JIMU: Jurnal Ilmiah Multi Disiplin
Vol. 03, N0. 01, Tahun 2025, Hal. 86 - 102, ISSN: 3031-9498 (Online)

Perancangan dan Analisis *Intrusion Detection System* Terhadap Serangan Probe Menggunakan Metode *Signature Based*

Muhammad Ardiansyah[1],Iwan Suhardi[2], Abdul Wahid[3]
[123]Universitas Negeri Makasssar
Email: muhammadardiansyah3703@gmail.com

Article Info

Article history:

Received September 25, 2024
Revised October 06, 2024
Accepted October 13, 2024

Keywords:

Network security, IP Tables, Port Knocking, Intrusion Detection System, Snort3.

ABSTRACT

This study designed and simulated a server network security system using signature-based methods, combining ip tables, port knocking, and analysis against probe and ICMP attacks with IDS Snort version 3 in inline mode and DAQ NFQ. The test results show that port knocking is effective in controlling access to the web server's ports, allowing only clients who know the combination of knock sequences to open ports. Snort version 3 can accurately detect probe attacks and ip tables drop attacks automatically, while the web server ports will be closed automatically after alert. Analysis of the ICMP attack shows that Snort processes a single request in 38 seconds with a packet processing speed of 3 packets per second, demonstrating adequate performance in detecting and dealing with threats.

Article Info

Article history:

Received September 25, 2024
Revised October 06, 2024
Accepted October 13, 2024

Keywords:

Kemanan Jaringan, Ip Tables, Port Knocking, Intrusion Detection System, Snort3.

ABSTRACT

Penelitian ini merancang dan mensimulasikan sistem keamanan jaringan server menggunakan metode signature-based, menggabungkan ip tables, port knocking, dan analisis terhadap serangan probe serta ICMP dengan IDS Snort versi 3 dalam mode inline dan DAQ NFQ. Hasil pengujian menunjukkan bahwa port knocking efektif dalam mengontrol akses ke port web server, hanya memungkinkan klien yang mengetahui kombinasi knock sequence untuk membuka port. Snort versi 3 dapat mendeteksi serangan probe secara akurat dan ip tables men-drop serangan otomatis, sementara port web server akan tertutup secara otomatis setelah alert. Analisis serangan ICMP dan Probe menunjukkan bahwa Snort memproses satu request dalam 38 detik dengan kecepatan pemrosesan paket 3 paket per detik, menunjukkan performa yang memadai dalam mendeteksi dan menangani ancaman.

Corresponding Author:

Nama penulis: Muhammad Ardiansyah
Universitas Negeri Makassar
Email: muhammadardiansyah3703@gmail.com

Pendahuluan

Keamanan jaringan sangatlah krusial untuk melindungi validitas dan integritas data serta menjamin ketersediaan layanan. Untuk itu, dibutuhkan sistem yang mampu memonitor setiap aktivitas pada jaringan komputer guna mengurangi risiko yang mungkin timbul[1] *Intrusion Detection System (*IDS*)* adalah sistem yang dirancang khusus untuk mengenali aktivitas mencurigakan pada jaringan atau sistem komputer. Sistem ini memantau lalu lintas jaringan dan aktivitas sistem guna mendeteksi tanda-tanda serangan atau kegiatan ilegal[2]. Namun demikian, terdapat permasalahan dalam mengenali pola paket data serangan di jaringan serta bentuk peringatan yang akan diberikan kepada administrator. Hal ini bertujuan untuk mempermudah administrator jaringan komputer dalam melakukan monitoring. Monitoring ini akan dilakukan menggunakan alat seperti Snort yang menerapkan metode berbasis tanda tangan (*signature-based*) untuk mengenali pola data normal dan pola data serangan dengan demikian, serangan di jaringan komputer yang dilakukan oleh penyerang dapat diketahui polanya dan dicegah secara otomatis[3].

Firewall adalah fitur keamanan pada jaringan komputer yang berfungsi mengatur lalu lintas paket data yang masuk dan keluar. Pada jaringan yang sederhana, *Firewall* umumnya hanya diimplementasikan pada komputer tertentu. Namun, *firewall* di komputer tersebut mungkin tidak cukup efektif untuk melindungi keamanan paket data yang diakses oleh penggunanya. Oleh karena itu, sebaiknya disiapkan *firewall* yang dapat melindungi pengguna dari akses baik internal maupun eksternal. Sistem *firewall* adalah salah satu perangkat lunak yang dapat mencegah beberapa serangan dari luar, namun *firewall* tidak selalu dapat memberikan peringatan untuk serangan yang lebih kompleks, seperti DdoS atau serangan pada *port* tertentu[4].

ip tables adalah alat yang digunakan untuk menyaring lalu lintas data pada sistem operasi Linux, serta mengatur lalu lintas data tersebut. IP Tables memiliki tiga jenis aturan dalam tabel penyaringan, yang dikenal sebagai firewall chain, yaitu *INPUT*, *OUTPUT*, dan *FORWARD*. Selain itu, *IP Tables* juga memiliki tiga tabel utama: *NAT*, *MANGLE*, dan *FILTER*. IP Tables adalah *firewall* populer dan kuat yang tersedia di sistem operasi Linux. Fungsinya mencakup konfigurasi, pemeliharaan, dan pemeriksaan tabel aturan (*rules tables*) untuk penyaringan paket IP yang terdapat di kernel Linux. Penyaringan ini dilakukan melalui *filter*[5].

Seperti yang dilakukan oleh Idrus dkk [6] yaitu *instrusion detection system* untuk keamanan *cloud* dengan menggunakan sucirata untuk mencegah ancaman informasi yang terkait dengan komputasi awan. Penelitian lain oleh Siswanto dkk [7] yaitu *instrusion detection system* untuk keamanan jaringan dengan menggunakan metode *signature based* untuk memitigasi serangan *blackhole*, dimana penulis memperoleh adalah *delay* sebesar 0.19 s atau 190 ms, throughput sebesar 95.18 kbbps dimna *delay* yang baik adalah nilainya dibawah 150 ms,pengujian skenario 2 menunjukkan hasil pengukuran ini sebesar 0.05 s atau 50 ms.

Selanjutnya penelitian dari Meiditra dkk [8] tentang *instrusion detection system* untuk pengamanan *private cloud storage* dengan metode *instrusion detection system* (IDS*)* dan *instrusion prevention system (*IPS*)* Hasil dari penelitian ini metode ids mampu mendeteksi penyusup dan mampu menganalisis lalu lintas *real-time* hal ini dapat mendeteksi berbagai jenis serangan masuk .pada *private cloud storage* menggunakan proxmox VE dapat dilakukan secara virtual dengan menggunakan VMware Workstation 16 sebagai *prompt console* nya dan dikonfigurasi di browser sebagai *cloud server* nya.

Namun demikian, terdapat permasalahan dalam mengenali pola paket data serangan di jaringan serta bentuk peringatan yang akan diberikan kepada administrator Hal ini bertujuan untuk mempermudah administrator jaringan komputer dalam melakukan monitoring. Monitoring ini akan dilakukan menggunakan alat seperti Snort yang menerapkan metode

berbasis tanda tangan (*signature-based*) untuk mengenali pola data normal dan pola data serangan dengan demikian, serangan di jaringan komputer yang dilakukan oleh penyerang dapat diketahui polanya dan dicegah secara otomatis [9].

