Jurnal Quancom
Vol.3, No.2, Desember 2025, pp. 9~18
e-ISSN: 2988-0998
P-ISSN: 2988-1005
DOI : https://doi.org/10.62375/jqc.v3i2.428

Penerapan Intrusion Detection Dan Prevention System Untuk Mendeteksi Serangan Metasploit Menggunakan Snort Dan Wireshark

Asri Yohana Sirait [1]*, Rivka Salvita Silalahi[2], Samuel Volder Naibaho[3]
[1*,2,3] Teknologi Rekayasa Perangkat Lunak, Institut Teknologi Del
asrisirait2004@gmail.com

Article Info

Article history:

Received 04 Desember 2025
Revised 17 Desember 2025
Accepted 31 Desember 2025

***Keyword*:** Intrusion Detection System, Snort, Wireshark, Metasploit Exploit

ABSTRACT

Network security is crucial as cyberattacks continue to rise. One common type of attack is a Remote Exploit, which exploits vulnerabilities in insecure protocols or open ports to remotely access a computer's operating system and steal data. A popular tool used for such attacks is the Metasploit Framework, which is widely used for network security testing. Metasploit includes a module called exploit/multi/handler, which facilitates receiving connections from a target after a security vulnerability is successfully exploited. Additionally, payloads such as linux/x64/meterpreter/reverse_tcp are often utilized to gain control of the target device. Meterpreter is a sophisticated payload that enables attackers to execute commands, extract data, or install malicious programs on the target computer. The reverse_tcp feature ensures the victim's computer establishes a connection back to the attacker, making access easier and less detectable. To protect networks against such threats, tools like Security Information and Event Management (SIEM) systems can be combined with Intrusion Detection Systems (IDS) such as Snort. IDS tools monitor network traffic, detect suspicious activities, and issue alerts when threats are identified. Additionally, tools like Wireshark can be used for analyzing network data, while VirusTotal helps check whether detected files or data contain viruses. By combining these tools, network security can be enhanced to better combat increasingly sophisticated threats.

PENDAHULUAN

Dalam era perkembangan teknologi yang semakin pesat, kebutuhan manusia untuk mengakses informasi terus meningkat. Teknologi tidak hanya mempermudah berbagai aspek kehidupan sehari-hari, tetapi juga meningkatkan efisiensi dalam berbagai pekerjaan. Namun, seiring dengan manfaat tersebut, muncul tantangan signifikan berupa ancaman serangan siber. Salah satu serangan yang cukup berbahaya adalah serangan Remote Exploit, yaitu serangan yang memanfaatkan celah pada port dan protokol yang terbuka untuk mengeksploitasi sistem operasi komputer target dari jarak jauh. Serangan ini dapat menyebabkan pencurian atau perusakan data pada sistem target.

Metasploit Framework sering digunakan dalam serangan ini untuk menciptakan backdoor dengan memanfaatkan modul seperti exploit/multi/handler dan payload seperti linux/x64/meterpreter/reverse_tcp guna mengendalikan sistem operasi target secara jarak jauh. Ancaman ini menuntut pengembangan sistem yang lebih andal untuk mendeteksi dan mencegah serangan siber, terutama melalui peningkatan keamanan jaringan.

Salah satu solusi yang dapat diterapkan adalah penggunaan Intrusion Detection System (IDS), sebuah sistem yang dirancang untuk memantau lalu lintas jaringan, mendeteksi aktivitas mencurigakan, dan memberikan peringatan berupa alert terhadap potensi gangguan atau

serangan pada jaringan. IDS mampu memonitor seluruh aktivitas jaringan dan memberikan respons secara cepat terhadap ancaman intrusi. Salah satu aplikasi IDS yang banyak digunakan adalah Snort. Snort memungkinkan administrator jaringan untuk membuat aturan sendiri guna mendeteksi ancaman berdasarkan tanda tangan (signature-based detection), sehingga dapat menyesuaikan dengan kebutuhan spesifik jaringan yang diawasi.

Snort dapat bekerja sama dengan Wireshark untuk menganalisa dan memonitor lalu lintas jaringan, sehingga meningkatkan efektivitas sistem deteksi serangan. Dengan kombinasi keduanya, administrator dapat mendapatkan gambaran yang lebih jelas tentang aktivitas jaringan dan merespons ancaman dengan lebih efektif.

Berdasarkan permasalahan yang ada, penelitian ini bertujuan untuk menerapkan sistem keamanan jaringan berbasis IDS dan IPS menggunakan Snort dan Wireshark. Pendekatan ini diharapkan mampu membantu administrator jaringan dalam mendeteksi, mencegah, dan melindungi sistem dari ancaman serangan seperti eksploitasi Metasploit, sehingga menjaga integritas dan keamanan data pada jaringan komputer.

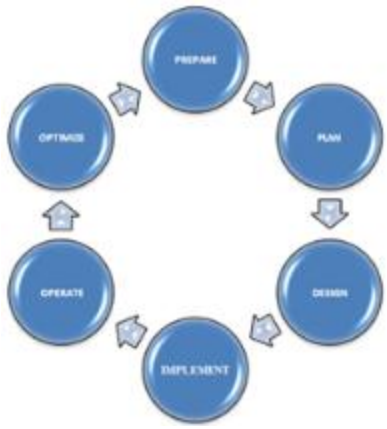
Pada penulisan jurnal ini tentunya menggunakan beberapa sumber jurnal yang mendukung dari internet. Jurnal yang pertama yang berjudul "Network Attack Detection Using Intrusion Detection System Utilizing Snort Based on Telegram" membahas penggunaan Snort sebagai Intrusion Detection System (IDS) untuk mendeteksi serangan jaringan seperti DDoS, SQL Injection, dan XSS, yang diintegrasikan dengan Telegram untuk memberikan peringatan secara real-time[1]. Sistem ini juga memungkinkan langkah mitigasi, seperti pemblokiran IP penyerang dan perubahan kata sandi server, melalui perintah di Telegram. Selain itu, sistem mengkategorikan tingkat risiko serangan menjadi Low, Medium, dan High, membantu administrator jaringan merespons dengan cepat dan mengamankan jaringan dari ancaman. Kedua, jurnal dengan judul "Wireless Network Security Design And Analysis Using Wireless Intrusion Detection System" membahas implementasi Intrusion Detection System (IDS) berbasis Snort untuk meningkatkan keamanan jaringan, terutama dalam mendeteksi serangan Denial of Service (DoS) pada jaringan nirkabel[2]. Dengan menggunakan Snort sebagai mesin sensor dan Iptables untuk menangani serangan, sistem ini mampu mengidentifikasi tindakan mencurigakan, seperti serangan ping dan flood, serta menyediakan langkah mitigasi untuk memblokir IP penyerang. Penelitian ini menekankan pentingnya analisis data lalu lintas jaringan untuk mendeteksi ancaman sejak dini, dengan fokus pada efisiensi sistem keamanan berbasis IDS dalam melindungi jaringan nirkabel.

