(2024), 2 (3): 218–238 Scientica 3021-8209
Jurnal Ilmiah Sain dan Teknologi

SISTEM MONITORING SERANGAN JARINGAN MENGGUNAKAN INTRUSION DETECTION SYSTEM (IDS) DENGAN NOTIFIKASI TELEGRAM

Givan Yandiputra Sunardi, Ade Kania Ningsih, Sigit Anggoro
Universitas Jenderal Achmad Yani
givanyandiputras18@if.unjani.ac.id

Abstract (English)
The ease of accessing the internet network has provided convenience in our daily lives, such as accessing information. However, the ease of accessing knowledge on the internet has also made it easier for individuals to engage in criminal activities, supported by the availability of various applications and tools to launch attacks on the internet. Every available internet network requires special monitoring, especially network servers that store important data and handle large data flows. Attacks can come in various forms, including brute force attacks, SQL injection, Denial of Service, and others. These attacks need to be monitored to identify the type of attack, the target, and other relevant information.Network server monitoring can be achieved using an Intrusion Detection System (IDS), which detects and records attacks occurring within the network in log files. These log files are in a text format that is easily understandable by an administrator. The monitoring system is built based on IDS logs, which facilitate the classification of attacks, tracking attack frequencies, identifying the attacker's IP address, and providing other relevant information. However, administrators may not always be present to monitor the network continuously. To address this, an automated alert system is implemented to notify administrators of attacks in real-time, even when they are not physically present. These alert messages are sent using a free messaging application, such as Telegram bot, to provide convenient access to monitoring the warning messages.

Article History
Submitted: 3 February 2024
Accepted: 12 February 2024
Published: 13 February 2024

Key Words
attack monitoring, real-time, intrusion detection system, telegram notification

Abstrak (Indonesia)
Kemudahan untuk mengakses jaringan internet, dapat membuat kemudahan kehidupan sehari hari seperti contohnya mengakses informasi. Namun semakin mudah pula untuk mengakses pengetahuan untuk melakukan kejahatan di internet, didukung dengan banyaknya aplikasi dan alat bantu untuk melancarkan serangan di internet. Setiap jaringan internet yang tersedia diperlukan sebuah monitoring khusus, terutama server jaringan yang memiliki data data penting didalamnya serta banyak aliran data. Serangan yang terjadi bisa berupa bruteforce, SQL Injection, Denial of Service dan serangan lainnya Serangan tersebut harus dimonitoring guna untuk mengetahui serangan, target serangan, jenis serangan dan informasi seputar serangan lainnya. Monitoring server jaringan dapat menggunakan metode IDS atau Intrusion Detection System yang berfungsi untuk medeteksi serangan yang terjadi di jaringan yang dicakup dan direkam menjadi sebuah file log. Berkas log tersebut berbentuk teks yang mudah dipahami oleh seorang administrator. Sistem monitoring tersebut dibuat berdasarkan log IDS, yang fungsinya untuk memudahkan seorang administrator dalam klasifikasi serangan, frekuensi terjadinya serangan, alamat penyerang dan informasi lainnya. Namun seorang administrator terkadang tidak selalu berada ditempat untuk monitoring, maka dari itu dibuatlah sebuah pesan peringatan otomatis jika terjadi serangan selama administrator tidak ditempat. Pesan peringatan dikirim secara real-time menggunakan aplikasi pesan gratis yang

Sejarah Artikel
Submitted: 3 February 2024
Accepted: 12 February 2024
Published: 13 February 2024

Kata Kunci
monitoring serangan, real-time, intrusion detection system, notifikasi telegram

tersedia yaitu telegram bot agar memudahkan akses monitoring pesan peringatan.

Pendahuluan

Berkembangnya teknologi informasi khususnya jaringan komputer beserta layanannya yang memudahkan pekerjaan manusia sehari-hari, seperti otomatisasi, kedokteran, kesehatan hingga bidang keuangan. Kita sudah merasakan kenyamanan yang didapatkan jika memakai teknologi tersebut. Namun disisi lain muncul masalah baru yaitu soal keamanan informasi tersebut. [1] Kejahatan yang dilakukan di internet biasanya seperti pengaksesan file secara ilegal, mengirimkan lalu lintas palsu, percobaan akses secara paksa, dan masih banyak lagi yang disebut sebagai intrusi. Setiap jaringan yang terkoneksi dengan internet perlu dipantau, terutama Server yang didalamnya memuat data data penting. Monitoring Server jaringan dilakukan untuk mempermudah seorang administrator untuk mengamati dan mengontrol system yang dibuat.

Server, sebagai pilar utama dalam infrastruktur teknologi informasi, menjadi sasaran potensial serangan siber yang beragam. Data sensitif, informasi pribadi, dan operasi bisnis krusial seringkali diolah dan disimpan dalam server. Karena itu, perlindungan dan pengawasan server sangat penting untuk menghindari kebocoran data, pengganggguan operasional, dan dampak serius lainnya akibat serangan.

Berdasarkan penelitian sebelumnya sistem pada sebuah Server terutama webserver biasanya diserang dengan beberapa macam metode. Contoh serangannya seperti eksplorasi port yang terbuka menggunakan Port Scanning, lalu ada Bruteforce yang mencoba memaksa masuk dengan segala username dan password yang dimiliki, Denial of Service dimana membanjiri terhadap service yang sedang berjalan dan masih banyak lagi[2]. Dengan mengetahui informasi serangan yang terjadi, selain kita mengamankan apa yang diserang, juga mengetahui darimana serangan berasal. Maka dari itu dibutuhkanlah sebuah monitoring server jaringan untuk mengamati lalu lintas data terhadap server tersebut.

Secara umum untuk memonitoring serangan jaringan, salah satunya yaitu dengan menggunakan cara Intrusion Detection System (IDS) atau sistem pedeteksi serangan. IDS mendeteksi serangan dan merekam data tersebut menjadikannya sebuah data log. Data log tersebut biasanya berbentuk teks yang berisikan alamat penyerang, alamat target, jenis serangan, pesan serangan dan masih banyak lagi. Namun dalam satu file teks log tersebut jika memiliki ribuan rekaman, cukup sulit untuk membaca lognya. Maka dari itu dibuatlah sistem monitoring serangan ini beserta visualisasi berbasis web. Visualisasi rekaman log ini dapat membantu seorang administrator untuk melihat serangan yang sering terjadi di jaringan, atau frekuensi serangan atau target yang sering diserang.[3]

Namun, kemampuan IDS dalam mendeteksi serangan sering kali bergantung pada aturan yang dikonfigurasi di dalamnya. Aturan ini membantu IDS memahami pola perilaku yang mencurigakan dan memberikan indikasi tentang serangan yang sedang terjadi. Meskipun IDS memiliki potensi untuk mengidentifikasi berbagai jenis serangan, terkadang penggunaan aturan yang tidak tepat atau kesalahan penulisan dapat berdampak negatif pada efektivitas deteksi. Kehadiran aturan yang spesifik adalah kunci untuk mengidentifikasi serangan dengan akurasi, tetapi pada saat yang sama, perlu diakui bahwa mengembangkan aturan yang optimal bukanlah tugas yang mudah. Maka dari itu dibutuhkanlah aturan IDS yang cukup spesifik untuk mendeteksi serangan yang terjadi.

Agar IDS dapat dipantau dimana saja, dibutuhkan sebuah pemberi pesan peringatan secara real-time saat serangan terjadi. Aplikasi berbasis pesan obrolan banyak sekali di internet dan memiliki fitur fitur yang berbeda setiap aplikasinya, Telegram merupakan salah satunya. Telegram memiliki fitur bot chat di aplikasinya. Bot chat dipakai tergantung dengan pemakainya, bisa dipakai untuk bermain game, membuat fitur baru dan masih banyak lagi. Telegram bot bisa digunakan untuk mengirimkan pesan peringatan dari IDS jika terjadi serangan. Dengan adanya IDS dapat mendeteksi serangan yang terjadi pada server. Serangan yang terjadi informasinya akan dikirimkan melalui telegram bot yang dibuat untuk memberikan pesan peringatan pada administrator, sehingga administrator dapat mengantisipasi jika terjadinya intrusi lebih dalam.

Metode Penelitian

Analisis dan Perancangan merupakan aktivitas yang dilakukan untuk mendesain dan membangun suatu sistem atau aplikasi berdasarkan pemahaman tentang kebutuhan pengguna dan tujuan yang ingin dicapai. Pada penelitian ini merancanga sebuah SISTEM MONITORING SERANGAN JARINGAN MENGGUNAKAN INTRUSION DETECTION SYSTEM (IDS) DENGAN NOTIFIKASI TELEGRAM untuk menganalisis apakah serangan jaringan dapat terdeteksi atau tidak serta jika serangan terjadi datanya akan diolah sebagai monitoring jaringan admin sekaligus pemberitahuan kepada admin bahwa serangan telah terjadi.

