KESATRIA: Jurnal Penerapan Sistem Informasi (Komputer & Manajemen)
Terakreditasi Nomor 204/E/KPT/2022 | Vol. 5, No. 2, April (2024), pp. 543-556

Analisis Perbandingan Optimalisasi Port Knocking Dan Honeypot dengan Iptables Pada Server Untuk Keamanan Jaringan

Anjun Dermawan[1], Yuhandri[2], Sumijan[3]
[1,2,3]Fakultas Ilmu Komputer, Universitas Putra Indonesia "YPTK" Padang, Indonesia
E-mail: [1]anjundermawan22@gmail.com, [2]yuyu@upiyptk.ac.id, [3]sumijan@upiyptk.ac.id

Abstract

Computer network systems are designed to share resources together, so that the security of resources on the server must be maintained and the resources used must be optimized. The aim of this research is to analyze the comparative level of optimization of Port knocking and Honeypot using the IPTables method for network security on servers with different CPU and memory resources. The security methods used in this research are Port knocking, Honeypot and IPTables. The data used includes ports that were successfully attacked as well as resource usage before and after IPTables implementation on a server with 2 CPU resources and 1507284KiB memory obtained from previous research. The results of this research show that 80% of ports cannot be attacked while 20% of ports, namely port 22, are designed to be attacked. The server CPU and Memory resource usage graph shows a decrease after implementing IPTables from Denial of Service (DoS) and Brute force testing. On a server with 1 CPU and 1015852KiB of memory resources, CPU usage decreased by 36%, and memory usage decreased by 41%. Meanwhile, on a server with 4 CPU resources and 6036624 KiB of memory, CPU usage decreased by 41%, and memory usage decreased by 46%. This shows increased effectiveness compared to using just the Port knocking and Honeypot methods. It is hoped that this research can be a guide in measuring server optimization in overcoming Denial of Service (DoS) and Brute force attacks.

Keywords: *Port Knocking, Honeypot, IPTables, Denial of Service, Brute force*

Abstrak

Tujuan dari penelitian ini adalah untuk menganalisis perbandingan tingkat optimalisasi Port knocking dan Honeypot menggunakan metode IPTables untuk keamanan jaringan pada server dengan sumber daya CPU dan memori yang berbeda. Metode keamanan yang digunakan pada penelitian ini yaitu Port knocking, Honeypot dan IPTables. Data yang digunakan meliputi port yang berhasil diserang serta penggunaan sumber daya sebelum dan sesudah implementasi IPTables pada server dengan sumber daya 2 CPU dan memori 1507284KiB yang diperoleh dari penelitian sebelumnya. Hasilnya pada penelitian ini menunjukkan bahwa 80% port tidak dapat diserang sedangkan 20% port, yaitu port 22 dirancang untuk dapat diserang. Grafik penggunaan sumber daya CPU dan Memori server menunjukkan penurunan setelah penerapan IPTables dari pengujian Denial of Service (DoS) dan Brute force. Pada server dengan sumber daya 1 CPU dan 1015852KiB memori, penggunaan CPU menurun sebesar 36%, dan penggunaan memori menurun sebesar 41%. Sedangkan pada server dengan sumber daya 4 CPU dan 6036624 KiB memori, penggunaan CPU menurun sebesar 41%, dan penggunaan memori menurun sebesar 46%. Hal ini menunjukkan peningkatan efektivitas dibandingkan hanya menggunakan metode Port knocking dan Honeypot saja. Penelitian ini diharapkan dapat menjadi panduan dalam mengukur optimalisasi server dalam mengatasi serangan Denial of Service (DoS) dan Brute force.

Kata Kunci: *Port Knocking, Honeypot, IPTables, Denial of Service, Brute force*

ISSN: 2720-992X

1. Pendahuluan

Keamanan sistem informasi adalah suatu langkah pencegahan terhadap tindakan penipuan pada sistem yang berbasis informasi berbentuk non-fisik dan dilakukan untuk memastikan bahwa data dalam suatu sistem terlindungi dari ancaman. Tidak hanya ancaman penipuan, bentuk ancaman dari serangan siber lainnya pun harus dicegah. Ancaman tersebut bisa berupa serangan seperti Malware, trojan, virus, dan pemindaian port [1].

Jaringan komputer merupakan media penghubung untuk berbagi data, informasi dan sumber daya untuk dipakai secara bersama [2][3][4]. Oleh karena itu dibutuhkan keamanan Jaringan untuk menjaga dan menjamin keamanan data [4][5][6]. Dari segala macam serangan dan usaha-usaha penyusupan atau pemindaian oleh pihak yang tidak bertanggung jawab[2][5][7][8].

Keamanan jaringan merupakan aspek terpenting sebuah sistem dalam menjaga validitas dan integritas data, serta menjamin ketersediaan layanan bagi penggunaannya. Sistem keamanan jaringan komputer harus dilindungi dari segala macam serangan dan usaha-usaha penyusupan atau pemindaian oleh pihak yang tidak berhak [5][6].

Kebutuhan akan jaringan komputer semakin bertambah penting, baik dalam pendidikan, pekerjaan maupun dalam sebuah permainan, dengan banyaknya akses ke jaringan tersebut maka akan banyak pula peluang kejahatan yang terjadi didalam jaringan ataupun adanya peretas yang dapat mematikan sumber daya pada server.

Pada penelitian ini peneliti menerapkan konfigurasi melalui *firewall*, dengan menerapkan *Port Knocking*. *Port knocking* adalah teknik keamanan yang digunakan untuk mengamankan akses ke sistem atau layanan jaringan dengan cara memungkinkan akses ke *port-port* tertentu hanya setelah urutan tertentu dari koneksi ke *port-port* lain telah terjadi [7][9][10]. Penting untuk diingat bahwa meskipun teknik ini dapat menambahkan lapisan keamanan tambahan tetapi kurang optimal dalam mendeteksi penyusupan [2], Oleh karena itu *port knocking* sering digunakan bersama dengan teknik keamanan lainnya untuk menciptakan sistem yang lebih aman.