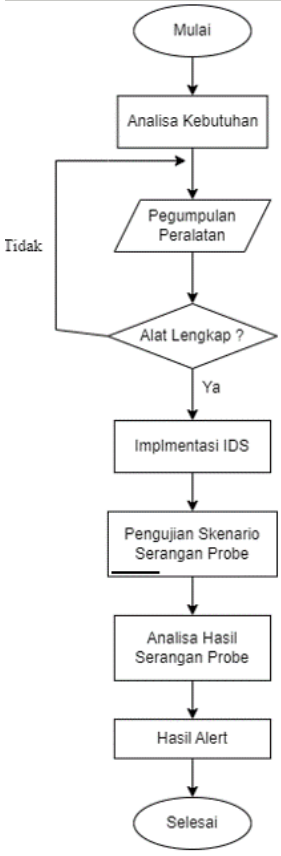
Penelitian ini akan memfokuskan pada perancangan dan anlisis intrusion detection system terhadap serangan probe menggunakan metode signature-based bertujuan yang mampu mendeteksi serangan *probe*, yaitu upaya yang dilakukan oleh pihak yang tidak sah untuk memetakan atau mengumpulkan informasi tentang sistem keamanan jaringan. Untuk mendeteksi intrusi pada sistem server jaringan, diperlukan metode berbasis tanda tangan (*signature-based*) yang menyadap paket data dan membandingkannya dengan database aturan *IDS* (yang berisi tanda tangan paket serangan)[10]. Jika paket data memiliki pola yang sama dengan salah satu pola dalam *database* aturan IDS, paket tersebut dianggap sebagai serangan. Jika tidak, paket tersebut dianggap bukan serangan. Mesin IDS kemudian membaca peringatan dari IDS (seperti jenis serangan dan alamat IP penyusup) dan memerintahkan *firewall* untuk memblokir akses koneksi dari penyusup. Selanjutnya, peringatan dari IDS akan dipantau dan notifikasi serangan dalam database snort 3 yang telah dikonfigurasi sebelumnya[11].

Metode

Metode penelitian yang digunakan yaitu metode eksperimen dimana dilakukan percobaan menggunakan sistem operasi Ubuntu dan kali linux.

1. Tahapan Penelitian

Flowchart dibawa ini merupakan tahapan dari proses penelitian.

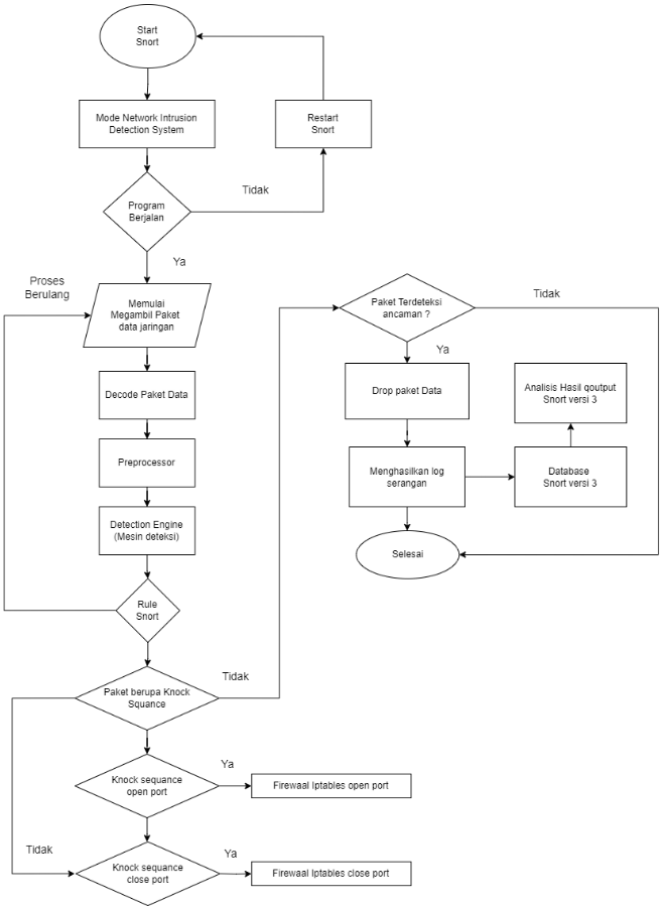


Gambar 1. Diagram Alir Perancangan dan Pengujian

Pada Gambar 1 menunjukkan tahapan penelitian yang dijabarkan flowchart penelitian tahpan dimulai dimulai dari analisa kebutuhan, di mana dilakukan identifikasi atas apa yang

dibutuhkan untuk menjalankan IDS[12]. Selanjutnya, peralatan yang diperlukan dikumpulkan. Jika alat tidak lengkap, proses kembali ke pengumpulan peralatan, namun jika peralatan sudah lengkap, implementasi IDS dapat dilakukan. Setelah IDS diimplementasikan, dilakukan pengujian dengan skenario serangan "probe" untuk menguji ketahanan sistem[13]. Hasil dari serangan tersebut dianalisis untuk melihat efektivitas IDS dalam mendeteksi intrusi, yang kemudian menghasilkan laporan peringatan (alert). Diagram ini berakhir setelah hasil alert dianalisis dan sistem kembali dipantau

2. Flowchart Sistem

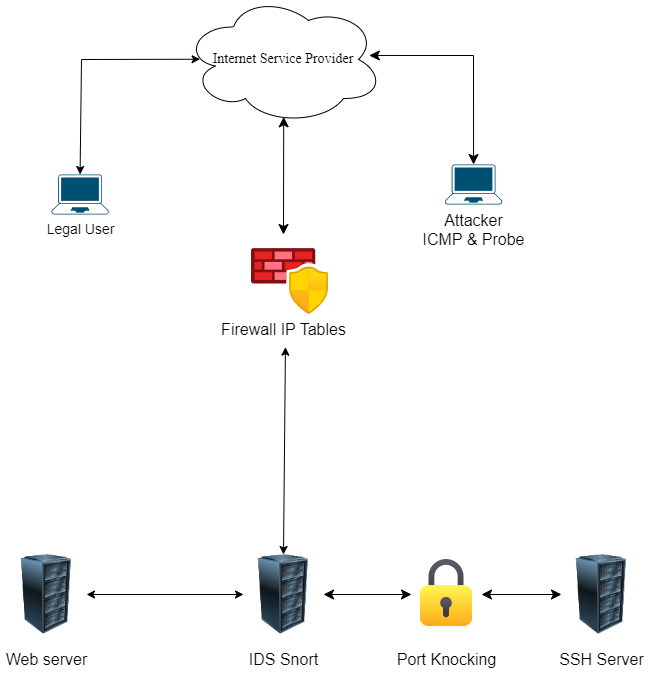


Gambar 2. Flowchart Sistem

Pada Gambar 2 menjelaskan alur kerja dari Snort, sebuah sistem deteksi intrusi jaringan (Network Intrusion Detection System atau NIDS), yang dimulai dengan memulai program dalam mode NIDS untuk memonitor lalu lintas jaringan. Program ini mengumpulkan paket data dan mendekode paket tersebut sebelum diproses lebih lanjut oleh preprocessor. Kemudian, data akan melewati mesin deteksi yang memeriksa ancaman berdasarkan aturan yang ditetapkan (Snort Rule) [14]. Jika ditemukan paket yang cocok dengan urutan knock sequence, sistem firewall melalui iptables akan membuka port yang diperlukan dan menutupnya kembali setelah urutan selesai. Jika paket tersebut terdeteksi sebagai ancaman, paket akan di-drop dan log serangan akan dihasilkan untuk analisis lebih lanjut [15]. Hasil dari deteksi ini kemudian disimpan di database Snort versi 3 untuk evaluasi dan tindakan selanjutnya.

3. Desain

Pemodelan yang digunakan pada penelitian ini adalah model arsitektur jaringan yang melibatkan berbagai elemen keamanan dan komunikasi antara perangkat yang berbeda melalui penyedia layanan internet (ISP).