Ketiga, jurnal dengan judul "Implementation of ids (intrusion detection system) di directorate innovation and business incubator using portsentry" membahas penerapan Intrusion Detection System (IDS) menggunakan Portsentry untuk meningkatkan keamanan jaringan pada Direktorat Inovasi dan Inkubator Bisnis (DIIB)[3]. Penelitian ini menunjukkan bahwa Portsentry dapat secara efektif melindungi website dan penyimpanan data dari serangan dengan memantau lalu lintas jaringan, memberikan laporan peringatan, dan memblokir akses yang tidak sah. Hal ini membantu mencegah akses yang berpotensi membahayakan sistem.

Keempat, jurnal dengan judul "Implementasi intrusion detection prevention system sebagai sistem keamanan jaringan komputer kejaksaan negeri pariaman menggunakan snort dan iptables berbasis linux" membahas penggunaan Snort sebagai IDS untuk mendeteksi serangan seperti DoS secara real-time dan IPTables sebagai IPS untuk memblokir serangan dengan aturan tertentu, menunjukkan efektivitas kombinasi keduanya dalam melindungi jaringan[4]. Kelima, jurnal dengan judul "Penerapan Intrusion Detection and Prevention System (IDPS) pada Jaringan komputer sebagai pencegahan serangan Port-Scanning" membahas penerapan Intrusion Detection and Prevention System (IDPS) dengan metode Port Scan Detection (PSD) pada router Mikrotik untuk mendeteksi dan mencegah serangan port scanning[5]. Dengan menetapkan aturan seperti pengaturan threshold dan parameter lalu lintas, sistem mampu mendeteksi aktivitas mencurigakan dan mencegah eksploitasi jaringan, relevan dengan pendekatan mencegah serangan menggunakan IDS/IPS. Keenam, jurnal dengan judul "Forensik Jaringan Terhadap Serangan ARP Spoofing Menggunakan Metode TAARA" membahas investigasi serangan ARP Spoofing menggunakan metode TAARA (Trigger, Acquire, Analysis, Report, Action) untuk memastikan bukti digital dapat dianalisis dan divalidasi[6]. Alat seperti Wireshark digunakan untuk mengamati lalu lintas jaringan.

METODE

Metode yang digunakan dalam penelitian ini adalah model PPDIOO (Prepare, Plan, Design, Implement, Operate, and Optimize). Tahapan penelitian pada metode PPDIOO dapat dijelaskan sebagai berikut [7]:



Gambar 1. Skema siklus PPDIO

A. Prepare

Pada tahap awal penelitian, langkah-langkah yang dilakukan bertujuan untuk memastikan semua persiapan berjalan dengan baik agar proses penelitian dapat dilakukan secara lancar dan terarah. Pertama, data dan informasi yang relevan dikumpulkan sebagai dasar untuk membangun sistem keamanan yang diinginkan. Proses ini mencakup pencarian sumber informasi, pengolahan data, dan analisis kebutuhan sistem. Setelah data terkumpul, dilakukan konfigurasi awal sistem untuk menguji fungsionalitasnya dan menemukan kekurangan yang mungkin ada. Jika ditemukan masalah, langkah selanjutnya adalah melakukan perbaikan dan penyesuaian agar sistem dapat bekerja secara optimal. Identifikasi masalah pada tahap ini sangat penting untuk mencegah kendala yang lebih besar di kemudian hari.

B. Plan

Dalam tahap ini, yang dilakukan adalah perancangan dan menentukan hardware serta software yang akan digunakan dalam penelitian ini. Proses ini melibatkan pemilihan perangkat keras yang mampu mendukung kinerja sistem keamanan yang diinginkan, seperti server yang memiliki spesifikasi memadai. Selain itu, pemilihan software juga sangat penting untuk memastikan bahwa alat yang digunakan dapat berfungsi dengan optimal dan sesuai dengan kebutuhan penelitian. Dengan perancangan yang matang, diharapkan sistem yang dikembangkan dapat secara efektif mendeteksi dan mencegah serangan siber yang mungkin terjadi.

C. Design

Dalam tahap desain ini, dibuat sebuah topologi jaringan yang sesuai dengan rencana yang telah disusun. Topologi ini menggambarkan bagaimana perangkat-perangkat dalam jaringan akan saling terhubung. Selain itu, desain ini juga mempertimbangkan faktor-faktor seperti kinerja dan keamanan jaringan. Dengan topologi yang baik, diharapkan sistem jaringan dapat berfungsi secara efisien dan efektif.

D. Implement

Dalam tahap ini, dilakukan implementasi sesuai dengan rencana yang telah disusun. Proses ini mencakup penerapan semua elemen yang telah didesain sebelumnya. Selain itu, perhatian diberikan pada setiap detail untuk memastikan semuanya berjalan dengan baik. Dengan implementasi yang tepat, diharapkan sistem dapat berfungsi sesuai harapan dan memenuhi tujuan yang ditetapkan.

E. Operate

Merupakan tahap untuk menjalankan sistem yang telah dibangun. Pada tahap ini, semua komponen dioperasikan untuk memastikan sistem berfungsi dengan baik. Pengujian dilakukan untuk mengidentifikasi dan mengatasi masalah yang mungkin muncul. Dengan pengoperasian yang efektif, diharapkan sistem dapat berjalan sesuai dengan yang diharapkan. dibangun.

F. Optimize

Optimasi dilakukan dengan menganalisis data dan melakukan evaluasi. Tujuan dari tahap ini adalah untuk meningkatkan kinerja sistem agar lebih baik. Proses ini mencakup identifikasi area yang perlu diperbaiki untuk mencapai efisiensi yang lebih tinggi. Dengan optimasi yang tepat, diharapkan sistem dapat berfungsi secara maksimal dan memenuhi kebutuhan yang ada.

Pada tahapan Prepare akan dilakukan secara bersamaan dengan tahap Plan, dikarenakan antara perancangan dan persiapan saling berhubungan satu dengan yang lain. Sehingga pada tahap selanjutnya akan menjadi lebih terarah.