Konfigurasi selanjutnya yang dilakukan yaitu *Honeypot*. *Honeypot* adalah sebuah sistem atau perangkat lunak yang didesain dengan sengaja untuk menarik perhatian penyerang atau pihak yang mencoba mengeksploitasi kelemahan keamanan dalam suatu jaringan komputer atau system [4][9]. *Honeypot* berfungsi sebagai umpan atau perangkap yang bertujuan untuk menarik serangan sehingga peneliti keamanan atau administrator jaringan dapat mempelajari metode serangan tersebut, mengidentifikasi ancaman baru, dan memahami taktik penyerang [5]. *Honeypot* digunakan sebagai alat pembelajaran dan pemahaman terhadap ancaman siber [11]. Implementasi *Honeypot* harus dilakukan dengan hati-hati dan diisolasi dengan baik agar tidak membahayakan keamanan sistem dan jaringan yang sebenarnya.

Konfigurasi selanjutnya yang dilakukan yaitu *Iptables*. *Iptables* adalah perangkat lunak *firewall* yang digunakan pada sistem operasi Linux untuk mengelola aturan-aturan keamanan jaringan [12]. Dengan menggunakan *Iptables*, pengguna dapat mengkonfigurasi aturan-aturan untuk mengontrol lalu lintas jaringan yang masuk dan keluar dari sistem, serta memungkinkan atau memblokir koneksi ke atau dari *port-port* tertentu. *Iptables* digunakan untuk mengamankan sistem Linux dari serangan jaringan, mengontrol akses ke layanan jaringan, dan melindungi sistem dari ancaman siber. *Iptables* bekerja dengan memeriksa paket-paket data yang melewati sistem dan membandingkannya dengan aturan-aturan yang telah ditetapkan [9].

Penelitian yang dilakukan Gunawan dkk meneliti tentang pendeteksi dan pencegah *malware*, menggunakan *S*ystem *Snort* dan *Honeypot*. Data yang digunakan diambil dari port scanning oleh *S*ystem *Snort* dan *Honeypot* terkumpul selama 1 bulan mulai bulan Februari 2020 – Maret 2020 menggunakan *Snort IDS* dan *Honeypot*. Hasil dalam penelitian ini mengungkapkan dapat mencegah 248.574 data serangan dengan 11 atribut, yang setiap atributnya dapat mendeteksi IP penyerang dan tanggal penyerangan [3].

Penelitian yang dilakukan Mardiansyah dkk meneliti tentang optimasi keamanan jaringan pada server, menggunakan metode *Port Knocking*, *Honeypot* dan *IPTables*. Data yang digunakan diambil dari berbagai konfigurasi yang dilakukan dan telah dilakukan pengujian. Hasil dari penelitian ini adalah penambahan metode *IPTables* dapat meningkatkan kinerja baik dari segi penggunaan CPU (38,4%) dan Memori (44,2%) pada server maupun keamanan jaringan dibandingkan dengan hanya menggunakan metode *Port knocking* dan *Honeypot* saja [5].

Penelitian yang dilakukan Novianto dkk meneliti tentang peningkatan keamanan mikrotik, menggunakan *Port Knocking* dan *Port Blocking*. Data yang digunakan yaitu hasil dari konfigurasi yang telah dilakukan. Hasil pengujian ini dapat mengamankan akses ke router mikrotik dengan cara terlebih dahulu mengirimkan paket *knocking* berupa ping ke *IP Address*, *Telnet*, dan *SSH* [6].

Penelitian yang dilakukan Rizal dkk meneliti tentang implementasi keamanan jaringan, menggunakan metode *Port Blocking* dan *Port Knocking* pada Mikrotik RB-941. Data yang digunakan diambil dari tiga tahap utama yang dilakukan yaitu analisis data port yang akan di uji. Hasil dalam pengimplementasian *Port Blocking* dan *Port Knocking* dalam pengamanan jaringan lokal mengungkapkan bahwa *Port Blocking* dan *Port Knocking* memiliki kelebihan dan kekurangannya masing-masing [10].

Penelitian yang dilakukan Rosyid dkk meneliti tentang deteksi malware pada jaringan lokal, menggunakan *Honeypot* dan *Yara*. Data yang digunakan diambil dari pengujian sistem dilakukan menggunakan 2 sensor *Honeypot* yang dipasang pada jaringan public dan server proaktif menggunakan 409 *Yara* rules *malware*. Hasil dari penelitian ini yaitu dalam mendeteksi *malware* telah berhasil dilakukan dan memiliki potensi sebagai mekanisme keamanan proaktif yang membantu meningkatkan kualitas mitigasi resiko keamanan siber di era industri 4.0 [11].

Penelitian yang dilakukan ini mengacu pada penelitian terdahulu yang dilakukan oleh Mardiansyah dkk dengan judul "Optimasi Port Knocking dan Honeypot menggunakan Iptables sebagai Keamanan Jaringan pada Server". Pada penelitian beliau meneliti tentang optimasi keamanan jaringan pada server, menggunakan metode *Port Knocking*, *Honeypot* dan *IPTables*. Hasil dari penelitian ini adalah penambahan metode *IPTables* dapat meningkatkan kinerja baik dari segi penggunaan CPU (38,4%) dan Memori (44,2%) pada server. Tujuan penelitian ini adalah analisis perbandingan tingkat optimalisasi *Port knocking* dan *Honeypot* menggunakan metode *IPTables* untuk keamanan jaringan pada *server* dengan sumber daya CPU dan Memori yang lebih rendah dan lebih tinggi.

2. Metodologi Penelitian

Berikut adalah bagan alir penelitian seperti yang ditunjukkan pada Gamabr 1 berikut.

Gambar 1. Bagan Alur Penelitian

2.1. Mengumpulkan Data

Data dalam penelitian ini diambil dari hasil studi literatur yang dilakukan pada penelitian terdahulu yang berkaitan dengan penerapan metode *Port Knocking*, *Honeypot* dan *IPTables*. Data yang digunakan yaitu data port yang berhasil diserang dan data penggunaan sumber daya CPU dan Memori server.

