ANALISIS KINERJA SIEM WAZUH DENGAN FIM, YARA, DAN VIRUSTOTAL DALAM MENDETEKSI MALWARE PADA UBUNTU SERVER

SKRIPSI

ROBBY AKBAR ABDULLAH

2107421004

PROGRAM STUDI TEKNIK MULTIMEDIA DAN JARINGAN

JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER

POLITEKNIK NEGERI JAKARTA

2025

ANALISIS KINERJA SIEM WAZUH DENGAN FIM, YARA, DAN VIRUSTOTAL DALAM MENDETEKSI MALWARE PADA UBUNTU SERVER

SKRIPSI

Dibuat untuk Melengkapi Syarat-Syarat yang Diperlukan

untuk Memperoleh Diploma Empat Politeknik

ROBBY AKBAR ABDULLAH

2107421004

PROGRAM STUDI TEKNIK MULTIMEDIA DAN JARINGAN

JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER

POLITEKNIK NEGERI JAKARTA

2025

SURAT PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan di bawah ini:

Nama	: Robby Akbar Abdullah
NIM	: 2107421004
Jurusan / Program Studi	: Teknik Informatika dan Komputer / Teknik Multimedia dan Jaringan
Judul Skripsi	: Analisis Kinerja SIEM Wazuh dengan FIM, YARA, dan VirusTotal dalam Mendeteksi Malware pada Ubuntu Server

Menyatakan dengan sebenarnya bahwa skripsi ini benar-benar merupakan hasil karya saya sendiri, bebas dari peniruan terhadap karya dari orang lain. Kutipan pendapat dan tulisan orang lain ditunjuk sesuai dengan cara-cara penulisan karya ilmiah yang berlaku.

Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa dalam skripsi ini terkandung ciri-ciri plagiat dan bentuk-bentuk peniruan lain yang dianggap melanggar peraturan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Depok, 11 Juli 2025

Yang membuat pernyataan,

(Robby Akbar Abdullah)

NIM. 2107421004

LEMBAR PENGESAHAN

Skripsi diajukan oleh:

Nama : Robby Akbar Abdullah

NIM : 2107421004

Program Studi : Teknik Multimedia dan Jaringan

Judul Skripsi : Analisis Kinerja SIEM Wazuh dengan FIM, YARA, dan VirusTotal dalam Mendeteksi Malware pada Ubuntu Server

Telah diuji oleh tim penguji dalam Sidang Skripsi pada hari Senin, tanggal 23, bulan Juni, tahun 2025 dan dinyatakan **LULUS.**

Disahkan oleh

Pembimbing I : Iik Muhamad Malik Matin, S.Kom., M.T. ( )

Penguji I : Asep Kurniawan, S.Pd., M.Kom. ( )

Penguji II : Fachroni Arbi Murad, S.Kom., M.Kom. ( )

Penguji III : Ariawan Andi Suhandana, S,Kom., M.Ti. ( )

Mengetahui:

Jurusan Teknik Informatika dan Komputer

Ketua

(Dr. Anita Hidayati, S.Kom., M.Kom.)

NIP. 197908032003122003

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Allah *subhanahu wa ta'ala* atas pertolonganNya dan karuniaNya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Analisis Kinerja SIEM Wazuh dengan FIM, YARA, dan VirusTotal dalam Mendeteksi Malware pada Ubuntu Server". Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk memperoleh gelar Diploma Empat Politeknik. Penulis menyadari bahwa tanpa bantuan dan dukungan dari berbagai pihak, penulis kesulitan dalam menyelesaikan skripsi ini. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Bapak Iik Muhammad Malik Matin, S.Kom., M.T. selaku dosen pembimbing yang telah banyak membantu, mendukung, dan memberikan masukan serta saran kepada penulis selama pengerjaan skripsi ini hingga selesai.
2. Bunda, ayah, dan kakak yang telah memberikan bantuan dan dukungan material maupun moral yang nilainya tak terhingga.
3. Sahabat dan teman-teman seperjuangan yang telah memberikan bantuan dan dukungan dalam pengerjaan skripsi ini.
4. Seseorang bernama "Zae" yang selalu mendoakan, memberikan semangat, dan menjadi motivasi dalam menyelesaikan skripsi ini.

Penulis meminta maaf sebesar-besarnya apabila ada kesalahan atau kekurangan dalam proses pengerjaan skripsi ini. Penulis juga mengucapkan terima kasih sebanyak-banyaknya kepada seluruh pihak dan semoga Allah membalas kebaikan kalian semua. Penulis berharap skripsi ini bermanfaat bagi pembaca dan menjadi rujukan untuk melakukan penelitian selanjutnya.

Jakarta, 4 Juni 2025

Penulis

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Politeknik Negeri Jakarta, saya bertanda tangan di bawah ini:

Nama	: Robby Akbar Abdullah
NIM	: 2107421004
Jurusan / Program Studi	: Teknik Informatika dan Komputer / Teknik Multimedia dan Jaringan

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Politeknik Negeri Jakarta Hak Bebas Royalti Non-Eksklusif atas karya ilmiah saya yang berjudul:

Analisis Kinerja SIEM Wazuh dengan FIM, YARA, dan VirusTotal dalam Mendeteksi Malware pada Ubuntu Server

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksklusif ini Politeknik Negeri Jakarta berhak menyimpan, mengalihmediakan/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan skripsi saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Depok, 11 Juli 2025

Yang menyatakan,

(Robby Akbar Abdullah)

NIM. 2107421004

ANALISIS KINERJA SIEM WAZUH DENGAN FIM, YARA, DAN VIRUSTOTAL DALAM MENDETEKSI MALWARE PADA UBUNTU SERVER

ABSTRAK

Seiring dengan pesatnya perkembangan teknologi, maka serangan siber juga semakin kompleks dan beragam. Serangan siber tersebut terjadi dalam berbagai sektor sehingga menyebabkan banyak permasalahan keamanan siber. Oleh karena itu, dibutuhkan suatu sistem yang dapat mendeteksi, mencegah, dan merespon serangan siber melalui Security Information and Event Management (SIEM). Solusi SIEM yang tepat adalah Wazuh karena bersifat zero-cost dan open-source. Namun, perlu dilakukan analisis kinerja SIEM Wazuh tersebut dalam mendeteksi serangan siber. Berdasarkan penelitian terdahulu, analisis kinerja SIEM hanya sebatas parameter tertentu dan menguji serangan web dan network attack saja sehingga perlu dianalisis dari sisi malware agar kinerjanya lebih baik. Analisis kinerja SIEM Wazuh menggunakan FIM, YARA, dan VirusTotal untuk mengetahui mana yang paling baik kinerjanya dalam mendeteksi malware dengan menggunakan parameter-parameter tertentu. Hasil dari penelitian ini adalah FIM tidak berhasil mendeteksi semua malware, YARA berhasil mendeteksi semua malware, dan VirusTotal hanya mendeteksi dua dari tiga malware karena bergantung kepada database threat intelligence VirusTotal.

***Kata Kunci**: FIM, Malware, VirusTotal, Wazuh, YARA*

DAFTAR ISI

SURAT PERNYATAAN BEBAS PLAGIARISME i
LEMBAR PENGESAHAN ii
KATA PENGANTAR iii
SURAT PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS iv
ABSTRAK v
DAFTAR ISI vi
DAFTAR GAMBAR ix
DAFTAR TABEL xiii
BAB I PENDAHULUAN 1
1.1 Latar Belakang Masalah 1
1.2 Perumusan Masalah 3
1.3 Batasan Masalah 3
1.4 Tujuan dan Manfaat 4
1.5 Sistematika Penulisan 5
BAB II TINJAUAN PUSTAKA 6
2.1 Tinjauan Pustaka 6
2.1.1 *Security Information and Event Management* (SIEM) 6
2.1.2 Wazuh 6
2.1.3 *Server* 6
2.1.4 *File Integrity Monitoring* (FIM) 7
2.1.5 YARA 7
2.1.6 VirusTotal 7
2.1.7 *Malware* 7

2.2 Penelitian Terkait 8

BAB III METODE PENELITIAN 11

3.1 Rancangan Penelitian 11

3.1.1 Model atau *framework* yang digunakan 11

3.1.2 Teknik Pengumpulan dan Analisis Data 11

3.2 Tahapan Penelitian 12

3.3 Objek Penelitian 13

BAB IV HASIL DAN PEMBAHASAN 14

4.1 Analisis Kebutuhan 14

4.1.1 Spesifikasi Sistem 14

4.1.2 Konfigurasi Jaringan 14

4.2 Perancangan Sistem 15

4.2.1 Arsitektur sistem 15

4.2.2 *Flowchart* sistem 16

4.3 Implementasi Sistem 17

4.3.1 Instalasi Wazuh 17

4.3.2 Instalasi Wazuh *Agent* 22

4.3.3 Instalasi Suricata pada Wazuh *Agent* 23

4.3.4 Intalasi *Web Server* pada Wazuh *Agent* 29

4.3.5 Konfigurasi *Firewall* pada Wazuh *Agent* 31

4.3.6 Penerapan FIM pada Wazuh *Agent* 33

4.3.7 Penerapan YARA pada Wazuh *Agent* 34

4.3.8 Penerapan VirusTotal pada Wazuh *Agent* 40

4.4 Pengujian 42

4.4.1 Deskripsi Pengujian 42

4.4.2 Prosedur Pengujian 42

4.4.2.1 Pengujian Serangan *Port Scanning* 42
4.4.2.2 Pengujian Serangan *Denial of Service* (DoS) 43
4.4.2.3 Pengujian Serangan SSH *Brute-force* 43
4.4.2.4 Pengujian Serangan *Malware* 45
4.4.3 Data Hasil Pengujian 48
4.4.3.1 Hasil Pengujian Serangan *Port Scanning* 48
4.4.3.2 Hasil Pengujian Serangan *Denial of Service* (DoS) 50
4.4.3.3 Hasil Pengujian Serangan SSH *brute-force* 52
4.4.3.4 Hasil Pengujian Serangan *Malware* 57
4.4.4 Analisis Data / Evaluasi Pengujian 63
4.4.4.1 Analisis Data Hasil Pengujian Serangan *Port Scanning* 63
4.4.4.2 Analisis Data Hasil Pengujian Serangan *Denial of Service* (DoS) 63
4.4.4.3 Analisis Data Hasil Pengujian Serangan SSH *Brute-force* 64
4.4.4.4 Analisis Data Hasil Pengujian Serangan *Malware* 64
BAB V PENUTUP 70
5.1 Kesimpulan 70
5.2 Saran 70
DAFTAR PUSTAKA 72
DAFTAR RIWAYAT HIDUP 73

DAFTAR GAMBAR

Gambar 1.1 Grafik Rekapitulasi Serangan *Malware* 1
Gambar 4.1 Arsitektur Sistem 15
Gambar 4.2 *Flowchart* Sistem 16
Gambar 4.3 Instalasi *Script* dan Konfigurasi Awal Wazuh 17
Gambar 4 4 Pengeditan *File* Konfigurasi Wazuh 17
Gambar 4.5 Isi *File* Konfigurasi Wazuh yang Telah Diedit 18
Gambar 4.6 Instalasi Wazuh 18
Gambar 4.7 Pengekstrakan *Password* Wazuh 19
Gambar 4.8 Instalasi Wazuh *Indexer* 19
Gambar 4.9 Inisiasi *Cluster* Wazuh 19
Gambar 4.10 Pengujian Keberhasilan Instalasi Wazuh 20
Gambar 4.11 Instalasi Wazuh *Server* 20
Gambar 4.12 Instalasi Wazuh *Dashboard* 21
Gambar 4.13 Tampilan Halaman *Login* Wazuh *Dashboard* 21
Gambar 4.14 Tampilan Halaman *Overview* Wazuh *Dashboard* 22
Gambar 4.15 Instalasi Wazuh *Agent* 22
Gambar 4.16 Pengaktifan Wazuh *Agent* 23
Gambar 4.17 Hasil Wazuh *Agent* yang Telah Berhasil Diinstalasi 23
Gambar 4.18 Penambahan Repositori Suricata 24
Gambar 4.19 Pembaruan Daftar Paket 24
Gambar 4.20 Instalasi Suricata 25
Gambar 4.21 Pengunduhan *Rules* Suricata 25
Gambar 4.22 Pengekstrakan *Rules* Suricata 26
Gambar 4.23 Pengubahan *Permission File Rules* Suricata 26
Gambar 4.24 Pengeditan *File Rules* Suricata 26
Gambar 4.25 Isi *File Rules* Suricata yang Telah Diedit 27
Gambar 4.26 Pengecekan Konfigurasi Jaringan 27
Gambar 4.27 Pembaruan Konfigurasi Suricata yang Telah Ditambahkan 27
Gambar 4.28 Pengeditan *File* Konfigurasi OSSEC 28

Gambar 4.29 Isi *File* Konfigurasi OSSEC yang Telah Diedit 28
Gambar 4.30 Pembaruan Konfigurasi OSSEC yang Telah Ditambahkan 28
Gambar 4.31 Hasil Suricata yang Telah Berhasil Diinstal pada Wazuh *Agent* 29
Gambar 4.32 Pembaruan Daftar Paket 29
Gambar 4.33 Instalasi *Web Server* Apache 30
Gambar 4.34 Pengecekan Keberhasilan Instalasi *Web Server* Apache 30
Gambar 4.35 Tampilan *Web Server* Apache pada Wazuh *Agent* 31
Gambar 4.36 Pengecekan dan Pengaktifan *Firewall* pada Wazuh *Agent* . 31
Gambar 4.37 Pengeditan *File* Konfigurasi SSH 31
Gambar 4.38 Isi *File* Konfigurasi SSH yang Telah Ditambahkan 32
Gambar 4.39 Pembaruan Konfigurasi SSH yang Telah Ditambahkan 32
Gambar 4.40 Hasil Konfigurasi *Firewall* yang Telah Ditambahkan 32
Gambar 4.41 Pengeditan *File* Konfigurasi OSSEC 33
Gambar 4.42 Penerapan FIM pada *File* Konfigurasi OSSEC 33
Gambar 4.43 Pembaruan Konfigurasi yang Telah Ditambahkan 33
Gambar 4.44 Pembaruan Daftar Paket 34
Gambar 4.45 Instalasi Dependensi 34
Gambar 4.46 Pengunduhan *Source Code* YARA 35
Gambar 4.47 Pengekstrakan *Source Code* YARA 35
Gambar 4.48 Proses *Build* dari *Source Code* YARA 36
Gambar 4.49 Pengecekan Keberhasilan Instalasi YARA 36
Gambar 4.50 Pembuatan Direktori untuk YARA *Rules* 36
Gambar 4.51 Pengunduhan YARA *Rules* 37
Gambar 4.52 Pembuatan *Script* untuk *Scanning* YARA 37
Gambar 4.53 Pengubahan *Owner* dan *Permission* pada *File Script* YARA 37
Gambar 4.54 Pembaruan Konfigurasi OSSEC yang Telah Ditambahkan 38
Gambar 4.55 Pengeditan *File* Konfigurasi *Local Rules* 38
Gambar 4.56 Isi *File* Konfigurasi *Local Rules* yang Telah Ditambahkan 38
Gambar 4.57 Pengeditan *File Local Decoder* 39

Gambar 4.58 Isi *File* Konfigurasi *Local Decoder* yang Telah Ditambahkan 39
Gambar 4.59 Pengeditan *File* Konfigurasi OSSEC 39
Gambar 4.60 Isi *File* Konfigurasi OSSEC yang Telah Ditambahkan..... 40
Gambar 4.61 Pembaruan Konfigurasi yang Telah Ditambahkan 40
Gambar 4.62 Pengambilan API *Key* VirusTotal 41
Gambar 4.63 Pengeditan *File* Konfigurasi OSSEC 41
Gambar 4.64 Penerapan VirusTotal pada *File* Konfigurasi OSSEC 41
Gambar 4.65 Pembaruan Konfigurasi OSSEC yang Telah Ditambahkan 42
Gambar 4.66 Serangan *Port Scanning* ke Wazuh *Agent*..... 42
Gambar 4.67 Serangan DoS ke Wazuh *Agent*..... 43
Gambar 4.68 Pengujian Pertama Serangan SSH *Brute-force* ke Wazuh *Agent*..... 43
Gambar 4.69 Pengujian Kedua Serangan SSH *Brute-force* ke Wazuh *Agent* 44
Gambar 4.70 Pengujian Ketiga Serangan SSH *Brute-force* ke Wazuh *Agent* 44
Gambar 4.71 Pengujian Keempat Serangan SSH *Brute-force* ke Wazuh *Agent*..... 45
Gambar 4.72 Pembuatan *Script Malware* Pertama pada Wazuh *Agent*.... 45
Gambar 4.73 Serangan *Malware* Pertama ke Wazuh *Agent* 46
Gambar 4.74 Pembuatan *Script Malware* Kedua pada Wazuh *Agent*..... 46
Gambar 4.75 Serangan *Malware* Kedua ke Wazuh *Agent*..... 47
Gambar 4.76 Pembuatan *Script Malware* Ketiga pada Wazuh *Agent* 48
Gambar 4.77 Serangan *Malware* Ketiga ke Wazuh *Agent*..... 48
Gambar 4.78 Hasil Serangan *Port Scanning* pada Wazuh *Agent*..... 49
Gambar 4.79 Hasil Deteksi Serangan *Port Scanning* pada Wazuh *Agent* 49
Gambar 4.80 Hasil *Traffic* Serangan DoS pada Wazuh *Agent* 50
Gambar 4.81 Dampak Serangan DoS pada *Web Server* Wazuh *Agent* Sebelum Diblokir *Firewall*..... 51
Gambar 4.82 Konfigurasi *Firewall* pada Wazuh *Agent*..... 51

Gambar 4.83 Dampak Serangan DoS pada *Web Server* Wazuh *Agent* Setelah Diblokir *Firewall* 52
Gambar 4.84 Hasil Serangan Pertama SSH *Brute-force* pada Wazuh *Agent* 52
Gambar 4.85 Hasil Deteksi Serangan Pertama SSH *Brute-force* pada Wazuh *Agent* 53
Gambar 4.86 Hasil Serangan Kedua SSH *Brute-force* pada Wazuh *Agent* 53
Gambar 4.87 Hasil Deteksi Serangan Kedua SSH *Brute-force* pada Wazuh *Agent* 54
Gambar 4.88 Hasil Serangan Ketiga SSH *Brute-force* pada Wazuh *Agent* 54
Gambar 4.89 Hasil Deteksi Serangan Ketiga SSH *Brute-force* pada Wazuh *Agent* 55
Gambar 4.90 Hasil Serangan Keempat SSH *Brute-force* pada Wazuh *Agent* 55
Gambar 4.91 Hasil Deteksi Serangan Keempat SSH *Brute-force* pada Wazuh *Agent* 56
Gambar 4.92 Pengujian *Login* SSH ke Wazuh *Agent* 56
Gambar 4.93 Hasil Deteksi Pengujian *Login* SSH ke Wazuh *Agent* 57
Gambar 4.94 Hasil Serangan *Malware* Pertama pada Wazuh *Agent* 57
Gambar 4.95 Hasil Serangan *Malware* Kedua pada Wazuh *Agent* 58
Gambar 4.96 Hasil Serangan *Malware* Ketiga pada Wazuh *Agent* 58
Gambar 4.97 Hasil Serangan *Malware* Pertama pada Wazuh *Agent* 59
Gambar 4.98 Hasil Serangan *Malware* Kedua pada Wazuh *Agent* 59
Gambar 4.99 Hasil Serangan *Malware* Ketiga pada Wazuh *Agent* 60
Gambar 4.100 Hasil Serangan *Malware* Pertama pada Wazuh *Agent* 60
Gambar 4.101 Detail *Malware* Pertama di VirusTotal 61
Gambar 4.102 Hasil Serangan *Malware* Kedua pada Wazuh *Agent* 61
Gambar 4.103 Detail *Malware* Kedua di VirusTotal 62
Gambar 4.104 Hasil Serangan *Malware* Ketiga pada Wazuh *Agent* 62
Gambar 4.105 Beban Sistem yang Digunakan 63

DAFTAR TABEL

Tabel 2.1 Penelitian Terkait 8
Tabel 4.1 Spesifikasi Sistem 14
Tabel 4.2 Kebutuhan NAT *Network* 14
Tabel 4.3 Kebutuhan *Port Forwarding* 15
Tabel 4.4 Analisis Data Hasil Pengujian Serangan *Malware* Menggunakan FIM 64
Tabel 4.5 Analisis Data Hasil Pengujian Serangan *Malware* Menggunakan YARA 65
Tabel 4.6 Analisis Data Hasil Pengujian Serangan *Malware* Menggunakan VirusTotal 67
Tabel 4.7 Analisis Data Hasil Pengujian Serangan *Malware* pada Wazuh *Agent* 69

BAB I
PENDAHULUAN

1.1 Latar Belakang Masalah

Era digital adalah era yang mengalami perkembangan teknologi dengan sangat pesat. Seiring dengan pesatnya perkembangan teknologi tersebut, maka serangan siber juga semakin kompleks dan beragam. Serangan siber tersebut ditujukan kepada infrastruktur teknologi dan informasi dalam berbagai sektor. Hal tersebut menyebabkan banyak sekali terjadinya permasalahan keamanan siber dikarenakan lemahnya sistem keamanan. (Husnul Khotimah, 2022)

Berdasarkan laporan tahunan *honeynet* yang diterbitkan oleh (Badan Siber dan Sandi Negara, 2024) jumlah serangan siber terutama *malware* setiap tahunnya mengalami peningkatan pada sistem *honeynet* secara keseluruhan. Data serangan *malware* yang tercatat pada sistem *honeynet* sebanyak 1.232.072 pada tahun 2024. Berikut adalah rekapitulasi serangan *malware* lima tahun terakhir pada sistem *honeynet* yang tercantum pada Gambar 1.1.

Gambar 1.1 Grafik Rekapitulasi Serangan Malware

Sumber: (Badan Siber dan Sandi Negara, 2024)

Berdasarkan permasalahan-permasalahan keamanan siber tersebut, maka dibutuhkan adanya suatu sistem yang dapat mendeteksi, mencegah, dan meresponnya. Oleh karena itu, dibutuhkan penerapan yang lebih efektif melalui *Security Information and Event Management* (SIEM). Teknologi SIEM ini dapat melakukan pengumpulan informasi keamanan yang berasal dari data *log* berbagai sumber secara *real-time*. (Husnul Khotimah, 2022)

Solusi SIEM yang tepat adalah menggunakan Wazuh karena bersifat *zero-cost* dan *open-source* sehingga mudah melakukan *custom* tanpa mengeluarkan biaya. Namun, perlu dilakukan analisis mengenai kinerja SIEM Wazuh tersebut dalam mendeteksi serangan siber. Penelitian yang dilakukan oleh (Oky Dwi Prasetyo, 2023) menganalisis kinerja SIEM Wazuh untuk menguji serangan *brute-force* dan *Denial of Service* (DoS) dengan parameter *signature-based detection* dan *anomaly-based detection*. Selain itu, terdapat penelitian yang serupa menganalisis kinerja SIEM IBM QRadar oleh (Adirosa, 2021) untuk menguji serangan *port scanning*, *password checking*, DoS, dan eksploit metasploitable 3 dengan parameter *flow source* dan *log source*. Selanjutnya, terdapat penelitian menganalisis kinerja SIEM Elastic Stack dan Splunk oleh (Alfandi, 2022) untuk menguji serangan *fingerprinting*, SQL *injection*, DoS, dan *port scanning* dengan parameter pengiriman notifikasi dan pengujian selama tiga hari.