Tabel 1 dan 2 persiapan kebutuhan perangkat pendukung software dan hardware yang digunakan untuk melakukan simulasi serangan Remot Exploit seperti berikut:

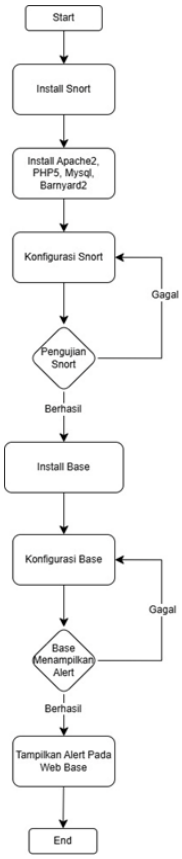
TABEL I. TABEL PERANGKAT LUNAK

No	Komponen	Versi	Keterangan
1	Ubuntu Desktop	20.04	Sebagai server OS
2	Snort	2.9.20	Digunakan untuk menangkap serangan yang masuk ke server
3	Base	1.4.5	Menyimpan alert ke dalam Web Base
4	Wireshark	3.2.3	Menganalisis jaringan komputer
5	Kali Linux	6.0	Sebagai Attacker
6	Metasploit	-	Aplikasi yang digunakan untuk melakukan serangan remot exploit ke komputer target
7	Virtualbox	7.0	Aplikasi virtual mesin untuk menjalankan OS.

TABEL II. TABEL PERANGKAT KERAS

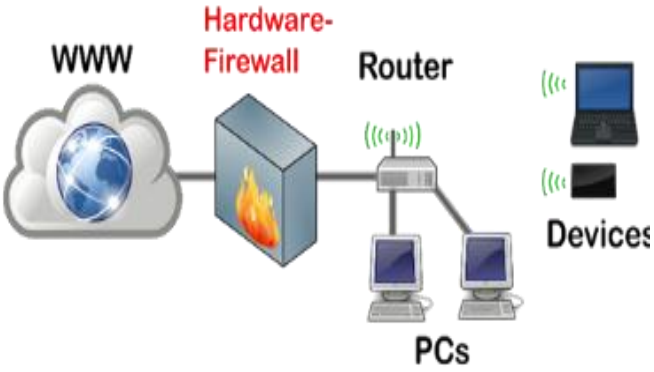
Server	IP Address
Server	192.168.56.1
Penyerang	192.168.56.1

Setelah selesai melakukan persiapan hardware dan software, maka dilanjutkan dengan design. Dalam tahap desain ini dilakukan sebuah desain topologi jaringan yang sesuai dengan perancangan penelitian.



Gambar 2. Diagram alir instalasi dan konfigurasi snort ids

Gambar 2 menunjukkan Alir instalasi dan konfigurasi Snort IDS. Dimulai dengan melakukan penginstalan Snort sekaligus akan di konfigurasi dan menambahkan Rules-rules Snort, kemudian melakukan penginstalan paket Apache2, MySQL server, Barnyard2. Selanjutnya melakukan konfigurasi Snort dengan paket-paket yang sudah diinstal dan akan melakukan pengujian jika konfigurasi gagal maka akan diulang dan jika berhasil Snort akan dilanjutkan konfigurasi Web BASE. Jika konfigurasi berhasil maka akan diarahkan ditampilan pada Web BASE jika gagal ulangi dari konfigurasi BASE.



Gambar 3. Topologi penelitian

Pada gambar 3 Design merupakan tahapan perancangan topologi jaringan yang digunakan. Pada gambar 3 dijelaskan setelah melakukan konfigurasi pada Snort IDS, peneliti akan melakukan serangan Remote Exploit menggunakan tools Metasploit untuk mengakses komputer target. Komputer target menjalankan Snort IDS sebagai alat untuk mendeteksi serangan dan menggunakan Wireshark untuk menganalisis serangan tersebut.

Fase Implement 'Implementasi Intrusion Detection System (IDS) untuk Mendeteksi serangan Metasploit Exploit Menggunakan Snort IDS dan Wireshark' diadakan sebagai langkah implementasi istilah keamanan Snort IDS dan analisis paket yang melalui jaringan server."

Pada gambar 3 Design merupakan tahapan perancangan topologi jaringan yang digunakan. Pada gambar 3 dijelaskan setelah melakukan konfigurasi pada Snort IDS, peneliti akan melakukan serangan Remote Exploit menggunakan tools Metasploit untuk mengakses komputer target. Komputer target menjalankan Snort IDS sebagai alat untuk mendeteksi serangan dan menggunakan Wireshark untuk menganalisis serangan tersebut.

Fase Implement 'Implementasi Intrusion Detection System (IDS) untuk Mendeteksi serangan Metasploit Exploit Menggunakan Snort IDS dan Wireshark' diadakan sebagai langkah implementasi istilah keamanan Snort IDS dan analisis paket yang melalui jaringan server.

HASIL DAN PEMBAHASAN

Pertama, dilakukan pengecekan statistics menggunakan wireshark dari file yang ingin dilakukan pengecekan. Pengecekan terhadap paket tingkat tinggi pada menu

Statistics → Capture File Properties.

Ada lebih dari 18.000 paket yang ditangkap dalam file ini. Ada sekitar 32 menit antara pengambilan paket pertama dan terakhir, dari pukul 17:09 hingga 17:42.

Statistics

Measurement	Captured
Packets	18189
Time span, s	1962.205
Average pps	9.3
Average packet size, B	642
Bytes	11674125
Average bytes/s	5,949
Average bits/s	47 k

Gambar 4. Pengecekan terhadap paket tingkat tinggi

Time

First packet:	2024-08-15 07:09:43
Last packet:	2024-08-15 07:42:26
Elapsed:	00:32:42

Gambar 5. Waktu (menit) antara pengambilan paket

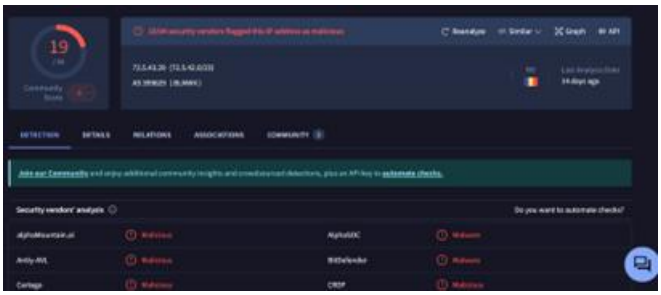
Dan pada menu Statistics → Conversations , melakukan proses pemantauan paket data untuk informasi umum tentang berkas tangkapan yang dimuat (seperti jumlah paket yang ditangkap), hingga statistik tentang protokol tertentu (misalnya statistik tentang jumlah permintaan dan respons HTTP yang ditangkap).