2.2. Analisis Masalah

Melakukan analisis masalah yang didapatkan dari penelitian terdahulu. Penelitian terdahulu menggunakan sumber daya server yang ada, bagaimana jika menggunakan

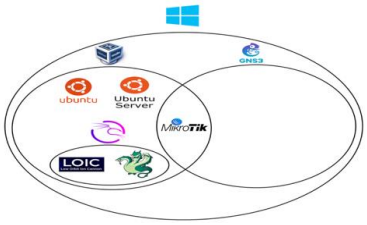
sumber daya yang lebih rendah dan lebih tinggi dari sumber daya pada penelitian sebelumnya.

2.3. Perancangan

Tahapan perancangan yang digunakan dalam penelitian ini adalah sebagai berikut:

2.3.1. Persiapan Software dan Hardware

Alat yang digunakan dalam perancangan sistem keamanan jaringan ini adalah Laptop PC dengan spesifikasi Processor AMD RYZEN 3 @2.3GHZ, 8 GB RAM, 1TB HDD, adapun bagan hubungan antar software yang digunakan adalah sebagai berikut.



Gambar 2. Bagan hubungan Software

Pada Gambar 2 dapat dilihat bahwa pada virtual box terdapat 3 buah system operasi yaitu Linux Ubuntu Server 16.4, Linux Ubuntu Dekstop 22.4 dan Kali Linux 2023, terdapat aplikasi yang digunakan antara lain loic dan hydra kemudian akan di-install pada sistem operasi Kali Linux yang divirtualisasikan pada Virtualbox. Kemudian langkah selanjutnya mengintegrasikan Virtualbox dengan simulator GNS3 agar sistem operasi yang telah ter-install sebelumnya dapat digunakan pada simulator GNS3. Selanjutnya melakukan installasi Mikrotik yang selanjutnya akan diintegrasikan dengan simulator GNS3 yang berjalan pada windows.

2.3.2. Instalasi Software

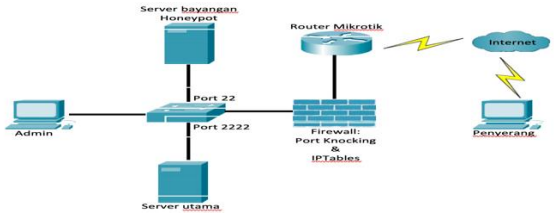
Pada tahap ini dilakukan instalasi software dengan menggunakan software GNS3, selanjutnya melakukan instalsi Virtualbox sebagai perangkat lunak untuk mengoperasikan/menginstal Operating System virtual seperti Linux Ubuntu Server 16.4, Linux Ubuntu Dekstop 22.4 dan Kali Linux 2023, selanjutnya perangkat beberapa System Operasi tersebut akan diintegrasikan dengan simulator GNS3.

2.3.3. Integrasi Perangkat Virtual

Pada tahapan ini dilakukanya integrasi perangkat virtual. Perangkat lunak GNS3 yang sudah di-install. Selanjutnya akan diintegrasikan dengan VirtualBox agar dapat menjalankan Operating System yang digunakan dan dapat terhubung dengan GNS3.

2.3.4. Membangun Topologi Jaringan

Melakukan perancangan dan memberikan gambaran mengenai sistem keamanan yang akan dibangun.



Gambar 3. Topologi Keamanan Jaringan

Pada Gambar 3 dapat dilihat topologi jaringan yang akan dibangun, dimana pada topologi jaringan tersebut terdapat 1 unit router mikrotik, 1 unit switch, 1 unit server, dan 2 unit pc yang digunakan sebagai penyerang dan admin dari server.

2.3.5. Konfigurasi Jaringan

Pada tahap ini dilakukan konfigurasi keamanan jaringan *Port Knocking*, *Honeypot* dan *IPTables* pada server utama. Pada konfigurasi pertama dilakukan instalasi *Port knocking* dan dilakukan konfigurasi untuk membuka dan menutup akses menuju *port* yang telah di *block* oleh *firewall*. pada konfigurasi kedua dilakukan instalasi *Honeypot* dan dilakukan konfigurasi untuk membuat server bayangan. Pada konfigurasi ketiga dilakukan instalasi *iptables* dan di lakukan konfigurasi untuk memfilter *port 22* yang merupakan *port SSH*.

2.4. Pengujian

Pengujian pada penelitian ini diterapkan pada server dengan sumber daya yang lebih rendah dan lebih tinggi dari penelitian terdahulu, terdapat dua metode penyerangan yaitu *Denial of Service* (*DoS*) dan *Brute force*. Skenario pengujian serangan menggunakan dua buah skenario pengujian, skenario pengujian pertama adalah sebelum menggunakan *IPTables* dan skenario kedua adalah setelah penambahan *IPTables*.

2.5. Analisis Hasil

Pada tahapan ini data yang didapat dari proses pengujian menggunakan *Denial of Service* (*DoS*) dan *Brute force* akan diuraikan dengan tujuan data tersebut dapat menjadi informasi atau *report* yang dapat dijadikan saran perbaikan untuk pengelola Jaringan. Analisa yang dilakukan adalah dengan menjelaskan bahwa penambahan *IPTables* dalam keamanan jaringan yang sebelumnya hanya menggunakan *Port knocking* dan *Honeypot* menjadi sangat signifikan dalam mengatasi serangan *Denial of Service* (*DoS*) dan *Brute force*.

2.6. Perbandingan

Pada tahapan ini akan membandingkan data yang dihasilkan dari proses pengujian penerapan *Port knocking* dan *Honeypot* sebelum dan setelah menggunakan *IPTables*, pada server dengan sumber daya yang lebih rendah dan lebih tinggi dari penelitian terdahulu. Data yang dibandingkan adalah data *port* yang berhasil diserang sebelum dan setelah menggunakan *IPTables* pada *server* dengan sumber daya rendah, tinggi dan menengah seperti yang ada pada penelitian terdahulu. Data selanjutnya yang akan dibandingkan adalah dampak dari penyerangan sebelum dan setelah menggunakan *IPTables* yang dilakukan terhadap sumber daya CPU dan Memori *server* yang lebih rendah, tinggi dan menengah seperti yang ada pada penelitian terdahulu.

3. Hasil dan Pembahasan

3.1. Data

Pada penelitian Mardiansyah dkk penerapan metode *Port knocking* dan *Honeypot* sebelum dan setelah penambahan *IPTables* untuk penyerangan menggunakan metode *Denial of Service* (*DoS*) dan *Brute force* menggunakan sumber daya server 2 CPUs dan 1507284KiB memori atau 1,5GB RAM [5].