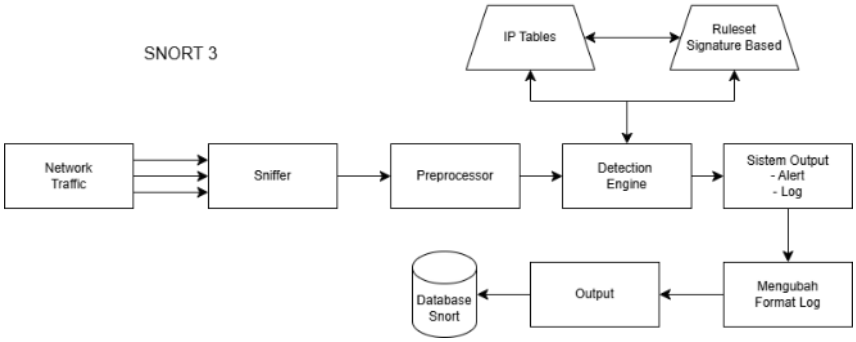


Gambar 3. Topologi Intrusion Detection System

Gambar 3 merupakan desain dari instrusion detection system Di bagian atas, dua komputer terhubung melalui jaringan yang dilindungi oleh firewall dan sistem keamanan, yang menjamin keamanan data selama komunikasi. Server yang berada di bagian bawah menunjukkan komponen penyimpanan atau hosting, dengan simbol gembok yang menunjukkan bahwa data yang dikirimkan ke dan dari server tersebut dienkripsi. Elemen-elemen ini menunjukkan adanya mekanisme pengamanan seperti firewall ip tables akan memblokir ketika terjadinya serangan jika serangan terdeteksi oleh detection engine snort 3 mendeteksi dan Salah satu server memiliki port knocking, yang menunjukkan bahwa data tersebut dapat di akases dengan menggetikkan knock sequance sehingga dapat membuka atau menutup lalu lintas request dienkripsi atau dilindungi untuk menjaga kerahasiaan dan integritas informasi. Diagram ini menekankan perlunya keamanan jaringan, firewall, dan enkripsi dalam proses komunikasi data.

a. Skema Pengujian ICMP

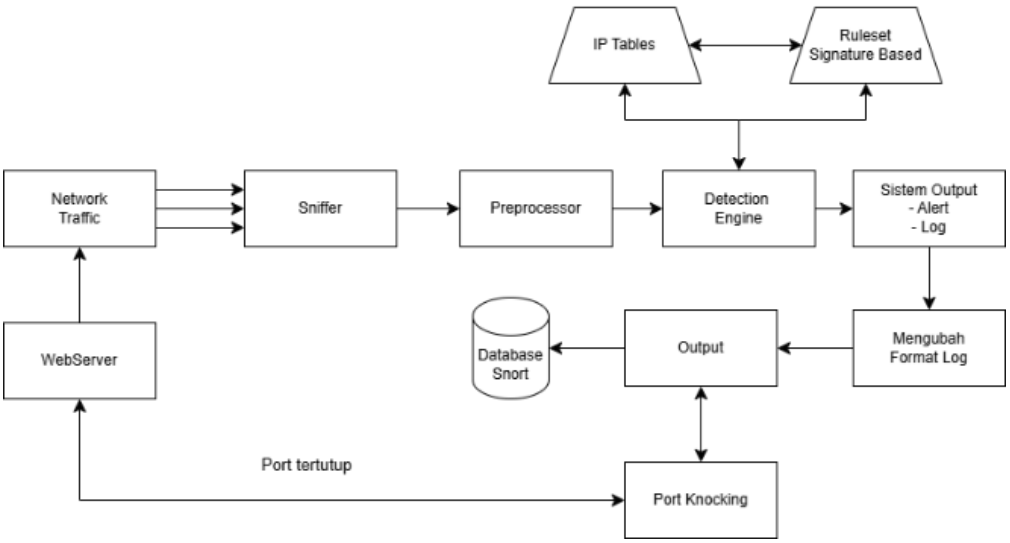
Pada bagian ini, terdapat skenario uji coba yang akan dilakukan pada pihak penyerang, yakni skenario uji coba serangan dengan menggunakan ICMP dengan menggunakan system operation kali linux dengan penyerang mencoba tekoneksi dengan server dengan ip address 192.168.1.40.



Gambar 4. Skema Pengujian ICMP

Gambar 4. menggambarkan alur kerja dari sebuah sistem Network Intrusion Detection System (NIDS) yang menggunakan perangkat lunak Snort untuk mendeteksi ancaman atau aktivitas berbahaya dalam jaringan komputer. Sniffer bertindak sebagai alat penangkap paket data di jaringan. Fungsinya adalah untuk menangkap setiap paket yang lewat, tanpa memandang apakah paket tersebut berisi data yang aman atau berbahaya. Data yang telah ditangkap oleh sniffer kemudian masuk ke tahap Preprocessor Detection Engine, yang merupakan inti dari sistem deteksi intrusi. Mesin deteksi ini menggunakan pendekatan signature-based detection, di mana ia mencocokkan paket-paket data yang telah diproses dengan database tanda tangan (signatures) serangan yang telah dikenal. Signature ini adalah pola atau karakteristik khusus yang menunjukkan adanya aktivitas berbahaya, seperti serangan icmp Hasil dari proses deteksi ini akan menghasilkan Sistem Output. Output ini bisa berupa Alert atau Log. Alert digunakan untuk memberikan peringatan langsung kepada administrator jaringan bahwa telah terjadi aktivitas mencurigakan atau serangan di dalam jaringan. Sedangkan Log merupakan catatan detil dari semua aktivitas yang telah dianalisis, baik itu aktivitas normal maupun mencurigakan.

b. Skema Pengujian ICMP



Gambar 5. Skema Pengujian ICMP

Gambar 5. Menggambarkan skema komprehensif dari sebuah sistem keamanan jaringan yang memanfaatkan berbagai komponen untuk mendeteksi, merekam, dan merespons ancaman dalam jaringan komputer secara efektif. Proses dimulai dengan Network Traffic dan WebServer, yang merupakan dua sumber utama data dalam jaringan ini. Network Traffic mencakup seluruh lalu lintas data yang melewati jaringan, baik yang berasal dari internal jaringan maupun dari luar, seperti internet. Tahap berikutnya adalah Preprocessor, yang

merupakan modul yang bertugas untuk melakukan pra-pemrosesan terhadap paket data yang telah ditangkap. Preprocessing melibatkan berbagai proses seperti fragmentasi paket, normalisasi, dan pembersihan data dari elemen-elemen yang tidak diperlukan. Misalnya, preprocessor dapat menyusun kembali paket-paket data yang terfragmentasi atau mengurai data agar sesuai dengan protokol tertentu. Secara keseluruhan, diagram ini menunjukkan alur kerja terperinci dari sistem deteksi dan respons ancaman dalam jaringan yang menggabungkan beberapa komponen penting untuk menjaga keamanan jaringan. Mulai dari penangkapan data hingga deteksi ancaman, perekaman, dan respons, skema ini menekankan pentingnya setiap tahap dalam menjaga integritas, kerahasiaan, dan ketersediaan jaringan. Integrasi dengan IP Tables dan Port Knocking juga memperlihatkan pendekatan proaktif dalam melindungi jaringan, memastikan bahwa sistem ini tidak hanya mendeteksi ancaman, tetapi juga mengambil tindakan pencegahan yang diperlukan untuk mencegah akses yang tidak sah.

Hasil

1. Analisa Kebutuhan

Dalam kebutuhan untuk perancangan dan analisis instrusion detection system terhadap serangan probe menggunakan metode signature based untuk mendeteksi serangan probe menggunakan metode *signature-based* mencakup identifikasi dan pemahaman mendalam tentang serangan probe yang akan dideteksi, serta karakteristik lingkungan jaringan tempat IDS akan diimplementasikan. Pertama, perlu dilakukan identifikasi jenis-jenis serangan *probe* seperti IP *sweep*, *port scan*, dan *vulnerability scan*, serta memahami pola dan tanda-tanda spesifik dari setiap serangan ini untuk membuat tanda tangan (*signature*) yang akurat.

2. Pengumpulan Peralatan

Dalam perancangan instrusion detection system menggunakan snort generasi terbaru yaitu versi 3 yang akan dilakukan terdapat beberapa peralatan, berikut detail spesikasi perangkat keras maupun perangkat lunak yang dibutuhkan pada penelitian ini dapat dilihat pada tabel 1.