Ethernet · 9 | IPv4 · 74 | IPv6 | TCP · 533 | UDP · 154

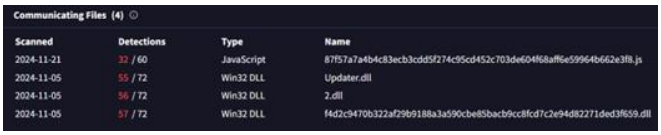
Address A	Address B	Packets	Bytes	Stream ID	Packets A → B
10.8.15.133	72.5.43.29	3,497	866 kB	45	1,844
10.8.15.133	10.8.15.4	2,248	579 kB	5	1,204
10.8.15.133	23.220.103.18	1,544	1 MB	11	469
10.8.15.133	23.220.103.8	1,531	1 MB	18	389
10.8.15.133	104.21.55.70	2,275	3 MB	37	262

Gambar 6. Memfilter tabel untuk menampilkan percakapan

Hasilnya menunjukkan bahwa host internal 10.8.15.133 memiliki komunikasi tertinggi, bertukar total 3.497 paket, dengan sebagian besar lalu lintas yang tampaknya mengalir dalam satu arah. akan mengidentifikasi 10.8.15.133 sebagai target yang mungkin karena semua alamat berasal dari IP ini. Selain itu, Alamat B tampaknya merupakan IP luar. Sedikit lebih rendah, alamat IP internal lain, 10.8.15.4, juga merupakan komunikator utama dengan 10.8.15.133. Saat ini, akan berfokus pada alamat IP eksternal 72.5.43.29.



Gambar 7. Deteksi malicious dalam analisis keamanan jaringan.



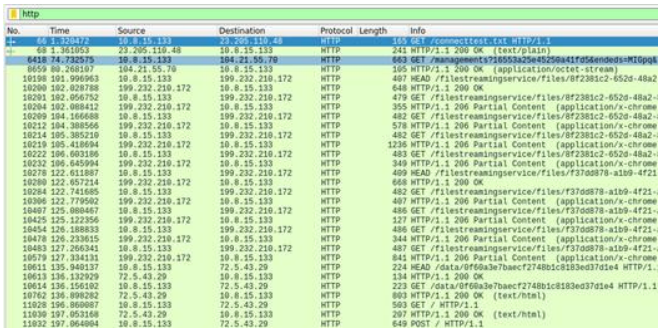
Gambar 8. Analisis file yang terdeteksi menunjukkan berbagai tingkat ancaman pada javascript dan DLL di sistem

Transmission Control Protocol	84.5	15367
Transport Layer Security	11.0	2009
NetBIOS Session Service	1.5	270
SMB2 (Server Message Block Protocol version 2)	1.5	267
Data	0.0	4
SMB (Server Message Block Protocol)	0.0	3
Lightweight Directory Access Protocol	1.4	256
Kerberos	0.3	50
Hypertext Transfer Protocol	3.6	658
Online Certificate Status Protocol	0.0	1
Media Type	0.0	9
Line-based text data	1.7	309
Distributed Computing Environment / Remote Procedure Call (DCE/RPC)	1.9	344
SAMR (pidl)	0.2	30
Microsoft Network Logon	0.1	10
Local Security Authority	0.1	12
DCE/RPC Endpoint Mapper	0.3	52
Active Directory Replication	0.8	138

Gambar 9. Rincian persentase dan volume data

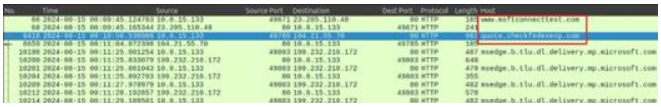
Dari gambar dapat dilihat bahwa lokasinya di Rumania. Kemudian, memeriksa koneksi, tampaknya beberapa pustaka tautan dinamis (DLL) dan file JavaScript berinteraksi dengan alamat IP ini, masing-masing menghasilkan banyak deteksi. Selanjutnya pada menu Statistics → Protocol Hierarchy, untuk meninjau protokol yang paling umum untuk menemukan peluang yang mudah untuk prospek yang akan datang.

Meskipun HTTP mewakili kurang dari 4% dari semua paket yang ditangkap, namun jumlahnya masih mencapai 600 paket. Karena HTTP tidak dienkripsi, HTTP menjadi titik awal yang cocok untuk memeriksa dan menganalisis paket-paket tersebut. Ketika memfilter HTTP, mengamati bahwa dua dari tiga alamat IP tujuan terhubung ke host sumber internal : 23.205.110.48, 199.232.210.172, dan 72.5.43.29.



Gambar 10. Log permintaan HTTP

Ketika menyertakan “Host” sebagai filter, sehingga bisa mengamati nama domain yang terselesaikan.

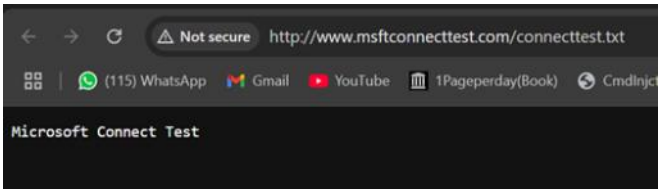


Gambar 11. Data jaringan HTTP

Selanjutnya, jelajahi domain-domain ini. Server internal memulai permintaan GET untuk berkas connecttest.txt. tidak mengetahui hal itu, sehingga akan menelitinya.

```
▾ Hypertext Transfer Protocol
  ▾ GET /connecttest.txt HTTP/1.1\r\n
      Request Method: GET
      Request URI: /connecttest.txt
      Request Version: HTTP/1.1
    Connection: Close\r\n
    User-Agent: Microsoft NCSI\r\n
    Host: www.msftconnecttest.com\r\n
    \r\n
    [Response in frame: 68]
    [Full request URI: http://www.msftconnecttest.com/connecttest.txt]
```

Gambar 12. Protokol transfer hiperteks



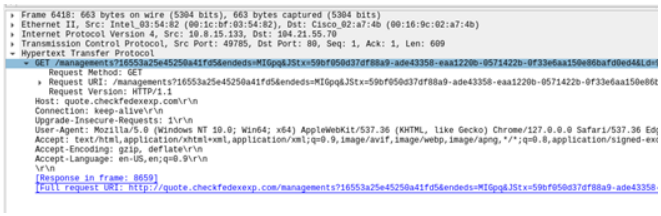
Gambar 13. Halaman uji koneksi microsoft

Setelah mengulik lebih lanjut, connectest.tst adalah file teks yang digunakan Network Connectivity Status Indicator (NCSI) untuk menentukan apakah komputer terhubung ke internet. Tampaknya itu asli.Host internal juga melakukan permintaan GET ke domain kedua. Satu hal yang baru saja diperhatikan adalah sebaiknya mencatat alamat MAC host sumber.

```
▸ Frame 66: 165 bytes on wire (1320 bits), 165 bytes captured (1320 bits)
▸ Ethernet II, Src: Intel_03:54:82 (00:1c:bf:03:54:82), Dst: Cisco_02:a7:4b (00:16:9c:02:a7:4b)
▸ Internet Protocol Version 4, Src: 10.8.15.133, Dst: 23.205.110.48
▸ Transmission Control Protocol, Src Port: 49671, Dst Port: 80, Seq: 1, Ack: 1, Len: 111
▸ Hypertext Transfer Protocol
```

Gambar 14. Detail paket jaringan TCP/IP

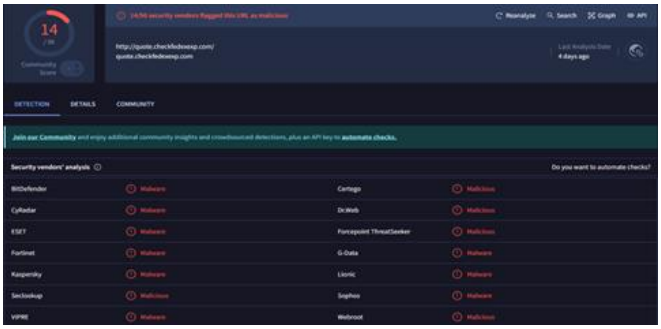
Sekarang beralih ke permintaan GET ke domain kedua setelah melakukan pemeriksaan konektivitas jaringan menggunakan Network Connectivity Status Indicator (NCSI).