Tabel 1. Data Penyerangan Port Knocking dan Honeypot Sebelum dan Setelah Penambahan IPTables Pada Penggunaan CPU dan Memori menggunakan Denial of Service (DoS) dan Brute force

Serangan	Port	Sebelum CPU	Setelah CPU	Sebelum Memori	Setelah Memori	Peningkatan CPU	Peningkatan Memori
Denial of Service	21	16,0 sy	0,9 sy	672,8	493,6	94 %	26 %
	22	0,9 sy	0,7 sy	900,8	476,4	22 %	47 %
	53	1,0 sy	0,8 sy	341,2	243,6	20 %	28 %
	80	0,8 sy	0,8 sy	684,6	101,8	0 %	85 %
	2222	0.7 sy	0.7 sy	0	0	0 %	0 %
Bruteforce	21	5,0 sy	0,8 sy	10983,4	101,6	84 %	99 %
	22	12,3 sy	12,0 sy	18763,3	17155,1	2 %	8 %

Serangan	Port	Sebelum CPU	Setelah CPU	Sebelum Memori	Setelah Memori	Peningkatan CPU	Peningkatan Memori
	53	3,6 sy	0,8 sy	274,6	132,6	77 %	51 %
	80	4,7 sy	0,7 sy	3755,6	49,6	85 %	98 %
	2222	0.7 sy	0.7 sy	0	0	0 %	0 %
Jumlah rata-rata peningkatan persentase:						384/10 = 38,4%	442/10 = 44,2%

Pada Tabel 1 penyerangan menggunakan *Denial of Service* dan *Brute force* menuju *port* 21,22,53,80 dan 2222 setelah penamabahan IPTables, penggunaan CPU time pada ruang kernel 7 dari 10 port yang ada mengalami penurunan dengan peningkatan performa sebesar 38,4% dan pada pengunaan Memori 8 dari 10 port yang ada mengalami penurunan dengan peningkatan performa sebesar 44,2%.

Tabel 2. Data Penyerangan Port pada Pengujian Port Knocking dan Honeypot Sebelum dan Setelah Penambahan IPTables menggunakan Denial of Service (DoS) dan Brute force

Serangan	Port	Sebelum Penambahan IPTables		Setelah Penambahan IPTables	
		Keterangan		Keterangan	
		Berhasil	Gagal	Berhasil	Gagal
Denial of Service	21	✓			✓
	22	✓		✓	
	53	✓			✓
	80	✓			✓
	2222		✓		✓
Persentase Serangan		80 %	20%	20 %	80%
Brute-force	21	✓			✓
	22	✓		✓	
	53		✓		✓
	80		✓		✓
	2222		✓		✓
Persentase Serangan		40 %	60%	20 %	80%
Rata-rata Persentase Serangan DoS & Bruteforce		60 %	40 %	20 %	80 %

Pada Tabel 2 penyerangan menggunakan *Denial of Service* dan *Brute force* menuju *port* 21,22,53,80 dan 2222 setelah penamabahan IPTables, hanya sebesar 20% penyerangan yang berhasil dilakukan oleh penyerang dan sebesar 80% penyerangan yang gagal dilakukan oleh penyerang.

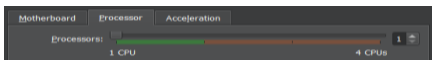
3.2. Analisis Masalah

Melakukan analisis masalah yang didapatkan dari penelitian Mardiansyah dkk, penelitian tersebut menggunakan sumber daya server 2 CPUs dan 1507284KiB memori, bagaimana tingkal optimalisasi server jika menggunakan sumber daya yang lebih rendah dan lebih tinggi dari sumber daya pada penelitian sebelumnya. Sumber daya server yang lebih rendah yaitu 1 CPUs dan 1015852KiB memori dan sumber daya server yang lebih tinggi yaitu 4 CPUs dan 6036624 KiB memori.

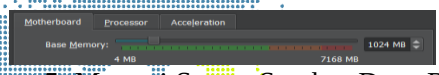
3.3. Perancangan

3.3.1. Integrasi Perangkat Virtual

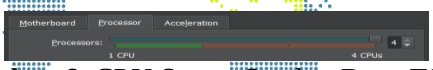
Proses integrase perangkat virtual ini yang perlu disiapkan yaitu setingan sumber daya server yang berbeda untuk jadi pembanding dengan penelitian terdahulu.



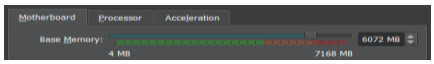
Gambar 4. CPU Server Sumber Daya Rendah



Gambar 5. Memori Server Sumber Daya Rendah



Gambar 6. CPU Server Sumber Daya Tinggi



Gambar 7. Memori Server Sumber Daya Tinggi

3.3.2. IP Address Ubuntu Server

Pada server perlu mengkonfigurasi TCP/IP. Untuk mengkonfigurasi TCP/IP pada server dapat menggunakan *Script command* sebagai berikut:

Nano /etc/network/interfaces

```
source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enp0s3
iface enp0s3 inet static
        address 192.168.1.200
        netmask 255.255.255.0
        network 192.168.1.0
```

Gambar 8. Network Interface

3.3.3. Port Knocking

Port Knocking diimplementasikan dengan mengkonfigurasi daemon guna memonitoring *log firewall* untuk permintaan koneksi dan menentukan apakah *client* telah melakukan urutan ketukan yang benar.

```
root@ubuntu:~# apt install knockd
Sedang membaca daftar paket... Selesai
Membangun pohon ketergantungan
Membaca informasi yang tersedia... Selesai
knockd is already the newest version (0.5-3ubuntu1).
0 dimutakhirkan, 0 baru terinstal, 0 akan dihapus dan 149 tidak akan dimutakhirkan.
root@ubuntu:~#
```

Gambar 9. Proses Pengistalan Port Knocking

```
GNU nano 2.9.3                  /etc/knockd.conf

[options]
        UseSyslog

[openSSH]
        sequence    = 7000,8000,9000
        seq_timeout = 5
        command     = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 2222 -j ACCEPT
        tcpflags    = syn

[closeSSH]
        sequence    = 9000,8000,7000
        seq_timeout = 5
        command     = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 2222 -j ACCEPT
        tcpflags    = syn
```