Instalasi & Konfigurasi Snort

Sebelum tahapan pengujian ada beberapa hal yang perlu dilakukan dalam pendeteksian snort apakah berjalan sesuai dengan diharapkan diantara nya dengan melakukan instalasi dan konfigurasi untuk pembuatan rule snort terkait adanya ancaman supaya dapat terdekteksi oleh snort.

Lakukan instalasi paket snort 3 dengan perintah berikut :

```
sudo apt install build-essential libpcap-dev libpcre3-dev libnet1-dev zlib1g-dev luajit hwloc libdnet-dev libdumbnet-dev bison flex liblzma-dev openssl libssl-dev pkg-config libhwloc-dev cmake cpputest libsqlite3-dev uuid-dev libcmocka-dev libnetfilter-queue-dev libmnl-dev autotools-dev libluajit-5.1-dev libunwind-dev
```

Buatlah sebuah folder dan masuk ke folder tersebut

```
mkdir snort_src
cd snort_src
```

Unduh dan install Snort Data Acquisition library (LibDAQ).

```
git clone https://github.com/snort3/libdaq.git
cd libdaq
./bootstrap
./configure
make
make install
```

Unduh dan install gperftools, berfungsi untuk optimalisasi alokasi memori dan memberikan performa untuk memori jauh lebih baik.

```
cd .. (untuk kembali ke folder snort_src)
wget https://github.com/gperftools/gperftools/releases/download/gperftools-2.9/gperftools-2.9.tar.gz
```

```
tar xzf gperftools-2.9.tar.gz
cd gperftools-2.9/
./configure
make
make install
```

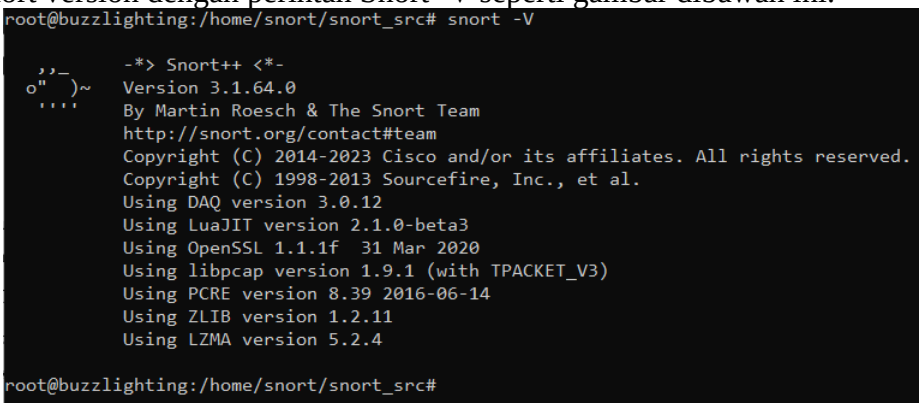
Unduh dan install Snort3. Untuk link unduh Snort 3 yang terbaru bisa cek ke link terbaru di web page snort sendiri.

```
cd .. (untuk kembali ke folder snort_src)
wget https://www.snort.org/downloads/snortplus/snort3-3.1.64.0.tar.gz
tar -xvzf snort3-3.1.64.0.tar.gz
cd snort3-3.1.64.0
./configure_cmake.sh --prefix=/usr/local --enable-tcmalloc
cd build
make
make install
```

Update libraries yang digunakan.

```
sudo ldconfig
```

Verifikasi snort version dengan perintah Snort -V seperti gambar dibawah ini.



Gambar 4 1 Snort Version

Selanjutnya konfigurasi snort supaya snort dapat bekerja sebagai IDS untuk mendeteksi serangan. Dalam konfigurasi snort ada hal yang penting perlu untuk dilaksanakan seperti untuk memasukan ip address client yang ingin dilindungin pada folder snort.lua.

```
nano /usr/local/etc/snort/snort.lua
```

```
-----------------------------------------------------
-- 1. configure defaults
-----------------------------------------------------

-- HOME_NET and EXTERNAL_NET must be set now
-- setup the network addresses you are protecting
HOME_NET = '192.168.100.55'

-- set up the external network addresses.
-- (leave as "any" in most situations)
EXTERNAL_NET = '!$HOME_NET'

include 'snort_defaults.lua'
```

Gambar 4 2 Konfigurasi Snort

Gambar diatas merupakan ip address yang akan dilindungi oleh snort yang HOME_NET dimana ip addressnya yaitu 192.168.100.55 (server). Selanjutnya menentukan dimana file untuk menyimpan aturan aturan tersebut. Masuk ke bagian nomor 5. Configure detection.

```
-----------------------------------------------------------------------
-- 5. configure detection
-----------------------------------------------------------------------

references = default_references
classifications = default_classifications

ips =
{
    -- use this to enable decoder and inspector alerts
    --enable_builtin_rules = true,

    -- use include for rules files; be sure to set your path
    -- note that rules files can include other rules files
    -- (see also related path vars at the top of snort_defaults.lua)

    variables = default_variables,
    rules = [[
    include /usr/local/etc/rules/snort3-community-rules/snort3-community.rules
    include /usr/local/etc/snort/block.rules
    ]]
}
```

Gambar 4 3 Konfigurasi Path Aturan Snort

Pada gambar diatas masukan file dimana aturan snort tersebut dibuat. Disini penliti menggunakan rules yang sudah tersedia yaitu snort3-community.rules dan block.rules untuk aturan baru untuk Denial of Service dipengujian nanti.
Agar rekam data snort dicetak dan dijadikan sebuah log yang berbentuk csv, ke bagian bawah lagi nomor 7 yaitu configure outputs.

```
-----------------------------------------------
-- 7. configure outputs
-----------------------------------------------

-- event logging
-- you can enable with defaults from the comma
-- uncomment below to set non-default configs
alert_csv = { file = true,
-- fields = 'timestamp rule pkt_num priority p
fields = 'timestamp rule pkt_num priority prot
}
```

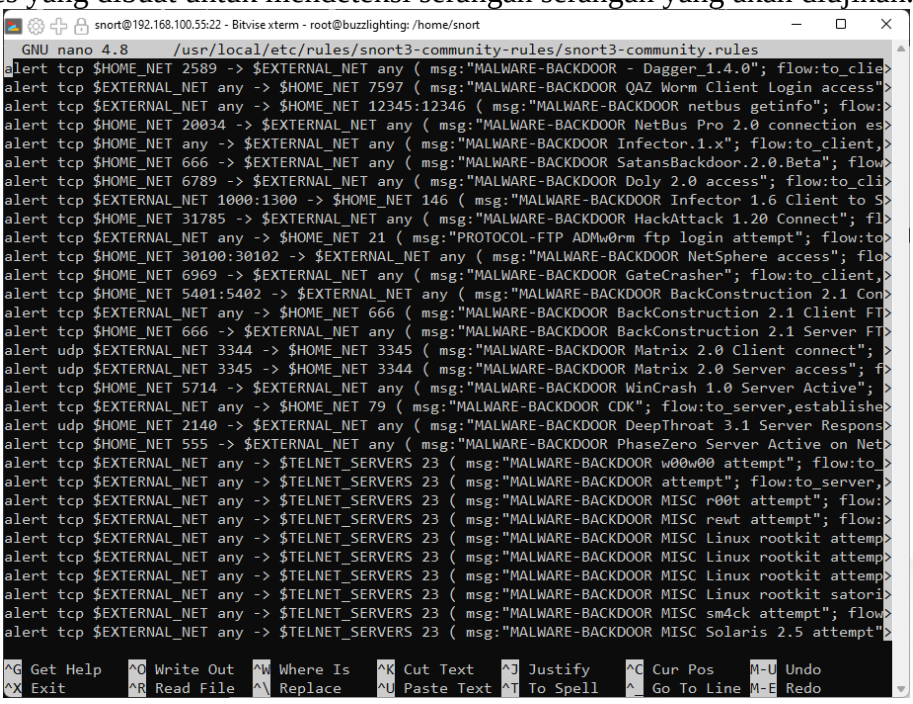
Gambar 4 4 Konfigurasi Path Output Snort

Untuk lengkapnya bagian fields yaitu :