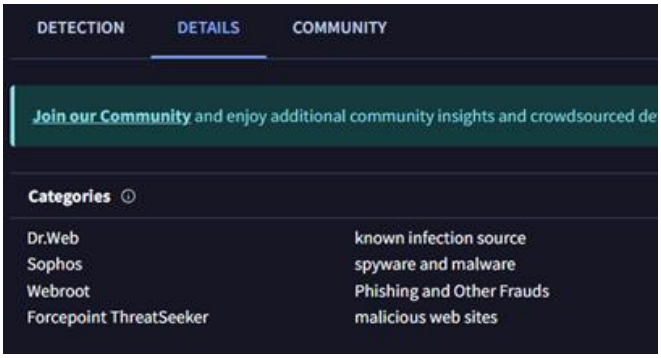
```
▸ Frame 6418: 663 bytes on wire (5304 bits), 663 bytes captured (5304 bits)
▸ Ethernet II, Src: Intel_03:54:82 (00:1c:bf:03:54:82), Dst: Cisco_02:a7:4b (00:16:9c:02:a7:4b)
▸ Internet Protocol Version 4, Src: 10.8.15.133, Dst: 104.21.55.70
▸ Transmission Control Protocol, Src Port: 49785, Dst Port: 80, Seq: 1, Ack: 1, Len: 609
▾ Hypertext Transfer Protocol
  ▾ GET /managements?16553a25e45250a41fd5&endeds=MIGpq&JStx=59bf050d37df88a9-ade43358-eaa1220b-0571422b-0f33e6aa150e86bafd0ed44&Ld=
      Request Method: GET
    ▸ Request URI: /managements?16553a25e45250a41fd5&endeds=MIGpq&JStx=59bf050d37df88a9-ade43358-eaa1220b-0571422b-0f33e6aa150e86
      Request Version: HTTP/1.1
    Host: quote.checkfedexexp.com\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.0.0 Safari/537.36 Ed
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-ex
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9\r\n
    \r\n
    [Response in frame: 8659]
    [Full request URI: http://quote.checkfedexexp.com/managements?16553a25e45250a41fd5&endeds=MIGpq&JStx=59bf050d37df88a9-ade43358
```

Gambar 15. Permintaan HTTP ke Server

Selanjutnya memverifikasi reputasi domain tersebut. Dari analisis yang ditampilkan, terlihat bahwa 14 dari 96 vendor keamanan menandai URL ini sebagai berbahaya atau "malicious".



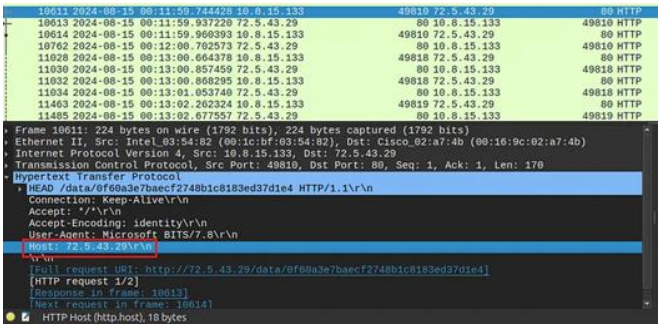
Gambar 16. Analisis keamanan URL



Gambar 17. Rincian kategori ancaman

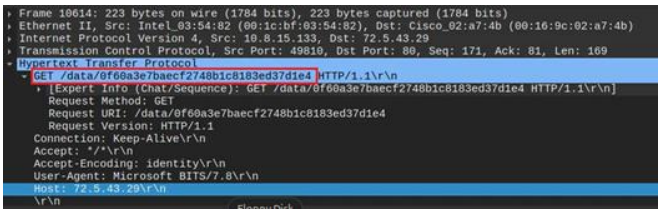
Seperti yang dilihat pada gambar, itu tampak tidak baik. Dari gambar dilihat bahwa hasil deteksi menunjukkan bahwa url memiliki banyak jenis malware yang berbahaya seperti Dr.Web, Sophos, Webroot, Forcepoint ThreatSeeker dan lain

sebagainya. Terakhir, mari meninjau diskusi yang melibatkan titik akhir komunikasi utama yaitu 72.5.43.29.



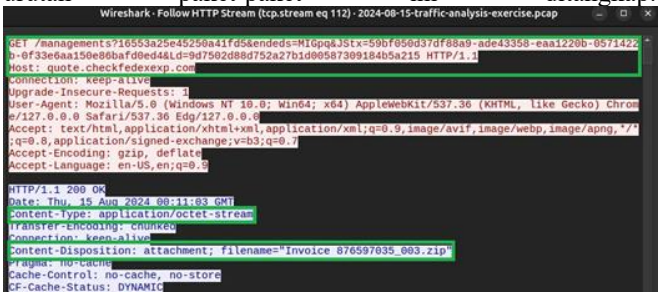
Gambar 18. Analisis Permintaan HTTP

Hasil tersebut mengkhawatirkan karena tidak menyelesaikan ke nama host. Selain itu, melihat bahwa ia membuat permintaan GET untuk - sesuatu.

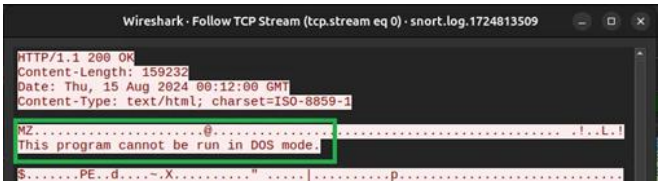


Gambar 19. Detail permintaan GET

Sejauh ini, telah mengidentifikasi dua permintaan GET yang misterius. Sekarang, saatnya untuk menyelidiki isinya. akan mulai dengan IP tujuan 199.232.210.172, mengikuti urutan paket-paket ini ditangkap.