Gambar 10. Proses Konfigurasi Knock

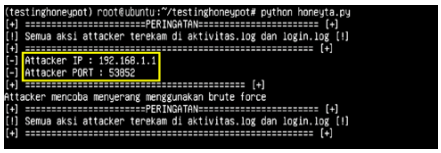
Pada Gambar 10 terdapat tiga bagian yaitu *options*, *open*SSH, dan *close*SSH. Pada *option* dapat digunakan untuk memberikan *log* aktivitas IP yang berusaha untuk masuk kedalam knock. Berikutnya pada *open*SSH terdapat "sequence = 7000, 8000 ,9000" yang merupakan ketukan sebagai autentifikasi untuk membuka *port* pada server SSH, berikutnya terdapat "command = /sbin/iptables -I INPUT -s %IP% -p tcp –d *port* 2222 - j ACCEPT" merupakan perintah yang akan dilakukan saat terdapat *client* yang berusaha masuk kedalam *port* 2222 yang dimana apabila ketukan yang dilakukan benar maka akan di terima. Pada *close*SSH terdapat "sequence = 9000, 8000, 7000" merupakan ketukan autentifikasi untuk menutup pengaksesan *port*, berikutnya terdapat "command=/sbin/iptables -D INPUT -s %IP% -p tcp –d*port* 2222 -j ACCEPT" merupakan perintah untuk menutup Kembali akses SSH *port* 2222 dengan ketukan autentifikasi yang telah di tentukan. Port 2222 merupakan port *rondom* yang dipilih untuk menggantikan SSH *default* (*port* 22) dikarenakan *port* 22 akan digunakan untuk mengakses *Honeypot*.

3.3.4. Honeypot

Honeypot merupakan keamanan yang dibuat untuk diserang oleh penyerang dimana pada penelitian ini *Honeypot* memiliki akses terhadap *port* 22 yang dimana *port* 22 tersebut pada awalnya merupakan *port* standar yang digunakan untuk mengakses SSH, tetapi pada penelitian ini *port* SSH yang asli telah di simpan pada *port* 2222 dan dilindungi oleh *Port knocking*.

Gambar 11. Melakukan Scanning Port Menggunakan Nmap

Pada Gambar 11 Menjelaskan proses untuk melakukan *scanning port* yang terbuka pada server menggunakan *tools* nmap. Dapat dilihat bahwa *status port* untuk layanan SSH dalam kondisi *open* atau terbuka dimana *port* SSH tersebut merupakan *port* dari *Honeypot* untuk mengelabui penyerang.

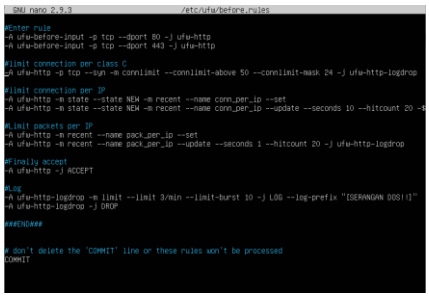


Gambar 12. Tampilan Honeypot Pada Server

Pada Gambar 12 Menjelaskan proses *Honeypot* yang berjalan pada server yang dimana pada tampilan tersebut menunjukkan seluruh aksi penyerangan yang dilakukan akan dicatat pada *file* "aktivitas.log" yang merupakan seluruh aktivitas yang dilakukan penyerang dalam server akan di catat dan "login.log" merupakan *file* yang dibuat untuk menyimpan hasil uji coba penggunaan *user* dan *password* dalam membobol server.

3.3.5. IPTables

Pada pengujian sistem yang telah dilakukan menggunakan *software* Hydra dan LOIC didapatkan hasil penyerangan menggunakan Hydra dapat di antisipasi oleh *Honeypot*, sedangkan penyerangan DoS menggunakan LOIC tidak dapat di antisipasi oleh server, sehingga untuk mengatasi serangan DoS dibuat *rules* IPTables yang berfungsi untuk memblokir dan mengijinkan akses masuk atau keluar server.

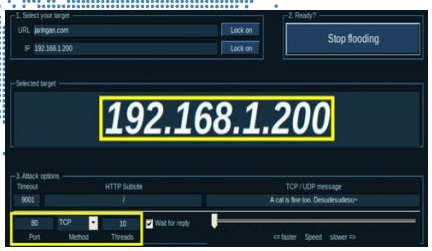


Gambar 13. Rules IPTables

3.4. Pengujian

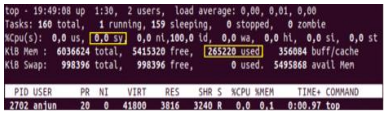
3.4.1. Pengujian Denial of Service

Penelitian ini dilakukan pengujian penyerangan *Denial of Service* terhadap server menggunakan *tool* LOIC. *Software* ini merupakan *tools* yang banyak digunakan pada saat ini untuk melakukan penyerangan DoS. Dengan *tools* ini dapat melakukan penyerangan berdasarkan IP dan menentukan *port* serangan yang diinginkan.

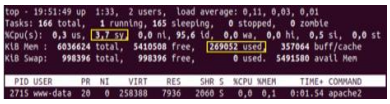


Gambar 14. Serangan DoS Menggunakan LOIC

Pada Gambar 14 Menunjukkan penyerangan yang menggunaakn *tools* LOIC dimana *url* yang akan diserang adalah "jaringan.com" dengan IP *address* "192.168.1.200" dengan tujuan *port* penyerangan yaitu *port* 80 dengan *menthod* TCP dan *thread* yang digunakan sebanyak 10. Penyerangan akan dilakukan dalam waktu 1 menit dan akan dilihat penggunaan CPU dan Memori pada *server* menggunakan perintah TOP yang berfungsi sebagai memanajemen proses yang berjalan pada *server*.