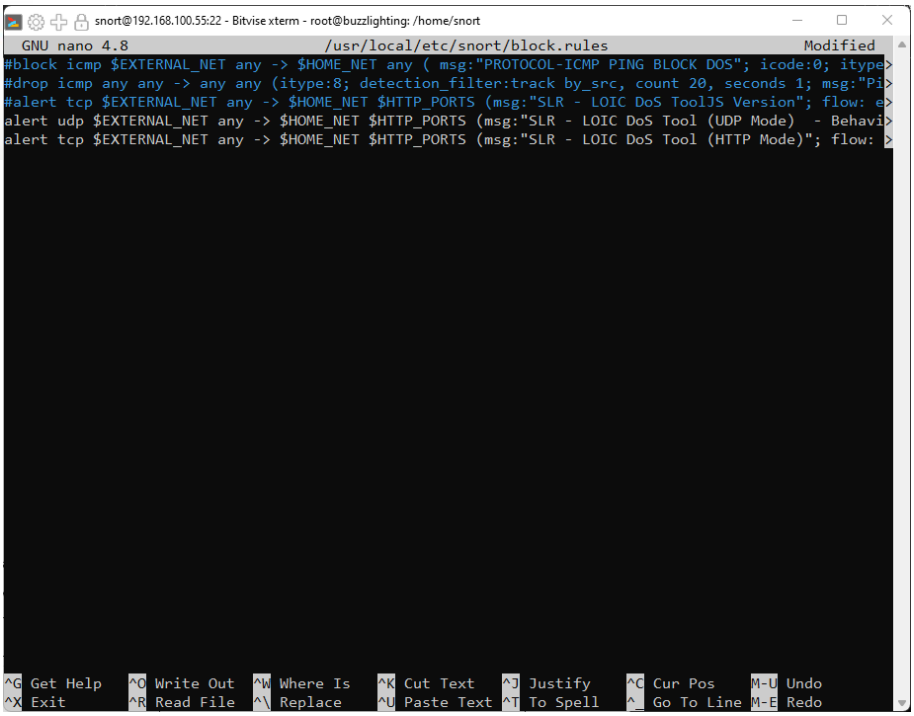
fields = 'timestamp rule pkt_num priority proto src_addr src_port dst_addr dst_port pkt_len class msg action'

Pada gambar diatas dijelaskan bagaimana konfigurasi Output Snortnya. File = true menyatakan bahwa alert_csv akan dicetak dan dijadikan sebuah file, namun tidak hanya itu saja fieldsnya ditentukan berdasarkan output apa saja yang ingin dibuat. Untuk fields apa saja yang dapat dimasukan bisa mengunduh snort reference dari website resminya. Lalu simpan konfigurasi tersebut.

Lalu buat aturan sesuai dengan Tabel 3-10 Perancangan Aturan Snort. Berikut dibawah ini beberapa rules yang dibuat untuk mendeteksi serangan serangan yang akan diujikan.



Gambar 4 5 Konfigurasi Rules Snort snort3-community.rules



Gambar 4 6 Konfigurasi Rules Snort block.rules

Aturan diatas merupakan aturan yang masuk ke snort yaitu snort3-community.rules.

Tabel 4 1 Implementasi Rules Snort

Nama	**Rules**
Port Scanning (NMAP)	alert tcp $EXTERNAL_NET any -> $HOME_NET 161 (msg:"PROTOCOL-SNMP request tcp"; flow:stateless; metadata:policy max-detect-ips drop,ruleset community; service:snmp; reference:bugtraq,4088; reference:bugtraq,4089; reference:bugtraq,4132; reference:cve,2002-0012; reference:cve,2002-0013; classtype:attempted-recon; sid:1418; rev:19;)
Brute Force SSH (Hydra)	alert tcp $EXTERNAL_NET any -> $HOME_NET $SSH_PORTS (msg:"INDICATOR-SHELLCODE ssh CRC32 overflow /bin/sh"; flow:to_server,established; content:"/bin/sh"; metadata:ruleset community; reference:bugtraq,2347; reference:cve,2001-0144; reference:cve,2001-0572; classtype:shellcode-detect; sid:1324; rev:12;)
Brute Force FTP (Hydra)	alert tcp $HOME_NET 21 -> $EXTERNAL_NET any (msg:"PROTOCOL-FTP Bad login"; flow:to_client,established; content:"530 ",fast_pattern,nocase; pcre:"/^530\s+(Login|User)/ims"; metadata:ruleset

	community; service:ftp; classtype:bad-unknown; sid:491; rev:15;)
Denial of Service (LOIC)	alert udp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS (msg:"SLR - LOIC DoS Tool (UDP Mode) - Behavior Rule "; detection_filter:track by_src, count 100 , seconds 5; reference: url, www.simpleweb.org/reports/loic-report.pdf ; classtype:misc-activity; sid:1234590; rev:1;)

Untuk menjalankan snort dapat menuliskan perintah sebagai berikut. Dimana perintahnya sebagai berikut :

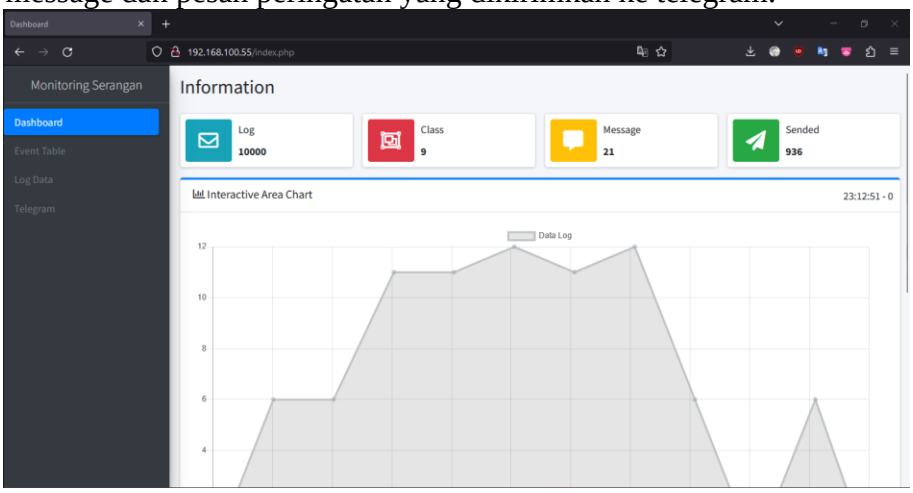
Tabel 4 2 Perintah Snort Run

snort -c /usr/local/etc/snort/snort.lua -i enp0s3 -s 65535 -k none -l /var/log/snort/

Tampilan Aplikasi

Dashboard

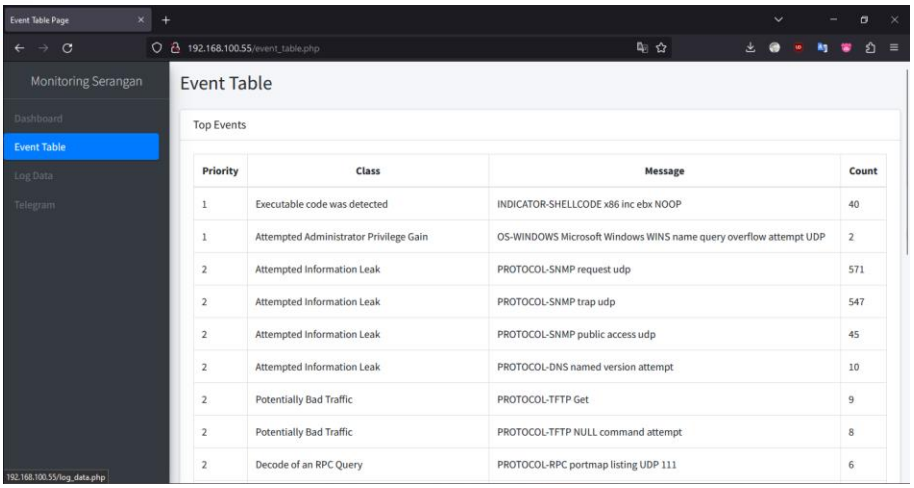
Pada tampilan halaman ini digunakan untuk menampilkan berapa banyak log yang tersimpan pada mysql, class, message dan pesan peringatan yang dikirimkan ke telegram.



Gambar 4 7 Implementasi Dashboard

Event Table

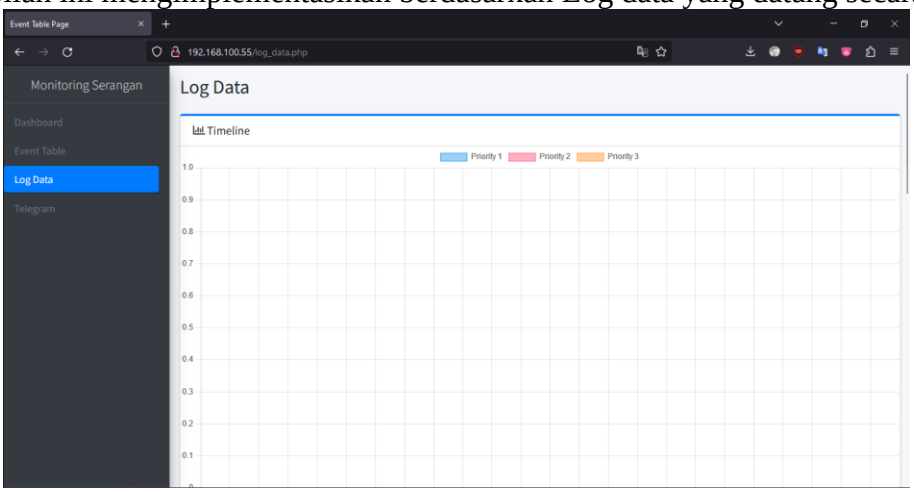
Pada tampilan ini adalah implementasi berdasarkan resikko, kelas serangan, pesan dan banyaknya serangan pada total keseluruhan data yanga ada.