Gambar 20. Analisis aliran HTTP



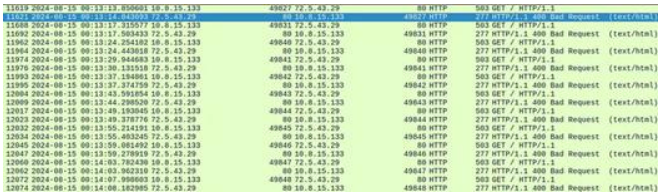
Gambar 21. File biner terdeteksi.

Dapat dilihat bahwa tipe kontennya adalah application/octet-stream, tetapi yang lebih penting lagi, ada lampiran .zip.Sekarang akan diperiksa aliran HTTP untuk host tujuan 72.5.43.29. Host internal digunakan untuk mengunduh file yang dapat dieksekusi seperti yang ditunjukkan oleh "Program ini tidak dapat dijalankan dalam mode DOS" di bawah header file, diikuti oleh teks seperti biner.

Hanya untuk mengonfirmasi, apakah header file MZ ini? Wikipedia memiliki daftar yang berguna tentang berbagai jenis header file dan nilai heksadesimalnya.

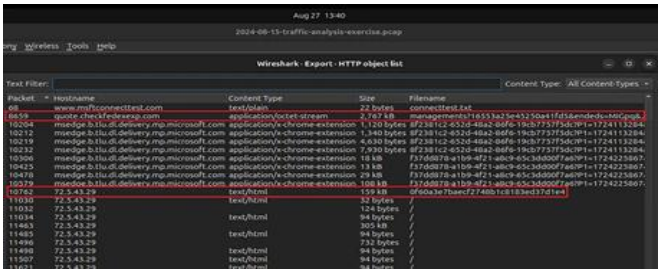
Gambar 22. Deteksi file eksekusi MZ

MZ adalah byte ajaib untuk file yang dapat dieksekusi, termasuk DLL. Ingat ketika memeriksa 72.5.43.29 pada VirusTotal? Bagian "*Communicating Files*" mengidentifikasi beberapa DLL yang dikaitkan dengan alamat ini. Melihat sisa percakapan ini, sebagian besar merupakan tanggapan "*Bad Request*" yang bolak-balik. Apakah ini komunikasi pasca-kompromi?



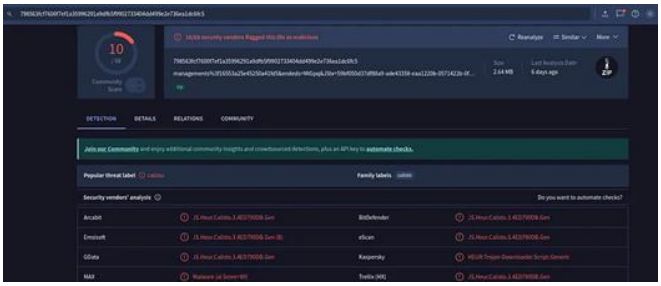
Gambar 23. Analisis permintaan HTTP mencurigakan.

Setelah mengidentifikasi dua kemungkinan indicators of compromise (IOCs), mari lanjutkan dengan mengekstrak berkas-berkas tersebut dan melakukan beberapa pemeriksaan. Untuk itu, akan melakukan/memeriksa untuk ***File Reputation Checks.*** Pertama, mari mengekspor objek HTTP. pada menu File → Export Objects → HTTP.

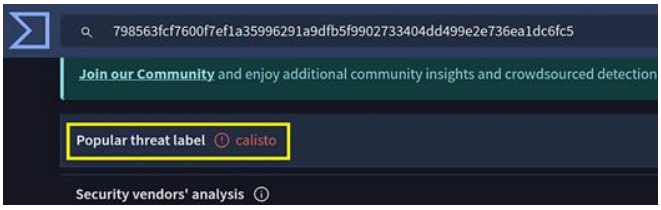


Gambar 24. Ekspor objek HTTP berbahaya

Dengan file yang telah diunduh, selanjutnya akan menghitung hash SHA256 dan menjalankannya melalui VirusTotal.

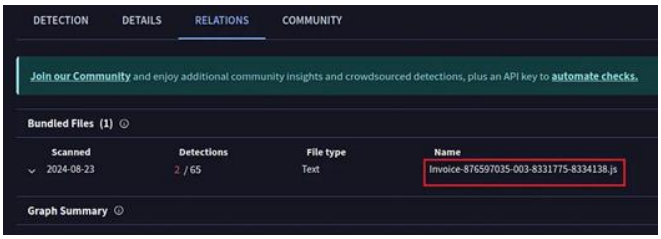


Gambar 25. Analisis keamanan deteksi ancaman permintaan HTTP



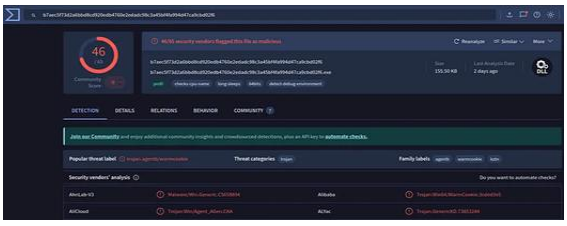
Gambar 26. Label Ancaman Terkemuka

Secara kronologis, akan mulai dengan memeriksa hash file .zip. Tampaknya Callisto adalah ancaman persisten tingkat lanjut (APT) yang berbasis di Rusia yang terkenal. Dengan memeriksa tab "Relasi" di VirusTotal, melihat bahwa file zip berisi file JavaScript.



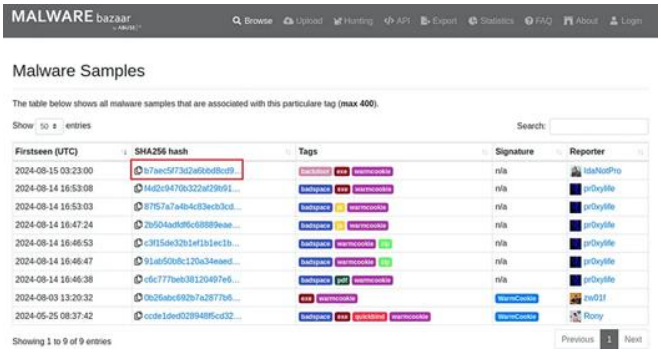
Gambar 27. Deteksi file teks

Beralih ke file kedua, melihat bahwa file ini memiliki tanda tangan yang jauh lebih kuat di antara mesin antivirus.