Berdasarkan beberapa penelitian tersebut, pengujian hanya dilakukan dari serangan *web* dan *network attack* saja sehingga perlu dianalisis dari sisi *malware* agar kinerjanya lebih baik. *Malware* atau *malicious software* dapat berdampak berbahaya bagi sistem, rusaknya data, tersebarnya infeksi, dan banyak kerugian lainnya. Oleh karena itu, perlu dilakukan analisis kinerja SIEM Wazuh dalam mendeteksi *malware* pada *server* agar kinerjanya lebih baik. SIEM Wazuh secara *default* hanya terdapat *File Integrity Monitoring* (FIM) sehingga harus diinstalasi dengan YARA dan VirusTotal agar kinerjanya lebih baik. Selanjutnya, dapat dianalisis kinerja tersebut berdasarkan parameter-parameter yang telah ditentukan, yaitu jumlah *malware* yang terdeteksi, kecepatan deteksi *malware*, tingkat kesalahan, jenis *malware* yang terdeteksi, dan beban sistem yang digunakan sehingga dapat lebih cepat dan tanggap dalam menangani serangan siber terutama *malware*.

1.2 Perumusan Masalah

Berdasarkan hal-hal yang telah disampaikan di atas, maka rumusan permasalahannya:

1. Bagaimana rancang bangun SIEM Wazuh untuk mendeteksi serangan *port scanning*, *DoS*, *brute-force*, dan *malware*?
2. Bagaimana penerapan FIM, YARA, dan VirusTotal ke dalam SIEM Wazuh untuk mendeteksi *malware*?
3. Bagaimana hasil analisis kinerja SIEM Wazuh dengan FIM, YARA, dan VirusTotal dalam mendeteksi *malware*?

1.3 Batasan Masalah

Penelitian ini berfokus pada:

1. SIEM yang digunakan terbatas pada Wazuh.
2. Penerapan *tools* dalam mendeteksi *malware* pada SIEM Wazuh terbatas pada FIM, YARA, dan VirusTotal.
3. *Server* yang dimonitoring terbatas pada sistem operasi Ubuntu *Server*.
4. Eksperimen yang dilakukan terbatas menggunakan *virtual machine* dalam ruang lingkup lokal dan berada pada jaringan yang sama.
5. Pengujian yang dilakukan terbatas pada serangan *port scanning*, *Denial of Service* (DoS), *brute-force*, dan *malware*.
6. Parameter-parameter yang diuji dalam mendeteksi *malware* terbatas pada jumlah *malware* yang terdeteksi, kecepatan deteksi *malware*, tingkat kesalahan, jenis *malware* yang dideteksi, dan beban sistem yang digunakan.
7. Analisis kinerja yang dilakukan pada SIEM Wazuh terbatas pada hasil pengujian serangan *port scanning*, *Denial of Service* (DoS), *brute-force*, dan *malware*.

1.4 Tujuan dan Manfaat

Penelitian ini memiliki tujuan yang akan dicapai, antara lain:

1. Merancang bangun SIEM Wazuh untuk mendeteksi serangan *port scanning*, *DoS*, *brute-force*, dan *malware*.
2. Menerapkan FIM, YARA, dan VirusTotal ke dalam SIEM Wazuh untuk mendeteksi *malware*.
3. Mengukur kinerja SIEM Wazuh dengan FIM, YARA, dan VirusTotal dalam mendeteksi *malware*.

Penelitian ini diharapkan dapat memberikan manfaat bagi berbagai pihak, antara lain:

1. Mahasiswa
   Penelitian ini dapat menjadi referensi dalam memahami dan mengembangkan penelitian di bidang keamanan siber.
2. Dosen
   Penelitian ini dapat menjadi bahan kajian tambahan dalam pengajaran topik-topik terkait keamanan siber.
3. Institusi
   Penelitian ini dapat menjadi bukti bahwa mahasiswa telah lulus kompetensi sesuai bidangnya.
4. Industri
   Penelitian ini dapat menjadi informasi teknis atau acuan dalam memilih metode deteksi yang sesuai untuk kebutuhan industri.
5. Masyarakat
   Penelitian ini dapat meningkatkan kesadaran akan bahayanya serangan siber dan pemahaman tentang pentingnya deteksi dini terhadap serangan siber.

1.5 Sistematika Penulisan

Adapun sistematika penulisan skripsi ini:

A. BAB I PENDAHULUAN

Bab I berisi penjelasan mengenai jawaban apa dan mengapa penelitian ini perlu dilakukan. Bagian ini meliputi latar belakang masalah, perumusan masalah, batasan masalah, serta tujuan dan manfaat analisis kinerja SIEM Wazuh dengan FIM, YARA, dan VirusTotal dalam mendeteksi *malware* pada Ubuntu *Server*.

B. BAB II TINJAUAN PUSTAKA

Bab II berisi penjelasan mengenai landasan teori atau studi literatur yang berhubungan antara karya peneliti sebelumnya dengan riset penelitian yang dilakukan saat ini.

C. BAB III METODOLOGI PENELITIAN

Bab III berisi penjelasan mengenai rancangan penelitian, tahapan penelitian, dan objek penelitian.

D. BAB IV HASIL DAN PEMBAHASAN

Bab IV berisi hasil dan pembahasan mengenai penelitian yang dilakukan meliputi analisis kebutuhan, perancangan, pengujian, dan hasil analisis pengujian.

E. BAB V PENUTUP

Bab V berisi penjelasan mengenai hasil akhir dari penelitian berupa kesimpulan dan saran secara singkat terhadap pembahasan yang telah diuraikan pada bagian isi.

BAB II
TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

2.1.1 *Security Information and Event Management* (SIEM)

Security Information and Event Management (SIEM) adalah sebuah sistem yang digunakan untuk mendeteksi serangan dan merespon keamanan melalui analisis *log* dari berbagai sumber data secara *real-time*. SIEM dapat mengumpulkan data dan menghubungkannya untuk dapat dianalisis apakah statusnya *valid threat* atau *false positive*. Analisis SIEM tersebut mencakup seluruh infrastruktur yang digunakan sebuah perusahaan atau organisasi. (Muhammad Rijal Kamal, 2021)

2.1.2 Wazuh

Wazuh adalah sebuah perangkat lunak *open source* yang berfungsi sebagai *Host Intrusion Detection System* (HIDS). Wazuh memiliki kemampuan XDR (*External Data Representation*) maupun SIEM (*Security Information and Event Management*). Wazuh menyediakan fitur visibilitas keamanan yang lebih dalam infrastruktur dengan memantau *host* pada sistem operasi dan juga tingkat aplikasi. (Azzah Shafiyyah, 2024)

2.1.3 *Server*

Server adalah sebuah sistem yang mempunyai kemampuan untuk memberikan fitur atau layanan dalam jaringan komputer. Berbeda dengan *Personal Computer* (PC), *server* biasanya mempunyai *processor* yang *scalable* dan *Random Access Memory* (RAM) yang besar. Selain itu, sistem operasi yang digunakan *server* berbeda dengan sistem operasi yang dipakai pada umumnya. (Prakoso, 2017)

2.1.4 *File Integrity Monitoring* (FIM)

File Integrity Monitoring (FIM) adalah salah satu solusi untuk memonitoring sebuah sistem yang menjadi fitur bawaan dari SIEM Wazuh. *File Integrity Monitoring* (FIM) adalah sebuah proses yang memvalidasi integritas suatu *file* pada perangkat lunak dengan menggunakan metode verifikasi. *File Integrity Monitoring* (FIM) dapat melindungi infrastruktur IT karena dapat memantau perubahan yang terjadi dalam sistem, jaringan, dan aplikasi. (Lintang, 2020)

2.1.5 YARA

YARA adalah sebuah *tool* yang dapat memindai *file* dalam bentuk *text* maupun biner berdasarkan suatu *rules*. YARA *rules* dibuat untuk mendeteksi *malware* pada sistem operasi berdasarkan sifat dan karakteristiknya. YARA *rules* dapat mengidentifikasi *malware* secara efektif tergantung kuantitas dan kualitas dalam pemindaian suatu *file*. (Nur Rohman Rosyid, 2023)

2.1.6 VirusTotal

VirusTotal adalah sebuah *tool threat intelligence* yang dapat menganalisis *file*, IP, dan URL yang mencurigakan. *Tool* ini mempunyai lebih dari 70 mesin pemindai dan menyediakan secara lengkap informasi tentang *malware* yang dianalisis. VirusTotal bekerja sama dengan vendor keamanan lainnya untuk menciptakan *database* gabungan yang sangat relevan tentang *file*, IP, dan URL yang dianalisis. (Nurul Qomariah, 2023)

2.1.7 *Malware*

Malware atau *malicious software* adalah suatu program yang memiliki tujuan untuk merusak, mencuri, atau mengubah data-data yang dimiliki orang lain untuk dijadikan tindak kejahatan. *Malware* bermacam-macam bentuknya yang dapat mengeksploitasi *file* penting dalam komputer. *Malware* biasanya disisipkan melalui perangkat lunak yang beredar di internet sehingga korban dikelabui untuk melakukan instalasi aplikasi yang telah disusupi *malware* tersebut. (Yuriansyah Ilhamdi, 2017)

2.2 Penelitian Terkait

Berikut adalah referensi penelitian terkait yang berhubungan antara karya peneliti sebelumnya dengan riset penelitian saat ini yang dapat dilihat pada Tabel 2.1.

Tabel 2.1 Penelitian Terkait

No.	Judul dan Penulis	Hasil	Perbedaan
1.	Analisis Kinerja *Security Information and Event Management* (SIEM) IBM QRADAR *Community Edition* dalam Mendeteksi Ancaman dan Serangan Siber pada *Server* (Adirosa, 2021)	Tingkat efektivitas identifikasi serangan sebesar 20% dan deteksi serangan sebesar 100% dengan *log source* Linux OS dan Windows *event log*.	• SIEM yang digunakan adalah IBM QRADAR *Community Edition*. • Parameter yang digunakan adalah *flow source* dan *log source*. • Pengujian yang dilakukan adalah serangan *port scanning*, *password checking*, DoS, dan eksploit metasploitable 3.
2.	Analisa *Security Information and Event Management* (SIEM) Menggunakan	SIEM Elastic Stack dan Splunk berhasil mendeteksi semua serangan yang terjadi selama tiga hari pengujian. Hasil deteksi serangan yang	• SIEM yang digunakan adalah Elastic Stack dan Splunk. • Parameter yang digunakan adalah pengiriman

	Elastic Stack dan Splunk (Alfandi, 2022)	telah terjadi akan disimpan oleh kedua SIEM tersebut sehingga administrator dapat melakukan analisis dan visualisasi serangan yang terjadi.	notifikasi dan pengujian selama tiga hari. • Pengujian serangan yang dilakukan terbatas pada *fingerprinting*, SQL *injection*, DoS, dan *port scanning*.
3.	Uji Kinerja *Host-Based Intrusion Detection System* Wazuh terhadap Serangan *Brute-force* dan DoS (Oky Dwi Prasetyo, 2023)	Sistem dapat melakukan pengujian terhadap *Host-Based Intrusion Detection System* (HIDS) Wazuh dengan mekanisme penyerangan *brute-force* dan DoS. Skema skenario penyerangan yang dilakukan dengan strategi berbeda membuahkan hasil yang konsisten sehingga menunjukkan kestabilan Wazuh dalam mendeteksi serangan secara *real-time*.	• SIEM yang digunakan adalah Wazuh HIDS. • Parameter yang digunakan adalah *signature-based detection* dan *anomaly-based detection*. • Pengujian yang dilakukan terbatas pada serangan *brute-force* dan DoS.
4.	*A Review of* Wazuh *Tool Capabilities for*	Menyajikan fungsionalitas SIEM Wazuh dalam	• SIEM yang digunakan adalah Wazuh HIDS.

	Detecting Attacks Based on Log Analysis (Stefan Stanković, 2022)	mendeteksi serangan pada *web server*. Serangan tersebut berhasil terdeteksi secara *real-time*.	• Parameter yang digunakan adalah *system performance*. • Pengujian yang dilakukan terbatas pada serangan SSH *brute-force*.
5.	*A Performance Analysis of Intrusion Detection with Snort and Security Information Management* (Thorarensen, 2021)	Melalui eksperimen ditemukan bahwa pada sistem yang sama, tingkat putus Snort berkurang dari 9,0% menjadi 1,1% dengan mengonfigurasi beberapa utas paket untuk lalu lintas 1,0 Gbit/s. Kedua, rangkaian aturan Snort *community* dan Proofpoint ET Open menunjukkan sekitar 1% dan 31% paket terputus masing-masing.	• Sistem yang digunakan adalah Snort, PulledPork, dan Elasticsearch. • Parameter yang digunakan adalah *throughput*, *network delay*, dan *drop rate*.

BAB III
METODE PENELITIAN

3.1 Rancangan Penelitian

3.1.1 Model atau *framework* yang digunakan

Model atau *framework* yang digunakan adalah menggunakan metode eksperimen yang merupakan bagian dari metode penelitian kuantitatif. Metode eksperimen dilakukan dengan cara melakukan pengujian langsung terhadap suatu variabel dalam kondisi yang dikendalikan. Metode ini diharapkan memperoleh data yang objektif mengenai performa masing-masing fitur dalam mendeteksi aktivitas berbahaya pada sistem terutama *malware*.

3.1.2 Teknik Pengumpulan dan Analisis Data

Teknik pengumpulan dan analisis data yang digunakan adalah berdasarkan parameter-parameter berikut ini:

a. Serangan *port scanning* dilakukan menggunakan *tool* Nmap dengan parameter berikut ini:
   - Mendapatkan *port* yang terbuka atau tidak.
   - Terdeteksi oleh Suricata atau tidak.

b. Serangan *Denial of Service* (DoS) dilakukan menggunakan *tool* Hping3 dengan parameter berikut ini:
   - Kondisi *web server* Apache sebelum serangan diblokir oleh *firewall*.
   - Kondisi *web server* Apache setelah serangan diblokir oleh *firewall*.
   - Terdeteksi oleh tcpdump atau tidak.

c. Serangan SSH *brute-force* dilakukan menggunakan *tool* Hydra dengan parameter berikut ini:
   - Percobaan kombinasi *password* dengan menggunakan *username*.
   - Percobaan kombinasi *password* dengan tingkat kesulitan mudah.

- Percobaan kombinasi *password* dengan tingkat kesulitan sedang.
- Percobaan kombinasi *password* dengan tingkat kesulitan sulit.
- Terdeteksi oleh SIEM atau tidak.

d. Serangan *malware* dilakukan dengan cara membuat tiga *script* untuk mengunduh tiga *malware* dengan parameter berikut ini:
   - Jumlah *malware* yang terdeteksi
   - Kecepatan deteksi *malware*
   - Tingkat kesalahan
   - Jenis *malware* yang terdeteksi
   - Beban sistem yang digunakan

3.2 Tahapan Penelitian

Tahapan penelitian yang akan dilakukan, yaitu:

1. Studi Literatur

   Studi literatur adalah melakukan identifikasi masalah dan analisis kebutuhan melalui pengumpulan referensi dari berbagai sumber terkait penelitian. Referensi tersebut dijadikan sebagai acuan untuk penelitian yang akan dilakukan.
2. Perancangan Sistem

   Sistem akan mulai dirancang sesuai kebutuhan dari referensi yang didapatkan. Proses perancangan ini akan menggunakan arsitektur dan *flowchart* yang dijadikan sebagai acuan untuk tahap berikutnya.
3. Pembuatan Sistem

   Sistem akan dibuat sesuai rancangan sistem pada tahap sebelumnya. Sistem harus selesai dibuat agar dapat diuji pada tahap selanjutnya.
4. Pengujian Sistem

   Sistem akan dilakukan pengujian berdasarkan parameter-parameter yang telah ditentukan, yaitu jumlah *malware* yang terdeteksi, kecepatan deteksi *malware*, tingkat kesalahan, jenis *malware* yang terdeteksi, dan beban sistem yang digunakan.

5. Evaluasi dan Analisis

   Sistem akan dilakukan evaluasi dan analisis dari hasil pengujian pada tahap sebelumnya. Evaluasi ini dilakukan agar mengetahui kekurangan sistem dan dapat dijadikan saran untuk penelitian selanjutnya.

6. Penulisan Laporan dan Dokumentasi

   Tahap terakhir adalah penulisan laporan disertai dengan dokumentasi. Laporan tersebut mengacu pada Pedoman Skripsi Mahasiswa Jurusan Teknik Informatika dan Komputer.

3.3 Objek Penelitian

Fokus dari penelitian ini adalah melakukan analisis kinerja SIEM Wazuh dengan FIM, YARA, dan VirusTotal dalam mendeteksi *malware* pada Ubuntu *server*. Sistem ini akan menganalisis seberapa efisien kinerja antara FIM, YARA, dan VirusTotal pada SIEM Wazuh. Setelah dianalisis, maka didapatkan hasil kinerja yang paling baik sehingga permasalahan keamanan siber dapat dicegah dan diatasi dengan lebih baik.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Analisis Kebutuhan

4.1.1 Spesifikasi Sistem

Sistem yang dibuat terbatas menggunakan *virtual machine* dalam ruang lingkup lokal. Tabel 4.1 menunjukkan data spesifikasi sistem yang dibutuhkan pada penelitian ini.

Tabel 4.1 Spesifikasi Sistem

Nama Virtual Mesin	Sistem Operasi	CPU	RAM	*Storage*
Wazuh *Server*	Ubuntu *Server* 22.04	6 *Cores*	10 GB	70 GB
Wazuh *Agent*	Ubuntu *Server* 22.04	2 *Cores*	2 GB	25 GB
Attacker	Kali Linux 2024	4 *Cores*	4 GB	60 GB

4.1.2 Konfigurasi Jaringan

Sistem yang menggunakan *virtual machine* dalam ruang lingkup lokal tersebut membutuhkan konfigurasi jaringan yang tepat agar dapat terisolasi dengan baik. Berikut adalah konfigurasi jaringan yang dibutuhkan pada penelitian ini.

a. NAT *Network*

NAT *network* adalah sebuah konfigurasi jaringan yang memungkinkan VM dapat mengakses internet dan dapat berkomunikasi antar-VM. Namun, VM dengan *host* tidak dapat berkomunikasi sehingga membutuhkan *port forwarding*.

Tabel 4.2 Kebutuhan NAT Network

Nama Virtual Mesin	IP *Address*	Segmentasi
Wazuh *Server*	10.1.2.12	10.1.2.0/24
Wazuh *Agent*	10.1.2.8	
Attacker	10.1.2.13	

b. *Port Forwarding*

Port forwarding adalah sebuah teknik yang digunakan untuk meneruskan lalu lintas jaringan dari satu *port* ke *port* lainnya.

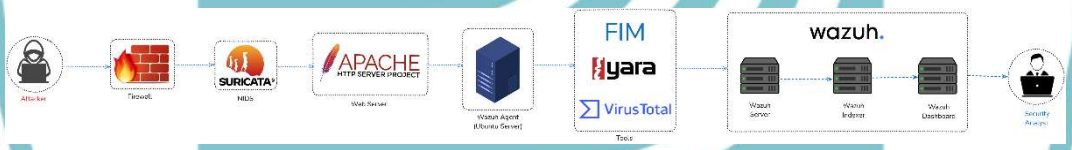
Tabel 4.3 Kebutuhan Port Forwarding

Nama Virtual Mesin	*Service*	*Port Default*	*Port Forwarding*
Wazuh *Server*	SSH	22	2201
	HTTPS	443	44301
Wazuh *Agent*	SSH	2200	22003
	HTTP	80	8002

4.2 Perancangan Sistem

Berikut ini adalah perancangan sistem yang dibutuhkan penelitian ini meliputi arsitektur sistem dan *flowchart* sistem.

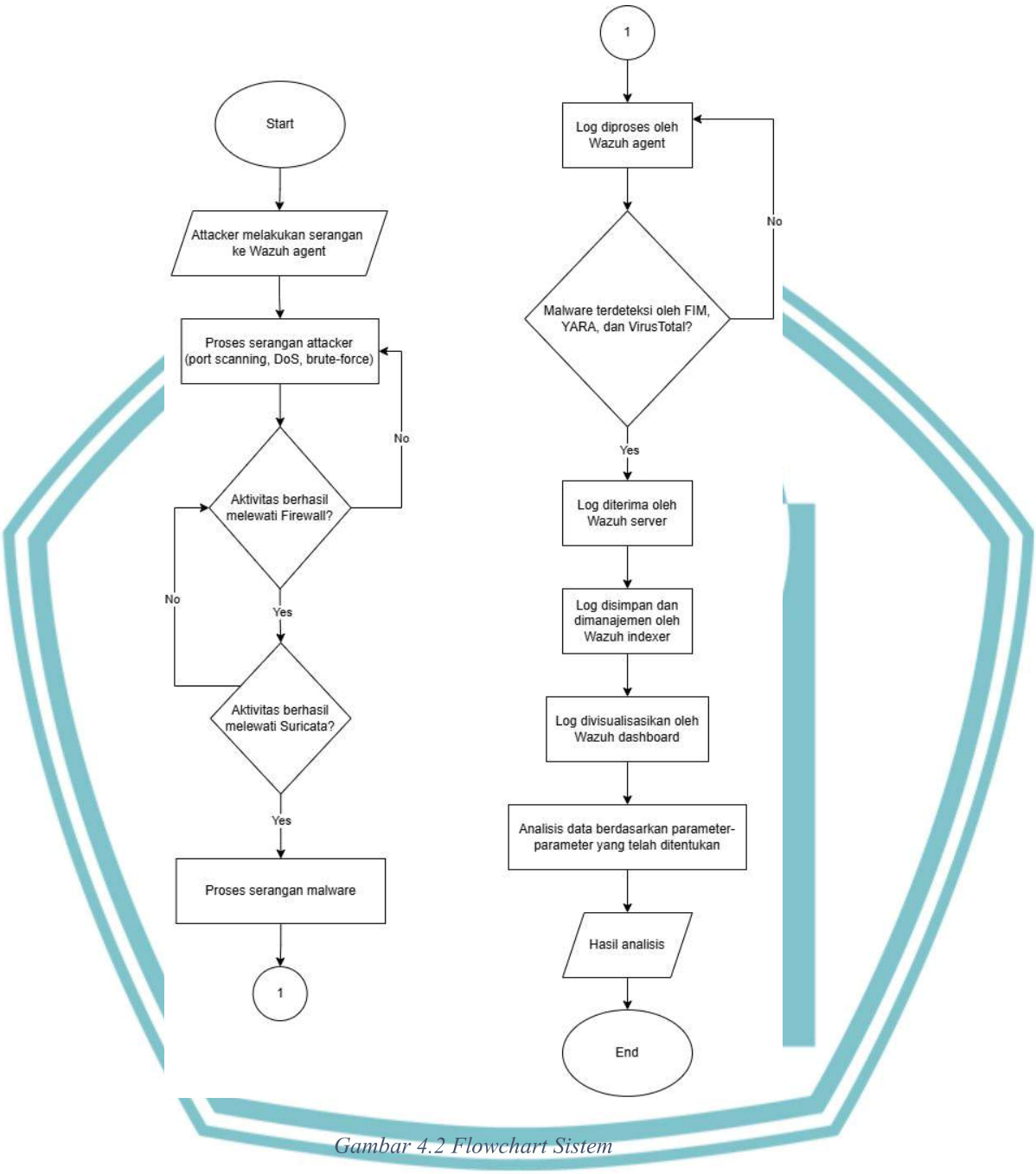
4.2.1 Arsitektur sistem



Gambar 4.1 Arsitektur Sistem

Gambar 4.1 menampilkan arsitektur sistem SIEM Wazuh yang telah dirancang. Dalam Wazuh *agent* terdapat *firewall*, Suricata, *web server* Apache, FIM, YARA, dan VirusTotal. Ketika *attacker* melakukan serangan, maka *log* yang ada di Wazuh *agent* akan dikirimkan ke Wazuh *server*. Kemudian, *log* yang berada di Wazuh *server* tersebut akan disimpan dan dimanajemen oleh Wazuh *indexer*. Lalu, data dari Wazuh *indexer* akan divisualisasikan menggunakan Wazuh *dashboard* sehingga dapat dianalisis oleh *security analyst*.