Tabel 1. Perangkat Keras

No	Spesikasi	Detail
1	Processor	*AMD Ryzen 5 4500U*
2	RAM	12 GB
3	Penyimpanan	SSD 512 GB

Berikut perangkat lunak yang digunakan terdapat nama dan versi perangkat lunak yang digunakan dalam penelitian ini dapat di lihat Tabel 2

Tabel 2. Perangkat Lunak

No	Nama	Versi	Fungsi
1	Snort	3.2.39.0	*Tool Intrusion Detection System*
2	Ubuntu	22.04	Sistem Operasi yang digunakan ids dan web server
3	Kali Linux	2022.2	Sistem operasi attacker
4	Probe	8.91	Tool untuk *peneration testing*

3. Implmentasi

A. Konfigurasi IDS Snort 3

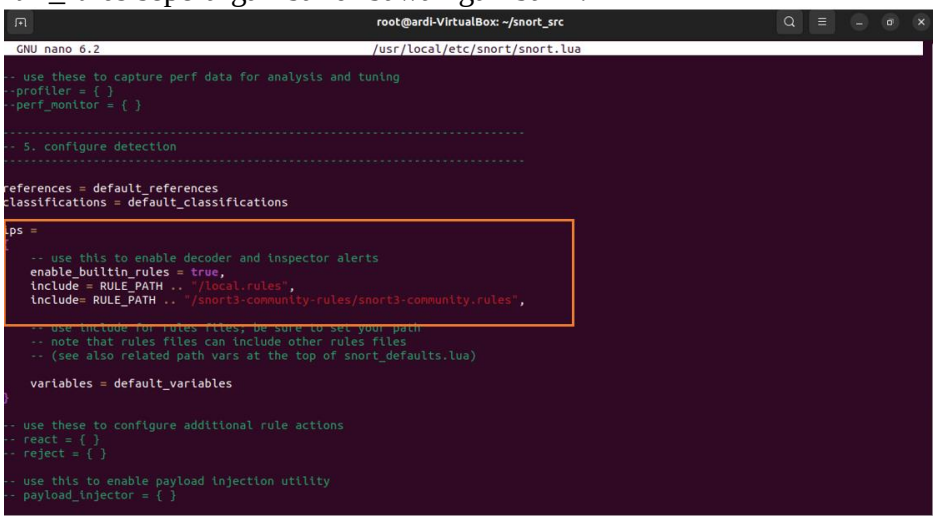
Snort intrusion detection system, IDS merupakan sumber terbuka terkemuka di dunia telah resmi merilis snort 3 pada bulan januari 2021 untuk melakukanya kita terlebih dahulu mengginstal plugin snort 3 di situs resmi https://www/snort.org dengan mengetikkan perintah di ubuntu server git clone jika sudah terinstall maka akan muncul seperti gambar 6.

```
root@ardi-VirtualBox:~/snort_src# snort -v
--------------------------------------------------
o")~   Snort++ 3.2.2.0
--------------------------------------------------
--------------------------------------------------
Network Policy : policy id 0 :
--------------------------------------------------
Inspection Policy : policy id 0 :
--------------------------------------------------
pcap DAQ configured to passive.
--------------------------------------------------
host_cache
    memcap: 33554432 bytes

Snort successfully validated the configuration (with 0 warnings).
o")~   Snort exiting
```

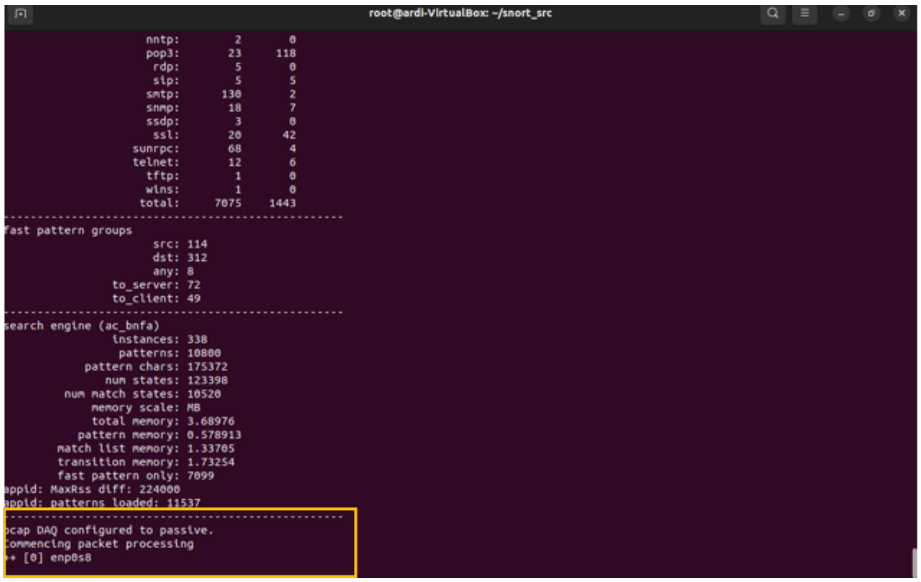
Gambar 6. Melihat Versi Snort 3

Setelah kita mengginstall snort 3 di situs resminya selanjutnya untuk menggaktifkan peringatan dekoder dan inspektur (lalu lintas berbahaya yang diidentifikasi oleh snort, disini kita menempatkan ruleset signature based ke dalam detection engine pada gambar 5. Agar ruleset terbaca ke dalam sistem cari ips pada bagian 182 yang di beri ips dan hapus komentar *enable_builtin_rules* seperti gambar di bawah gambar 7.



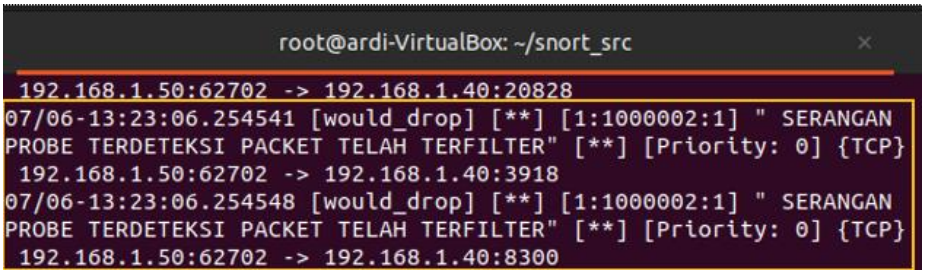
Gambar 7. Validasi *Open Appid Snort 3*

Setelah menjalankan perintah untuk mendeteksi serangan melalui IDS/IPS *snort* 3 di dapatkan serangan berupa *alert* di *local.rules* yang sudah terlebih dahulu kita masukkan *alert. Alert* dapat dilihat pada gambar 8.



Gambar 8. Pendeteksian Serangan Snort 3

Setelah menjalankan perintah untuk mendeteksi serangan melalui IDS/IPS *snort* 3 di dapatkan serangan berupa *alert* di *local.rules* yang sudah terlebih dahulu kita masukkan *alert. Alert* dapat dilihat pada gambar 8



Gambar 8. Output Pendeteksi Snort 3

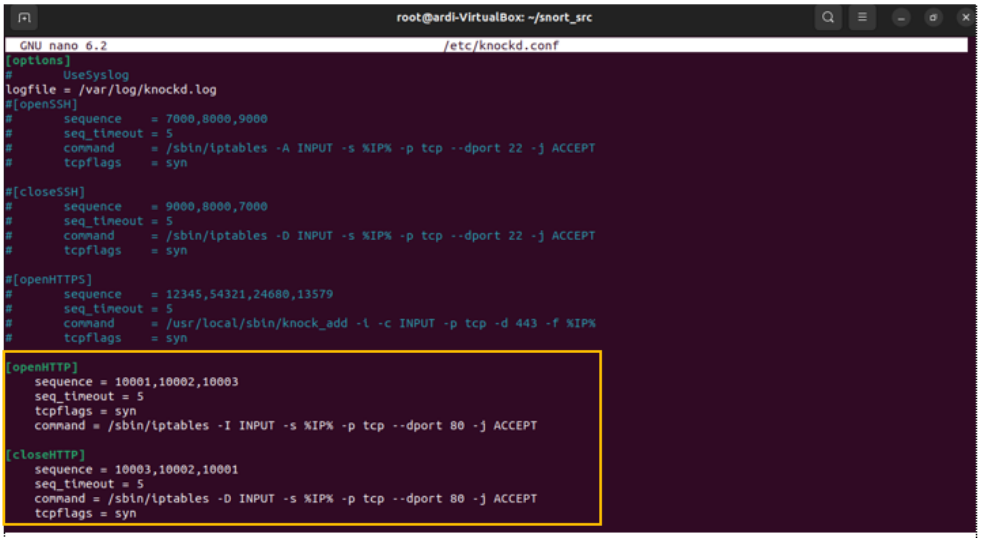
B. Konfigurasi Port Knocking

Pada konfigurasi *port knocking* telah diimplmentasikan di server dengan menggunakan knockd. *Port* knocking memberikan lapisan keamanan tambahan dengan memastikan bahwa hanya *legal user* yang mengetahui urutan port yang benar yang dapat mengakses layanan tertentu. Pada langkah pertama kita terlebih dahulu mengginstall knock off dari repositori perangkat lunak *default* seperti Gambar 9.

```
root@ardi-VirtualBox:~/snort_src# apt-get install knockd -y
```