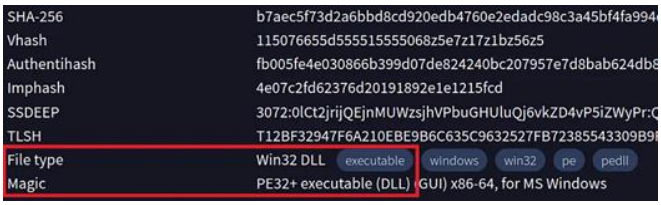
Gambar 28. Deteksi ancaman file

Itu juga menunjukkan bahwa mengidentifikasi ancaman yang terkenal adalah "trojan" dan "warm cookie". Dengan memeriksa Malware Bazaar sebagai sumber lain. Di bagian atas, menemukan hash file bersama delapan hash tambahan yang sesuai dengan tanda tangan pintu belakang "warm cookie".



Gambar 29. Sample malware terdaftar

Kembali ke VirusTotal, memeriksa Detail menunjukkan bahwa file tersebut memang sebuah DLL.



Gambar 30. Detail file win32 DLL

Ada banyak informasi sandboxing yang harus disaring dari file-file ini, tetapi satu hal yang jelas: hos internal kemungkinan besar telah disusupi.

Dengan wawasan baru ini, mari coba untuk menulis aturan deteksi untuk memperingatkan lalu lintas jaringan yang mungkin mengandung aktivitas berbahaya. **Snort** sangat populer untuk analisis jaringan karena menggabungkan analisis lalu lintas waktu nyata dengan deteksi berbasis tanda tangan, yang membantunya mengidentifikasi dan merespons berbagai ancaman jaringan. Yang sangat bagus untuk situasi adalah yang dimana bisa memberikan file pcap yang tersimpan pada Snort dan menguji deteksi di luar lingkungan langsung dengan menggunakan mode IDS/IPS-nya.

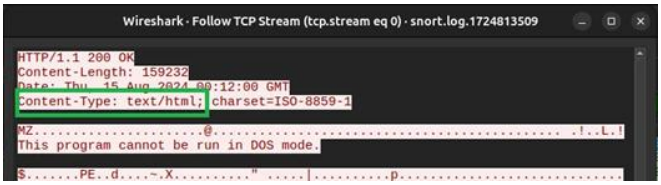
Bagian yang sulit adalah tidak ada satu pun permintaan GET yang memiliki ekstensi yang mengidentifikasikannya sebagai berkas yang dapat dieksekusi, termasuk berkas zip.

Dengan ini, dapat mencoba mendeteksi berdasarkan Content-Type, tetapi itu juga tidak dapat diandalkan. Sebagai contoh, berkas zip memiliki Tipe Konten application/octet-stream, yang tidak eksklusif untuk berkas yang dapat

dieksekusi. Selain itu, file zip tidak secara inheren berbahaya - file zip dapat berisi semua jenis file. Memblokir file zip mungkin bukan merupakan kepentingan terbaik bagi organisasi.

Gambar 31. Analisis aliran HTTP terperinci

Untuk file DLL, Tipe-Content bahkan lebih tidak dapat diandalkan, melaporkannya sebagai teks/html.



Gambar 32. Analisis aliran TCP HTML

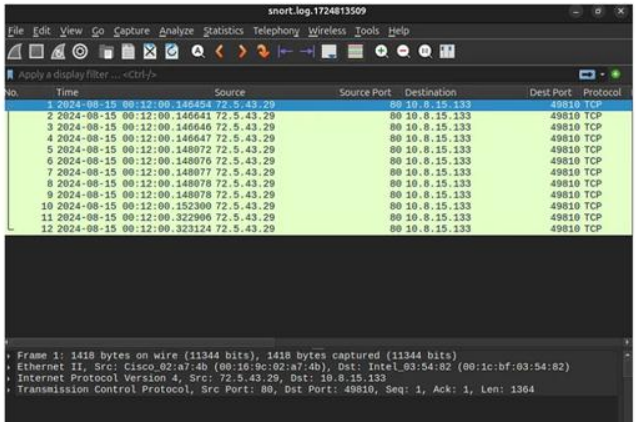
Solusi cepatnya mungkin dengan memblokir alamat IP tujuan, tetapi ingin menulis sebuah aturan yang bisa bertahan dan diterapkan pada banyak situasi lainnya. Untuk itu, akan fokus pada file yang dapat dieksekusi daripada file zip untuk memperhitungkan unduhan langsung dari file yang berpotensi berbahaya.

Salah satu metode yang dapat diandalkan (meskipun dapat dilewati) adalah menulis aturan yang mencari byte ajaib dari setiap file. Sebagai contoh, file yang dapat dieksekusi ditandai dengan “MZ”, seperti yang dilihat di atas, yang sesuai dengan nilai heksadesimal 4D 5A, seperti yang tertera di Wikipedia. Dengan mempertimbangkan semua itu, membuat aturan Snort akan terlihat seperti ini:

Gambar 33. Log analisis jaringan kesalahan

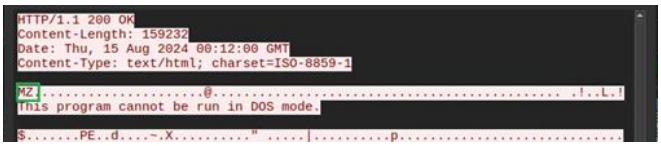
Menjalankan Snort terhadap berkas pcap, dan membuka log Snort di Wireshark untuk melihat paket yang memicu peringatan.

```
sudo wireshark /var/log/snort/snort.log.17248
```



Gambar 34. Log wireshark snort TCP

Dengan mengikuti aliran TCP dalam login, dapat dilihat dan mengonfirmasi header file MZ di awal file. Seperti pada gambar, itu bekerja dengan baik.



Gambar 35. Deteksi file mencurigakan

SIMPULAN

Pada tahap ini, telah terdeteksi dan diverifikasi malware pada mesin dengan alamat IP 10.8.15.133. Ini menandakan poin penting dalam penyelidikan, di mana perlu dipahami bagaimana malware tiba di host ini. Sangatlah penting untuk melakukan referensi silang terhadap log dari SIEM, EDR, atau file PCAP lainnya untuk mendapatkan pemahaman yang lebih lengkap tentang perkembangan serangan. Secara signifikan, alamat IP lain dalam subnet yang sama, 10.8.15.4, juga berkomunikasi dengan host yang disusupi, yang menunjukkan kemungkinan adanya upaya pergerakan lateral oleh penyerang.

Penyelidikan dimulai dengan perspektif umum, menyempurnakan fokus dengan memeriksa data lalu lintas dan protokol spesifik untuk mendeteksi ketidakberesan. Melalui analisis aliran paket HTTP, ekspor objek HTTP, dan pemantauan garis waktu dari pengunduhan file .zip pertama hingga pengunduhan file DLL berikutnya, rute dan tindakan malware diungkap.