4.2.2 *Flowchart* sistem



Gambar 4.2 Flowchart Sistem

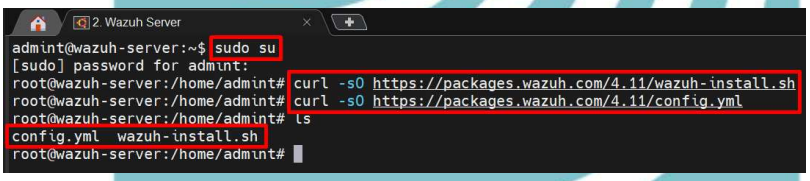
Alur kerja sistem pada Gambar 4.2 diawali dengan *attacker* melakukan serangan berupa *port scanning*, DoS, dan *brute-force* yang melewati *firewall* dan Suricata hingga mendapatkan akses ke Wazuh *agent*. Setelah mendapatkan akses, *attacker* mengunduh *malware* menggunakan *script* ke Wazuh *agent*. Jika kondisinya Wazuh *agent* yang terdapat FIM, YARA, dan VirusTotal mendeteksi bahwa ada serangan *malware*, maka *log* tersebut

akan dikirimkan ke Wazuh *server*. Jika tidak, maka *log* akan berhenti di Wazuh *agent*. Selanjutnya, semua *log* yang dikirimkan oleh Wazuh *agent* akan diterima oleh Wazuh *server*. *Log* tersebut akan disimpan dan dimanajemen oleh Wazuh *indexer*. Setelah itu, *log* akan divisualisasikan oleh Wazuh *dashboard* untuk dianalisis berdasarkan parameter-parameter yang telah ditentukan.

4.3 Implementasi Sistem

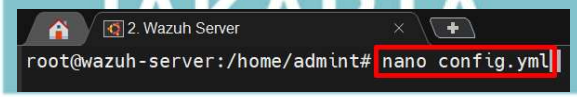
4.3.1 Instalasi Wazuh

Perintah sudo su digunakan untuk mendapatkan akses penuh ke dalam sistem sebagai root. Perintah curl digunakan untuk mengunduh *file* instalasi *script* dan konfigurasi awal Wazuh. Perintah ls untuk melihat isi *file* yang telah diunduh tersebut.



Gambar 4.3 Instalasi Script dan Konfigurasi Awal Wazuh

File config.yml yang telah diunduh sebelumnya diedit. Perintah nano digunakan untuk mengedit *file* konfigurasi Wazuh. *File* tersebut digunakan untuk manajemen otomatis Wazuh.



Gambar 4.4 Pengeditan File Konfigurasi Wazuh

Konfigurasi yang diubah adalah menggunakan IP *address* yang sama karena diinstalasi dalam satu *server* dengan keterangan sebagai berikut:

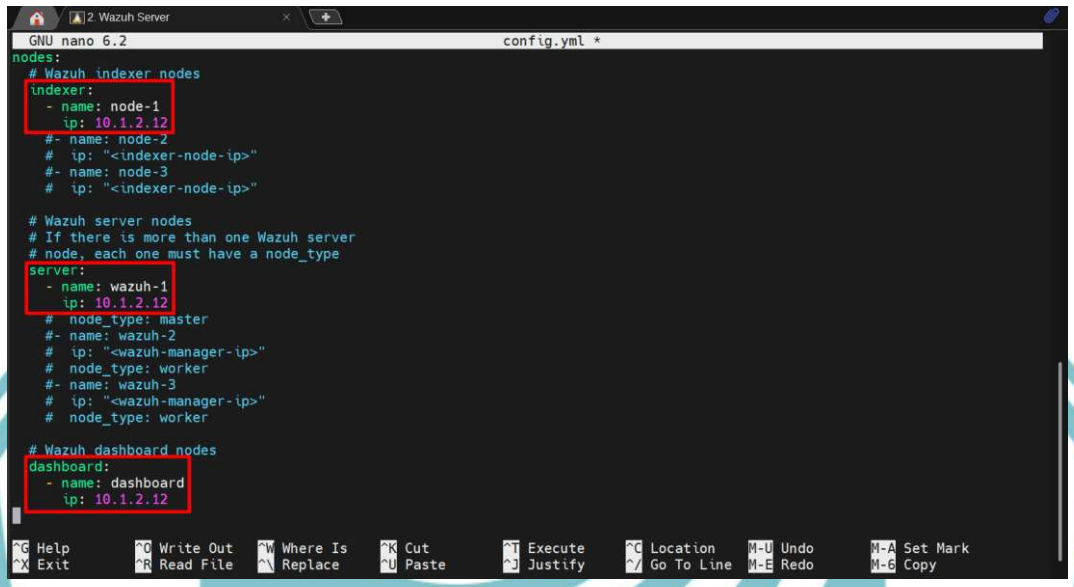
a. Wazuh *indexer*
   - *Name* : node-1
   - IP : 10.1.2.12

b. Wazuh *server*
   - *Name* : wazuh-1

- IP : 10.1.2.12

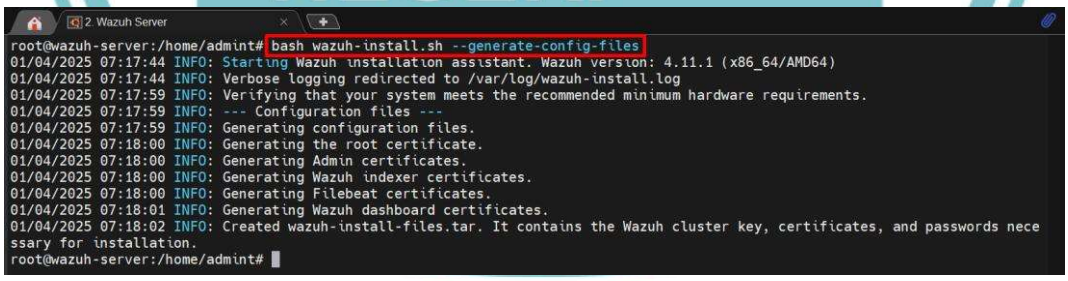
c. Wazuh *dashboard*

- *Name* : dashboard
- Wazuh : 10.1.2.12



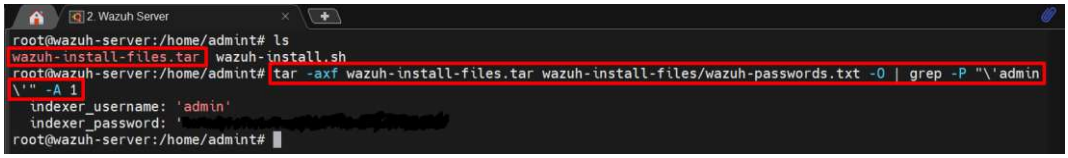
Gambar 4.5 Isi File Konfigurasi Wazuh yang Telah Diedit

Perintah bash digunakan untuk menjalankan *file shell script*. *File* tersebut digunakan untuk instalasi *script* Wazuh. *Option* menggunakan parameter –generate-config-files digunakan untuk menghasilkan *cluster key*, *certificates*, dan *password* yang dibutuhkan untuk instalasi Wazuh.



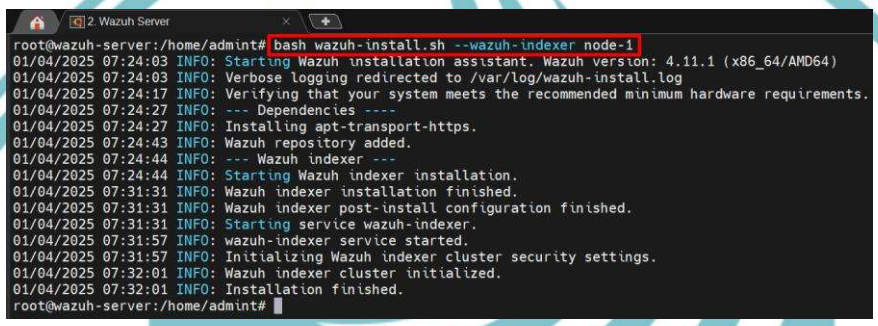
Gambar 4.6 Instalasi Wazuh

Password yang digunakan untuk mengakses Wazuh *dashboard* diekstrak dengan cara menjalankan perintah tar -axf. *Login* Wazuh *dashboard* dapat menggunakan *username* admin dan *password* yang telah didapatkan tersebut.



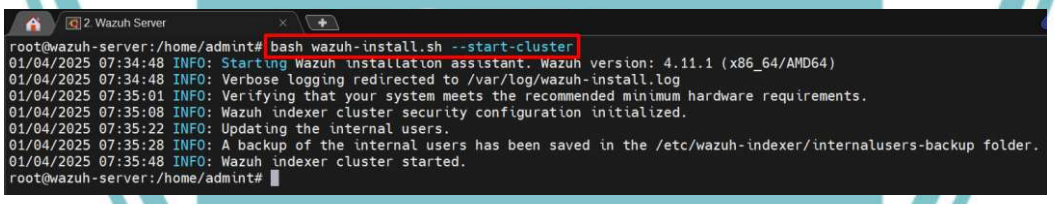
Gambar 4.7 Pengekstrakan Password Wazuh

Perintah bash digunakan untuk menjalankan *file shell script*. *File* tersebut digunakan untuk instalasi *script* Wazuh. *Option* menggunakan parameter –wazuh-indexer digunakan untuk instalasi Wazuh *indexer*, serta nama *node* disesuaikan pada *file* config.yml.



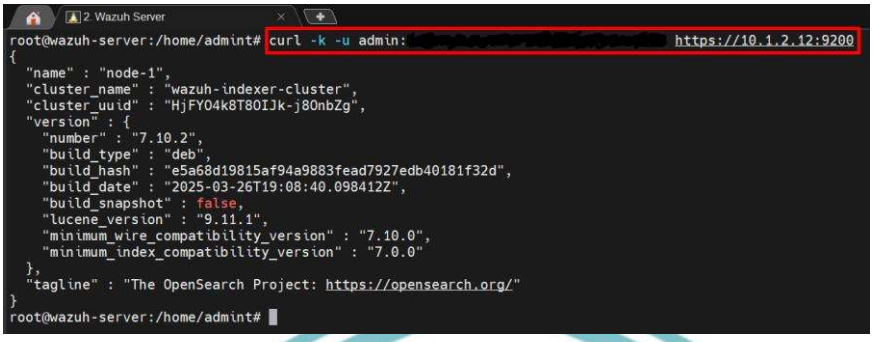
Gambar 4.8 Instalasi Wazuh Indexer

Perintah bash digunakan untuk menjalankan *file shell script*. *File* tersebut digunakan untuk instalasi *script* Wazuh. *Option* menggunakan parameter –start-cluster digunakan untuk inisiasi *cluster*.



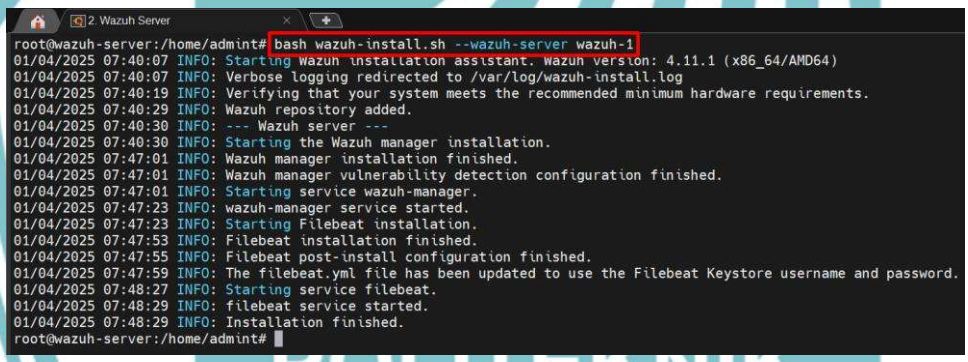
Gambar 4.9 Inisiasi Cluster Wazuh

Perintah curl -k -u dengan *user*, *password*, IP *address* dan *port* dari Wazuh untuk menguji keberhasilan instalasi Wazuh *indexer*. Wazuh yang telah berhasil diinstal akan menampilkan keterangan yang dapat dilihat pada Gambar 4.10.



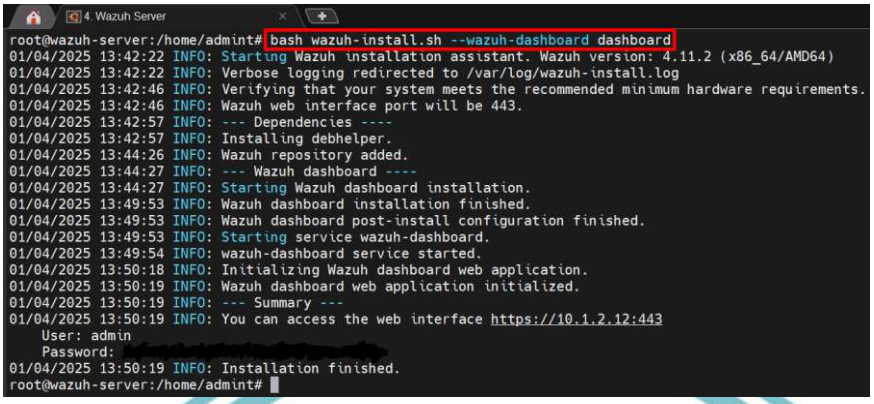
Gambar 4.10 Pengujian Keberhasilan Instalasi Wazuh

Perintah bash digunakan untuk menjalankan *file shell script*. *File* tersebut digunakan untuk instalasi *script* Wazuh. *Option* menggunakan parameter –wazuh-server digunakan untuk instalasi Wazuh *server*, serta nama *node* disesuaikan pada *file* config.yml.



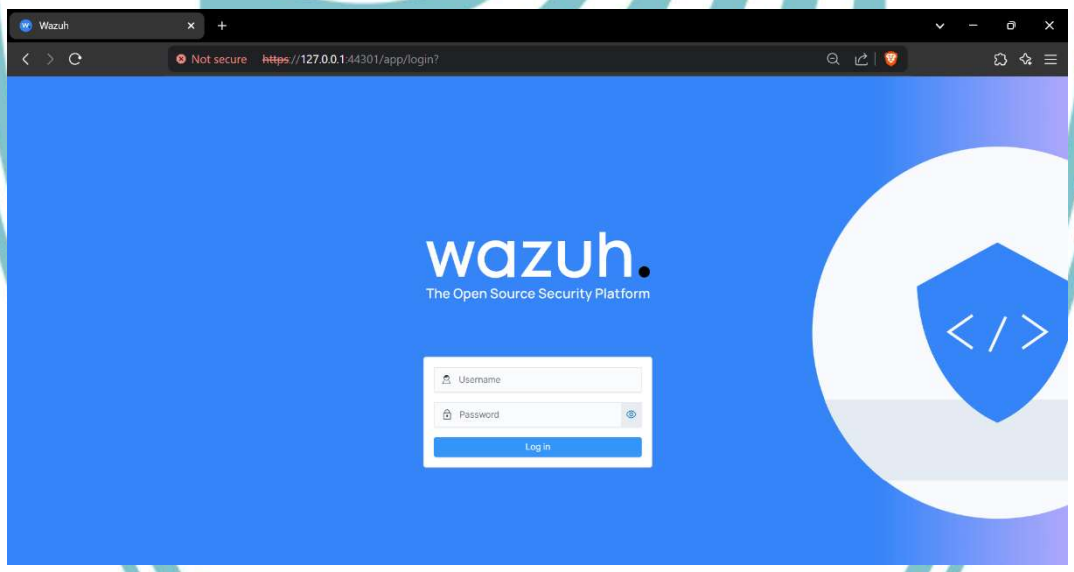
Gambar 4.11 Instalasi Wazuh Server

Perintah bash digunakan untuk menjalankan *file shell script*. *File* tersebut berisi instalasi *script* Wazuh. *Option* menggunakan parameter –wazuh-dashboard digunakan untuk instalasi Wazuh *dashboard*, serta nama *node* disesuaikan pada *file* config.yml.



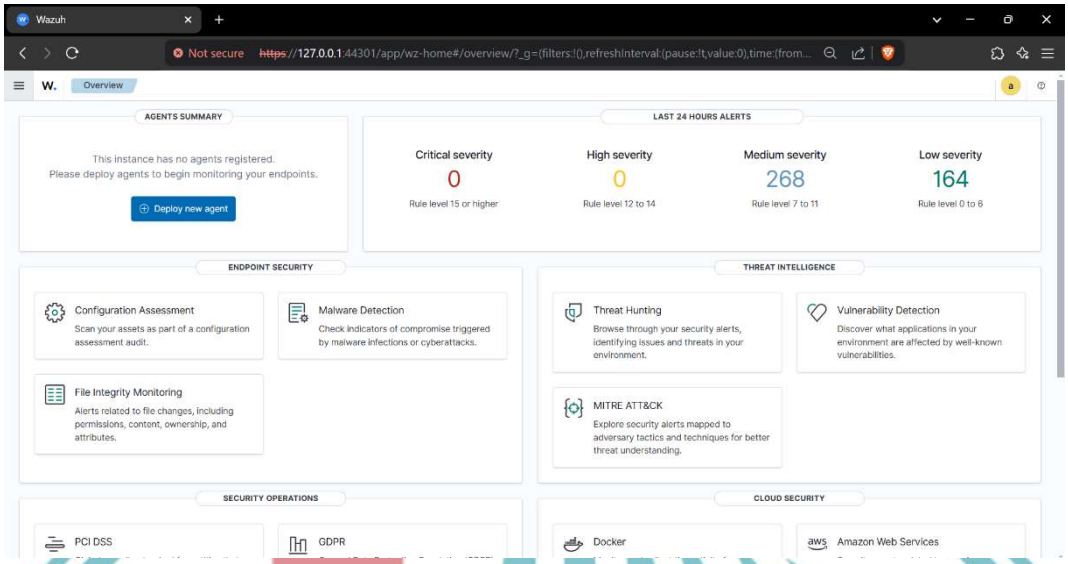
Gambar 4.12 Instalasi Wazuh Dashboard

Akses Wazuh *dashboard* dengan cara membuka IP *address localhost*, yaitu 127.0.0.1 dan *port forwarding* HTTPS, yaitu 44301 melalui *browser*. Berikut ini adalah tampilan halaman *login* dari Wazuh *dashboard* yang dapat dilihat pada Gambar 4.13.



Gambar 4.13 Tampilan Halaman Login Wazuh Dashboard

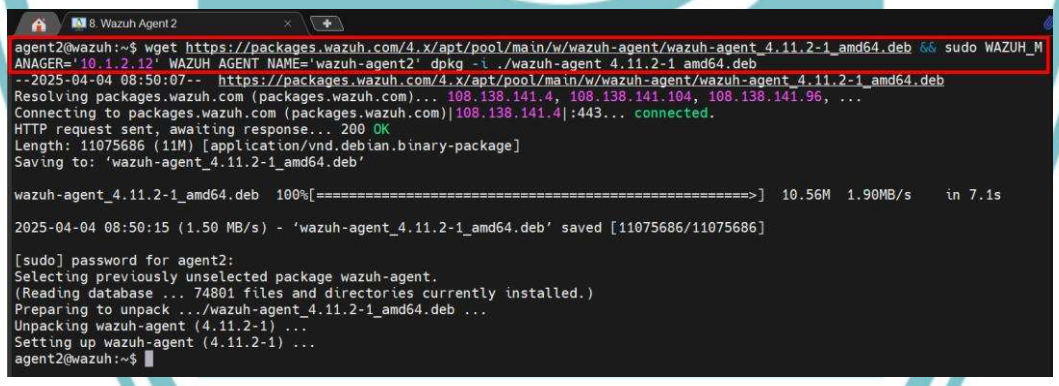
Berikut ini adalah tampilan Wazuh *dashboard* yang telah berhasil *login* yang dapat dilihat pada Gambar 4.14. Halaman *overview* menampilkan *agent's summary*, *last* 24 *hours alerts*, *endpoint security*, *threat intelligence*, dan sebagainya sebagai ringkasan informasi yang penting.



Gambar 4.14 Tampilan Halaman Overview Wazuh Dashboard

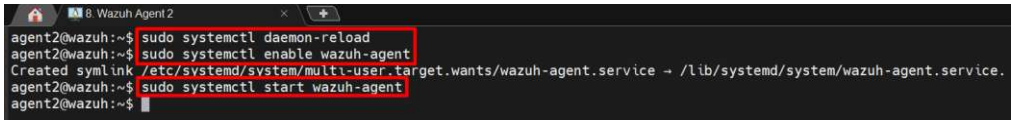
4.3.2 Instalasi Wazuh *Agent*

Wazuh *agent* diinstal dengan cara menggunakan perintah wget untuk mengunduh paket .deb. Instalasi Wazuh *agent* dengan *environment variables* seperti WAZUH_MANAGER dan WAZUH_AGENT_NAME.



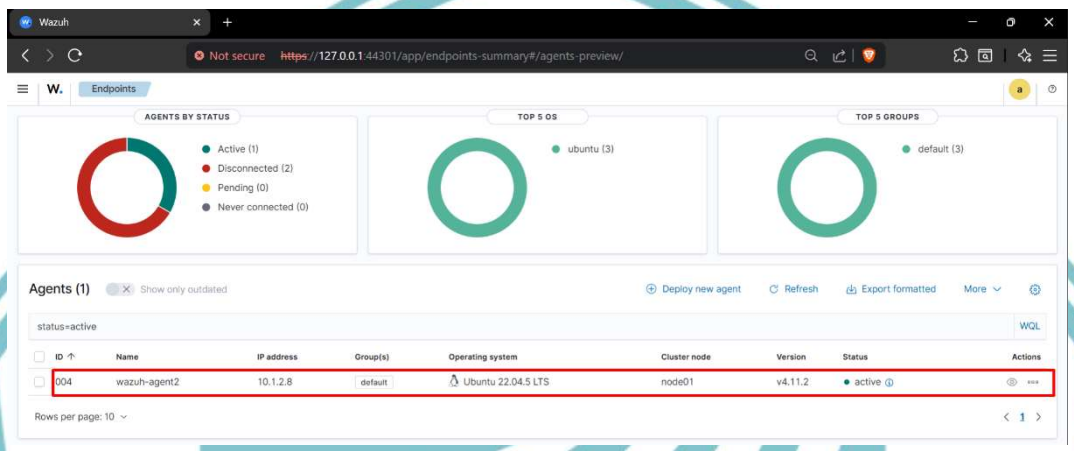
Gambar 4.15 Instalasi Wazuh Agent

Perintah systemctl daemon-reload untuk memuat ulang konfigurasi unit *file* systemd. Perintah systemctl enable wazuh-agent untuk mengatur segala *service* tersebut otomatis berjalan saat proses *booting*. Perintah systemctl start wazuh-agent untuk menjalankan *service* tersebut sekarang juga.