Gambar 4 8 Implementasi Event Table

Log Data

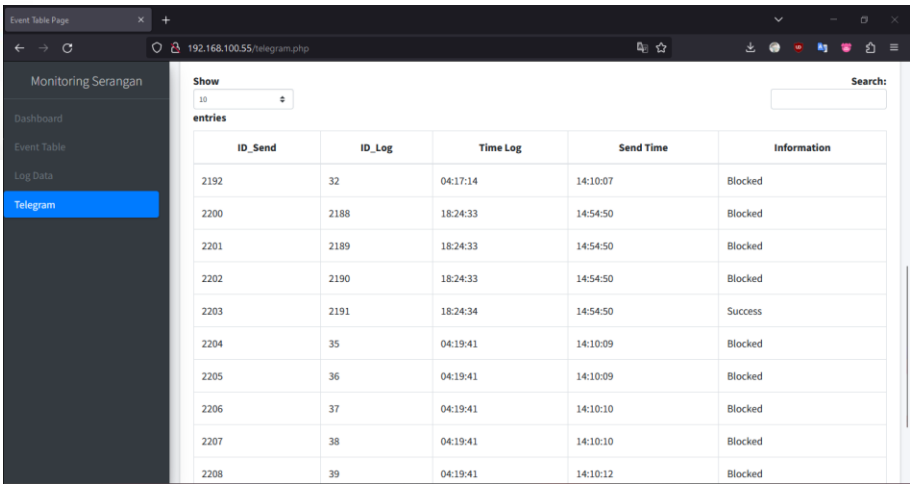
Tampilan ini mengimplementasikan berdasarkan Log data yang datang secara realtime.



Gambar 4 9 Implementasi Log Data

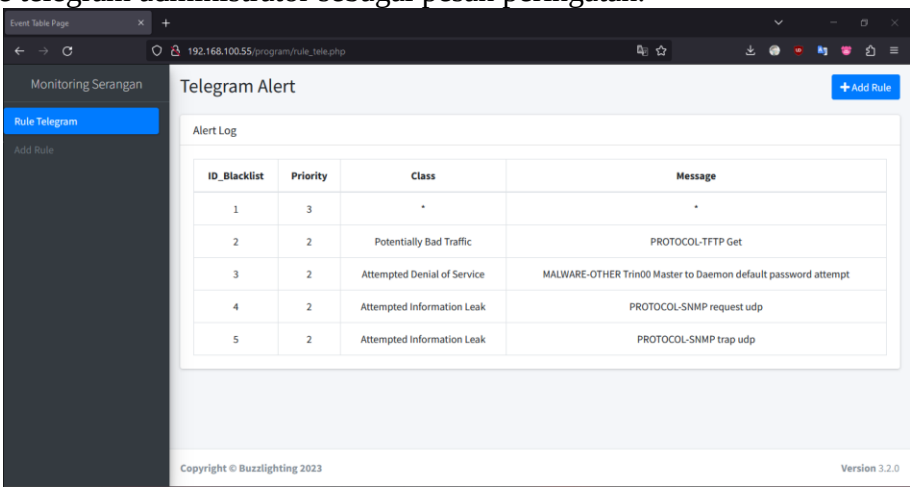
Telegram

Tampilan ini menampilkan data dari pesan pesan yang dikirim atau tidaknya ke telegram. Semuanya tercatat dalam halaman ini.



Gambar 4 10 Implementasi Telegram

Tampilan ini menampilkan data data class dan pesan yang di blacklist atau tidak usah dikirimkan ke telegram administrator sebagai pesan peringatan.



Gambar 4 11 Implementasi Rule List Telegram

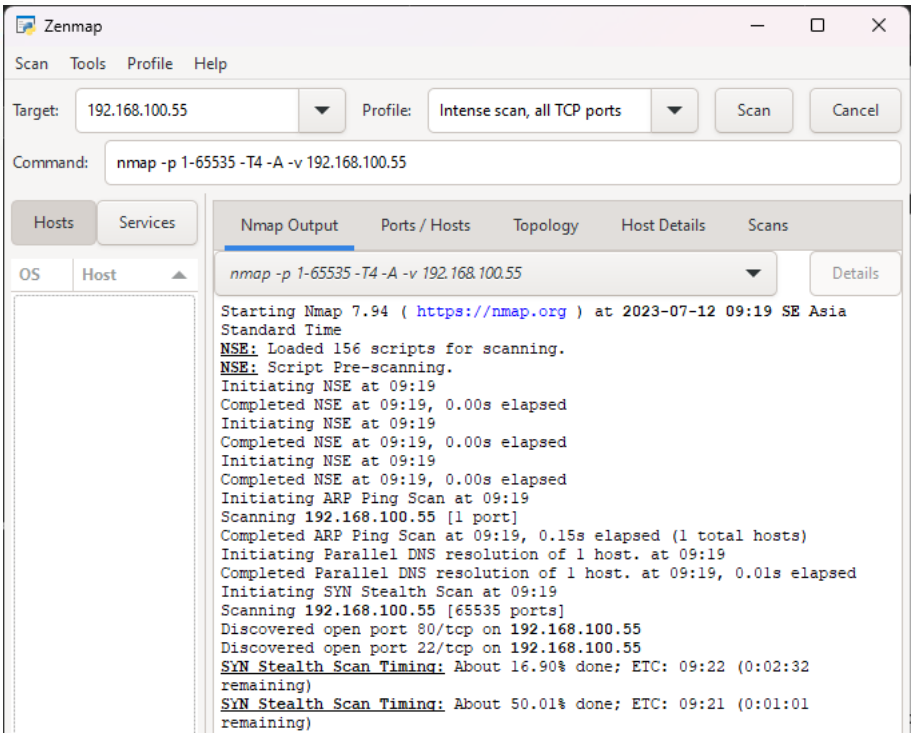
Tampilan ini form untuk menambahkan aturan yang akan di blacklist berdasarkan alert yang sudah disimpan dalam mysql.

Tahapan Pengujian

Pengujian NMAP

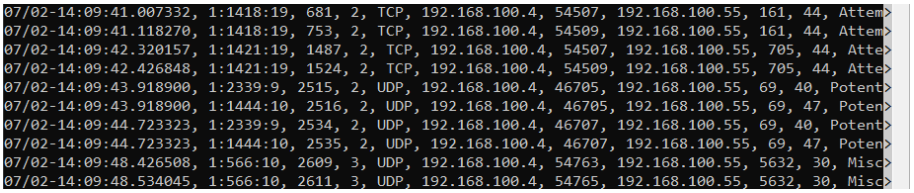
Komputer yang sudah terpasang snort akan dicoba untuk melakukan Metode penyerangan nmap yaitu serangan dengan port scanning dari komputer penyerang. NMAP merupakan sebuah aplikasi atau tools yang berguna untuk audit dan eksplorasi suatu keamanan jaringan. Berikut gambar dibawah ini aplikasi nmap pada NMAP untuk melakukan pengujian ke snort apakah snort dapat membaca serangan nmap.

(2024), 2 (3): 218–238 Scientica 3021-8209
Jurnal Ilmiah Sain dan Teknologi



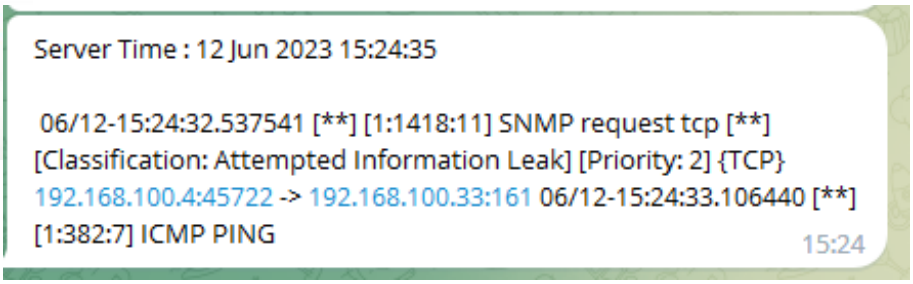
Gambar 4 12 Port Scanning ke Server menggunakan NMAP

Pada gambar diatas dapat dilihat setelah melakukan port scanning ke jaringan client maka jaringan client memiliki 2 port yang terbuka yang dimana 22 ssh dan port 80 http. Sehingga ini merupakan sebuah celah dimana penyerang dapat memasuki atau pun lebih mendalami untuk melakukan penyerangan kedalam jaringan. Pada gambar dibawah ini merupakan tampilan linux ubuntu yang telah dipasang snort maka snort akan mendeteksi adanya serangan atau aktivitas yang tidak wajar didalam jaringan. sehingga snort akan menampilkan peringatan pendeteksi snort seperti berikut ini.