Gambar 15. Proses TOP Pada Saat Belum Diserang



Gambar 16. Proses TOP Pada Saat Diserang

Pada Gambar 15 dan Gambar 16 menujukkan hasil proses dari perintah TOP yang berjalan pada *server* sebelum ditambahkan IPTables. terlihat jumlah waktu yang dihabiskan dalam ruang kernel (sy) pada CPU mengalami peningkatan setelah dilakukan penyerangan DoS menggunakan LOIC sebesar (3.7 sy). Berikutnya pada memori menunjukkan peningkatan penggunaan memori sebesar (3832 kibibita). Pada Gambar 17 dan Gambar 18 Dibawah menunjuknan hasil pengujian setelah ditambahkan IPTables.

Gambar 17. Proses TOP Pada Saat Belum Diserang

Gambar 18 Proses TOP Pada Saat Diserang

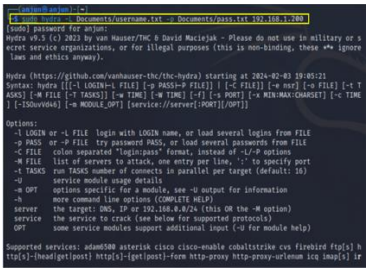
Pada Gambar 17 dan Gambar 18 Menujukkan hasil proses dari perintah TOP yang berjalan pada *server* setelah ditambahkan IPTables. Terlihat waktu yang dihabiskan dalam ruang kernel (sy) pada CPU peningkatan setelah dilakukan penyerangan DoS menggunakan LOIC pada Dengan jumlah (0.3 sy). Berikutnya pada memori dapat dilihat jumlah memori sebelum dilakukan penyerangan sebesar (269052 kibibita) dan pada saat penyerangan sebesar (270032 kibibita) menunjukkan penggunaan memori sebesar (980 kibibita).

3.4.2. Pengujian Bruteforce

Pada penelitian ini dilakukan penyerangan *brute force* menggunakan hydra terhadap *port* 22 yang merupakan *port Honeypot* dan *port* 2222 yang merupakan *port* SSH, dimana penyerangan ini bertujuan untuk memperoleh *username* dan *password* pada server. perintah yang digunakan dalam penyerangan ini adalah:

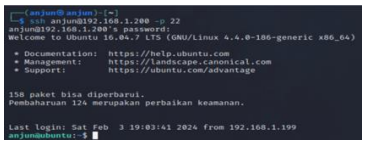
```
sudo hydra -L Documents/username.txt -P Documents/pass.txt 192.168.1.200 ssh
```

"sudo hydra" berfungsi untuk menjalankan *tool hydra* pada kali linux, "- L Documents/username.txt" merupaka perintah untuk membuka *file userlist* yang telah di buat dengan nama username.txt pada *file Documents*, berikutnya perintah "-P Documents/pass.txt" merupakan perintah untuk membuka *file password list* pada *file documents* dengan nama *file* pass.txt dan dilanjutkan dengan IP *address* dari server dan *port* yang akan diserang (SSH). Sehingga di peroleh hasil pengujian terhadap penyerangan menggunakan *brute force* yang bertujuan untuk memperoleh *username* dan *password* dari server. hasil dapat dilihat pada Gambar 19.



Gambar 19. Proses Penyerangan Menggunakan Brute force

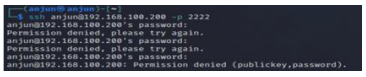
Selanjutnya dilakukan uji coba untuk mengakses server menggunakan perintah "ssh username@host -p 22" pada uji coba penyerangan sebelumnya yang dilakukan menggunakan *bruteforce*, penyerang mendapatkan hasil yaitu *username*, *host* dan *password* dalam mengakses server dan dapat digunakan untuk mengakses server seperti pada Gambar 20.



Gambar 20. Melakukan Remote Ke Server Honeypot

Pada Gambar 20 menjelaskan proses dimana penyerang melakukan *remote* server *Honeypot* menggunakan layanan SSH dengan menjalankan perintah "ssh anjun@192.168.1.200 -p 22" dimana "192.168.1.200" adalah *Host* dari server sehingga penyerang berhasil masuk ke server *Honeypot* yang dimana tampilan dari server *Honeypot* ini merupakan tampilan dari *code* yang telah dibuat pada *file Honeypot*.py. sehingga penyerang merasa telah berhasil masuk kedalam server utama.

Pada penyerangan yang telah dilakukan oleh penyerang tidak menutup kemungkinan penyerang mengetahui *port* 22 yang pada umumnya merupakan *port* SSH bukan *port* SSH yang asli. Sehingga penyerang akan mencoba untuk menyerang *port -port* yang lain. Apabila penyerang telah mengetahui *port* 2222 merupakan *port* SSH dari server maka penyerang akan mencoba untuk melakukan penyerangan *brute force* terhadap *port* 2222 seperti pada Gambar 21.



Gambar 21. Penyerangan Brute Force Menuju Port 2222

Pada Gambar 21 dilakukan penyerangan menuju *port* 2222 yang merupakan *port* SSH dari server dimana penyerangan tersebut mengalami "*Permission denied connection to 192.168.1.200"* atau penyerangan gagal terhubung dikarenakan pada *port* 2222 telah dilindungi oleh *Port Knocking* yang dimana untuk mengakses *port* 2222 dibutuhkan autentifikasi untuk membuka *port* tersebut.

3.5. Analisis Hasil

Berdasarkan hasil pengujian kepada server dengan sumber daya CPU dan Memori rendah dan server dengan sumber daya CPU dan Memori tinggi menggunakan metode *Port Knocking, Honeypot* sebelum dan setelah menggunakan *IPTables* menuju *port* (21,22,53,80 dan 2222) sebanyak 1 menit dikali 10 kali uji coba penyerangan menggunakan metode *Denial of Service* (*DoS*) dan *Brute Force* didapatkan hasil rata-rata seperti pada Tabel 3, Tabel 4, Tabel 5 dan Tabel 6.