Gambar 4.16 Pengaktifan Wazuh Agent

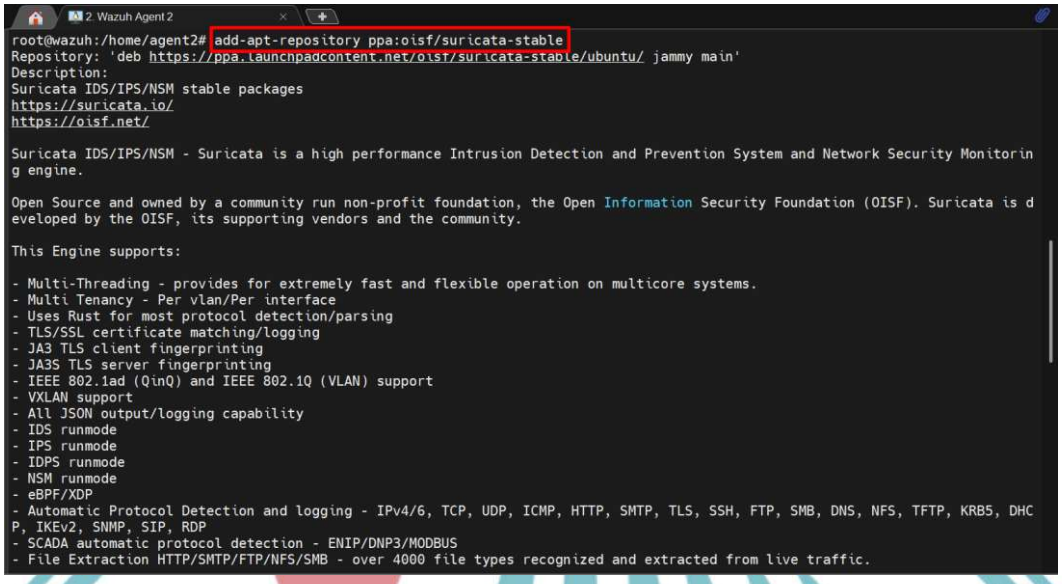
Gambar 4.17 menampilkan hasil Wazuh *agent* yang telah berhasil diinstal, yaitu dengan IP *address* 10.1.2.8.



Gambar 4.17 Hasil Wazuh Agent yang Telah Berhasil Diinstalasi

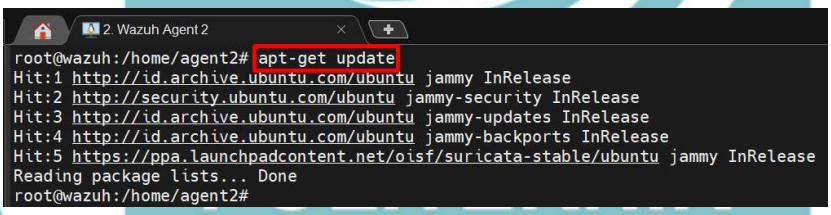
4.3.3 Instalasi Suricata pada Wazuh *Agent*

Instalasi Suricata pada Wazuh *agent* dengan cara menambahkan repositori Suricata terlebih dahulu. Perintah tersebut digunakan untuk menambahkan PPA (*Personal Package Archive*) ke sistem berbasis APT. Repositori Suricata yang diambil merupakan versi stabil dan resmi dari OISF (*Open Information Security Foundation*).



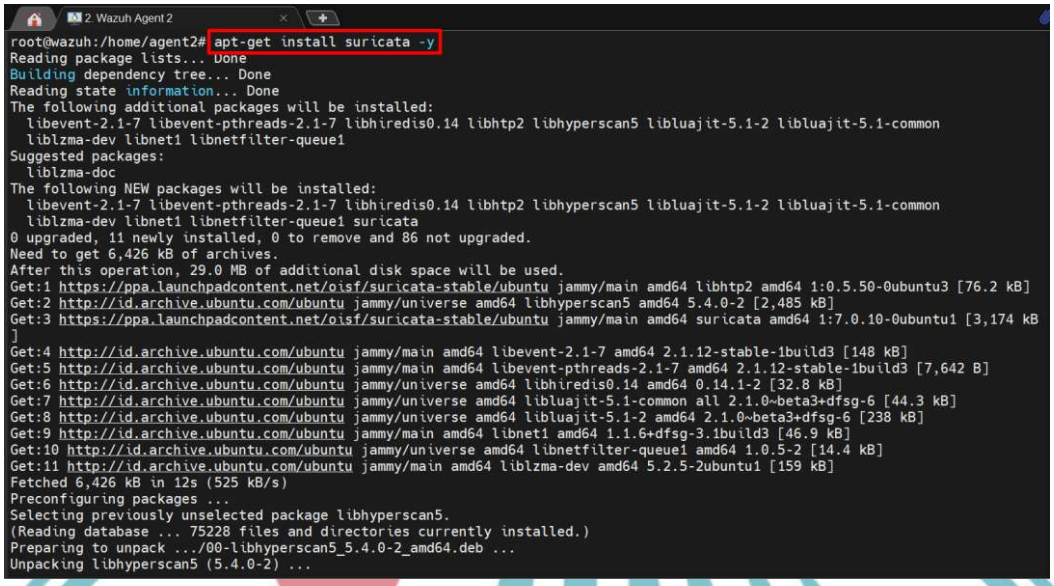
Gambar 4.18 Penambahan Repositori Suricata

Perintah apt-get merupakan *tool* manajemen paket untuk mengelola *software*. Perintah *update* merupakan opsi untuk mengunduh ulang informasi terbaru tentang semua paket dari repositori yang terdaftar.



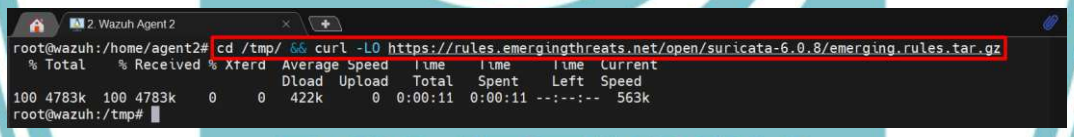
Gambar 4.19 Pembaruan Daftar Paket

Perintah apt-get merupakan *tool* manajemen paket untuk mengelola *software*. Perintah *install* untuk menginstal satu atau lebih paket. Paket yang diinstal adalah Suricata dengan opsi -y agar terinstal secara otomatis.



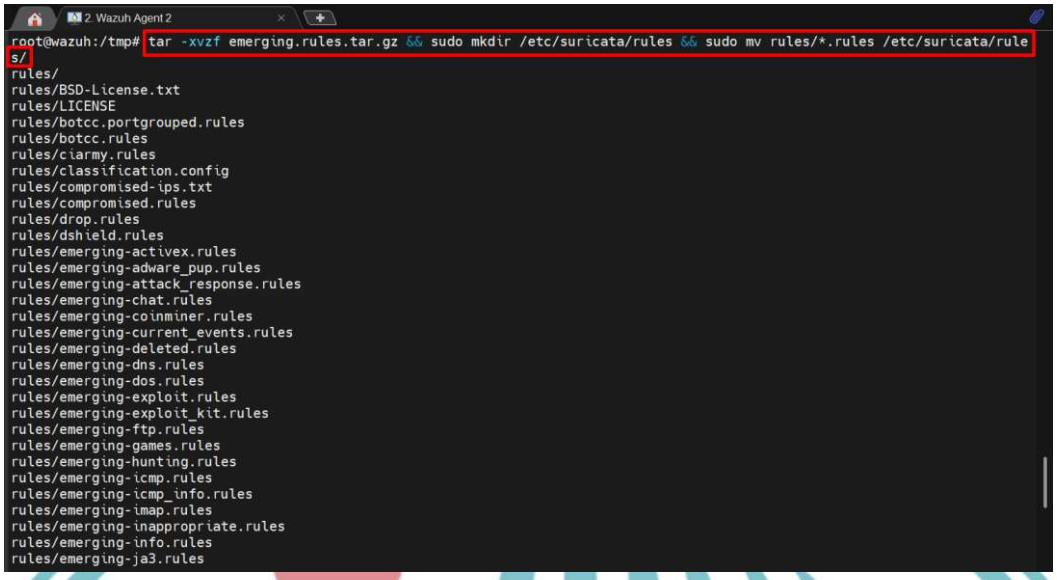
Gambar 4.20 Instalasi Suricata

Perintah cd untuk berpindah direktori dan perintah curl untuk mengambil data atau *file* dari URL. File *emerging rules* tersebut adalah *ruleset open-source* untuk Suricata yang disediakan oleh *Emerging Threats*.



Gambar 4.21 Pengunduhan Rules Suricata

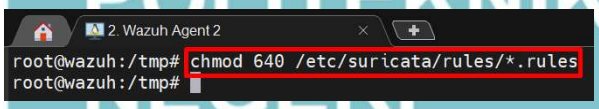
Perintah tar untuk mengekstrak *rules* Suricata yang telah diunduh dari *Emerging Threats*. Perintah mkdir untuk membuat direktori baru dan perintah mv untuk memindahkan *file*. Semua *file rules* dari *Emerging Threats* dipindahkan ke direktori *rules* Suricata.



Gambar 4.22 Pengekstrakan Rules Suricata

Perintah chmod untuk mengubah *permission* pada semua *file* .rules di dalam direktori tersebut. Nilai *permission* dalam bentuk bilangan oktal 640 tersebut maksudnya adalah sebagai berikut:

- *Owner* : 6 (*read and write*)
- *Group* : 4 (*read*)
- *Others* : 0 (*empty*)

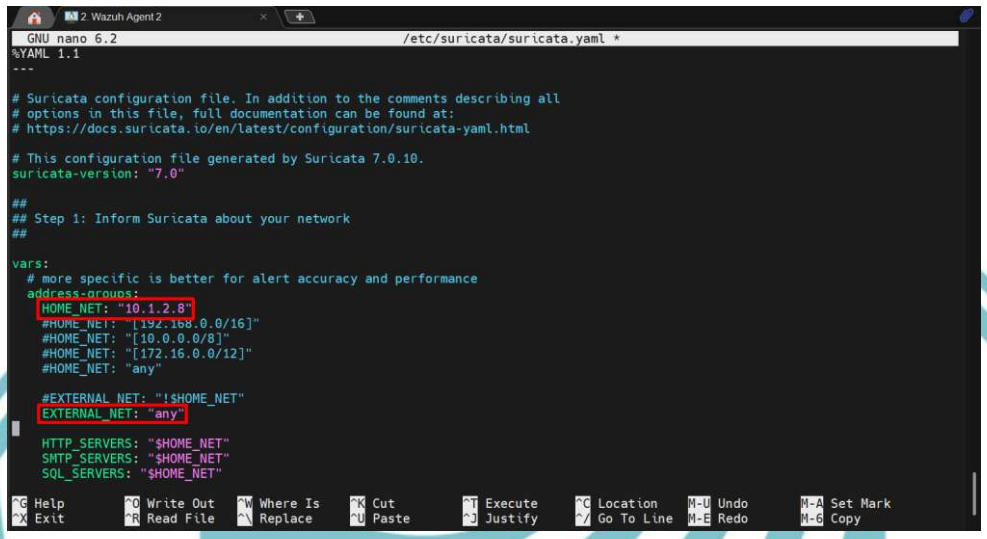


Gambar 4.23 Pengubahan Permission File Rules Suricata

Perintah nano digunakan untuk mengedit *file rules* Suricata. *File* suricata.yaml adalah *file* konfigurasi utama yang digunakan Suricata untuk melakukan berbagai konfigurasi.

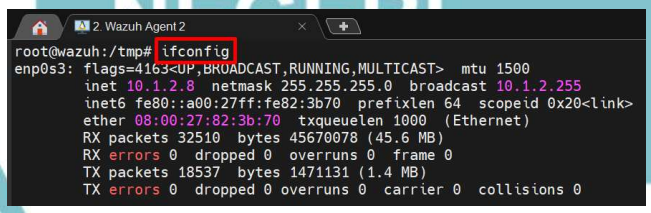
Gambar 4.24 Pengeditan File Rules Suricata

Mengedit beberapa bagian pada *file rules* Suricata. Bagian-bagian yang diedit adalah HOME_NET, EXTERNAL_NET, *default-rule-path*, *rule-files*, *stats enabled*, dan *interface*. Setelah diedit, konfigurasi tersebut dapat disimpan.



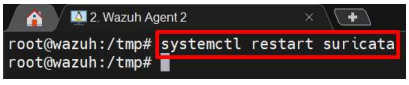
Gambar 4.25 Isi File Rules Suricata yang Telah Diedit

Perintah ifconfig digunakan untuk melakukan pengecekan konfigurasi jaringan yang telah diubah pada *file rules* Suricata sebelumnya. Ifconfig dapat menampilkan status dan konfigurasi *interface* jaringan, serta mengaktifkan dan menonaktifkan *interface* jaringan.



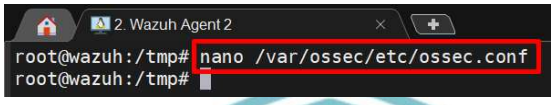
Gambar 4.26 Pengecekan Konfigurasi Jaringan

Perintah systemctl untuk mengelola *service* atau layanan di sistem Linux berbasis systemd. Perintah *restart* untuk menghentikan lalu memulai kembali *service* tersebut. *Service* yang dimaksud adalah Suricata karena sebelumnya telah menambahkan konfigurasi.



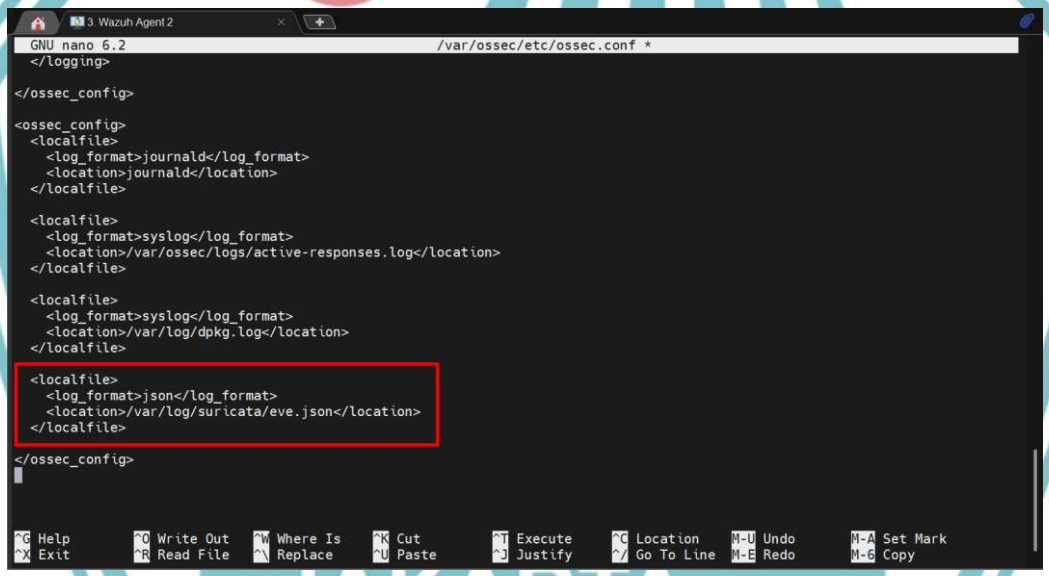
Gambar 4.27 Pembaruan Konfigurasi Suricata yang Telah Ditambahkan

Perintah nano digunakan untuk mengedit *file* konfigurasi OSSEC. *File* tersebut merupakan *file* konfigurasi utama untuk Wazuh dan *file* ini menentukan semua pengaturan Wazuh.



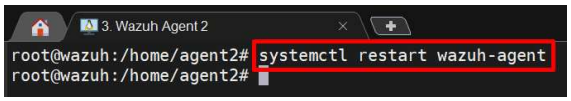
Gambar 4.28 Pengeditan File Konfigurasi OSSEC

Menambahkan *log* JSON pada *file* konfigurasi OSSEC tersebut. Konfigurasi tersebut digunakan untuk mengumpulkan *log* dari *file* eve.json milik Suricata. Tujuannya adalah agar Wazuh dapat membaca dan menganalisis *output log* dari Suricata.



Gambar 4.29 Isi File Konfigurasi OSSEC yang Telah Diedit

Perintah systemctl untuk mengelola *service* atau layanan di sistem Linux berbasis systemd. Perintah *restart* untuk menghentikan lalu memulai kembali *service* tersebut. *Service* yang dimaksud adalah Wazuh *agent* karena sebelumnya telah menambahkan konfigurasi.



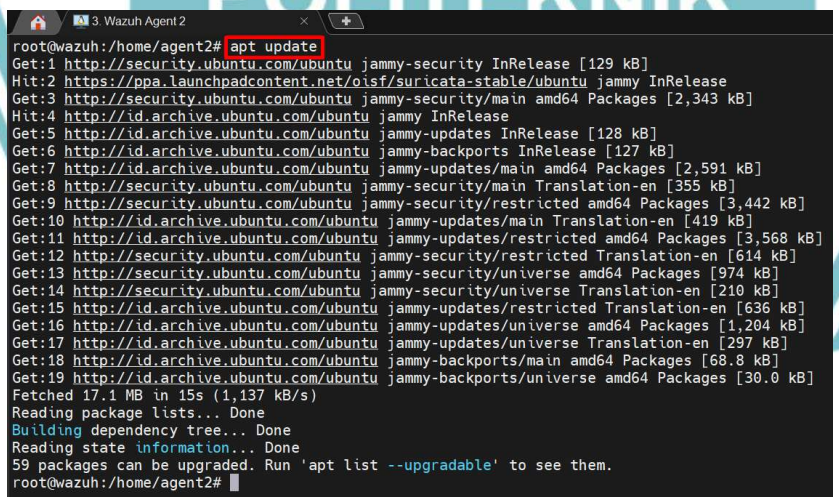
Gambar 4.30 Pembaruan Konfigurasi OSSEC yang Telah Ditambahkan

Berikut ini adalah hasil dari Suricata yang telah berhasil diinstalasi pada Wazuh *agent* yang dapat dilihat pada Gambar 4.31. *Rules* yang terpicu adalah Suricata: Alert - GPL ICMP_INFO PING *NIX.

Gambar 4.31 Hasil Suricata yang Telah Berhasil Diinstal pada Wazuh Agent

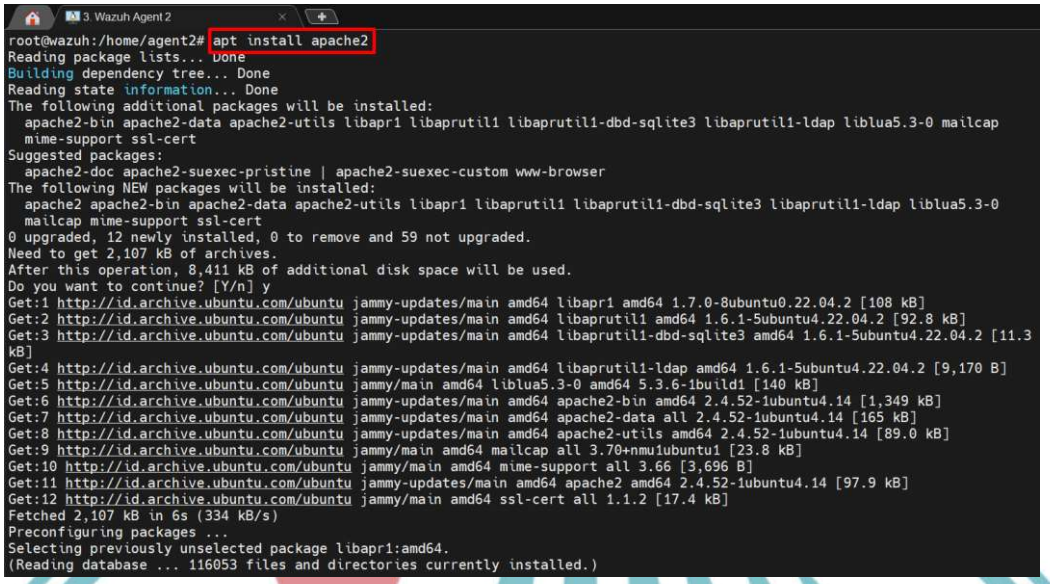
4.3.4 Intalasi *Web Server* pada Wazuh *Agent*

Perintah apt-get merupakan *tool* manajemen paket untuk mengelola *software*. Perintah *update* merupakan opsi untuk mengunduh ulang informasi terbaru tentang semua paket dari repositori yang terdaftar.



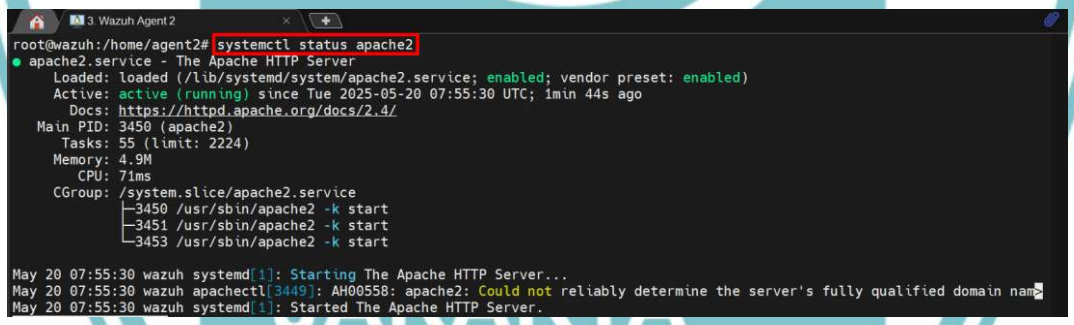
Gambar 4.32 Pembaruan Daftar Paket

Perintah apt-get merupakan *tool* manajemen paket untuk mengelola *software*. Perintah *install* untuk menginstal satu atau lebih paket. Paket yang diinstal adalah *web server* Apache.