Gambar 4 13 Terdapat Percobaan Scanning Network ke Server

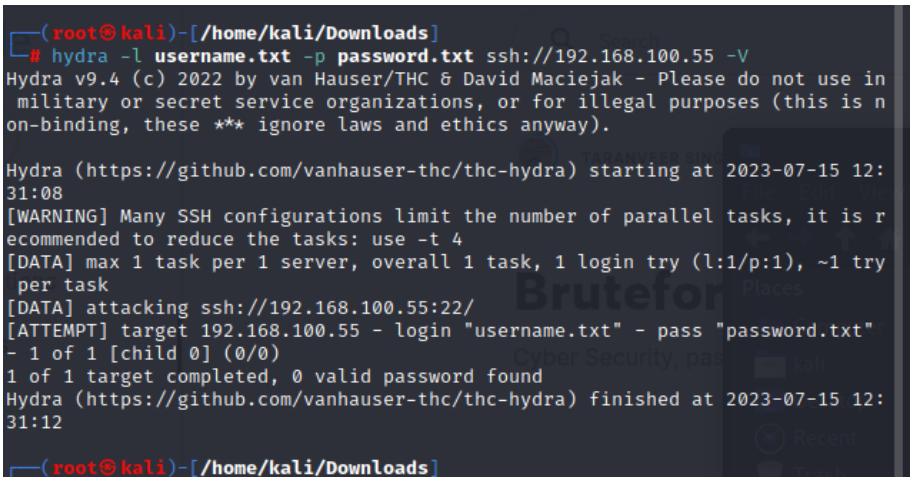
Lalu pada snort juga mendeteksi adanya serangan atau ancaman snort akan memberikan peringatannya adanya port scanning ke telegram administrator. Seperti pada gambar dibawah ini.



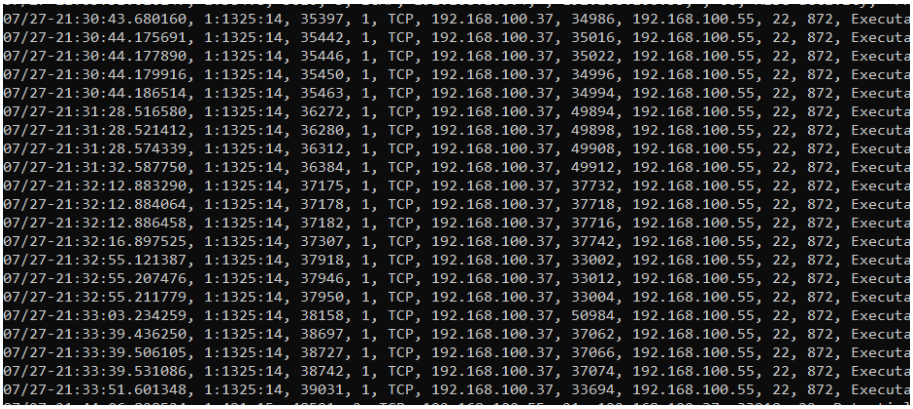
Gambar 4 14 Telegram Alert - Port Scanning ke Server

Pengujian Bruteforce SSH

Pada gambar di bawah merupakan kondisi dimana penyerang (kali linux) menggunakan aplikasi hydra untuk melakukan penetration testing terhadap korban yang terlihat pada gambar dibawah ini.



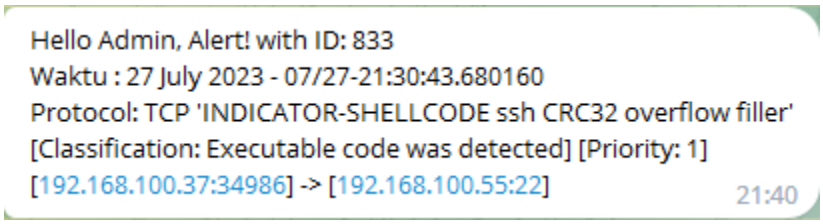
Gambar 4 15 Pengujian Bruteforce SSH



Gambar 4 16 Hasil Deteksi Bruteforce SSH

07/27-21:30:43.680160, 1:1325:14, 35397, 1, TCP, 192.168.100.37, 34986, 192.168.100.55, 22, 872, Executable code was detected, "INDICATOR-SHELLCODE ssh CRC32 overflow filler", allow

Gambar 4 17 Detail Hasil Deteksi Bruteforce SSH

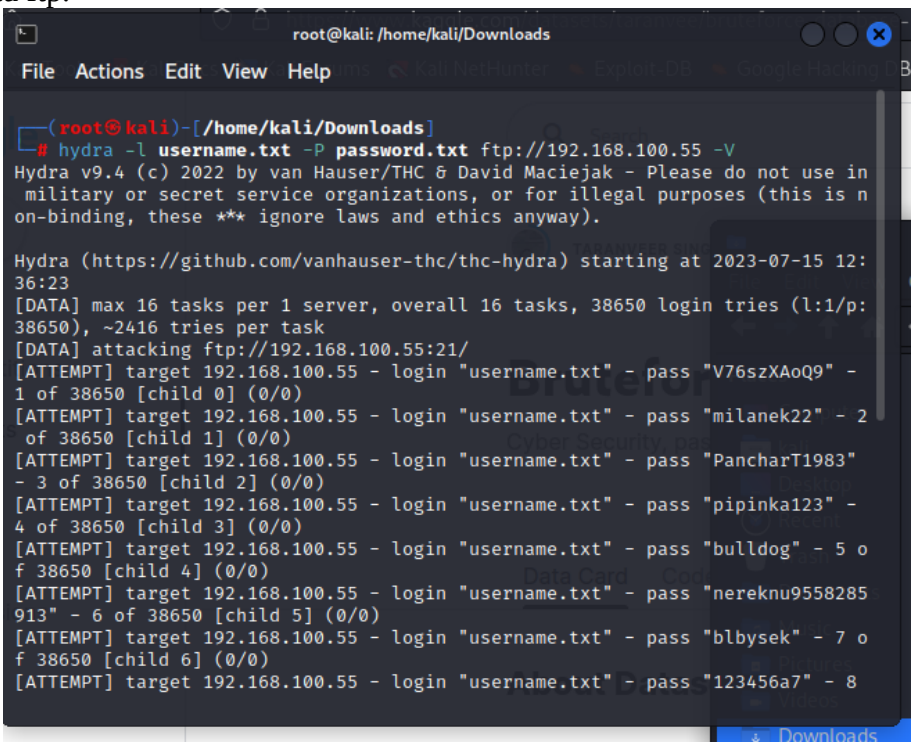


Gambar 4 18 Telegram Alert - Bruteforce SSH ke Server

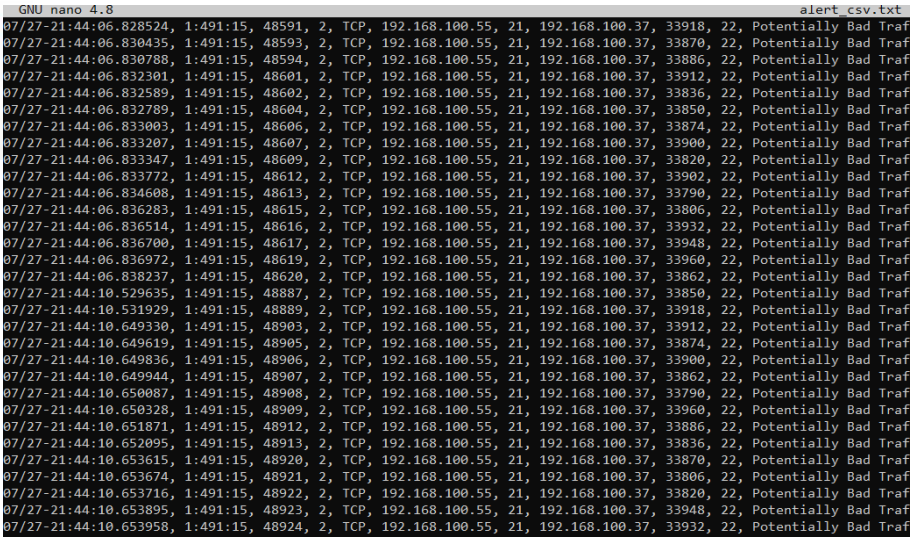
Pada gambar diatas merupakan gambar hasil deteksi dari bruteforce yang terjadi dengan port tujuan 22 (default port ssh). Ada yang mencoba akses ke dalam aplikasi ssh yang diinstal oleh server.

Pengujian Bruteforce FTP

Pada gambar dibawah merupakan kondisi dimana penyerang (kali linux) menggunakan aplikasi hydra juga untuk melakukan penetration testing terhdap korban dengan perubahan serangan yaitu ftp.