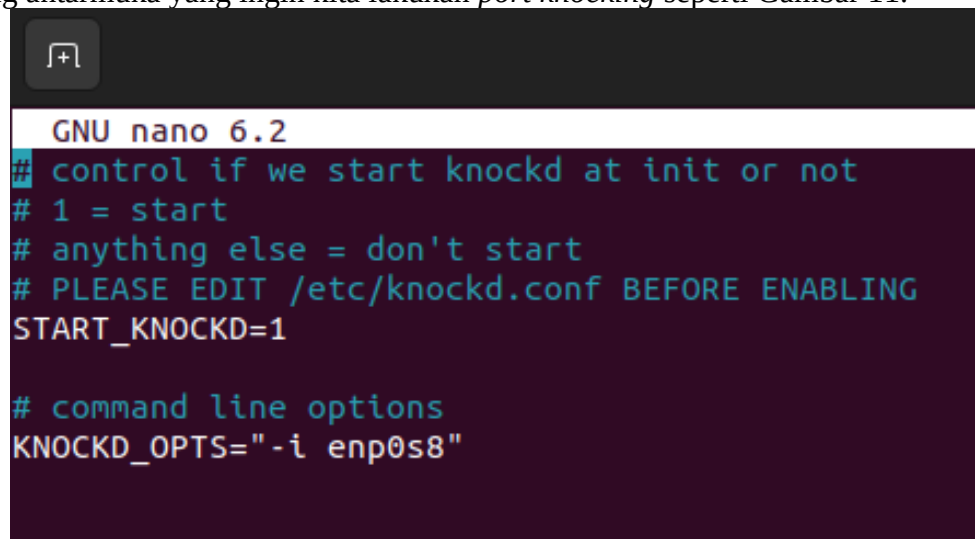
Gambar 9. Konfigurasi Install Port Knocking

Langkah selanjutnya kita konfigurasi *port knocking* dengan editor teks baris perintah *sudo nano /etc/knockd.conf* untuk masuk ke dalam pengaturan *port knocking* kita harus perlu mengubah tiga item dalam konfigurasi seperti Gambar 10.



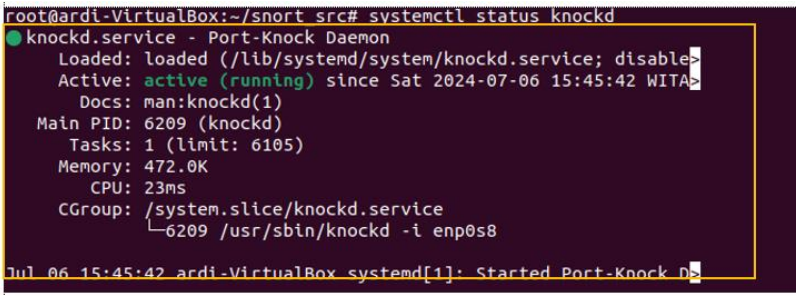
Gambar 10. Konfigurasi Port Knocking

Pada tahapan selanjutnya buka konfigurasi *default knocked* dengan mengetikkan perintah sudo nano /etc/default/knockd temukan baris *start_knockd=0* lalu ubah menjadi 1 untuk mengaktifkan *autostart* saat sistem *boot* setelah kita harus mengubah *knockd_OPTS= I enp0s8* (*interface network ubuntu*) kita harus menyesuaikan antarmuka agar nantinya dapat terhubung antarmuka yang ingin kita lakukan *port knocking* seperti Gambar 11.



Gambar 12. Konfigurasi *Default Port Knocking*

Langkah selajutnya kita mulai otomatis *port knocking* dengan perintah *sudo systemctl start knockd*, periksa status port knocking *systemctl status knockd* seperti Gambar 12.



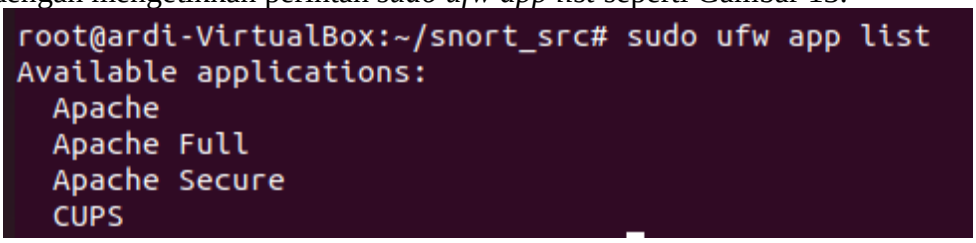
Gambar 12. Status *port knocking*

Pada Konfigurasi *port knocking* bahwa *port* yang dilindungi tetap tertutup dan tidak terlihat oleh pengguna yang tidak mengetahui urutan yang benar, sehingga meningkatkan keamanan terhadap percobaan akses tidak sah Gambar 13.

Gambar 12. *Rule iptables*

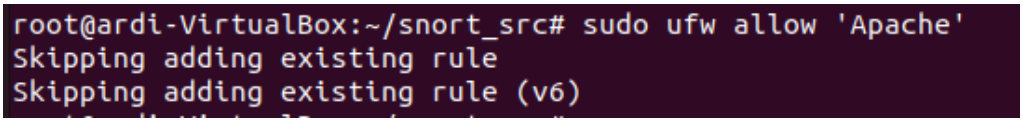
C. Konfigurasi Web Server

Pada konfigurasi web server melakukan install repositori perangkat lunak dengan perintah *sudo apt install apache2* sebelum menguji *apache*, kita terlebih dahulu perlu memodifikasi pengaturan *firewall* untuk mengizinkan akses dari luar ke *port web* ,selama instalasi apache mendaftarkan drinya dengan ufw untuk menyediakan beberapa profil aplikasi yang dapat digunakan untuk mengaktifkan atau menonaktifkan akses ke *apache* melalui *firewall* dengan mengetikkan perintah *sudo ufw app list* seperti Gambar 13.



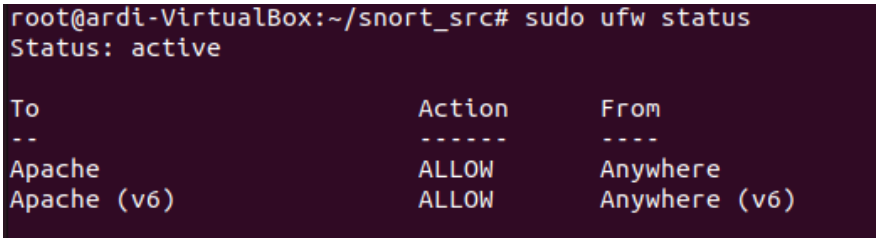
Gambar 13. *Ufw Firewall Apache*

Selanjutnya kita mengaktifkan profil paling ketat yang akan mengizinkan lalu lintas yang telah sudah dikonfigurasikan dengan mengetikkan perintah *sudo ufw allow 'Apache'* seperti Gambar 14.



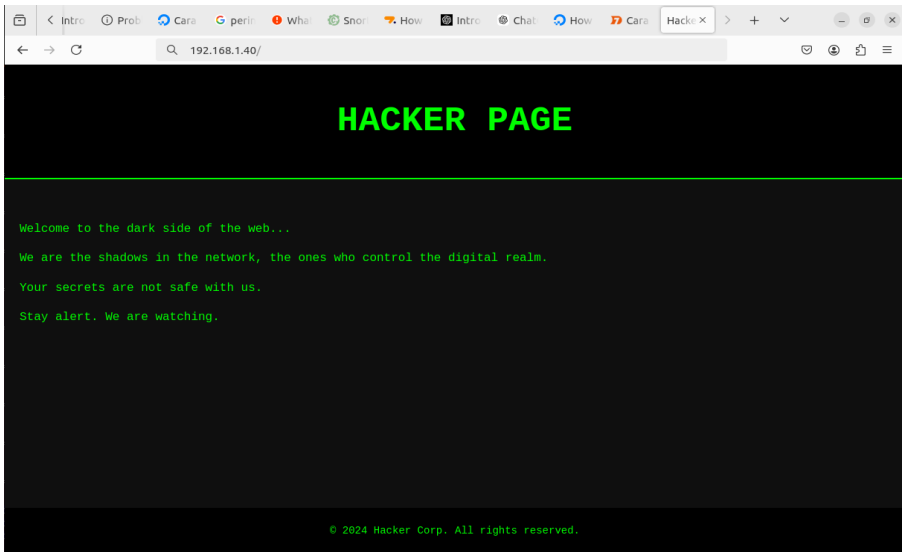
Gambar 14. Mengizinkan Lalu Lintas

Untuk dapat memverifikasi perubahan kita harus mengetikkan perintah *sudo ufw status* seperti Gambar 15.