Tabel 3. Hasil Pengujian Server Sumber Daya CPU Dan Memori Rendah Dengan Port Knocking Dan Honeypot Sebelum Dan Setelah Penambahan IPTables Menggunakan Denial Of Service (Dos) Dan Brute Force

Serangan	Port	Sebelum CPU	Setelah CPU	Sebelum Memori	Setelah Memor	Peningkatan CPU	Peningkatan Memori
Denial of Service	21	20,21 sy	5,37 sy	683,9	516,7	73 %	24 %
	22	5,48 sy	4,4 sy	914,9	493,4	20 %	46 %
	53	5,47 sy	4,51 sy	456,7	361,7	18 %	21 %
	80	0,87 sy	0,87 sy	726,7	135	0 %	81 %
	2222	0,85 sy	0.85 sy	0	0	0 %	0 %
Bruteforce	21	6,02 sy	1 sy	11715,5	252,7	83 %	98 %
	22	18,61 sy	18,47 sy	18823,7	17622	1 %	6 %
	53	4,33 sy	0,98 sy	350,8	206,8	77 %	41 %
	80	5,56 sy	0,89 sy	4196,7	175	84 %	96 %
	2222	0.85 sy	0.85 sy	0	0	0 %	0 %
Jumlah rata-rata peningkatan persentase:						365/10 = 36%	414/10 = 41%

Pada Tabel 3 penyerangan menggunakan *Denial of Service* dan *Brute Force* menuju *port* 21,22,53,80 dan 2222 kepada *server* dengan sumber daya rendah didapatkan hasil yaitu penggunaan CPU mengalami peningkatan performa sebanyak 36% dan memori mengalami peningkatan performa sebanyak 41%.

Tabel 4. Hasil Pengujian Port Pada Keamanan Server Sumber Daya CPU Dan Memori Rendah Dengan Port Knocking Dan Honeypot Sebelum Dan Setelah Penambahan IPTables Menggunakan Denial Of Service(DOS) Dan Brute Force

Serangan	Port	Sebelum Penambahan IPTables		Setelah Penambahan IPTables	
		Keterangan		Keterangan	
		Berhasil	Gagal	Berhasil	Gagal
Denial of Service	21	✓			✓
	22	✓		✓	
	53	✓			✓
	80	✓			✓
	2222		✓		✓
Persentase Serangan		80 %	20%	20 %	80%
Brute-force	21	✓			✓
	22	✓		✓	
	53		✓		✓
	80		✓		✓
	2222		✓		✓
Persentase Serangan		40 %	60%	20 %	80%
Rata-rata Persentase Serangan DoS & Bruteforce		60 %	40 %	20 %	80 %

Pada Tabel 4 penyerangan menggunakan *Denial of Service* dan *Brute Force* menuju *port* 21,22,53,80 dan 2222 kepada *server* dengan sumber daya rendah didapatkan hasil yaitu sebelum penggunaan *IPTables* 60% penyerangan berhasil dilakukan dan seteleh penggunaan *IPTables* hanya 20% penyerangan yang berhasil dilakukan, penyerangan tersersebut hanya berhasil ke *port* 22 yaitu *port server* bayangan *Honeypot.*

Tabel 5. Hasil Pengujian Server Sumber Daya CPU Dan Memori Tinggi Dengan Port Knocking Dan Honeypot Sebelum Dan Setelah Penambahan IPTables Menggunakan Denial Of Service (DOS) Dan Brute Force

Serangan	Port	Sebelum CPU	Setelah CPU	Sebelum Memori	Setelah Memori	Peningkatan CPU	Peningkatan Memori
Denial of Service	21	14,14 sy	0,59 sy	523,4	354,9	96 %	32 %
	22	0,85 sy	0,59 sy	800,8	379,4	31 %	53 %
	53	0,95 sy	0,74 sy	314,2	211,6	22 %	33 %
	80	0,78 sy	0,78 sy	605,9	80,9	0 %	87 %
	2222	0,6 sy	0.6 sy	0	0	0 %	0 %
Bruteforce	21	4,47 sy	0,6 sy	9486,1	80,3	87 %	99 %
	22	10,96 sy	10,14 sy	17895,9	16259,4	7 %	9 %
	53	3 sy	0,61 sy	232,7	109,4	80 %	53 %
	80	4,38 sy	0,56 sy	3275,9	26,1	87 %	99 %
	2222	0.6 sy	0.6 sy	0	0	0 %	0 %
Jumlah rata-rata peningkatan persentase:						409/10 = 41%	465/10 = 46%

Pada Tabel 5 penyerangan menggunakan *Denial of Service* dan *Brute Force* menuju *port* 21,22,53,80 dan 2222 kepada *server* dengan sumber daya rendah didapatkan hasil yaitu penggunaan CPU mengalami peningkatan performa sebanyak 41% dan memori mengalami peningkatan performa sebanyak 46%.

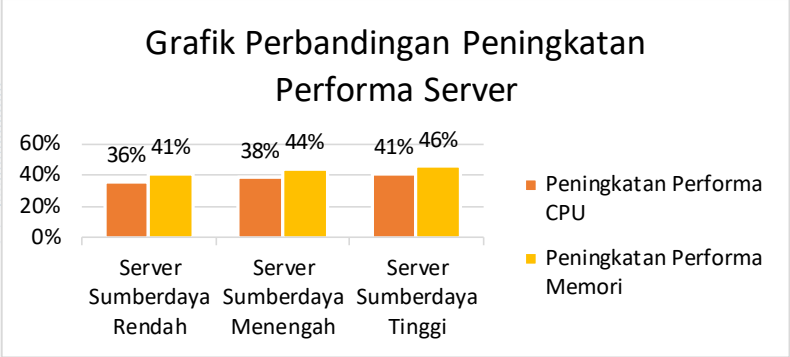
Tabel 6. Hasil Pengujian Port Pada Keamanan Server Sumber Daya CPU Dan Memori Tinggi Dengan Port Knocking Dan Honeypot Sebelum Dan Setelah Penambahan IPTables Menggunakan Denial Of Service(DOS) Dan Brute Force

Serangan	Port	Sebelum Penambahan IPTables		Setelah Penambahan IPTables	
		Keterangan		Keterangan	
		Berhasil	Gagal	Berhasil	Gagal
Denial of Service	21	✓			✓
	22	✓		✓	
	53	✓			✓
	80	✓			✓
	2222		✓		✓
Persentase Serangan		80 %	20%	20 %	80%
Brute-force	21	✓			✓
	22	✓		✓	
	53		✓		✓
	80		✓		✓
	2222		✓		✓
Persentase Serangan		40 %	60%	20 %	80%
Rata-rata Persentase Serangan DoS & Bruteforce		60 %	40 %	20 %	80 %