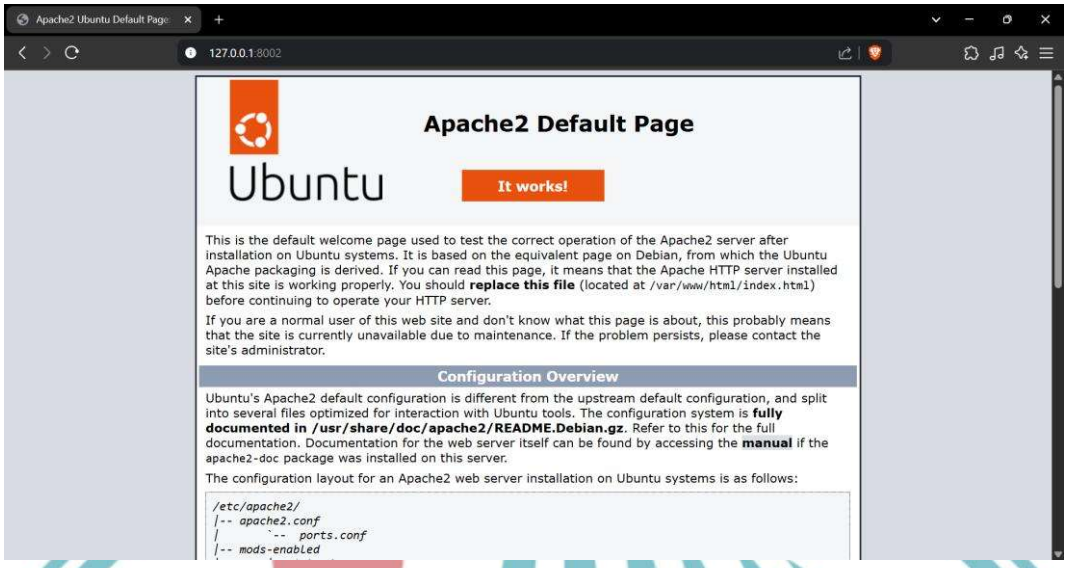
Gambar 4.33 Instalasi Web Server Apache

Perintah systemctl untuk mengelola *service* atau layanan di sistem Linux berbasis systemd. Perintah *status* untuk melakukan pengecekan keberhasilan instalasi *service* tersebut. *Service* yang dimaksud adalah Apache2 karena sebelumnya telah diinstalasi.



Gambar 4.34 Pengecekan Keberhasilan Instalasi Web Server Apache

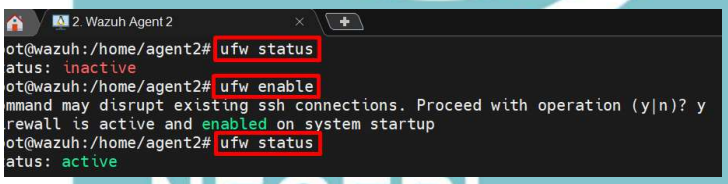
Tampilan *web server* Apache Wazuh *agent* dapat diakses dengan cara membuka IP *address localhost*, yaitu 127.0.0.1 dan *port forwarding*, yaitu 8001 melalui *browser*. Berikut ini adalah tampilan *default* dari *web server* Apache pada Wazuh *agent* yang dapat dilihat pada Gambar 4.35.



Gambar 4.35 Tampilan Web Server Apache pada Wazuh Agent

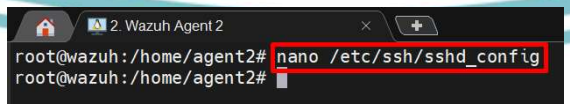
4.3.5 Konfigurasi *Firewall* pada Wazuh *Agent*

Melakukan pengecekan status *firewall*, mengaktifkan *firewall*, dan melakukan pengecekan kembali status *firewall* setelah diaktifkan pada Wazuh *agent*.



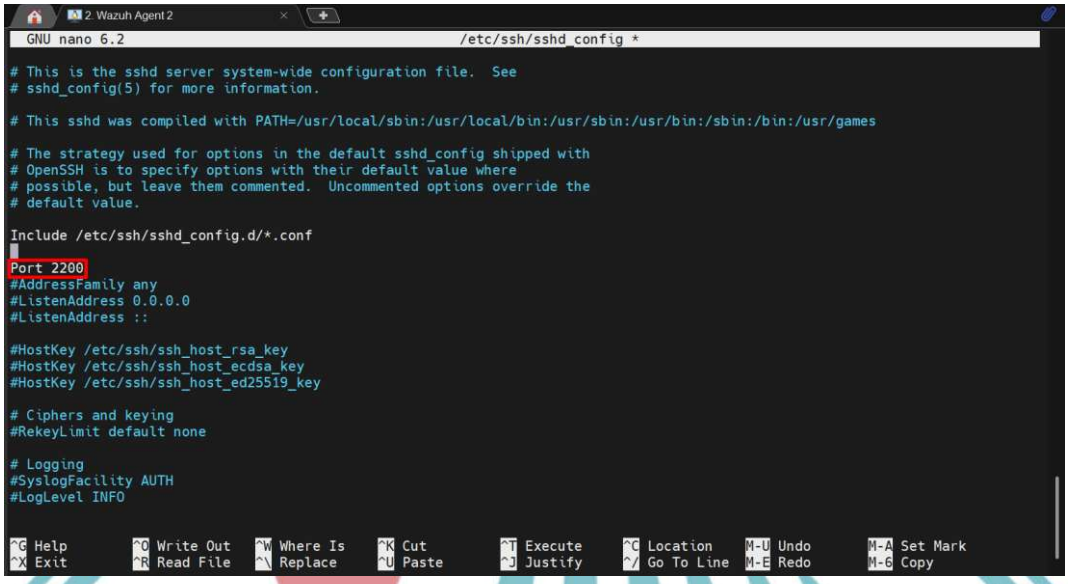
Gambar 4.36 Pengecekan dan Pengaktifan Firewall pada Wazuh Agent

Perintah nano digunakan untuk mengubah *file* konfigurasi SSH. *File* tersebut merupakan *file* konfigurasi utama SSH *server* (sshd) karena *file* ini mengatur bagaimana *server* SSH bekerja.



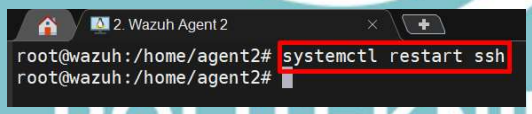
Gambar 4.37 Pengeditan File Konfigurasi SSH

Mengubah isi *file* konfigurasi SSH pada bagian *port* 22 menjadi *port* 2200. Mengubah *port default* SSH tersebut menjadi *port* lain pada Wazuh *agent*, tujuannya adalah agar tidak mudah diketahui oleh *attacker*.



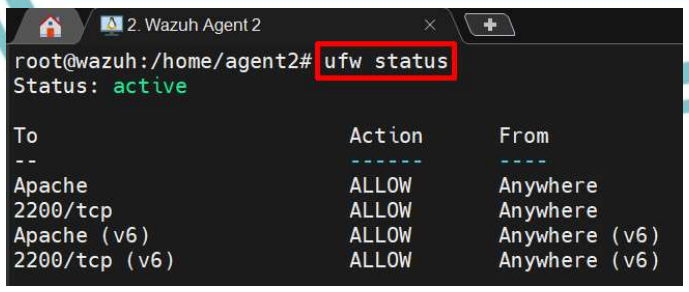
Gambar 4.38 Isi File Konfigurasi SSH yang Telah Ditambahkan

Perintah systemctl untuk mengelola *service* atau layanan di sistem Linux berbasis systemd. Perintah *restart* untuk menghentikan lalu memulai kembali *service* tersebut. *Service* yang dimaksud adalah SSH karena sebelumnya telah menambahkan konfigurasi.



Gambar 4.39 Pembaruan Konfigurasi SSH yang Telah Ditambahkan

Perintah ufw status untuk melakukan pengecekan status *firewall*. Gambar 4.40 menunjukkan hasil *port* SSH yang telah diubah pada *firewall* dan statusnya *allow*.



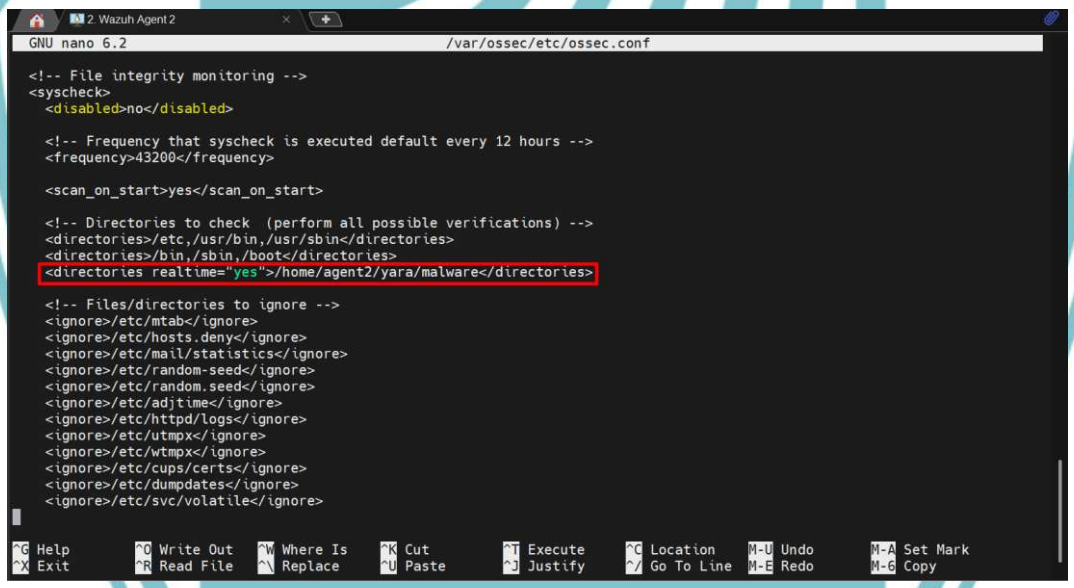
Gambar 4.40 Hasil Konfigurasi Firewall yang Telah Ditambahkan

4.3.6 Penerapan FIM pada Wazuh *Agent*

Perintah nano digunakan untuk mengedit *file* konfigurasi OSSEC. *File* tersebut merupakan *file* konfigurasi utama untuk Wazuh dan *file* ini menentukan semua pengaturan Wazuh.

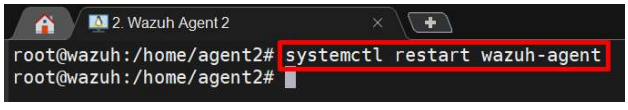
Gambar 4.41 Pengeditan File Konfigurasi OSSEC

Menerapkan FIM pada *file* konfigurasi OSSEC. Caranya adalah menambahkan direktori yang ingin dimonitoring oleh FIM, yaitu /home/agent2/yara/malware. FIM akan memonitoring direktori tersebut secara *real-time*.



Gambar 4.42 Penerapan FIM pada File Konfigurasi OSSEC

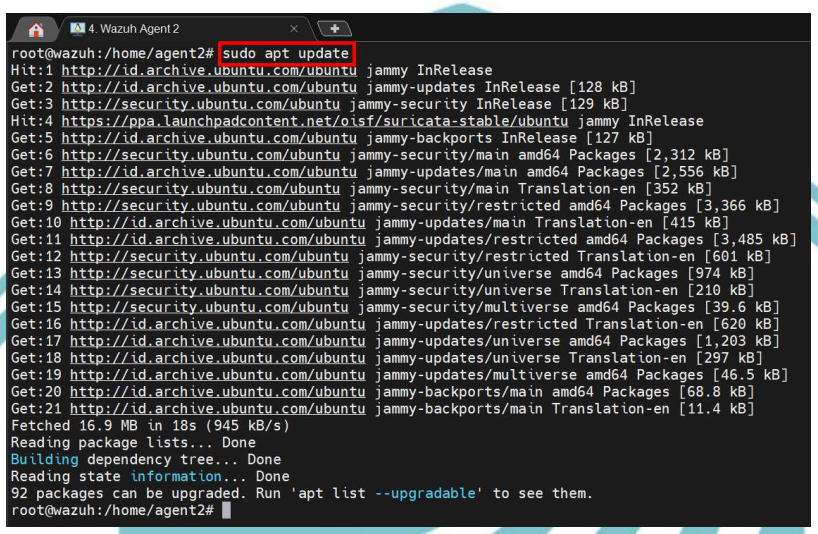
Perintah systemctl untuk mengelola *service* atau layanan di sistem Linux berbasis systemd. Perintah *restart* untuk menghentikan lalu memulai kembali *service* tersebut. *Service* yang dimaksud adalah Wazuh *agent* karena sebelumnya telah menambahkan konfigurasi.



Gambar 4.43 Pembaruan Konfigurasi yang Telah Ditambahkan

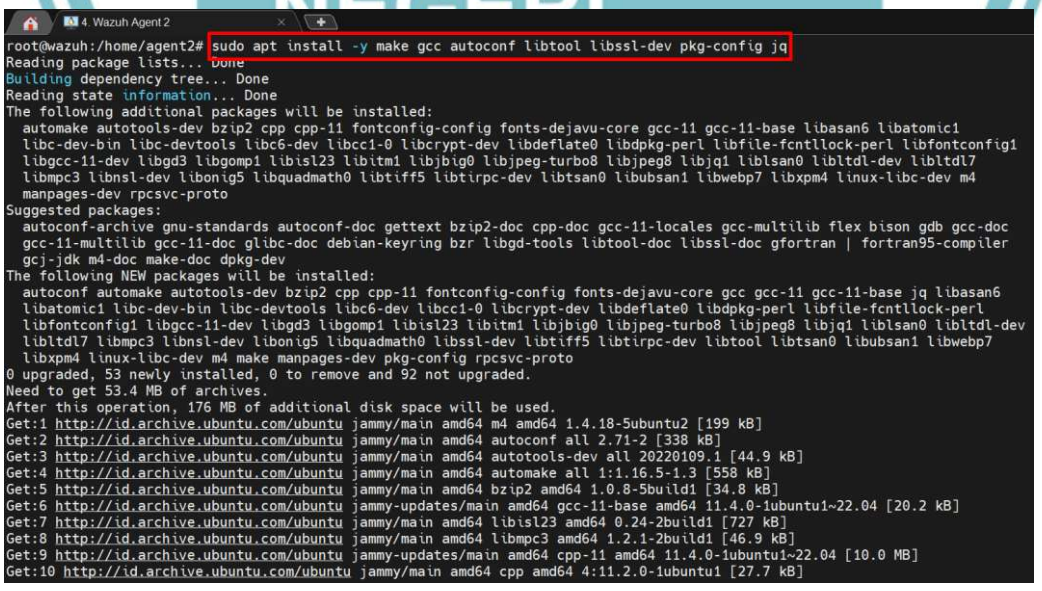
4.3.7 Penerapan YARA pada Wazuh *Agent*

Perintah apt-get merupakan *tool* manajemen paket untuk mengelola *software*. Perintah *update* merupakan opsi untuk mengunduh ulang informasi terbaru tentang semua paket dari repositori yang terdaftar.



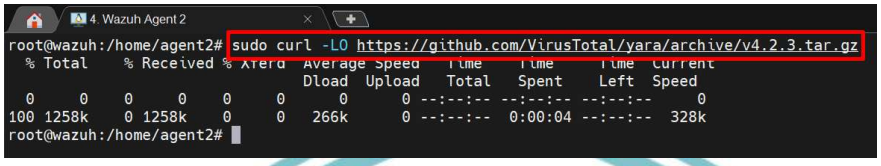
Gambar 4.44 Pembaruan Daftar Paket

Melakukan instalasi dependensi yang dibutuhkan. Perintah tersebut untuk menginstal sejumlah paket penting di Linux yang dibutuhkan untuk membangun (*compile*) dan menjalankan aplikasi dari *source code*, serta untuk memanipulasi data JSON.



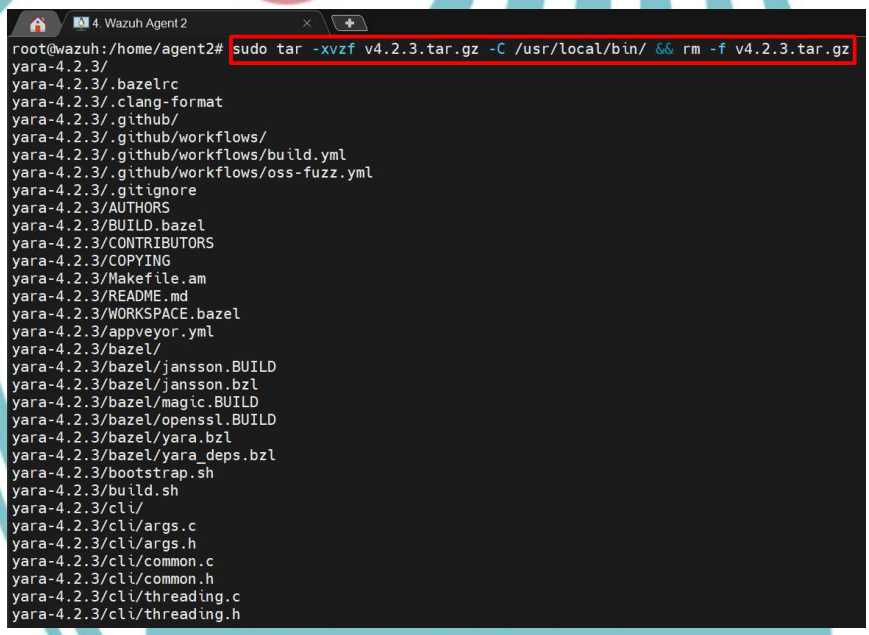
Gambar 4.45 Instalasi Dependensi

Mengunduh *source code* yang dibutuhkan, yaitu YARA. *Source code* yang diunduh bersumber dari repositori Github YARA. *Source code* tersimpan di direktori tempat menjalankan perintah tersebut.



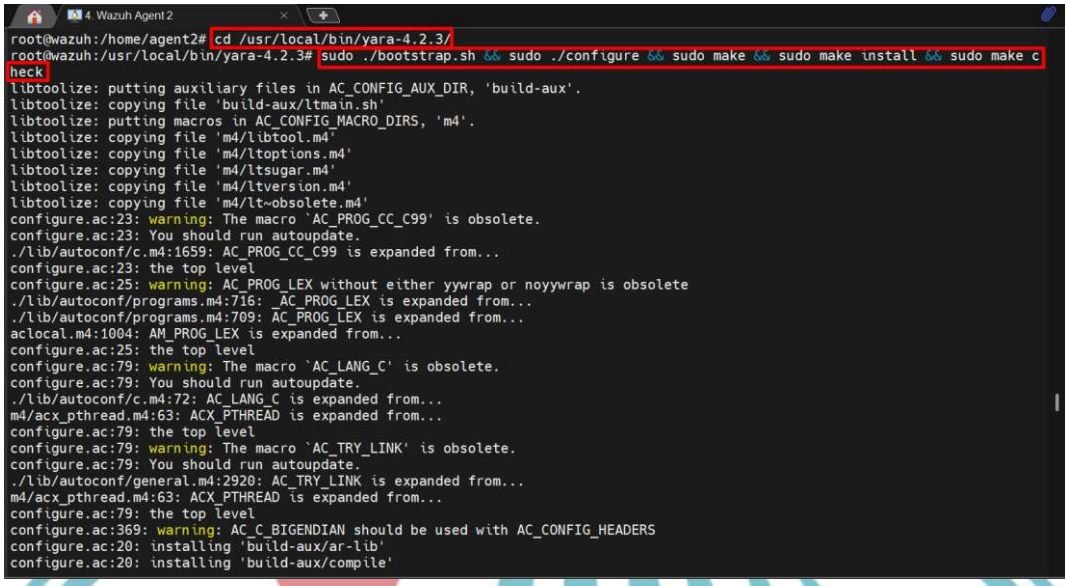
Gambar 4.46 Pengunduhan Source Code YARA

Mengekstrak *source code* YARA yang telah diunduh sebelumnya. Perintah tersebut bertujuan untuk mengekstrak arsip v.4.2.3.tar.gz ke dalam direktori /usr/local/bin. Lalu, menghapus *file* arsipnya setelah selesai.



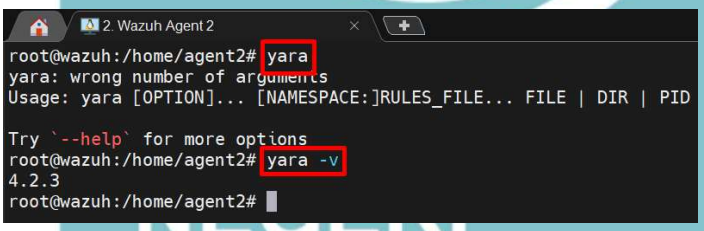
Gambar 4.47 Pengekstrakan Source Code YARA

Gambar 4.48 menunjukkan perintah untuk membangun (*build*) dan menginstal YARA versi 4.2.3 dari *source code*. Perintah ./boostrap.sh untuk menyiapkan lingkungan agar *source code* siap dikompilasi Perintah ./configure untuk menyiapkan proses kompilasi berdasarkan konfigurasi sistem. Perintah *make* untuk melakukan kompilasi *source code* YARA menjadi *binary*. Perintah *make install* untuk menginstal hasil *build* ke sistem. Perintah *make check* untuk memastikan bahwa hasil *build* bekerja dengan benar.



Gambar 4.48 Proses Build dari Source Code YARA

Melakukan pengujian keberhasilan instalasi YARA. Perintah yara adalah perintah dasar untuk menampilkan panduan penggunaan yang menjelaskan opsi dan format penggunaannya. Perintah yara -v untuk menampilkan versi YARA yang terinstal pada sistem.

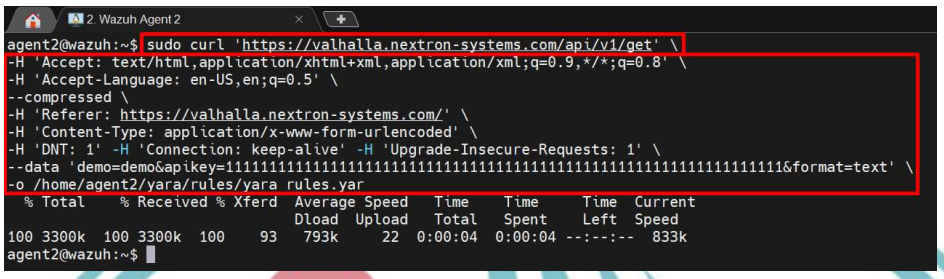


Gambar 4.49 Pengecekan Keberhasilan Instalasi YARA

Perintah mkdir digunakan untuk membuat direktori baru dan parameter -p untuk membuat direktori dalam jalur yang belum ada. Direktori /home/agent2 merupakan direktori *home* milik *user* agent2. Direktori /yara untuk menyimpan *file* yang berkaitan dengan YARA dan direktori /rules untuk menyimpan *file rules* YARA.

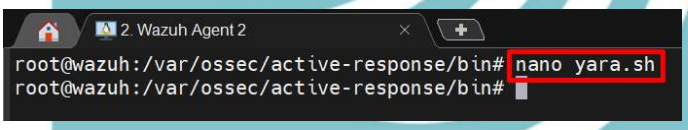
Gambar 4.50 Pembuatan Direktori untuk YARA Rules

Gambar 4.51 menunjukkan perintah untuk mengunduh kumpulan YARA *rules* dari layanan Valhalla API milik Nextron Systems. *File* tersebut disimpan sebagai .yar ke dalam direktori yang telah dibuat sebelumnya. *Rules* YARA tersebut akan digunakan untuk mendeteksi *malware*.