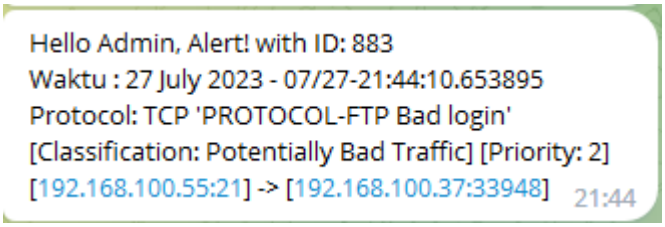
Gambar 4 19 Pengujian Bruteforce FTP



Gambar 4 20 Pengujian Bruteforce FTP

07/27-21:44:10.653895, 1:491:15, 48923, 2, TCP, 192.168.100.55, 21, 192.168.100.37, 33948, 22, Potentially Bad Traffic, "PROTOCOL-FTP Bad login", allow

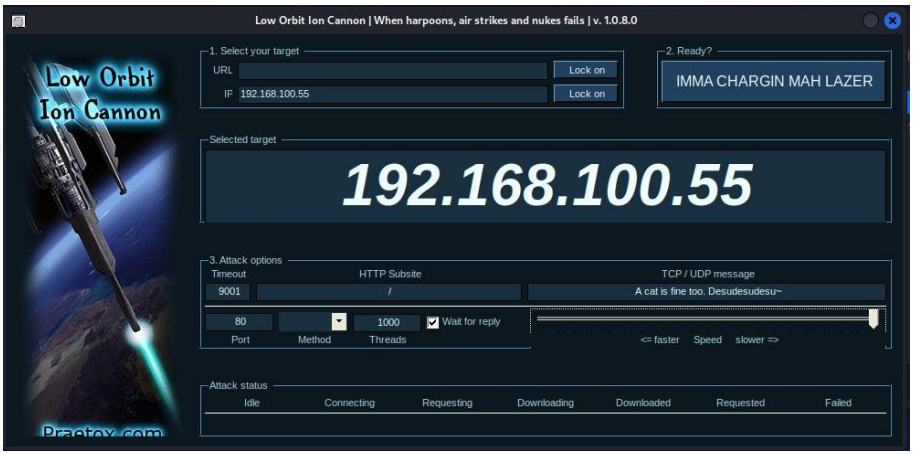
Gambar 4 21 Detail Pengujian Bruteforce FTP



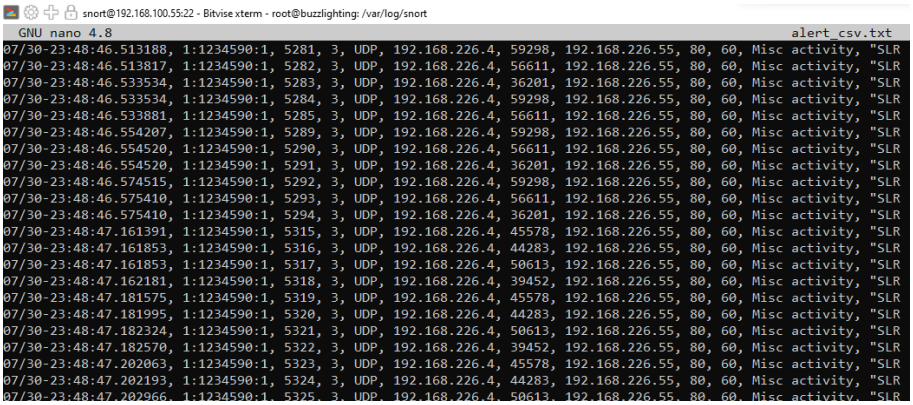
Gambar 4 22 Telegram Alert - Bruteforce FTP ke Server

Pengujian Denial of Service

Pada gambar dibawah merupakan kondisi dimana penyerang (kali linux) menggunakan aplikasi LOIC juga untuk melakukan penetration testing terhdap korban dengan perubahan serangan yaitu DoS dengan service protocol UDP.



Gambar 4 23 Pengujian LOIC dengan Protocol UDP



Gambar 4 24 Hasil Pengujian LOIC dengan Protocol UDP

07/30-23:48:47.223617, 1:1234590:1, 5327, 3, UDP, 192.168.226.4, 45578, 192.168.226.55, 80, 60, Misc activity, "SLR - LOIC DoS Tool (UDP Mode) - Behavior Rule ", allow

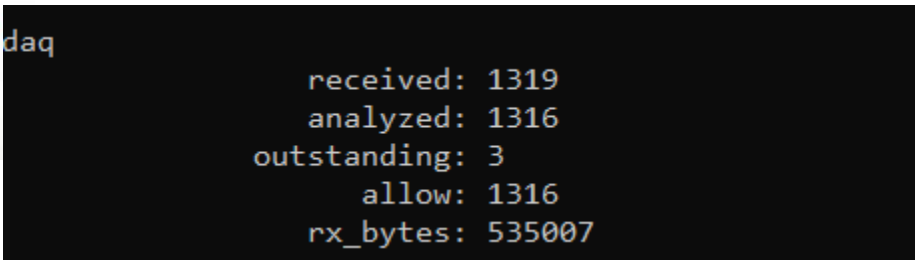
Gambar 4 25 Detail Hasil Pengujian LOIC dengan Protocol UDP

Hasil Pengujian Denial of Service yang dilakukan LOIC terhadap server dengan protocol UDP tidak akan memberikan notifikasi ke telegram, dikarenakan risk value atau prioritas yang diberikan berdasarkan yang dibuat yaitu bernilai 3, yang dimana tidak begitu beresiko.

Hasil Pengujian

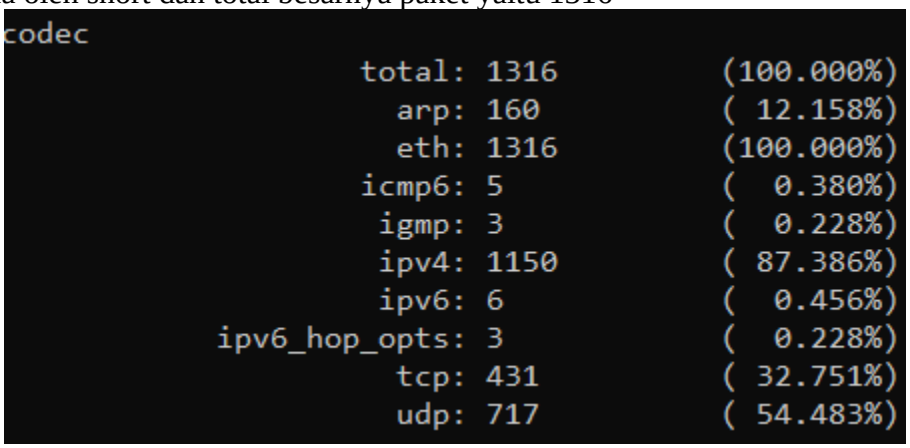
Analisis Snort

Pada hasil pengujian menggunakan snort terbukti bahwa snort dapat melakukan pendeteksian adanya aktivitas jaringan yang tidak wajar. Dimana pada dibawah ini merupakan analisis dari snort itu sendiri terhadap adanya serangan-serangan.



Gambar 4 26 Analisis Snort

Dapat dilihat pada gambar diatas ini snort yang menangkap sejumlah serangan dari peyerang ke client. Dimana snort menerima paket 1319 dan snort menganalisis paket serangan sehingga menghasilkan 1316 dapat dianalisis dan 3 tidak dapat dianalisis, jadi hanya 1316 yang dapat diterima oleh snort dan total besarnya paket yaitu 1316

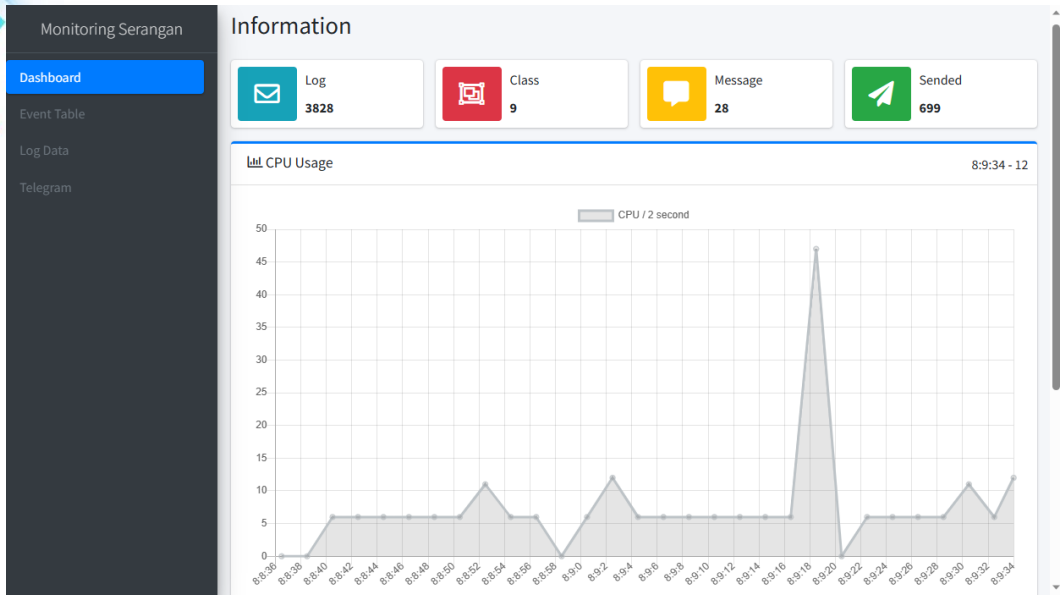


Gambar 4 27 Analisis detail packet Snort

Total detail packet yang snort analisis yaitu 1316, dengan ARP sebanyak 160, eth sebanyak 1316, ICMP6 sebanyak 5, IGMP sebanyak 3, IPv4 sebanyak 1150, IPv6 sebanyak 6, IPv6_hop_opts sebanyak 3, TCP sebanyak 431 dan UDP sebanyak 717. Dengan persentase yang terdapat pada di gambar.