Pada Tabel 6 penyerangan menggunakan *Denial of Service* dan *Brute Force* menuju *port* 21,22,53,80 dan 2222 kepada server dengan sumber daya rendah didapatkan hasil yaitu sebelum penggunaan *IPTables* 60% penyerangan berhasil dilakukan dan seteleh penggunaan *IPTables* hanya 20% penyerangan yang berhasil dilakukan, penyerangan tersersebut hanya berhasil ke *port* 22 yaitu *port server* bayangan *Honeypot.*

3.6. Perbandingan

Grafik pada Gambar 22 menyatakan bahwa *Port Knocking* dan *Honeypot* setelah menggunakan *IPTables* lebih baik diterapkan pada *Server* dengan Sumberdaya yang lebih tinggi dikarenakan pada gambar diatas dapat terlihat persentase peningkatan performa CPU 41% dan Memori 46% pada *Server* dengan Sumberdaya terbesar.



Gambar 22. Grafik Persentase Penyerangan Port Knocking dan Honeypot menggunakan Denial of Service dan Bruteforce

4. Kesimpulan

Keamanan jaringan *Port Knocking* dan *Honeypot* dapat ditingkatkan keamanannya dengan melakukan penambahan konfigurasi *IPTables* pada firewall, untuk mengatasi serangan *Denial of Service* (*DoS*) dan *Brute force* menuju *port* 21, 22, 53, 80 dan 2222, pada server dengan Sumber daya CPU dan Memori yang lebih rendah dan lebih tinggi mempunyai tingkat keamanan port yang sama yaitu 80% uji penyerangan gagal dan hanya 20% penyerangan yang berhasil yaitu menuju port 22 yaitu port dari server bayangan *Honeypot*. Penggunaan sumber daya CPU dan Memori server mengalami penurunan penggunaan setelah penambahan *IPTables* dari pengujian serangan *Denial of Service* (*DoS*) dan *Brute force*, dibandingkan sebelum penambahan *IPTables*, dengan penggunaan menurun CPU sebesar 36% dan penggunaan memori juga menurun sebesar 41% pada *server* dengan Sumberdaya 1 CPUs dan 1015852KiB memori dibandingkan dengan *server* dengan sumber daya 2 CPUs dan 1507284KiB memori penggunaan CPU menurun sebesar 38% dan penggunaan memori juga menurun sebesar 44% dan dibandingkan dengan *server* dengan sumber daya 4 CPUs dan 6036624 KiB memori penggunaan CPU menurun sebesar 41% dan penggunaan memori juga menurun sebesar 46% dibandingkan dengan menggunakan metode *Port knocking* dan *Honeypot* saja.

Daftar Pustaka

[1] Brades, T., & Irwansyah. (2022). Pemanfataan Metode Port Knocking Dan Blocking Untuk Keamanan Jaringan Bpkad Provinsi Sumsel. pp. 99–107.

[2] Ernawati, T., & Rachmat, F, F, F. (2021). Keamanan Jaringan dengan Cowrie Honeypot dan Snort Inline-Mode sebagai Intrusion Prevention System. *J, RESTI*, Vol. 5, No. 1, pp. 180-186.

[3] Gunawan, A, R., Sastra, N, P., & Wiharta, D, M. (2021). Penerapan Keamanan Jaringan Menggunakan Sistem Snort dan Honeypot Sebagai Pendeteksi dan Pencegah Malware. *Majalah. Ilmiah. Teknologi. Elektro*, Vol. 20, No. 1, pp. 81–88.

[4] Wibawa, G, H, P., Sasmita, I, G, M, A., & Raharja, I, M, S. (2020). Analisis Data Log Honeypot Menggunakan Metode K-Means Clustering. *J. Ilmiah. Merpati*, Vol. 8, No. 1, pp. 13–21.

[5] Mardiansyah, A, Z., Abdussyakur, Y, M., & Jatmika, A, H. (2021). Optimasi Port Knocking Dan Honeypot Menggunakan Iptables Sebagai Keamanan Jaringan Pada Server. *JTIK*, Vol. 3, No. 2, pp. 189-199.

[6] Novianto, D., Japriadi, Y, S., & Tommy, L. (2023). Implementasi Multiple Port knocking dan Port blocking Untuk Peningkatan Keamanan Hak Akses Administrator Pada Routerboard Mikrotik. *JTKSI*, Vol. 6, No. 1, pp. 94–101.

[7] Pratama, R., Wijaya, A., Fatoni., & Suryayusra. (2022). Strategi Pengamanan Akses Jaringan Dengan L2TP Over IP Security Pre-shared Key. *J. JUPITER*, Vol. 14 No. 2, pp. 306–316.

[8] Anggreni, N, K., A, S., & Jasa, L. (2022). Literatur Review Analisis metode De-Militarized Zone (DMZ) dan Switch Port security Sebagai Metode Keamanan Jaringan. *Majalah. Ilmiah. Teknologi. Elektro*, Vol. 21, No. 2, pp. 195–200.

[9] Amien, J, A. (2020). Implementasi Keamanan Jaringan Dengan Iptables Sebagai Firewall Menggunakan Metode Port Knocking. *J. FASILKOM*, Vol. 10, No. 2, pp. 159–165.

[10] Rizal, R., Ruuhwan., & Nugraha, K, A. (2020). Implementasi Keamanan Jaringan Menggunakan Metode Port Blocking dan Port Knocking Pada Mikrotik RB-941. *J. ICT*, Vol. 19, No. 1, pp. 1-8.

[11] Rosyid, N, R., Murti, B, B., Prayudha, B., Ramadloni, A, F., & Subekti, L. (2023). Deteksi Malware pada Jaringan Lokal Berbasis Honeypot dan Yara. *J. Sistem Informasi*, Vol. 12, No. 1, pp. 186–193.

[12] Nida, N., & Adrian, R. (2023). Analisis Perbedaan Pengaruh Penggunaan Iptables Chains dalam Mencegah Denial of Service (DoS) pada Jaringan IoT. Journal of Internet and Software Engineering (JISE), Vol. 4, No.1.