Gambar 4.51 Pengunduhan YARA Rules

Perintah nano digunakan untuk membuat *file script* yara.sh. *File* tersebut dibuat pada direktori /var/ossec/active-response/bin. *Script* tersebut digunakan YARA untuk melakukan *scanning*.



Gambar 4.52 Pembuatan Script untuk Scanning YARA

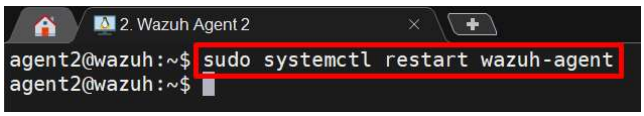
Perintah chown digunakan untuk mengatur kepemilikan *file script* yara.sh dengan ketentuan *user* root dan grup wazuh. Perintah chmod digunakan untuk mengatur izin akses *file script* yara.sh. Nilai *permission* dalam bentuk bilangan oktal 750 tersebut maksudnya adalah sebagai berikut:

- *Owner* : 7 (*read, write, execute*)
- *Group* : 5 (*read, execute*)
- *Others* : 0 (*empty*)

Gambar 4.53 Pengubahan Owner dan Permission pada File Script YARA

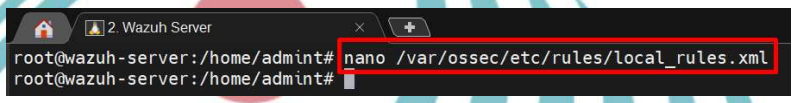
Perintah systemctl untuk mengelola *service* atau layanan di sistem Linux berbasis systemd. Perintah *restart* untuk menghentikan lalu memulai

kembali *service* tersebut. *Service* yang dimaksud adalah Wazuh *agent* karena sebelumnya telah menambahkan konfigurasi.



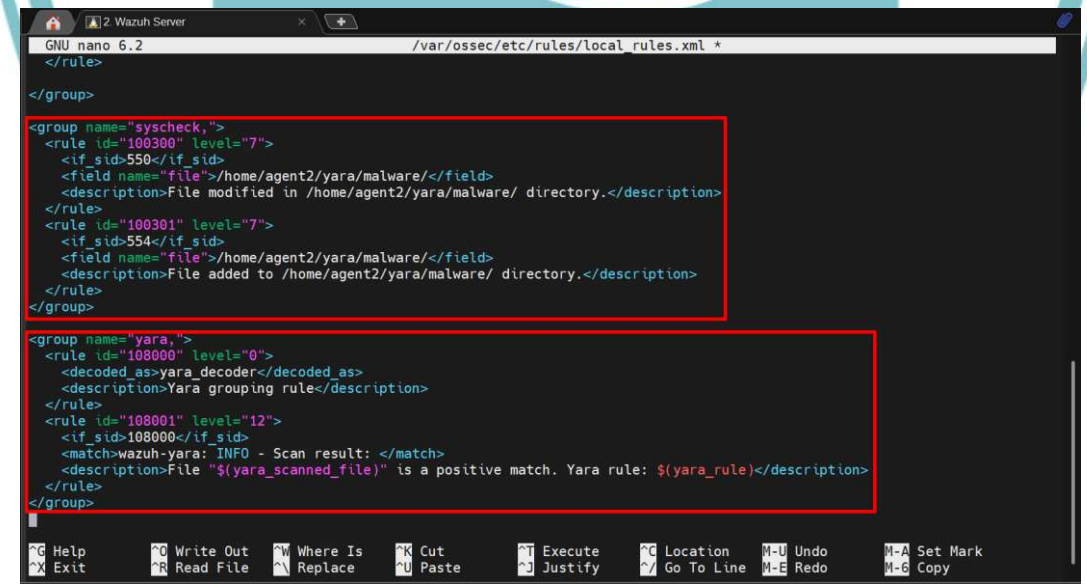
Gambar 4.54 Pembaruan Konfigurasi OSSEC yang Telah Ditambahkan

Perintah nano digunakan untuk mengedit *file* konfigurasi local_rules.xml. *File* tersebut digunakan untuk mendefinisikan aturan deteksi tambahan atau *custom* yang tidak tersedia dalam *rules default*.



Gambar 4.55 Pengeditan File Konfigurasi Local Rules

Menambahkan konfigurasi pada *file* local_rules.xml. Konfigurasi ini untuk mendeteksi direktori yang dimonitoring sehingga YARA dapat mendeteksi *malware* dari suatu *file*. Berikut isi konfigurasi yang ditambahkan pada *file* local_rules.xml yang dapat dilihat pada Gambar 4.56.



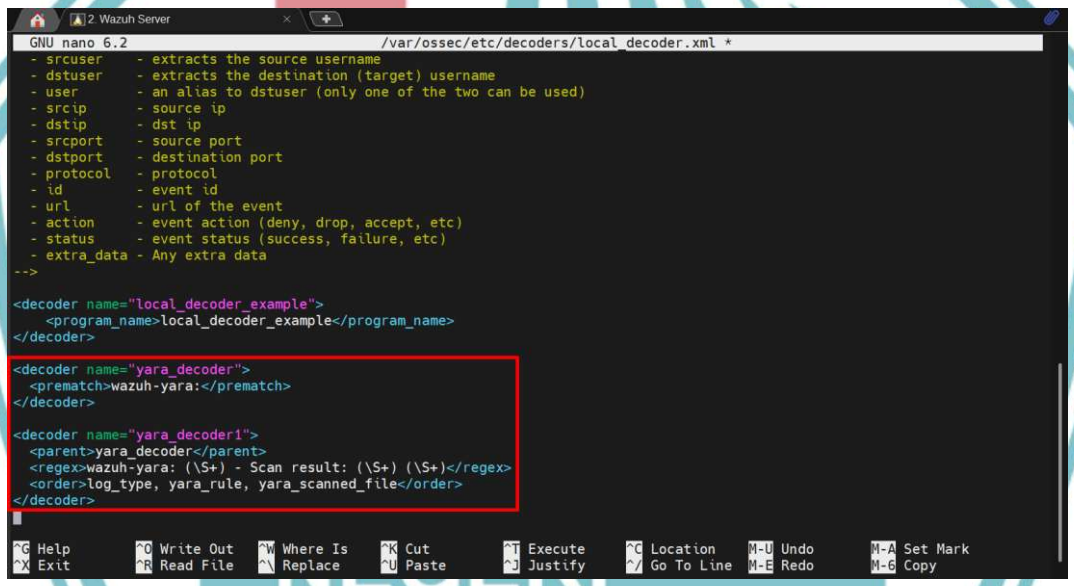
Gambar 4.56 Isi File Konfigurasi Local Rules yang Telah Ditambahkan

Perintah nano digunakan untuk mengedit *file* konfigurasi local_decoder.xml. *File* tersebut digunakan untuk membuat atau memodifikasi *custom decoder*,

yaitu komponen yang bertugas melakukan *parsing log* mentah menjadi *field* terstruktur sebelum dicocokkan dengan *rule* (misalnya di local_rules.xml).

Gambar 4.57 Pengeditan File Local Decoder

Menambahkan konfigurasi pada *file* local_decoder.xml. Konfigurasi ini untuk mengizinkan ekstrak informasi dari hasil *scanning* YARA. Berikut isi konfigurasi yang ditambahkan pada *file* local_decoder.xml yang dapat dilihat pada Gambar 4.58.

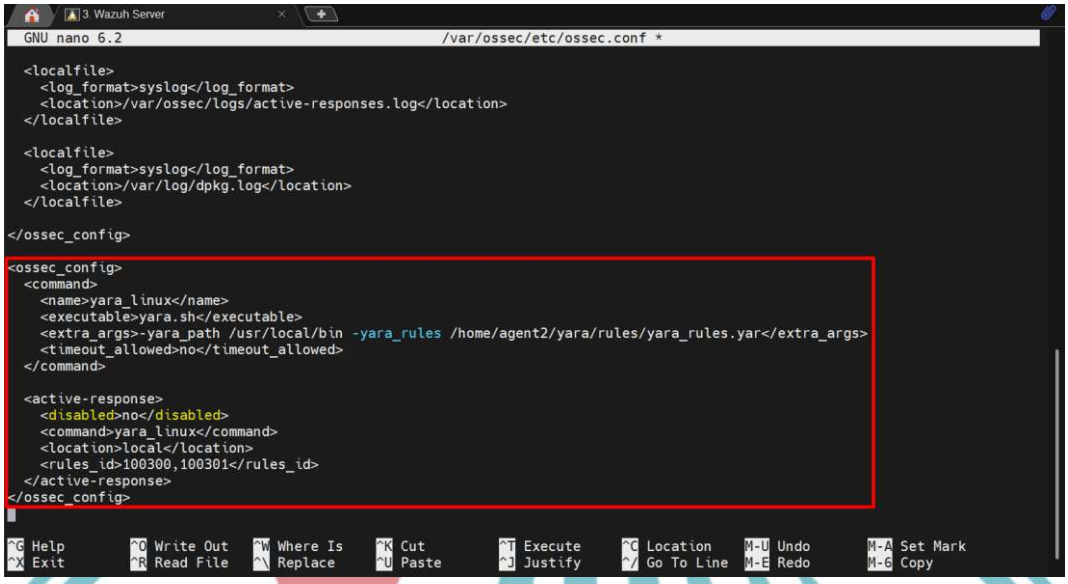


Gambar 4.58 Isi File Konfigurasi Local Decoder yang Telah Ditambahkan

Perintah nano digunakan untuk mengedit *file* konfigurasi OSSEC. *File* tersebut merupakan *file* konfigurasi utama untuk Wazuh dan *file* ini menentukan semua pengaturan Wazuh.

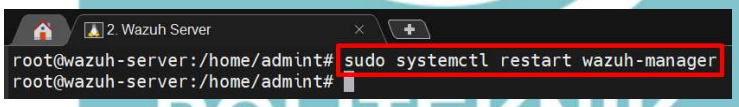
Gambar 4.59 Pengeditan File Konfigurasi OSSEC

Menambahkan konfigurasi pada *file* ossec.conf. Konfigurasi ini untuk memicu *rules* 100300 dan 100301 agar aktif. Berikut isi konfigurasi yang ditambahkan pada *file* ossec.conf yang dapat dilihat pada Gambar 4.60.



Gambar 4.60 Isi File Konfigurasi OSSEC yang Telah Ditambahkan

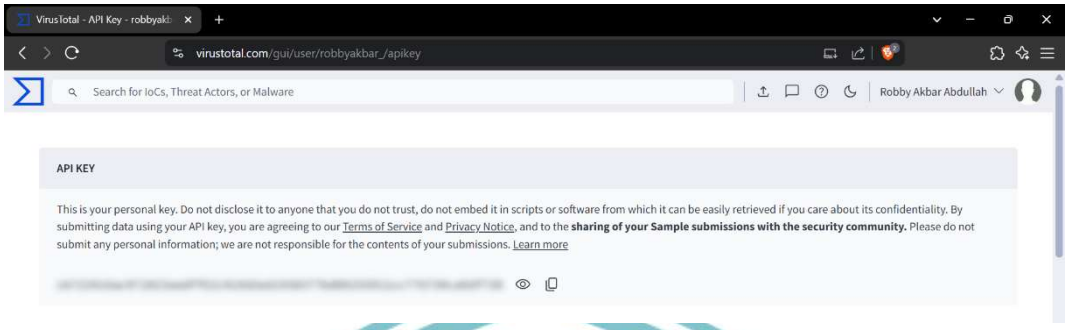
Perintah systemctl untuk mengelola *service* atau layanan di sistem Linux berbasis systemd. Perintah *restart* untuk menghentikan lalu memulai kembali *service* tersebut. *Service* yang dimaksud adalah Wazuh *manager* karena sebelumnya telah menambahkan konfigurasi.



Gambar 4.61 Pembaruan Konfigurasi yang Telah Ditambahkan

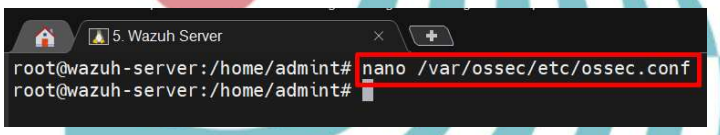
4.3.8 Penerapan VirusTotal pada Wazuh *Agent*

VirusTotal memiliki API *key public* yang dapat digunakan siapa saja. API *key* ini didapatkan dengan cara membuat akun terlebih dahulu. Berikut ini adalah contoh pengambilan API *key* dari VirusTotal yang dapat dilihat pada Gambar 4.62.



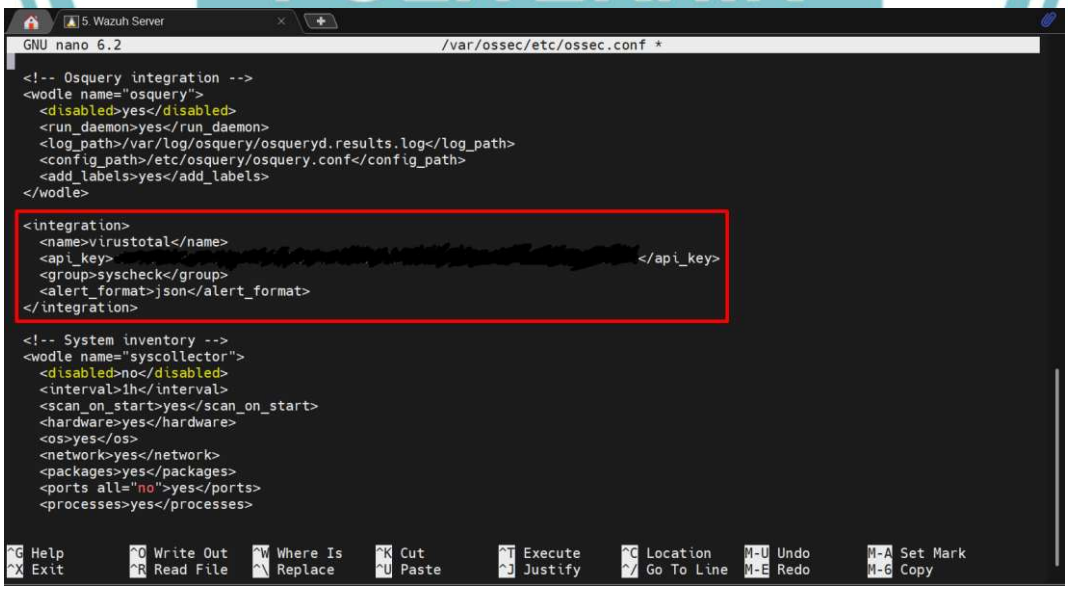
Gambar 4.62 Pengambilan API Key VirusTotal

Perintah nano digunakan untuk mengedit *file* konfigurasi OSSEC. *File* tersebut merupakan *file* konfigurasi utama untuk Wazuh dan *file* ini menentukan semua pengaturan Wazuh.



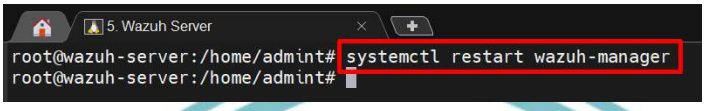
Gambar 4.63 Pengeditan File Konfigurasi OSSEC

Menerapkan VirusTotal pada *file* konfigurasi OSSEC. Caranya adalah menambahkan API *key* yang telah didapatkan dari VirusTotal. VirusTotal akan mencocokan *file* yang terdeteksi pada SIEM Wazuh dengan *database* yang ada di VirusTotal untuk mengetahui *file* tersebut *malicious* atau tidak.



Gambar 4.64 Penerapan VirusTotal pada File Konfigurasi OSSEC

Perintah systemctl untuk mengelola *service* atau layanan di sistem Linux berbasis systemd. Perintah *restart* untuk menghentikan lalu memulai kembali *service* tersebut. *Service* yang dimaksud adalah Wazuh *manager* karena sebelumnya telah menambahkan konfigurasi.



Gambar 4.65 Pembaruan Konfigurasi OSSEC yang Telah Ditambahkan

4.4 Pengujian

4.4.1 Deskripsi Pengujian

Pengujian dilakukan dengan cara melakukan serangan *port scanning* terlebih dahulu untuk mendapatkan *port* yang terbuka. Kemudian, melakukan serangan *Denial of Service* (DoS) untuk menguji ketersediaan sistem. Lalu, melakukan serangan SSH *brute-force* untuk mendapatkan akses ke Wazuh *agent*. Selanjutnya, melakukan *download malware* ke Wazuh *agent*. Setelah itu, dilakukan analisis kinerja SIEM Wazuh dengan FIM, YARA, dan VirusTotal dalam mendeteksi *malware* pada Ubuntu *server* dengan menggunakan parameter-parameter yang telah ditentukan.

4.4.2 Prosedur Pengujian

4.4.2.1 Pengujian Serangan *Port Scanning*

Nmap adalah *tool* untuk melakukan serangan *port scanning* yang bertujuan untuk mendapatkan *port* yang terbuka. Caranya adalah melakukan perintah nmap 10.1.2.8 ke Wazuh *agent*. *Port* yang terbuka tersebut dapat dimanfaatkan untuk melakukan serangan lebih lanjut.