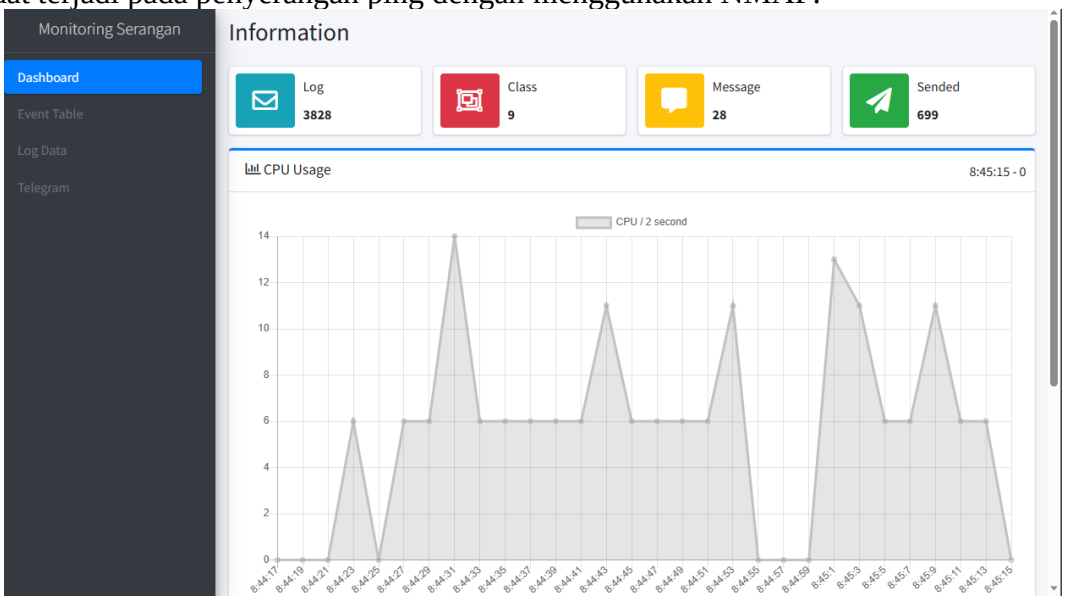
Hasil Monitoring Perangkat Server

Sebelum terjadinya penyerangan pada gambar dibawah merupakan tampilan CPU server saat sebelum terjadinya penyerangan. Dapat dilihat pada CPU history dan network history CPU masih berjalan seperti biasa.



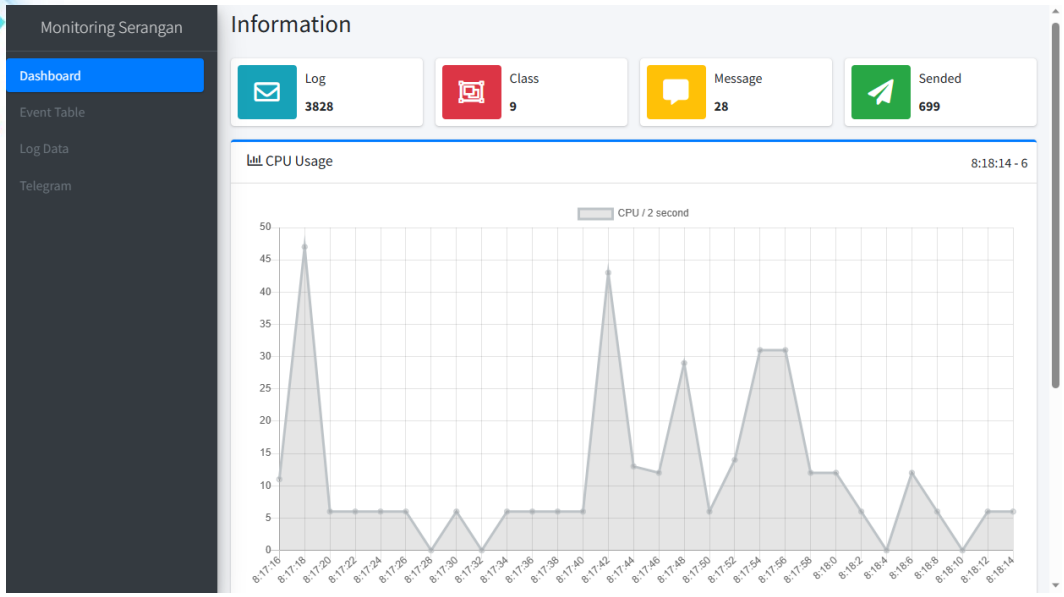
Gambar 4 28 CPU Server saat tidak terjadi penyerangan

Sesaat terjadinya penyerangan dengan Port Scanning dibawah ini merupakan tampilan CPU saat terjadi pada penyerangan ping dengan menggunakan NMAP.



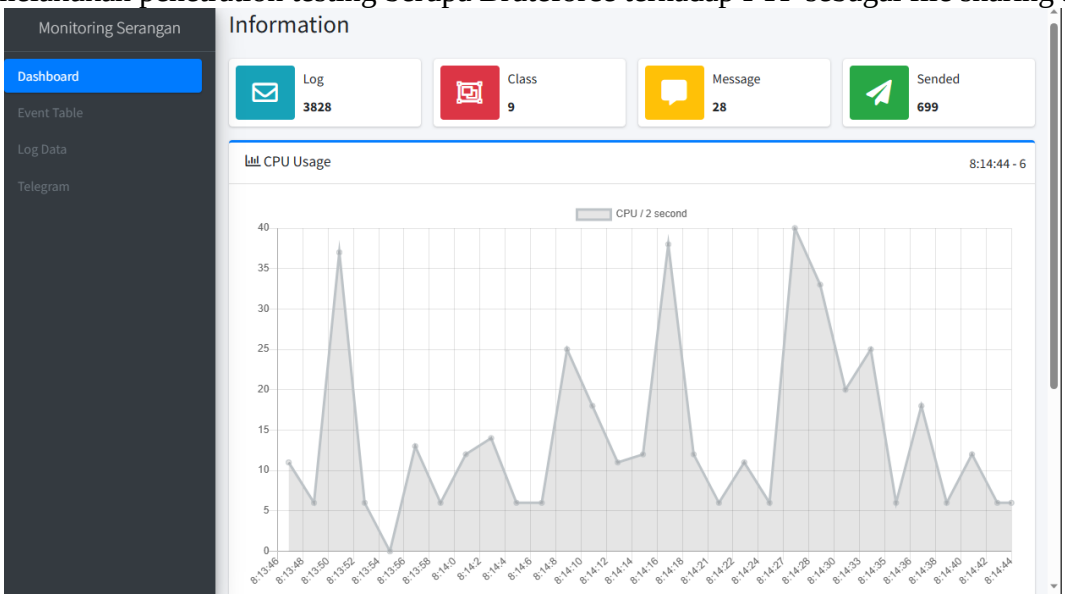
Gambar 4 29 CPU Server saat terjadinya NMAP

Berikut ini sesaat terjadinya penyerangan dengan menggunakan HYDRA melakukan penetration testing yaitu Bruteforce terhadap SSH sebagai remote server.



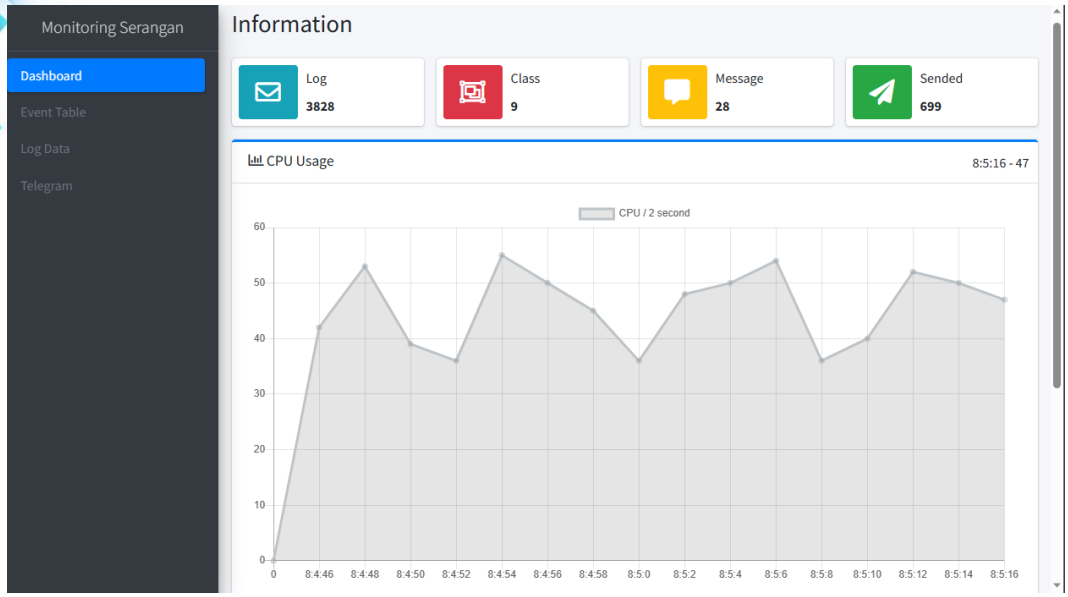
Gambar 4 30 CPU Server saat terjadinya Bruteforce SSH

Lalu ini juga hasil penyerangan dengan menggunakan aplikasi yang sama yaitu HYDRA untuk melakukan penetration testing berupa Bruteforce terhadap FTP sebagai file sharing server.