Gambar 15. Validasi Perubahan Web Server

Setelah konfigurasi selesai, server harus diuji untuk memastikan semua pengaturan bekerja dengan benar dan situs *web* dapat diakses oleh pengguna.yang nantinya setelah kita lakukan konfiguras *web* server terseburt kita dapat melihatnya di *browser* dengan mengetikkan https:/192.168.1.40 sesuai IP ubuntu server kita seperti Gambar 20.

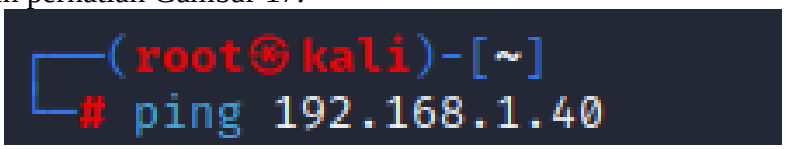


Gambar 16. Web Server di Ubuntu Server

D. Pengujian Skenario Serangan ICMP dan Probe

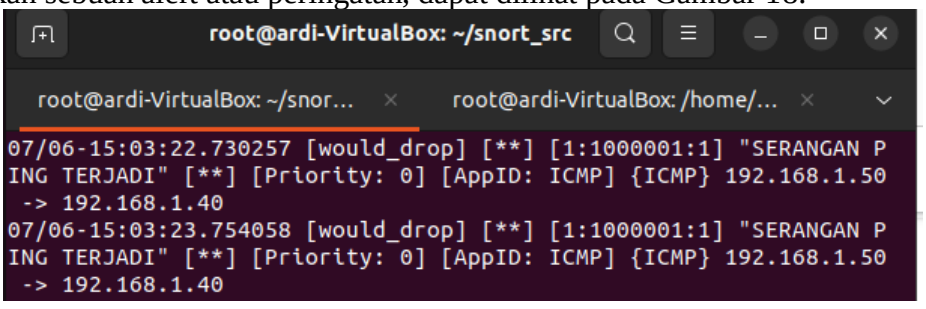
a. ICMP

Pada komputer yang telah dilengkapi dengan perangkat lunak *Snort* akan berusaha untuk mendeteksi adanya serangan atau aktivitas yang tidak biasa dalam jaringan yang terhubung ke server tersebut. ICMP, atau *Internet Control Message Protocol*, adalah protokol yang digunakan untuk mengirimkan pesan kesalahan dan informasi mengenai kondisi jaringan yang memerlukan perhatian Gambar 17.



Gambar 17. Ping Ip Address di ubuntu Server

Pada Gambar 17 attacker mencoba meminta request dengan mencoba ping ke server dengan ip 192.168.1.40 ,dan berhasil terhubung dengan server. Pada Gambar 22 menggambarkan bagaimana tampilan peringatan ini muncul pada antarmuka Snort, memberikan informasi yang jelas tentang jenis aktivitas yang terdeteksi berdasarkan aturan yang telah diatur sebelumnya. Ketika Snort mendeteksi aktivitas ping yang sesuai dengan pola atau kriteria tertentu yang telah didefinisikan dalam aturan (rules) yang telah dibuat, Snort akan menghasilkan sebuah alert atau peringatan, dapat dilihat pada Gambar 18.



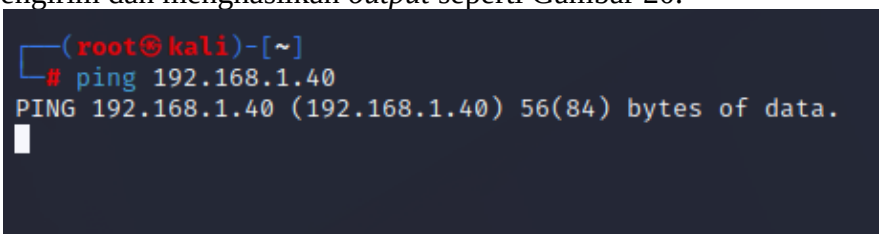
Gambar 18. Hasil Alert Snort Mendeteksi Serangan ICMP

Pada snort sudah menerapkan IP *Tables* di dalamnya yang dikonfigurasi menggunakan ufw pada sistem Linux untuk mengontrol lalu lintas jaringan yang masuk. Aturan-aturan ini termasuk menolak koneksi TCP dengan *flags* SYN, ACK/SYN, ACK, yang menunjukkan upaya serangan atau pemindaian port, mengizinkan semua lalu lintas yang sudah ada dalam

koneksi terkait atau sudah ada (*ctstate RELATED, ESTABLISHED*), dan menolak koneksi dari 192.168.1.40 dengan mengirim pesan ICMP *'port unreachable'*. Aturan terakhir secara spesifik menolak semua lalu lintas ICMP ,yang mungkin dianggap sebagai sumber serangan. Gambar 20 di bawah merupakan konfigurasi untuk mendrop serangan icmp.

Gambar 19. Konfigurasi IP Tables di Dalam Ids Snort

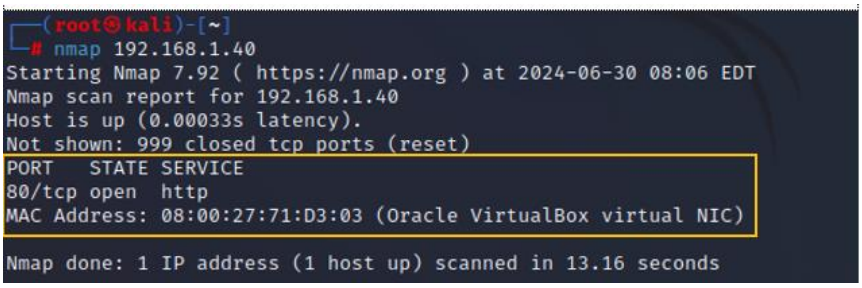
Pada Gambar 19 menunjukkan aktivitas ping 192.168.1.40 yang dieksekusi dari terminal di mesin Kali Linux, mengirimkan paket ICMP *Echo Request* ke alamat IP 192.168.1.40. Dalam konteks ini, Snort3 aktif dan dikonfigurasi untuk mendeteksi paket ICMP, sehingga dapat menghasilkan *alert* untuk setiap upaya pengiriman ICMP *Echo Request* berdasarkan aturan yang telah ditentukan. Sementara itu, *iptables* diatur untuk memblokir paket ICMP, baik untuk *Echo Request* masuk maupun *Echo Reply* keluar, yang berarti meskipun perintah ping mengirimkan paket, *respons* dari alamat tujuan akan diblokir dan tidak diterima oleh mesin pengirim dan menghasilkan *output* seperti Gambar 20.