Gambar 4.66 Serangan Port Scanning ke Wazuh Agent

4.4.2.2 Pengujian Serangan *Denial of Service* (DoS)

Hping3 adalah *tool* untuk melakukan serangan *Denial of Service* (DoS) yang bertujuan untuk membanjiri *traffic* pada *server*. Caranya adalah melakukan perintah hping3 -S --flood -V -p 80 10.1.2.8 ke Wazuh *agent*. Dampaknya adalah *server* mengalami kelambatan atau *down*.

```
┌──(root㉿kali-linux)-[/home/attacker]
└─# hping3 -S --flood -V -p 80 10.1.2.8
using eth0, addr: 10.1.2.13, MTU: 1500
HPING 10.1.2.8 (eth0 10.1.2.8): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

Gambar 4.67 Serangan DoS ke Wazuh Agent

4.4.2.3 Pengujian Serangan SSH *Brute-force*

Pengujian pertama serangan SSH *brute-force* dilakukan menggunakan *username* dan *password* yang sama dengan *username*. *Tool* yang dilakukan untuk melakukan *brute-force* adalah Hydra. Hydra ini melakukan *brute-force* tersebut ke *port* SSH dan IP *address* yang dituju sampai mendapatkan *username* dan *password* yang valid.

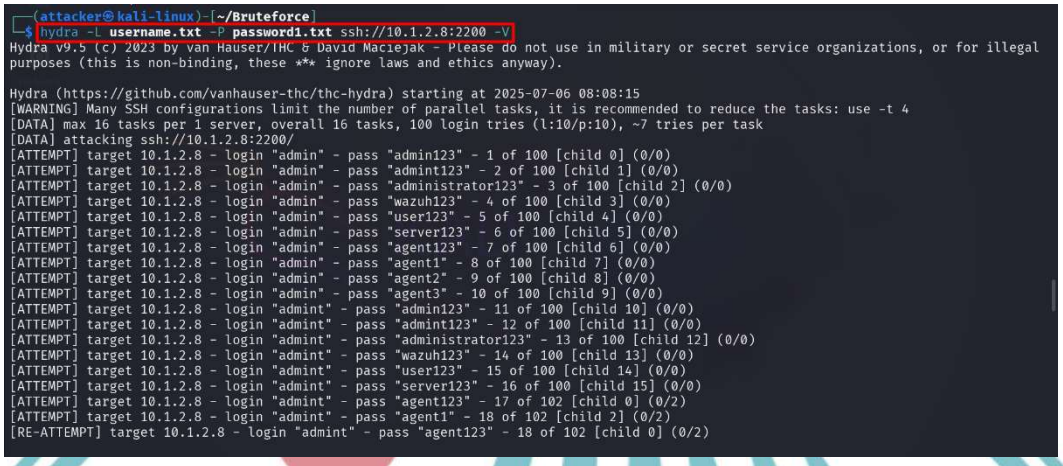
```
┌──(attacker㉿kali-linux)-[~/Bruteforce]
└─$ hydra -L username.txt -P username.txt ssh://10.1.2.8:2200 -V
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal
purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-07-06 07:58:54
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 100 login tries (l:10/p:10), ~7 tries per task
[DATA] attacking ssh://10.1.2.8:2200/
[ATTEMPT] target 10.1.2.8 - login "admin" - pass "admin" - 1 of 100 [child 0] (0/0)
[ATTEMPT] target 10.1.2.8 - login "admin" - pass "admint" - 2 of 100 [child 1] (0/0)
[ATTEMPT] target 10.1.2.8 - login "admin" - pass "administrator" - 3 of 100 [child 2] (0/0)
[ATTEMPT] target 10.1.2.8 - login "admin" - pass "wazuh" - 4 of 100 [child 3] (0/0)
[ATTEMPT] target 10.1.2.8 - login "admin" - pass "user" - 5 of 100 [child 4] (0/0)
[ATTEMPT] target 10.1.2.8 - login "admin" - pass "server" - 6 of 100 [child 5] (0/0)
[ATTEMPT] target 10.1.2.8 - login "admin" - pass "agent" - 7 of 100 [child 6] (0/0)
[ATTEMPT] target 10.1.2.8 - login "admin" - pass "agent1" - 8 of 100 [child 7] (0/0)
[ATTEMPT] target 10.1.2.8 - login "admin" - pass "agent2" - 9 of 100 [child 8] (0/0)
[ATTEMPT] target 10.1.2.8 - login "admin" - pass "agent3" - 10 of 100 [child 9] (0/0)
[ATTEMPT] target 10.1.2.8 - login "admint" - pass "admin" - 11 of 100 [child 10] (0/0)
[ATTEMPT] target 10.1.2.8 - login "admint" - pass "admint" - 12 of 100 [child 11] (0/0)
[ATTEMPT] target 10.1.2.8 - login "admint" - pass "administrator" - 13 of 100 [child 12] (0/0)
[ATTEMPT] target 10.1.2.8 - login "admint" - pass "wazuh" - 14 of 100 [child 13] (0/0)
[ATTEMPT] target 10.1.2.8 - login "admint" - pass "user" - 15 of 100 [child 14] (0/0)
[ATTEMPT] target 10.1.2.8 - login "admint" - pass "server" - 16 of 100 [child 15] (0/0)
[ATTEMPT] target 10.1.2.8 - login "admint" - pass "agent" - 17 of 101 [child 1] (0/1)
[ATTEMPT] target 10.1.2.8 - login "admint" - pass "agent1" - 18 of 101 [child 9] (0/1)
[ATTEMPT] target 10.1.2.8 - login "admint" - pass "agent2" - 19 of 101 [child 4] (0/1)
```

Gambar 4.68 Pengujian Pertama Serangan SSH Brute-force ke Wazuh Agent

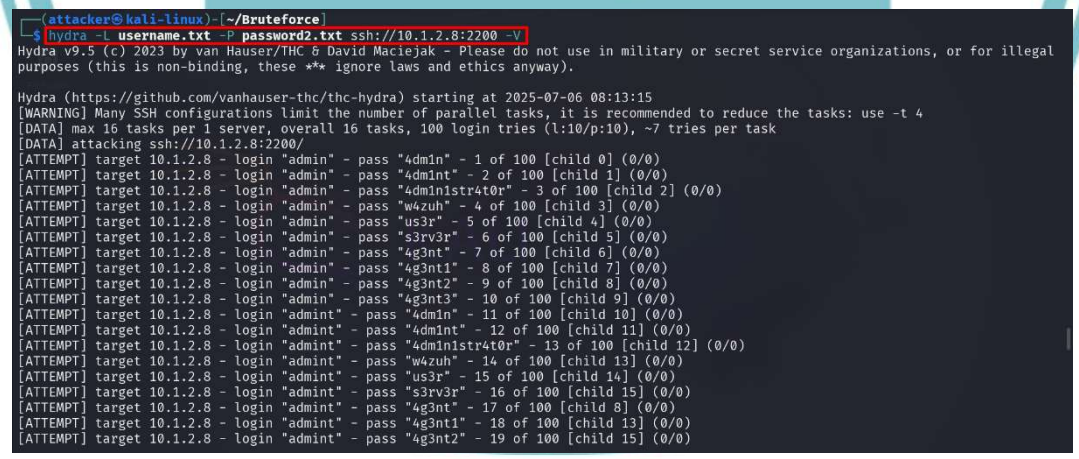
Pengujian kedua serangan SSH *brute-force* dilakukan menggunakan *username* dan *password* dengan tingkat kesulitan mudah. *Tool* yang dilakukan untuk melakukan *brute-force* adalah Hydra. Hydra ini

melakukan *brute-force* tersebut ke *port* SSH dan IP *address* yang dituju sampai mendapatkan *username* dan *password* yang valid.



Gambar 4.69 Pengujian Kedua Serangan SSH Brute-force ke Wazuh Agent

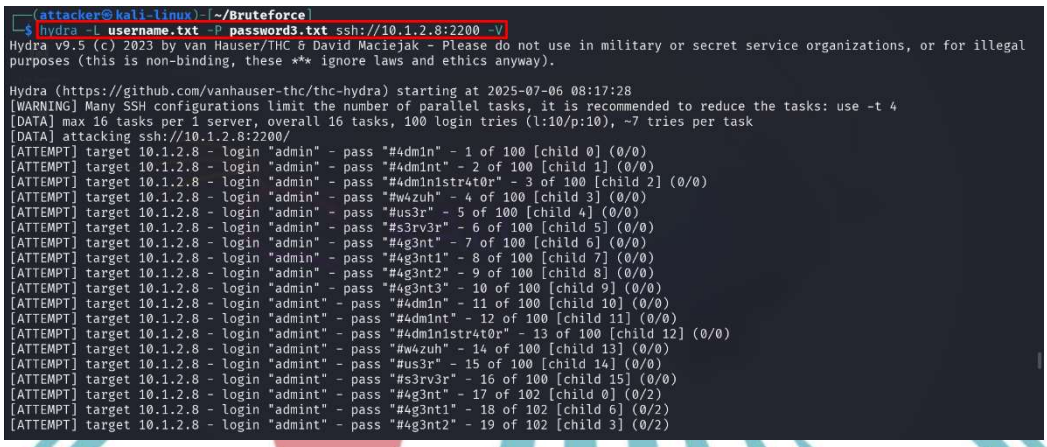
Pengujian ketiga serangan SSH *brute-force* dilakukan menggunakan *username* dan *password* dengan tingkat kesulitan sedang. *Tool* yang dilakukan untuk melakukan *brute-force* adalah Hydra. Hydra ini melakukan *brute-force* tersebut ke *port* SSH dan IP *address* yang dituju sampai mendapatkan *username* dan *password* yang valid.



Gambar 4.70 Pengujian Ketiga Serangan SSH Brute-force ke Wazuh Agent

Pengujian keempat serangan SSH *brute-force* dilakukan menggunakan *username* dan *password* dengan tingkat kesulitan sulit. *Tool* yang dilakukan untuk melakukan *brute-force* adalah Hydra. Hydra ini melakukan *brute-force* tersebut ke *port* SSH dan

IP *address* yang dituju sampai mendapatkan *username* dan *password* yang valid.



Gambar 4.71 Pengujian Keempat Serangan SSH Brute-force ke Wazuh Agent

4.4.2.4 Pengujian Serangan *Malware*

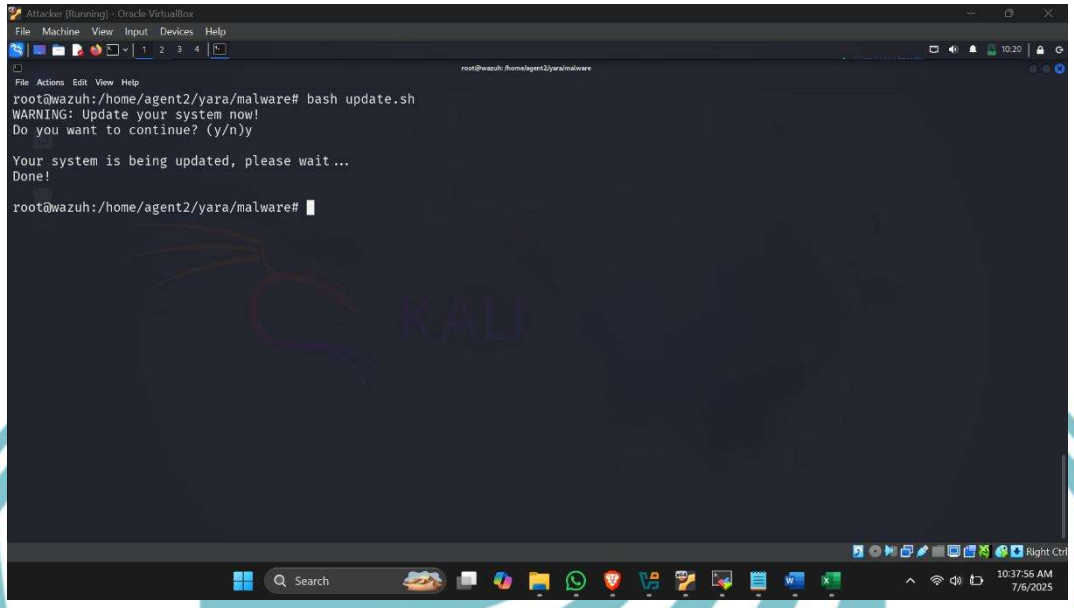
- Serangan *malware* pertama

Membuat *script* dalam Wazuh *agent* dengan nama yang tidak mencurigakan seperti update.sh. *Script* tersebut berisi perintah untuk mengunduh *malware* "Mirai". Mirai adalah *malware botnet* yang menargetkan perangkat Linux untuk menghabiskan *resources* sistem.

Gambar 4.72 Pembuatan Script Malware Pertama pada Wazuh Agent

Menjalankan *script* yang telah dibuat dengan instruksi yang tidak mencurigakan. Instruksi seperti untuk memperbarui sistem sekarang

dan terdapat *loading*-nya juga hingga keterangan selesai. Padahal proses yang sebenarnya terjadi adalah *malware* telah berhasil diunduh.



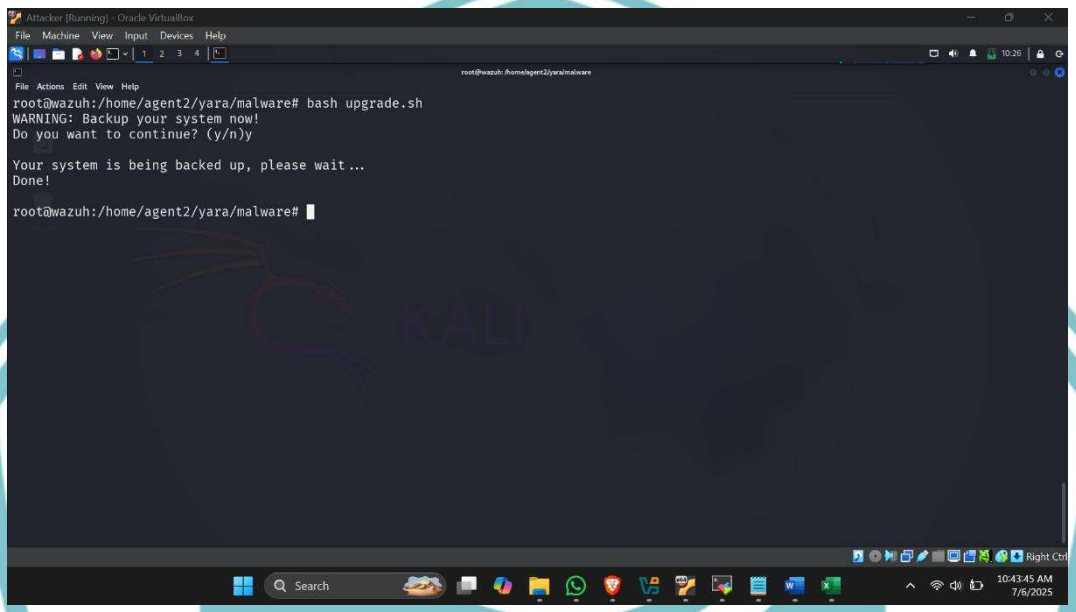
Gambar 4.73 Serangan Malware Pertama ke Wazuh Agent

- Serangan *malware* kedua

Membuat *script* dalam Wazuh *agent* dengan nama yang tidak mencurigakan seperti upgrade.sh. *Script* tersebut berisi perintah untuk mengunduh *malware* "Webshell". Webshell adalah *malware* yang sering digunakan sebagai *backdoor* untuk pengambilalihan *server*.

Gambar 4.74 Pembuatan Script Malware Kedua pada Wazuh Agent

Menjalankan *script* yang telah dibuat dengan instruksi yang tidak mencurigakan. Instruksi seperti untuk meningkatkan sistem sekarang dan terdapat *loading*-nya juga hingga keterangan selesai. Padahal proses yang sebenarnya terjadi adalah *malware* telah berhasil diunduh.



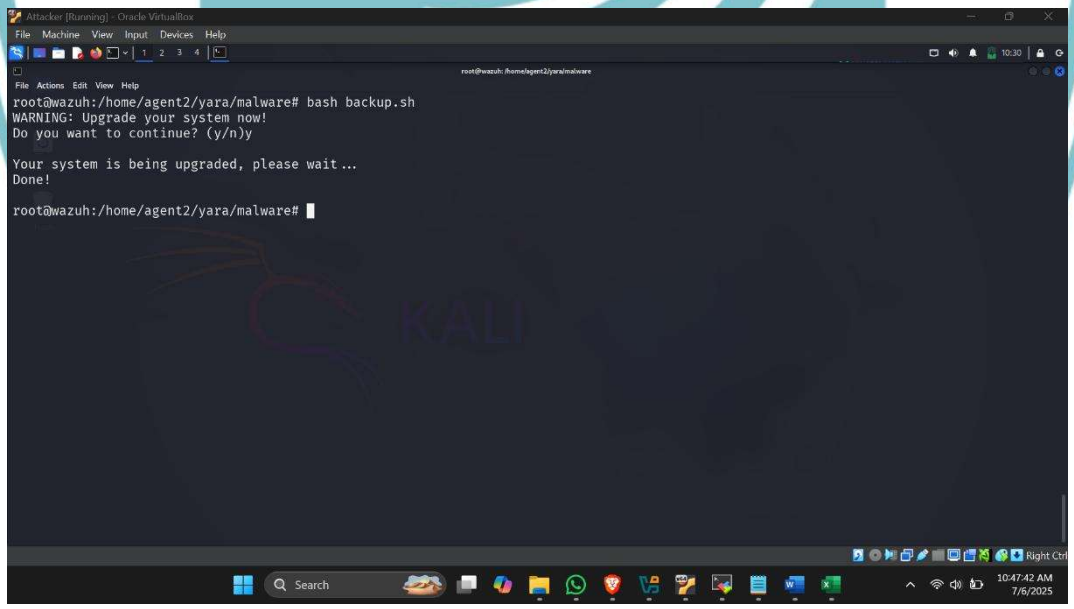
Gambar 4.75 Serangan Malware Kedua ke Wazuh Agent

- Serangan *malware* ketiga

Membuat *script* dalam Wazuh *agent* dengan nama yang tidak mencurigakan seperti backup.sh. *Script* tersebut berisi perintah untuk mengunduh *malware* "Xbash". Xbash adalah *malware* multifungsi yang menggabungkan kemampuan *ransomware*, *cryptojacker*, dan *botnet*.

Gambar 4.76 Pembuatan Script Malware Ketiga pada Wazuh Agent

Menjalankan *script* yang telah dibuat dengan instruksi yang tidak mencurigakan. Instruksi seperti untuk mencadangkan sistem sekarang dan terdapat *loading*-nya juga hingga keterangan selesai. Padahal proses yang sebenarnya terjadi adalah *malware* telah berhasil diunduh.



Gambar 4.77 Serangan Malware Ketiga ke Wazuh Agent

4.4.3 Data Hasil Pengujian

4.4.3.1 Hasil Pengujian Serangan *Port Scanning*

Serangan *port scanning* **berhasil** mendapatkan *port* yang terbuka, yaitu:

- ***Port* 80 (HTTP)**
- ***Port* 2200 (ICI)**

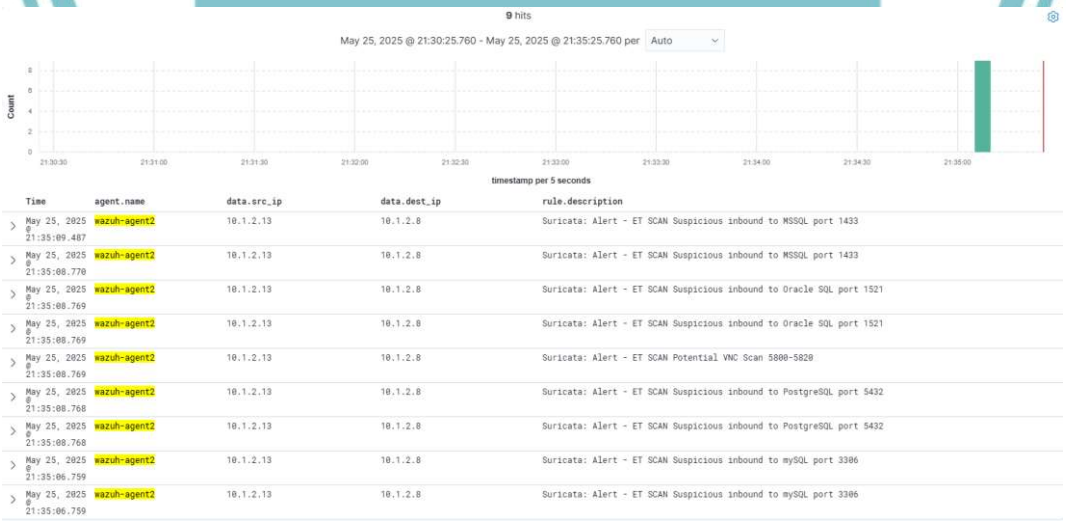
```
Nmap scan report for 10.1.2.8
Host is up (0.0021s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT     STATE SERVICE
80/tcp   open  http
2200/tcp open  ici
MAC Address: 08:00:27:82:3B:70 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 6.76 seconds
```

Gambar 4.78 Hasil Serangan Port Scanning pada Wazuh Agent

Serangan *port scanning* **berhasil** terdeteksi pada SIEM Wazuh dengan *rules* dari Suricata sebagai berikut:

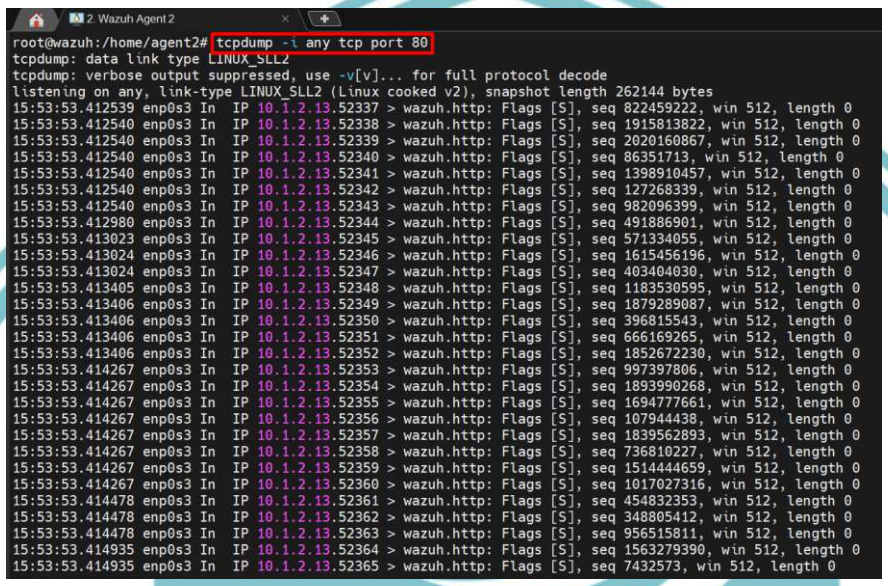
- Suricata: Alert - ET SCAN Potential VNC Scan 5800-5820
- Suricata: Alert - ET SCAN Suspicious inbound to mySQL port 3306
- Suricata: Alert - ET SCAN Suspicious inbound to MSSQL port 1433
- Suricata: Alert - ET SCAN Suspicious inbound to Oracle SQL port 1521
- Suricata: Alert - ET SCAN Suspicious inbound to PostgreSQL port 5432



Gambar 4.79 Hasil Deteksi Serangan Port Scanning pada Wazuh Agent

4.4.3.2 Hasil Pengujian Serangan *Denial of Service* (DoS)

Hasil *traffic* Wazuh *agent* dari serangan DoS yang dimonitoring menggunakan tcpdump. Bagian yang dimonitoring adalah protokol TCP dan *port* 80 (HTTP). *Attacker* membanjiri *traffic* pada target *server*.

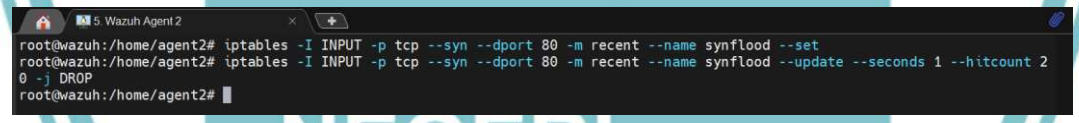


Gambar 4.80 Hasil Traffic Serangan DoS pada Wazuh Agent

Dampak dari serangan DoS tersebut adalah *web server* Apache mengalami **kelambatan sistem atau layanan karena belum terblokir oleh *firewall***. Hal ini disebabkan oleh paket yang diterima *server* sangat banyak dan tidak berhenti.

Gambar 4.81 Dampak Serangan DoS pada Web Server Wazuh Agent Sebelum Diblokir Firewall

Melakukan konfigurasi pada *firewall* untuk memblokir serangan DoS tersebut. Konfigurasi tersebut menggunakan iptables untuk melindungi *server* dari serangan SYN *flood*. Cara kerjanya adalah *firewall* akan mencatat dan memblokir IP yang terlalu sering mengirim paket SYN ke *port* 80 dalam waktu yang singkat.



Gambar 4.82 Konfigurasi Firewall pada Wazuh Agent

Berikut ini adalah hasil tampilan *web server* Apache yang berjalan **normal walaupun mendapatkan serangan DoS** yang dapat dilihat pada Gambar 4.83. Hal tersebut karena serangan DoS telah diblokir oleh *firewall*.

Gambar 4.83 Dampak Serangan DoS pada Web Server Wazuh Agent Setelah Diblokir Firewall

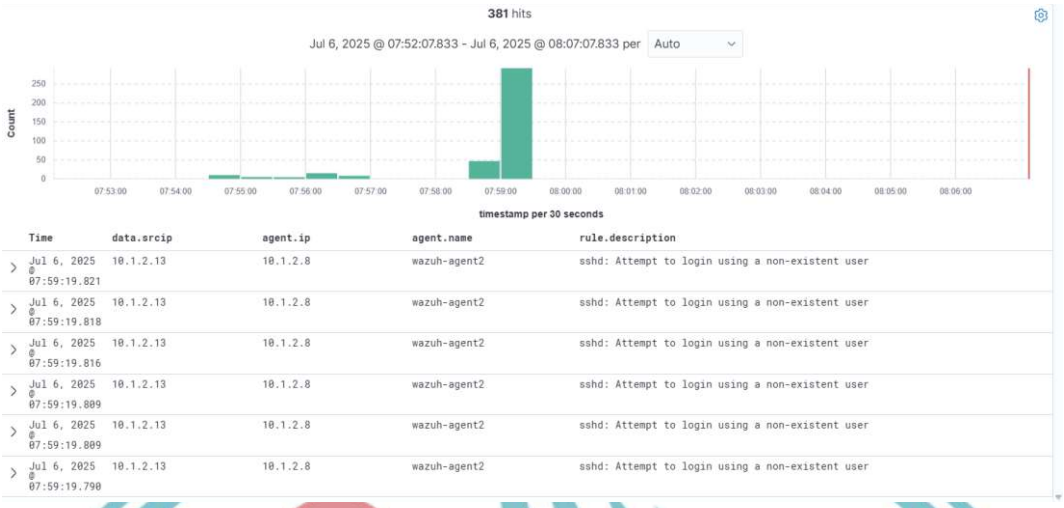
4.4.3.3 Hasil Pengujian Serangan SSH *brute-force*

Pengujian pertama serangan SSH *brute-force* dilakukan menggunakan *username* dan *password* yang sama dengan *username* hasilnya adalah **tidak berhasil** mendapatkan *username* dan *password* yang valid.