Gambar 4 31 CPU Server saat terjadinya Bruteforce FTP

Pengujian terakhir melakukan serangan Denial of Service dengan menggunakan aplikasi LOIC (Low Orbit Ion Cannon) untuk membebani server pada protocol yang diserang.



Gambar 4 32 CPU Server saat terjadinya Denial of Service

Hasil Penetration Test

Hasil dari metode penetration test secara keseluruhan dari waktu pengujian keamanan Server. Dapat dilihat dibawah ini.

Tabel 4 3 Waktu Pengujian

No	Jenis Serangan	Waktu		
		Awal Serangan	Waktu deteksi	Terkirim
1	DoS (LOIC)	23:48:47	23:48:50	23:48:52
2	Port Scanning (NMAP)	15:24:00	15:24:02	15:24:04
3	Bruteforce SSH (Hydra)	20.30.20	20.30.21	20.30.30
4	Bruteforce FTP (Hydra)	20.37.31	20.37.40	20.37.58

Sehingga hasil penetration test dalam pengujian dapat dilihat pada tabel dibawha ini.

Tabel 4 4 Hasil Pengujian

No	Jenis Serangan	Hasil Pengujian Sistem	Kesimpulan
1	DoS (LOIC)	Terdeteksi	Berhasil
2	Port Scanning (NMAP)	Terdeteksi	Berhasil
3	Bruteforce SSH (Hydra)	Terdeteksi	Berhasil
4	Bruteforce FTP (Hydra)	Terdeteksi	Berhasil

Hasil Analisis

Pada hasil Monitoring Perangkat Server yaitu CPU memliki hasil yang berupa dari mulai kondisi normal hingga mengganggu CPU Usage.

Tabel 4 5 CPU Usage Serangan

No	Jenis Serangan	CPU	
		CPU Usage/Menit	Persentase Naik dari Kondisi Normal
1	Normal	6,9 %	0 %
2	DoS (LOIC)	45,7%	38,8%
3	Port Scanning (NMAP)	8,26 %	1,36 %
4	Bruteforce SSH (Hydra)	12,1 %	5,2 %
5	Bruteforce FTP (Hydra)	11,90%	4,93 %

Dapat dilihat pada tabel 4-6, bahwa serangan yang terjadi pada saat menyerang menggunakan Denial of Service, persentase penggunaan CPU yang signifikan sebesar 45,7%, yang merupakan peningkatan sebesar 38,8% dari kondisi normal. Ini menunjukkan serangan DoS yang dapat mengganggu kinerja jaringan, walaupun jika dalam aturan IDSnya resiko pengancamannya yaitu dengan kondisi 3 (rendah) setiap terdeteksi 100 kali dalam 5 detik. Serangan Port Scanning (NMAP) memiliki penggunaan CPU sebesar 8,26%, yang merupakan peningkatan sebesar 1,36% dari kondisi normal. Meskipun tidak seintensif serangan DoS, ini tetap merupakan aktivitas mencurigakan yang perlu diawasi. Serangan Bruteforce SSH memiliki penggunaan CPU sebesar 12,1%, yang merupakan peningkatan sebesar 5,2% dari kondisi normal. Ini adalah tanda serangan bruteforce terhadap protokol SSH yang perlu segera diidentifikasi dan dicegah. Terakhir serangan Bruteforce FTP memiliki penggunaan CPU sebesar 11,90%, yang merupakan peningkatan sebesar 4,93% dari kondisi normal. Mirip dengan serangan SSH, serangan bruteforce terhadap protokol FTP juga memerlukan tindakan pengamanan yang cepat dikarenakan jika penyerang dapat memasuki FTP, maka terdapat ancaman pada data penyimpanannya.

Dapat dilihat pada tabel 4-4 dan 4-5 dimana seluruh pengujian mendapatkan hasil yang sesuai dengan yang diharapkan atau dapat terdeteksi dengan baik, namun untuk pengiriman pesan melewati telegram terdapat delay.

Tabel 4 6 Jeda dan Rata rata Log

No	Jenis Serangan	Waktu		
		Awal Serangan	Waktu Terkirim	Jeda Terkirim dari dicatat log hingga ke telegram
1	DoS (LOIC)	23:48:47	23:48:52	5 Detik
2	Port Scanning (NMAP)	15:24:00	15:24:04	4 Detik
3	Bruteforce SSH (Hydra)	20.30.20	20.30.30	10 Detik
4	Bruteforce FTP (Hydra)	20.37.31	20.37.58	27 Detik

Waktu yang didapatkan saat terjadinya serangan Denial of Service, jeda tekirimnya tercatat 5 detik, sistem cukup responsif dalam mengirimkan pesannya. Port Scanning memiliki jeda 4 detik saja, namun Bruteforce SSH memiliki jeda 10 detik. Dan terakhir Bruteforce FTP memiliki jeda 27 detik, waktu yang cukup lambat dari pertimbangan serangan yang diujikan yang lain, seperti DoS, Bruteforce SSH ataupun Port Scanning.

Kesimpulan

Berdasarkan hasil pengujian dari bab IV penulis menarik simpulan sebagai berikut ini.

1. IDS dapat memonitoring serangan jaringan yang terjadi pada server secara realtime walaupun terjadi sedikit delay yang berbeda beda antara serangan terjadi, tercatat oleh IDS hingga terkirim ke telegram
2. IDS dapat mendeteksi serangan yang diujikan seperti Port Scanning, Denial of Service, Bruteforce SSH dan Bruteforce FTP berdasarkan aturan yang dibuat.
3. Intrusion Detection System dapat mengoptimalkan tingkat keamanan jaringan komputer melalui pendeteksian serangan sehingga administrator dapat melakukan tindakan pencegahan.

Referensi

[1] Z. Munawar, M. Kom, and N. I. Putri, "KEAMANAN JARINGAN KOMPUTER PADA ERA BIG DATA."

[2] H. X. Dau, N. T. T. Trang, and N. T. Hung, "A Survey of Tools and Techniques for Web Attack Detection," *J. Sci. Technol. Inf. Secur.*, vol. 1, no. 15, pp. 109–118, Jun. 2022, doi: 10.54654/isj.v1i15.852.

[3] A. L. Ginting, J. Napitupulu, and J. Jamaluddin, *Sistem Monitoring Pendeteksian Penyusup Menggunakan Snort pada Jaringan Komputer Fakultas Ekonomi Universitas Methodist Indonesia*. SNASTIKOM, 2015.

[4] F. Panjaitan and R. Syafari, "PEMANFAATAN NOTIFIKASI TELEGRAM UNTUK MONITORING JARINGAN," *J. SIMETRIS*, vol. 10, no. 2, 2019.

[5] B. Rahardjo, "Keamanan Sistem Informasi Berbasis Internet," 1998.

[6] C. Chazar, "STANDAR MANAJEMEN KEAMANAN SISTEM INFORMASIBERBASIS ISO/IEC 27001:2005," 2015.

[7] J. Gondohanindijo, "Sistem Untuk Mendeteksi Adanya Penyusup (IDS : Intrusion Detection System)," 2011.

[8] The Graylog Team, "Visualize and Correlate IDS Alerts with Open Source Tools," Jul. 13, 2020. https://www.graylog.org/post/visualize-and-correlate-ids-alerts-with-open-source-tools/ (accessed Jan. 19, 2023).

[9] U. Asghar *et al.*, "A Survey of Intrusion Detection & Prevention Techniques," 2011.

[10] D. Utomo *et al.*, "Membangun Sistem Mobile Monitoring Keamanan Web Aplikasi Menggunakan Suricata dan Bot Telegram Channel," vol. 2, 2017.

[11] D. T. Atmaja, E. Budhy Prasetya, and P. E. Kresnha, "NOTIFIKASI ADANYA SERANGAN PADA JARINGAN KOMPUTER DENGAN MENGIRIM PESAN MELALUI APLIKASI TELEGRAM DAN KONTROL SERVER," 2018.

[12] J. J. M. Molina, M. A. H. Ruíz, M. G. Pérez, G. M. Pérez, and A. F. G. Skarmeta, "Event-driven architecture based on patterns for detecting complex attacks," *Int. J. Crit. Comput. Syst.*, vol. 1, no. 4, pp. 283–309, 2010, doi: 10.1504/IJCCBS.2010.036602.

[13] Asep Fauzi Mutaqin, "Rancang Bangun Sistem Monitoring Keamanan Jaringan Prodi Teknik Informatika Melalui SMS Alert dengan Snort," 2016.