Gambar 20. Pemblokiran ICMP Ping ke Client oleh IP Tables

b. Probe

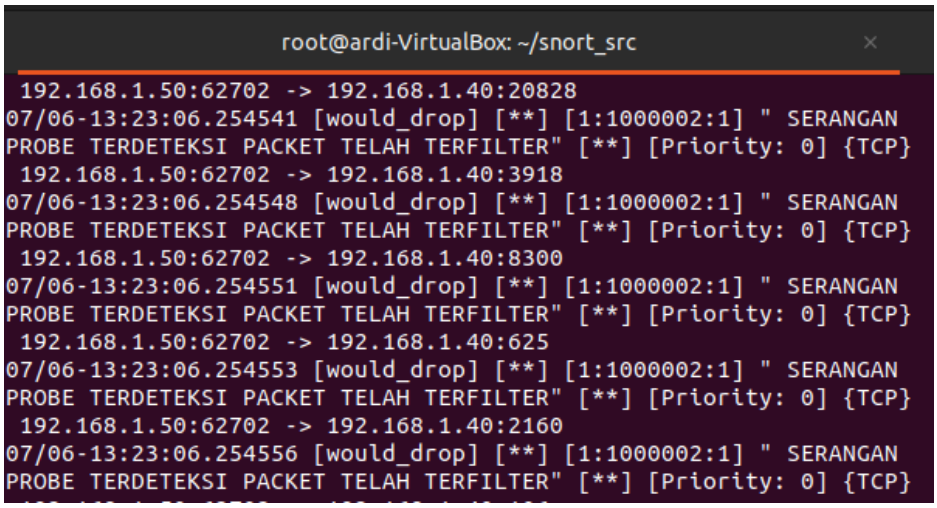
Komputer yang sudah terpasang snort 3 akan dicoba untuk melakukan metode penyerangan probe menggunakan nmap yaitu serangan *port scanning* dari komputer penyerang. Nmap alat yang digunakan untuk pemetaan jaringan dan keamanan, memungkinkan administrator sistem dan profesional keamanan untuk memindai dan menganalisis jaringan untuk menemukan perangkat aktif, mengidentifikasi layanan yang berjalan pada perangkat tersebut, dan mendeteksi potensi kerentanannya. Pada gambar 22 dapat dilihat hasil setelah melakukan *port scanning* terhadap jaringan server. Dari hasil pemindaian ini, ditemukan bahwa jaringan server memiliki satu *port* yang terbuka, yaitu *port* 80, yang digunakan untuk layanan HTTP. *Port* 80 yang terbuka ini menunjukkan adanya celah keamanan di mana penyerang potensial dapat mencoba masuk ke dalam jaringan melalui layanan *web* yang berjalan pada port tersebut.



Gambar 20. *Port Scanning Ke* Ip Client Dengan Menggunakan Nmap

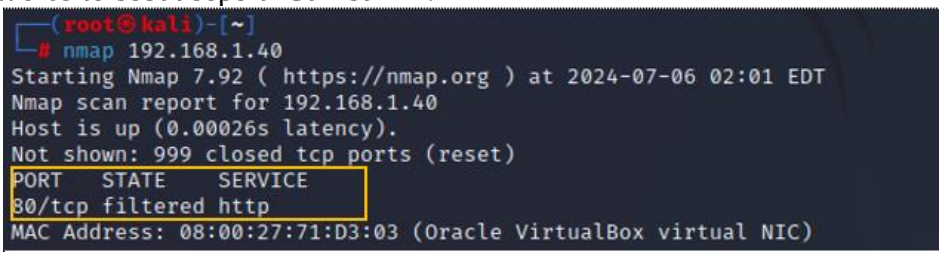
Snort versi 3 mendeteksi adanya serangan terhadap jaringan. Setelah *Snort* mendeteksi aktivitas mencurigakan yang sesuai dengan aturan yang telah dikonfigurasi, *Snort* segera

mengirimkan peringatan kepada administrator jaringan. Peringatan ini berfungsi sebagai notifikasi cepat agar administrator dapat segera mengambil tindakan yang diperlukan untuk mengatasi ancaman tersebut. Hasil dari deteksi ini menunjukkan bahwa serangan berhasil dideteksi oleh *Snort*, dan sebagai respons otomatis, *Snort* men-*drop* atau menghentikan paket yang mencurigakan tersebut, sehingga mencegahnya mencapai tujuannya. Di tunjukkan pada Gambar 20



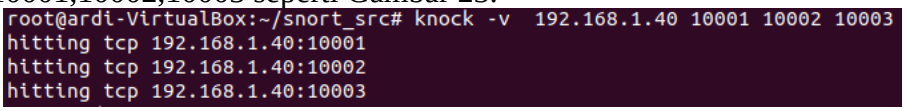
Gambar 21. Hasil Serangan *Probe* Menggunakan Metode TCP

Proses ini menunjukkan betapa pentingnya memiliki sistem IDS yang andal untuk menjaga keamanan dan integritas jaringan dari berbagai ancaman siber.dan menghasilkan *Output* ip *tables* tersebut seperti Gambar 22.



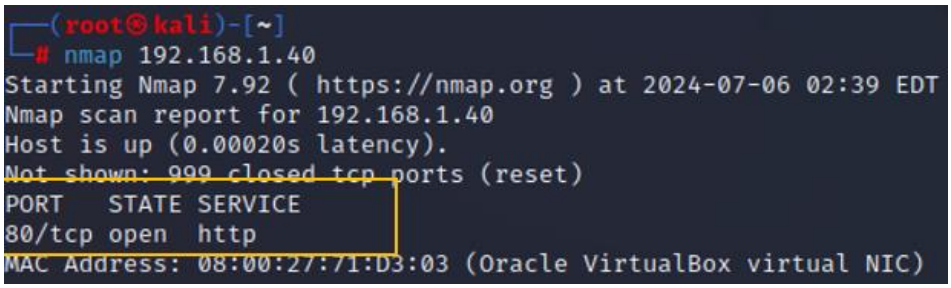
Gambar 22. Penutupan *Port* oleh iptables

Selanjutnya untuk membuka port yang tertutup oleh karena terjadinya serangan *probe* untuk membuka port ketikkan perintah yang sudah di lakukan sebelumnya dengan menuliskan kombinasi 10001,10002,10003 seperti Gambar 23.



Gambar 23. Penutupan *Port* oleh iptables

Ketika *client legal* mengecek kembali apakah *port sequance* berhasil dengan cara mengecek melalui terminal kali linux dan menghasilkan *port* yang terbuka seperti Gambar 24



Gambar 23. *Knock Sequance* Untuk Membuka Port

Pembahasan

Pada penelitian ini dilakukan perancangan dan analisis intrusion detection system terhadap serangan probe menggunakan metode signature based . intrusion detection system digunakan untuk mendeteksi serangan sementara signature based metode keamanan yang mendeteksi ancaman atau aktivitas berbahaya dengan menggunakan tanda tangan atau pola yang telah dikenal. Tanda tangan ini merupakan representasi unik dari suatu ancaman, seperti urutan karakter atau bit yang menunjukkan adanya virus, worm, atau jenis serangan tertentu. Sistem keamanan membandingkan aktivitas yang sedang berlangsung dengan tanda tangan yang ada dalam database untuk mendeteksi ancaman yang sudah dikenal dan pada penelitian ini dikombinasikan keamanan menggunakan port knocking, ip tables port knocking digunakan untuk megamankan akses ke port HTTPS sementara ip tables digunakan untuk mereject atau mendrop serangan

Hasil analisis pengujian serangan ICMP seperti yang terlihat di tabel 3. menunjukkan hasil pengujian penanganan serangan ICMP *(Internet Control Message Protocol)* pada alamat IP 192.168.1.50, yang dilakukan untuk mengevaluasi efektivitas deteksi dan mitigasi serangan menggunakan *snort* dan IP *Tables.* Tabel mencakup beberapa kombinasi status serangan ICMP (aktif atau tidak aktif), status *snort* (aktif atau tidak aktif), dan status IP *Tables* (aktif atau tidak aktif), serta memberikan keterangan tentang hasil dari setiap kombinasi. Ketika serangan ICMP aktif dan kedua mekanisme pertahanan *(snort* dan IP *Tables)* juga aktif, hasilnya menunjukkan bahwa serangan berhasil dideteksi dan di-*drop,* menandakan bahwa kombinasi ini efektif dalam menangani serangan, Saat ICMP aktif, *snort* tidak aktif, tetapi IP *Tables* aktif, serangan tidak terdeteksi dan tidak di-*drop,* menunjukkan bahwa snort memiliki peran penting dalam mendeteksi serangan sementara IP *Tables* hanya dapat memblokir serangan yang terdeteksi, Jika ICMP aktif, *snort* aktif, tetapi IP *Tables* tidak aktif, serangan berhasil dideteksi oleh s*nort* namun tidak di-*drop*, mengindikasikan bahwa IP *Tables* berperan penting dalam aksi pencegahan terhadap serangan yang terdeteksi oleh *snort,* Untuk kondisi di mana ICMP tidak aktif dan IP *Tables* aktif tetapi *snort* tidak aktif, serangan tidak terdeteksi atau di-*drop*, karena tidak ada lalu lintas serangan ICMP yang harus ditangani.