Langkah-langkah berikut ini meliputi evaluasi tingkat insiden. Harus dilakukan pemeriksaan host tambahan untuk mengetahui kemungkinan kompromi, berkonsentrasi pada interaksi, protokol, dan portalnya. Memanfaatkan informasi

yang terkumpul—termasuk nama host, alamat IP, dan nama file—memungkinkan untuk mendapatkan wawasan tentang skala infeksi dan mengatasi ancaman lebih lanjut. Dalam konteks ini, implementasi Intrusion Detection System (IDS) seperti Snort dan alat analisis jaringan seperti Wireshark dapat berfungsi untuk mendeteksi serangan Remote Exploit. Dengan integrasi kedua alat ini, server dapat memberikan alert saat terdeteksi serangan, serta melakukan langkah pencegahan dengan memblokir alamat IP atau port yang teridentifikasi sebagai ancaman.

UCAPAN TERIMA KASIH

Kami ingin mengucapkan terima kasih kepada semua pihak yang telah berkontribusi dalam penelitian ini. Pertama, kami mengucapkan terima kasih yang tulus kepada dosen pembimbing kami, Bapak Rudy Chandra, S.Kom., M.Kom dan Ibu Mei Pane, S.Tr.Kom, yang telah memberikan bimbingan, arahan, dan masukan berharga sepanjang proses penelitian. Dukungan akademis dan motivasi dari kedua beliau sangat membantu kami dalam menghasilkan penelitian ini. Kerjasama tim kami sangat memperkaya proses penelitian ini.

Akhirnya, kami mengucapkan terima kasih kepada keluarga kami atas dukungan moral dan semangat yang tiada henti, yang selalu mendorong kami untuk menyelesaikan penelitian ini dengan baik. Tanpa dukungan dan kontribusi dari semua pihak, penelitian ini tidak akan dapat terwujud.

DAFTAR PUSTAKA

[1] J. Fernandes, “Analisis Keamanan Jaringan Wireles Landi Dinas Perpustakan Dan Kearsipan Kota Pekanbaru,” pp. 1–79, 2021.

[2] S. Maesaroh, L. Kusumaningrum, N. Sintawana, D. P. Lazirkha, and R. D. O., “Wireless Network Security Design And Analysis Using Wireless Intrusion Detection System,” Int. J. Cyber IT Serv. Manag., vol. 2, no. 1, pp. 30–39, 2022, doi: 10.34306/ijcitsm.v2i1.74.

[3] R. N. Dasmen and A. Abdul, “IMPLEMENTATION OF IDS (INTRUSION DETECTION SYSTEM) DI DIRECTORATE INNOVATION AND BUSINESS INCUBATOR USING PORTSENTRY,” vol. xx, no. x, pp. 1–6, 2013.

[4] H. Awal, “Implementasi Intrusion Detection Prevention System Sebagai Sistem Keamanan Jaringan Komputer Kejaksaan Negeri Pariaman Menggunkan Snort Dan Iptables Berbasis Linux,” J. Sains Inform. Terap., vol. 2, no. 1, pp. 38–44, 2023, doi: 10.62357/jsit.v2i1.184.

[5] Nuroji, “Penerapan Intrusion Detection andPrevention System (IDPS) padaJaringan komputer sebagaipencegahan serangan Port-Scanning,” J. Data Sci. Inf. Syst., vol. 1, no. 2, pp. 41–49, 2023, [Online]. Available: https://doi.org/10.58602/dimis.v1i2.35

[6] A. WIJAYANTO, “Forensik Jaringan Terhadap Serangan Arp Spoofing Menggunakan Metode Taara,” 2023, [Online]. Available: https://dspace.uii.ac.id/handle/123456789/42627

[7] B. A. B. Iii and M. Penelitian, “Model Pengembangan Sistem Jaringan PPDIOO,” pp. 26–34, [Online]. Available: https://repository.atmaluhur.ac.id/bitstream/handle/123456789/3384/BAB III.pdf?isAllowed=y&sequence=4

[8] W. Seo and W. Pak, “Real-Time Network Intrusion Prevention System Based on Hybrid Machine Learning,” IEEE Access, vol. 9, pp. 46386–46397, 2021, doi: 10.1109/ACCESS.2021.3066620.

[9] F. Riza, “Sistem Deteksi Intrusi pada Server secara Realtime Menggunakan Seleksi Fitur dan Firebase Cloud Messaging,” J. Sistim Inf. dan Teknol., vol. 5, pp. 7–9, 2022, doi: 10.37034/jsisfotek.v5i1.161.

[10] A. Nugraha and D. A. Gustian, “Deteksi Malware Dridex Menggunakan Signature-based Snort,” Indones. J. Comput. Sci., vol. 10, no. 1, pp. 54–64, 2022, doi: 10.33022/ijcs.v10i1.3068.

[11] G. Pradita and A. Pramono, “Implementasi Monitoring Keamanan Jaringan Pada Server Ubuntu Menggunakan Snort Intrusion Detection Prevention System (Idps) Dan Telegram Bot Sebagai Media Notifikasi Di Pt Ss Utama,” JATI (Jurnal Mhs. Tek. Inform., vol. 8, no. 4, pp. 5827–5834, 2024, doi: 10.36040/jati.v8i4.10069.

[12] E. Stephani, Fitri Nova, and Ervan Asri, “Implementasi dan Analisa Keamanan Jaringan IDS (Intrusion Detection System) Menggunakan Suricata Pada Web Server,” JITSI J. Ilm. Teknol. Sist. Inf., vol. 1, no. 2, pp. 67–74, 2020, doi: 10.30630/jitsi.1.2.10.

[13] M. N. Hafizh, I. Riadi, and A. Fadlil, “Forensik Jaringan Terhadap Serangan ARP Spoofing menggunakan Metode Live Forensic,” J. Telekomun. dan Komput., vol. 10, no. 2, p. 111, 2020, doi: 10.22441/incomtech.v10i2.8757.

[14] J. A. Dharma and Rino, “Network Attack Detection Using Intrusion Detection System Utilizing Snort Based on Telegram,” bit-Tech, vol. 6, no. 2, pp. 118–126, 2023, doi: 10.32877/bt.v6i2.943.

[15] I. sri wahyuningsi Manguling and J. M. Parenreng, “Security System Analysis Using the HTTP Protocol Against Packet Sniffing Attacks,” Internet Things Artif. Intell. J., vol. 3, no. 4, pp. 325–340, 2023, doi: 10.31763/iota.v3i4.612.

[16] P. B. Pramudya, “Implementation of signature-based intrusion detection system using SNORT to prevent threats in network servers,” J. Soft Comput. Explor., vol. 3, no. 2, pp. 93–98, 2022, doi: 10.52465/joscex.v3i2.80.

[17] “View of Implementasi Intrusion Detection System (IDS) untuk Mendeteksi serangan Metasploit Exploit Menggunakan Snort dan Wireshark.pdf.”