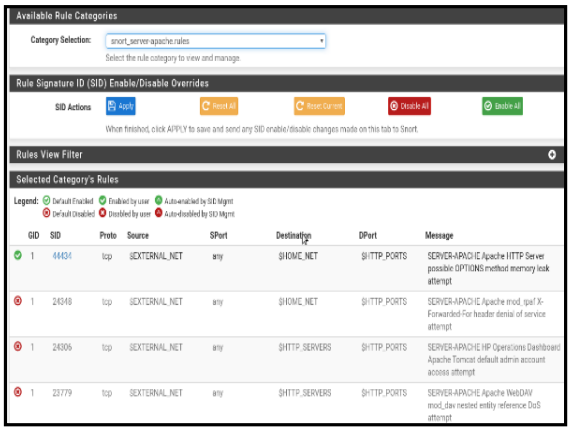


Gambar 7. Durasi berapa lama pemblokiran

Pada gambar 7 perlu diberlakukan *check list* pada *general settings* pada bagian *Remove Blocked Hosts Interval*. Ini akan menyebabkan berapa lama pemblokiran pada ip *address* yang melakukan penyerangan.

4) *Rule* Pada Pfsense

Didalam *snort* pfsense terdapat banyak aturan yang perlu diaktifkan, untuk melakukan konfigurasi masuk ke menu *snort interface* , pilih bagian *action* dan edit, akan terlihat beberapa kategori yang ingin diaktifkan guna mendeteksi serangan yang menyerang *web server* terlihat pada gambar 8.



Gambar 8. Menu *Rule* pada *snort* pfsense

4.2 Hasil

Berdasarkan hasil pembahasan maka hasil yang didapat adalah sebagai berikut :

1. Bahwa *snort* mampu mengenali jenis serangan pada *port* tcp dan DDoS dengan jenis serangan *slowloris*.
2. Setiap serangan yang dikenali akan tersimpan di *log snort* dengan *log* tersebut pfsense mengambil tindakan untuk melakukan pemblokiran dalam durasi yang telah ditetapkan.

V. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, ada beberapa hal yang penulis simpulkan, yaitu:

1. Hasil yang diperoleh adalah snort mampu mengenali jenis serangan, agar *snort* mudah mengenali jenis serangan maka harus mengaktifkan *automatic update* untuk pembaharuan *database snort.*
2. Untuk perlindungan pada *web server* maka cukup mengaktifkan *port* yang lazim digunakan pada web server yaitu 80 dan 443, sedangkan untuk *upload* dokumen gunakan FTP dengan *port* 21 tetapi harus menggunakan *secure socket layer* (SSL) guna enkripsi *user* dan *password* pada saat *login* dan transfer data..

VI. SARAN

Hal yang dapat dikembangkan adalah mencobanya dalam ruang lingkup topologi yang berbeda misalnya dalam ruang lingkup virtualisasi.

VII. DAFTAR PUSTAKA

[1] S. Patton, D. Doss, and W. Yurcik, "Open source versus commercial firewalls: Functional comparison," in *Conference on Local Computer Networks*, 2000, pp. 223–224, doi: 10.1109/LCN.2000.891032.

[2] M. Arunwan, T. Laong, and K. Atthayuwat, "Defensive performance comparison of firewall systems," in *2016 Management and Innovation Technology International Conference, MITiCON 2016*, 2017, pp. MIT221–MIT224, doi: 10.1109/MITICON.2016.8025212.

[3] D. Kumar and M. Gupta, "Implementation of Firewall & Intrusion Detection System Using pfSense To Enhance Network Security," in *International Journal of Electrical Electronics & Computer Science Engineering*, 2018, pp. 131–137.

[4] H. F. El-Sofany, S. A. El-Seoud, and I. A. T. F. Taj-Eddin, "A case study of the impact of denial of service attacks in cloud applications," *J. Commun.*, vol. 14, no. 2, pp. 153–158, 2019, doi: 10.12720/jcm.14.2.153-158.

[5] G. A. Jaafar, S. M. Abdullah, and S. Ismail, "Review of Recent Detection Methods for HTTP DDoS Attack," *J. Comput. Networks Commun.*, vol. 2019, p. 10, 2019, doi: 10.1155/2019/1283472.

[6] H. Beitollahi and G. Deconinck, "Analyzing well-known countermeasures against distributed denial of service attacks," *Comput. Commun.*, vol. 35, no. 11, pp. 1312–1332, 2012, doi: 10.1016/j.comcom.2012.04.008.

[7] K. Subramanian, P. Gunasekaran, and M. Selvaraj, "Two layer defending mechanism against DDoS attacks," *Int. Arab J. Inf. Technol.*, vol. 12, no. 4, pp. 317–324, 2015.

[8] X. Yuan, C. Li, and X. Li, "DeepDefense: Identifying DDoS Attack via Deep Learning," in *2017 IEEE International Conference on Smart Computing, SMARTCOMP 2017*, 2017, pp. 1–8, doi: 10.1109/SMARTCOMP.2017.7946998.

[9] K. Singh, P. Singh, and K. Kumar, "Application layer HTTP-GET flood DDoS attacks: Research landscape and challenges," *Comput. Secur.*, vol. 65, no. October, pp. 344–372, 2017, doi: 10.1016/j.cose.2016.10.005.

[10] T. Ni, X. Gu, H. Wang, and Y. Li, "Real-time detection of application-

layer DDoS attack using time series analysis," *J. Control Sci. Eng.*, vol. 2013, 2013, doi: 10.1155/2013/821315.

[11] J. Wang, M. Zhang, X. Yang, K. Long, and J. Xu, "HTTP-sCAN: Detecting HTTP-flooding attack by modeling multi-features of web browsing behavior from noisy web-logs," *China Commun.*, vol. 12, no. 2, pp. 118–128, 2015, doi: 10.1109/CC.2015.7084407.

[12] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDOS) flooding attacks," *IEEE Commun. Surv. Tutorials*, vol. 15, no. 4, pp. 2046–2069, 2013, doi: 10.1109/SURV.2013.031413.00127.

[13] N. Hoque, D. K. Bhattacharyya, and J. K. Kalita, "Botnet in DDoS Attacks: Trends and Challenges," *IEEE Commun. Surv. Tutorials*, vol. 17, no. 4, pp. 2242–2270, 2015, doi: 10.1109/COMST.2015.2457491.

[14] I. Sreeram and V. P. K. Vuppala, "HTTP flood attack detection in application layer using machine learning metrics and bio inspired bat algorithm," *Appl. Comput. Informatics*, vol. 15, no. 1, pp. 59–66, 2019, doi: 10.1016/j.aci.2017.10.003.

[15] A. Rai and R. K. Challa, "Survey on recent DDoS mitigation techniques and comparative analysis," in *Proceedings - 2016 2nd International Conference on Computational Intelligence and Communication Technology, CICT 2016*, 2016, pp. 96–101, doi: 10.1109/CICT.2016.27.

[16] S. Suroto, "A Review of Defense Against Slow HTTP Attack," *JOIV Int. J. Informatics Vis.*, vol. 1, no. 4, p. 127, 2017, doi: 10.30630/joiv.1.4.51.

[17] J. Tiso and D. Teare, *Designing Cisco Network Service Architectures (ARCH): Foundation Learning Guide*, Third. Indianapolis: ciscopress.com, 2011.