Hasil analisis Pengujian serangan Probe seperti yang terlihat di tabel 4.5 menunjukkan hasil pengujian serangan Probe pada IP Address 192.168.1.50, dengan berbagai kombinasi status serangan (aktif/tidak aktif), status Snort/IP Tables (aktif/tidak aktif), dan penggunaan teknik Port Knocking (benar/salah). Setiap baris dalam tabel ini mencerminkan skenario pengujian yang berbeda dan keterangan hasil yang dicapai. Dalam skenario pertama, di mana serangan Probe aktif, Snort dan IP Tables juga aktif, dan teknik Port Knocking dilakukan dengan benar, hasilnya menunjukkan bahwa serangan berhasil dideteksi dan di-drop, sementara client legal berhasil mengakses urutan knock yang benar, mengindikasikan mekanisme

pertahanan bekerja optimal. Pada skenario kedua, serangan Probe tidak aktif, Snort aktif tetapi IP Tables tidak aktif, dan teknik Port Knocking salah, hasil menunjukkan bahwa Snort berhasil tetapi tidak di-drop karena urutan knock sequence salah, mengindikasikan bahwa walaupun serangan tidak aktif, konfigurasi yang salah dapat menyebabkan kegagalan dalam mengamankan akses. Di skenario ketiga, serangan Probe aktif, Snort aktif, tetapi IP Tables tidak aktif, dan teknik Port Knocking salah, hasilnya menunjukkan bahwa serangan terdeteksi tetapi tidak di-drop, dan client tidak dapat mengakses knock sequence yang benar, menunjukkan bahwa IP Tables berperan penting dalam aksi drop serangan. Dalam skenario keempat, di mana serangan Probe tidak aktif, Snort aktif, tetapi IP Tables tidak aktif, dan teknik Port Knocking benar, hasilnya menunjukkan bahwa Snort berhasil mendeteksi serangan tetapi tidak di-drop, menunjukkan bahwa Snort dapat mendeteksi serangan tetapi tanpa IP Tables, aksi drop tidak dilakukan. Pada skenario kelima, serangan Probe tidak aktif, Snort tidak aktif, dan IP Tables aktif dengan teknik Port Knocking benar, hasilnya menunjukkan bahwa serangan tidak terdeteksi atau di-drop, tetapi knock sequence berhasil, menunjukkan bahwa tanpa Snort, deteksi serangan tidak dilakukan meskipun IP Tables dan Port Knocking aktif. Skenario keenam menunjukkan serangan Probe tidak aktif, Snort tidak aktif, IP Tables aktif, dan Port Knocking salah, hasilnya menunjukkan bahwa tidak ada deteksi atau drop, mengindikasikan bahwa konfigurasi Port Knocking yang salah menghambat mekanisme keamanan. Di skenario ketujuh, serangan Probe aktif, Snort aktif, dan IP Tables aktif dengan teknik Port Knocking benar, hasilnya menunjukkan bahwa serangan berhasil dideteksi dan di-drop, dan knock sequence benar, menunjukkan kombinasi optimal dari Snort dan IP Tables dengan konfigurasi Port Knocking yang tepat. Skenario kedelapan menunjukkan serangan Probe aktif, Snort aktif, IP Tables tidak aktif, dan teknik Port Knocking salah, hasilnya menunjukkan bahwa Snort berhasil mendeteksi tetapi tidak di-drop, dan knock sequence salah, menunjukkan pentingnya IP Tables dalam mekanisme drop.

Kesimpulan

Berdasarkan hasil Pengujian dari Instrusion Detection System terhadap serangan Probe menggunakan metode signature Probe sebagai berikut ini:

1. Hasil pengujian mendeteksi serangan probe dengan menerapkan snort 3 yang mampu mendeteksi ketika terjadi serangan dan untuk membuka atau menutup port dilakukan kombinasi knock sequance yang hanya di berikan akses kepada client legal ketika terjadinya serangan ip tables yang sudah dimasukkan ke dalam snort 3 akan melakukan drop otomatis ketika terjadinya serangan
2. Snort 3 berfungsi dalam mengatasi serangan dengan menerapkap rules untuk mendeteksi icmp, ddos yang terintegrasi dengan database log snort 3 ketika terjadinya serangan dan masuk ke dalam log snort tersebut akan dilakukan drop secara otomatis dengan mengkombinasikan port knocking yang berfungsi untuk menutup atau membuka port pada web server sehingga hanya client legal yang dapat megakses web server tersebut
3. Hasil pengujian snort 3 terbukti mendeteksi serangan dengan mengintegrasikan log serangan snort di dalam pengaturan appid yang tedapat ke dalam pengaturan qoutput intrusion detection system di snort.lua lalu dengan direktori snort.log yang nantinya ketika terjadinya serangan snort dapat meyimpan catatan serangan yang telah terjadi.

Daftar Pustaka

[1] S. Alviana and I. D. Sumitra, "Analisis Pengukuran Penggunaan Sumber Daya Komputer Pada Intrusion Detection System Dalam Meminimalkan Serangan Jaringan," *Komputa J. Ilm. Komput. Dan Inform.*, vol. 7, no. 1, pp. 27–34, Mar. 2018, doi: 10.34010/komputa.v7i1.2533.
[2] M. Sulfarita, "Implementasi Dan Analisis Network Intrusion Detection System (Nids) Untuk Monitoring Jaringan Intranet," 2023.
[3] K. A. Fikri, "Keamanan Jaringan Menggunakan Switch Port Security," vol. 5, 2021.
[4] M. A. M. Herri Setiawan, "Penggunaan Metode Signatured Based Dalam Pengenalan Pola Serangan Di Jaringan Komputer," *08 Juni 2021*, vol. 8, no. 3, pp. 517–524, 2019.
[5] M. A. Gangwar and M. S. Sahu, "A survey on anomaly and signature based intrusion detection system (IDS)," vol. 4, no. 4, 2022.
[6] A. Idrus, L. Sugiyanta, and M. Nugraheni, "Design and Implementation of Sucirata-Based Instrusion Detection System as a Network Security System Cloud Computers," vol. 7, no. 158, 2023.
[7] I. A. Siswanto, M. T. Kurniawan, and A. Widjajarto, "Sistem Keamanan Wireless Sensor Network Menggunakan Signature Based Intrusion Detection System Dan System Shutdown Untuk Memitigasi Serangan Blackhole," 2020.
[8] I. Meiditra, "Analisis Dan Perancangan Private Cloud Storage Menggunakan Metode Pengamanan Ids (Intrusion Detection System) Dan Ips(Intrusion Prevention System) (Studi Kasus: Diskominfo Kota Padang Panjang)," 2023.
[9] M. Simanjuntak, T. Pasaribu, and S. Rahmadilla, "Implementasi Algoritma Merkle Hellman untuk Keamanan Database," *MEANS Media Inf. Anal. Dan Sist.*, pp. 46–50, Jun. 2019, doi: 10.54367/means.v4i1.327.
[10] C. N. A. Prasetyo, I. R. Lie, and M. A. Naufal, "Implementasi Cloud Storage OwnCloud pada Debian VirtualBox," 2022.
[11] T. Purnama, Y. Muhyidin, and D. Singasatia, "Implementasi Intrusion Detection System (Ids) Snort Sebagai Sistem Keamanan Menggunakan Whatsapp Dan Telegram Sebagai Media Notifikasi," *J. Teknol. Inf. DAN Komun.*, vol. 14, no. 2, pp. 358–369, Sep. 2023, doi: 10.51903/jtikp.v14i2.726.
[12] I. G. W. Bangga and S. M. Ladjamuddin, "Simulasi Snort Sebagai Alat Pendeteksi Intrusi Pada Web Damn Vulnerable Web Application," vol. 11, 2022.
[13] M. A. Fauzi, "Sistem Deteksi Intrusi Menggunakan Algoritma Genetik Pada Serangan Dos Di Protokol Tcp Dan Udp".
[14] Adam Dwi Ralianto and S. Cahyono, "Perbandingan Nilai Akurasi Snort dan Suricata dalam Mendeteksi Intrusi Lalu Lintas di Jaringan," *Info Kripto*, vol. 15, no. 2, pp. 69–75, Aug. 2021, doi: 10.56706/ik.v15i2.10.
[15] F. H. Utami, "Aplikasi Pelayanan Antrian Pasien Menggunakan Metode FCFS Menggunakan PHP dan MySQL," 2022.