```
[ATTEMPT] target 10.1.2.8 - login "agent2" - pass "server" - 86 of 101 [child 10] (0/1)
[ATTEMPT] target 10.1.2.8 - login "agent2" - pass "agent" - 87 of 101 [child 0] (0/1)
[ATTEMPT] target 10.1.2.8 - login "agent2" - pass "agent1" - 88 of 101 [child 14] (0/1)
[ATTEMPT] target 10.1.2.8 - login "agent2" - pass "agent2" - 89 of 101 [child 5] (0/1)
[RE-ATTEMPT] target 10.1.2.8 - login "agent2" - pass "agent1" - 89 of 101 [child 14] (0/1)
[ATTEMPT] target 10.1.2.8 - login "agent2" - pass "agent3" - 90 of 101 [child 3] (0/1)
[RE-ATTEMPT] target 10.1.2.8 - login "agent3" - pass "agent1" - 90 of 101 [child 14] (0/1)
[ATTEMPT] target 10.1.2.8 - login "agent3" - pass "admin" - 91 of 101 [child 15] (0/1)
[RE-ATTEMPT] target 10.1.2.8 - login "agent3" - pass "admin" - 91 of 101 [child 15] (0/1)
[ATTEMPT] target 10.1.2.8 - login "agent3" - pass "admint" - 92 of 101 [child 9] (0/1)
[ATTEMPT] target 10.1.2.8 - login "agent3" - pass "administrator" - 93 of 101 [child 6] (0/1)
[ATTEMPT] target 10.1.2.8 - login "agent3" - pass "wazuh" - 94 of 101 [child 7] (0/1)
[ATTEMPT] target 10.1.2.8 - login "agent3" - pass "user" - 95 of 101 [child 4] (0/1)
[RE-ATTEMPT] target 10.1.2.8 - login "agent3" - pass "administrator" - 95 of 101 [child 6] (0/1)
[RE-ATTEMPT] target 10.1.2.8 - login "agent3" - pass "administrator" - 95 of 101 [child 6] (0/1)
[ATTEMPT] target 10.1.2.8 - login "agent3" - pass "server" - 96 of 101 [child 8] (0/1)
[ATTEMPT] target 10.1.2.8 - login "agent3" - pass "agent" - 97 of 101 [child 13] (0/1)
[RE-ATTEMPT] target 10.1.2.8 - login "agent3" - pass "server" - 97 of 101 [child 8] (0/1)
[ATTEMPT] target 10.1.2.8 - login "agent3" - pass "agent1" - 98 of 101 [child 12] (0/1)
[ATTEMPT] target 10.1.2.8 - login "agent3" - pass "agent2" - 99 of 101 [child 2] (0/1)
[RE-ATTEMPT] target 10.1.2.8 - login "agent3" - pass "agent" - 99 of 101 [child 13] (0/1)
[ATTEMPT] target 10.1.2.8 - login "agent3" - pass "agent3" - 100 of 101 [child 1] (0/1)
[REDO-ATTEMPT] target 10.1.2.8 - login "admint" - pass "admint" - 101 of 101 [child 10] (1/1)
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-07-06 07:59:19
```

Gambar 4.84 Hasil Serangan Pertama SSH Brute-force pada Wazuh Agent

Pengujian pertama serangan SSH *brute-force* dilakukan menggunakan *username* dan *password* yang sama dengan *username* hasilnya adalah **berhasil** terdeteksi pada SIEM Wazuh.



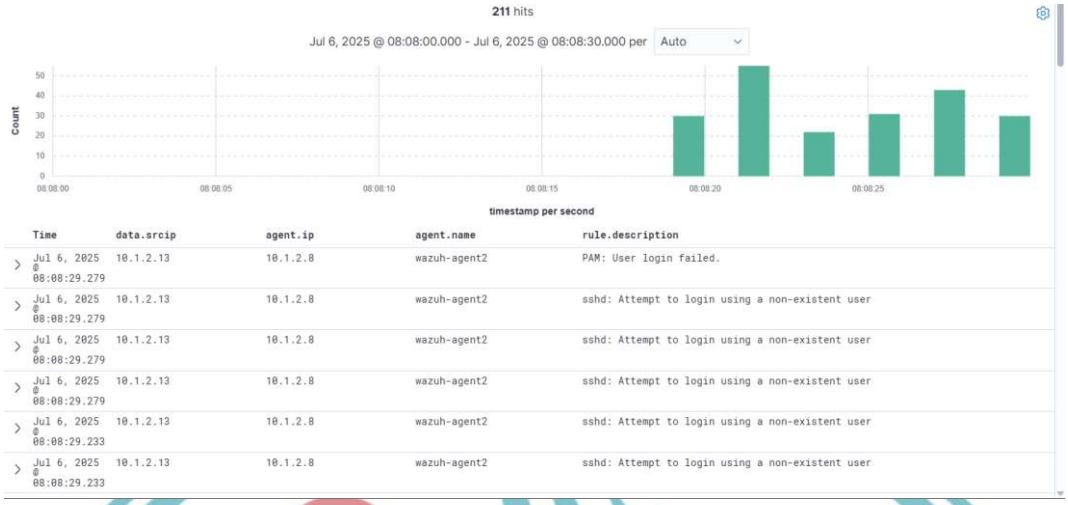
Gambar 4.85 Hasil Deteksi Serangan Pertama SSH Brute-force pada Wazuh Agent

Pengujian kedua serangan SSH *brute-force* dilakukan menggunakan *username* dan *password* dengan tingkat kesulitan mudah hasilnya adalah **tidak berhasil** mendapatkan *username* dan *password* yang valid.

```
[RE-ATTEMPT] target 10.1.2.8 - login "agent2" - pass "administrator123" - 84 of 102 [child 10] (0/2)
[ATTEMPT] target 10.1.2.8 - login "agent2" - pass "user123" - 85 of 102 [child 7] (0/2)
[ATTEMPT] target 10.1.2.8 - login "agent2" - pass "server123" - 86 of 102 [child 12] (0/2)
[ATTEMPT] target 10.1.2.8 - login "agent2" - pass "agent123" - 87 of 102 [child 8] (0/2)
[RE-ATTEMPT] target 10.1.2.8 - login "agent2" - pass "user123" - 87 of 102 [child 7] (0/2)
[ATTEMPT] target 10.1.2.8 - login "agent2" - pass "agent1" - 88 of 102 [child 3] (0/2)
[ATTEMPT] target 10.1.2.8 - login "agent2" - pass "agent2" - 89 of 102 [child 4] (0/2)
[RE-ATTEMPT] target 10.1.2.8 - login "agent2" - pass "agent1" - 89 of 102 [child 3] (0/2)
[ATTEMPT] target 10.1.2.8 - login "agent2" - pass "agent3" - 90 of 102 [child 11] (0/2)
[ATTEMPT] target 10.1.2.8 - login "agent3" - pass "admin123" - 91 of 102 [child 15] (0/2)
[ATTEMPT] target 10.1.2.8 - login "agent3" - pass "admint123" - 92 of 102 [child 1] (0/2)
[ATTEMPT] target 10.1.2.8 - login "agent3" - pass "administrator123" - 93 of 102 [child 5] (0/2)
[RE-ATTEMPT] target 10.1.2.8 - login "agent3" - pass "administrator123" - 93 of 102 [child 5] (0/2)
[ATTEMPT] target 10.1.2.8 - login "agent3" - pass "wazuh123" - 94 of 102 [child 0] (0/2)
[RE-ATTEMPT] target 10.1.2.8 - login "agent3" - pass "wazuh123" - 94 of 102 [child 0] (0/2)
[ATTEMPT] target 10.1.2.8 - login "agent3" - pass "user123" - 95 of 102 [child 2] (0/2)
[ATTEMPT] target 10.1.2.8 - login "agent3" - pass "server123" - 96 of 102 [child 6] (0/2)
[ATTEMPT] target 10.1.2.8 - login "agent3" - pass "agent123" - 97 of 102 [child 9] (0/2)
[ATTEMPT] target 10.1.2.8 - login "agent3" - pass "agent1" - 98 of 102 [child 10] (0/2)
[ATTEMPT] target 10.1.2.8 - login "agent3" - pass "agent2" - 99 of 102 [child 15] (0/2)
[ATTEMPT] target 10.1.2.8 - login "agent3" - pass "agent3" - 100 of 102 [child 12] (0/2)
[REDO-ATTEMPT] target 10.1.2.8 - login "admint" - pass "wazuh123" - 101 of 102 [child 8] (1/2)
[REDO-ATTEMPT] target 10.1.2.8 - login "admint" - pass "user123" - 102 of 102 [child 4] (2/2)
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-07-06 08:08:36
```

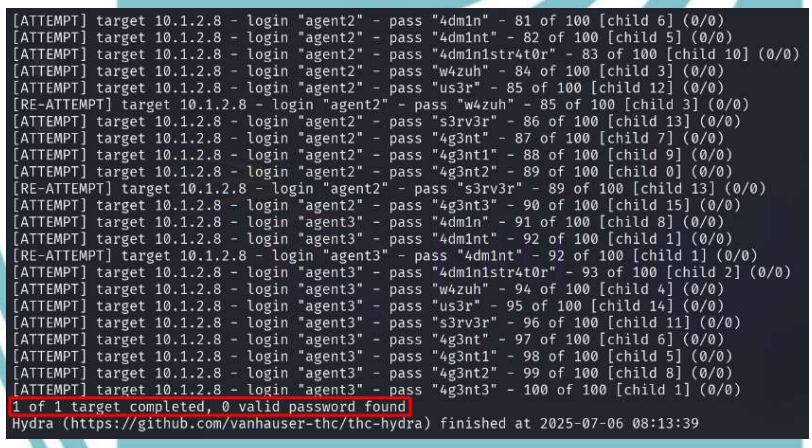
Gambar 4.86 Hasil Serangan Kedua SSH Brute-force pada Wazuh Agent

Pengujian kedua serangan SSH *brute-force* dilakukan menggunakan *username* dan *password* dengan tingkat kesulitan mudah hasilnya adalah **berhasil** terdeteksi pada SIEM Wazuh.



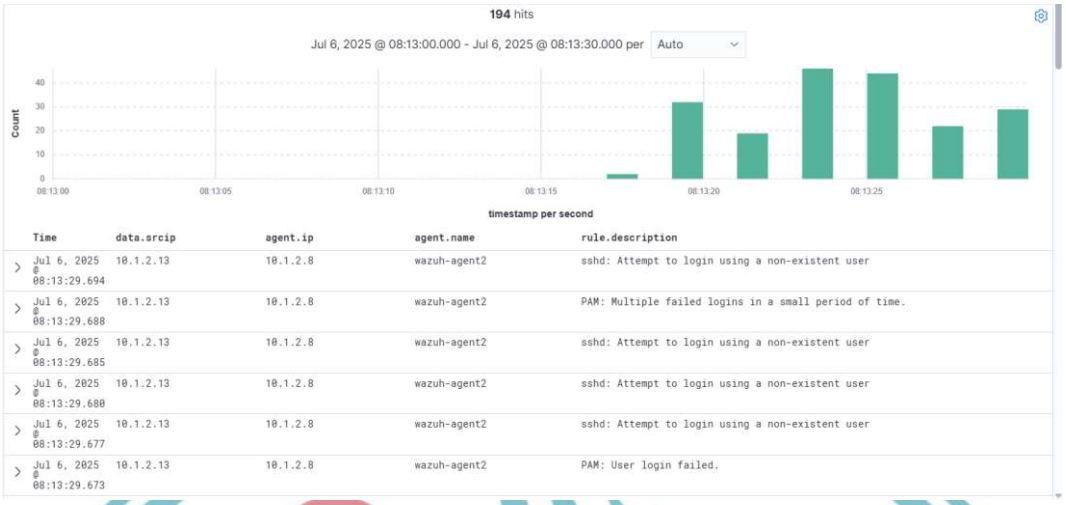
Gambar 4.87 Hasil Deteksi Serangan Kedua SSH Brute-force pada Wazuh Agent

Pengujian ketiga serangan SSH *brute-force* dilakukan menggunakan *username* dan *password* dengan tingkat kesulitan sedang hasilnya adalah **tidak berhasil** mendapatkan *username* dan *password* yang valid.



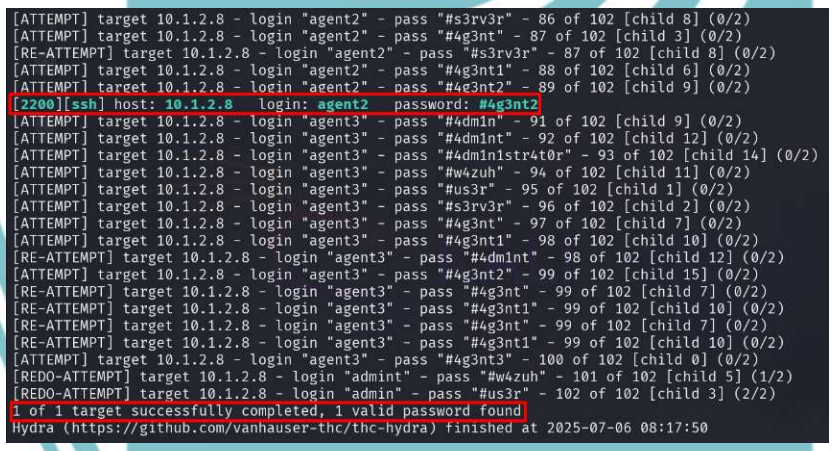
Gambar 4.88 Hasil Serangan Ketiga SSH Brute-force pada Wazuh Agent

Pengujian ketiga serangan SSH *brute-force* dilakukan menggunakan *username* dan *password* dengan tingkat kesulitan sedang hasilnya adalah **berhasil** terdeteksi pada SIEM Wazuh.



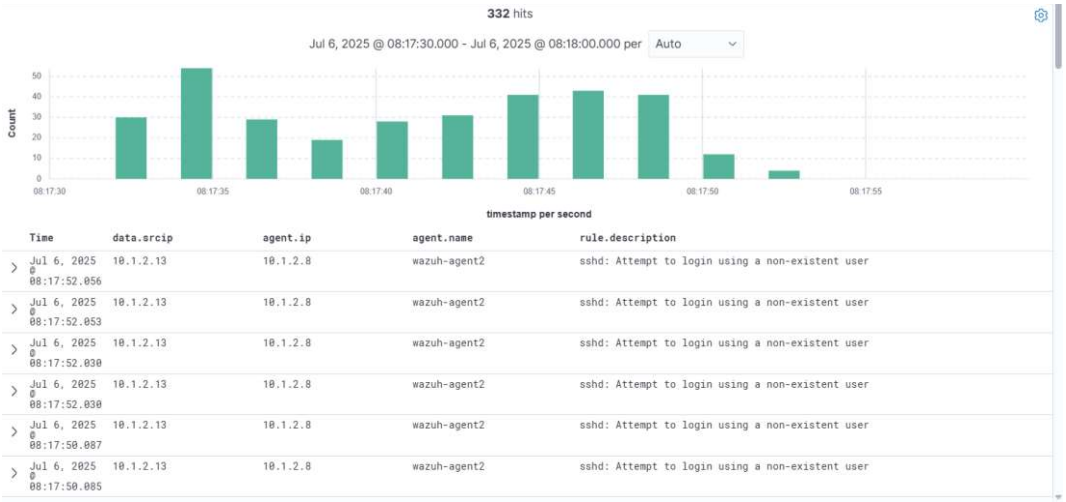
Gambar 4.89 Hasil Deteksi Serangan Ketiga SSH Brute-force pada Wazuh Agent

Pengujian keempat serangan SSH *brute-force* dilakukan menggunakan *username* dan *password* dengan tingkat kesulitan sulit hasilnya adalah **berhasil** mendapatkan *username* dan *password* yang valid.



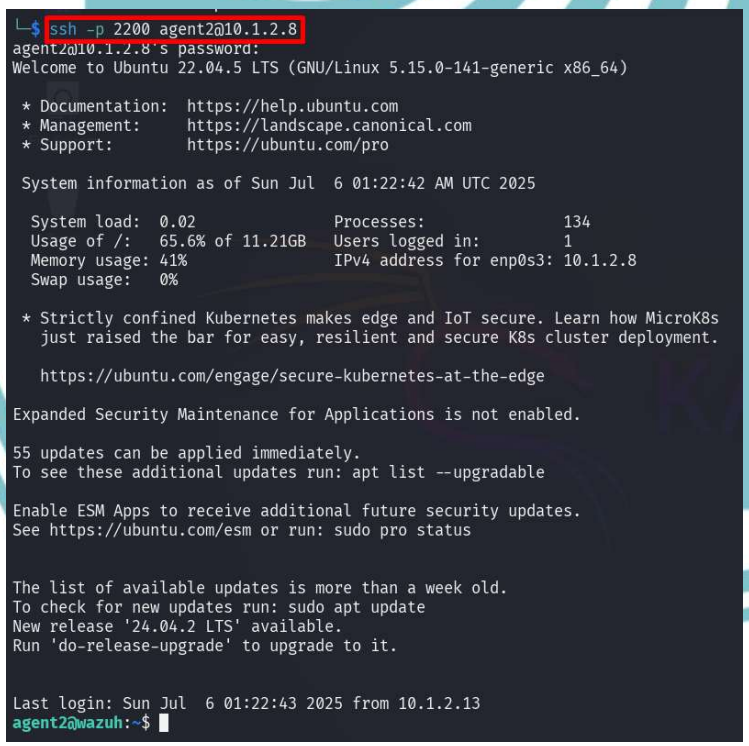
Gambar 4.90 Hasil Serangan Keempat SSH Brute-force pada Wazuh Agent

Pengujian keempat serangan SSH *brute-force* dilakukan menggunakan *username* dan *password* dengan tingkat kesulitan sulit hasilnya adalah **berhasil** terdeteksi pada SIEM Wazuh.



Gambar 4.91 Hasil Deteksi Serangan Keempat SSH Brute-force pada Wazuh Agent

Pengujian *login* SSH ke Wazuh *agent* dilakukan menggunakan *username* dan *password* yang telah didapatkan dari hasil *brute-force* dan hasilnya adalah **berhasil** *login*.



Gambar 4.92 Pengujian Login SSH ke Wazuh Agent

Pengujian *login* SSH ke Wazuh *agent* **berhasil** terdeteksi pada SIEM Wazuh dengan keterangan “sshd: authentication success”.

Gambar 4.93 Hasil Deteksi Pengujian Login SSH ke Wazuh Agent

4.4.3.4 Hasil Pengujian Serangan *Malware*

- Hasil serangan *malware* pertama

File Integrity Monitoring (FIM) hanya mendeteksi *file* yang ditambahkan (mirai) dan perubahan *hash* saja. Oleh karena itu, FIM **tidak berhasil** mendeteksi langsung *file* tersebut sebagai *malware*.

Gambar 4.94 Hasil Serangan Malware Pertama pada Wazuh Agent

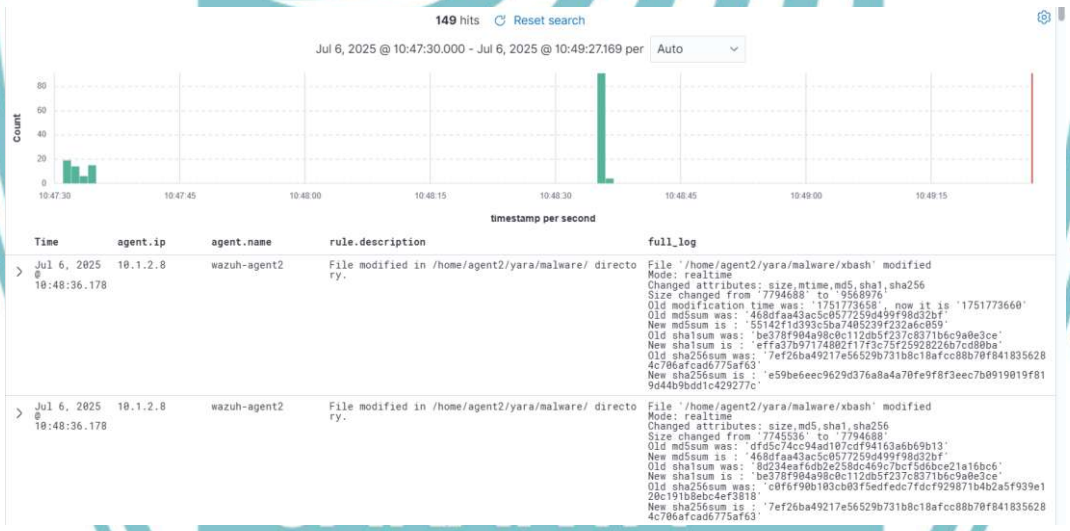
- Hasil serangan *malware* kedua

File Integrity Monitoring (FIM) hanya mendeteksi *file* yang ditambahkan (webshell) saja dan tidak mendeteksi perubahan *hash*. Oleh karena itu, FIM **tidak berhasil** mendeteksi langsung *file* tersebut sebagai *malware*.

Gambar 4.95 Hasil Serangan Malware Kedua pada Wazuh Agent

- Hasil serangan *malware* ketiga

File Integrity Monitoring (FIM) hanya mendeteksi *file* yang ditambahkan (xbash) dan perubahan *hash* saja. Oleh karena itu, FIM **tidak berhasil** mendeteksi langsung *file* tersebut sebagai *malware*.

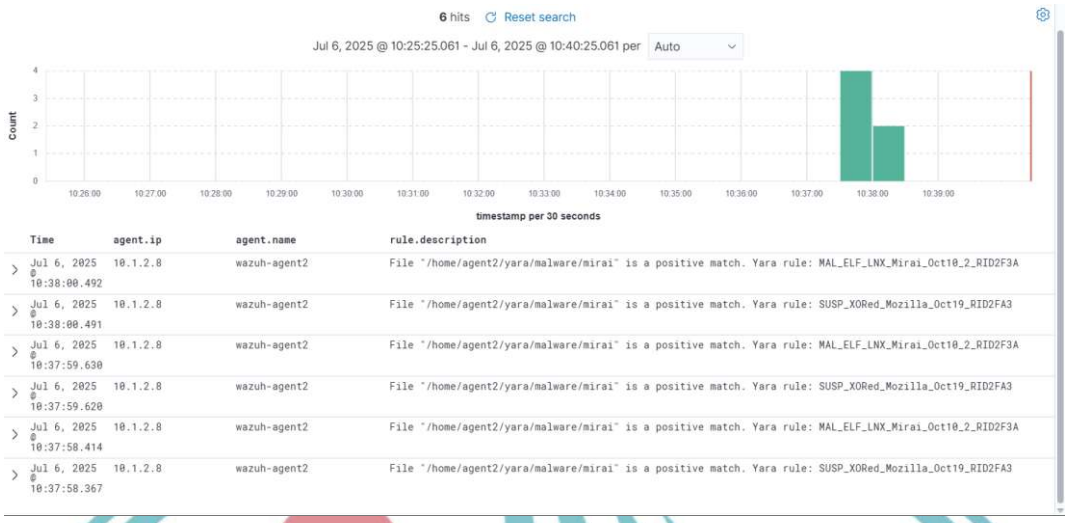


Gambar 4.96 Hasil Serangan Malware Ketiga pada Wazuh Agent

- Hasil serangan *malware* pertama

YARA **berhasil** mendeteksi *file* yang ditambahkan (mirai) sebagai *malware* dan cocok dengan *rules* YARA, yaitu:

- MAL_ELF_LNX_Mirai_Oct10_2_RID2F3A
- SUSP_XORed_Mozilla_Oct19_RID2FA3

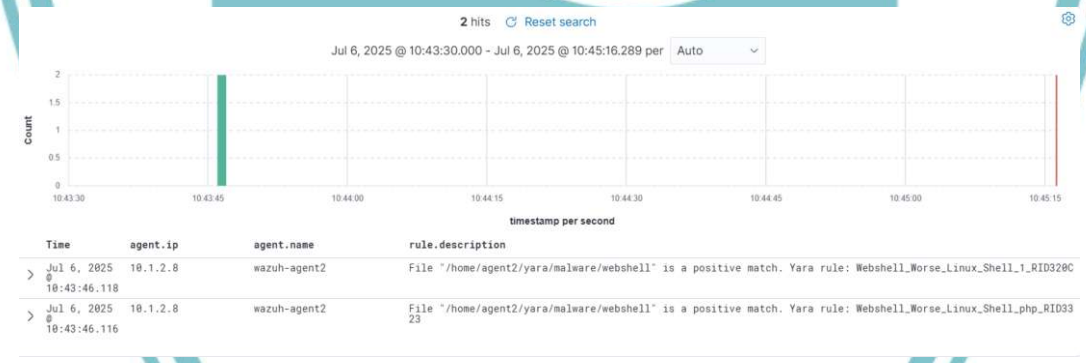


Gambar 4.97 Hasil Serangan Malware Pertama pada Wazuh Agent

- Hasil serangan *malware* kedua

YARA **berhasil** mendeteksi *file* yang ditambahkan (webshell) sebagai *malware* dan cocok dengan *rules* YARA, yaitu:

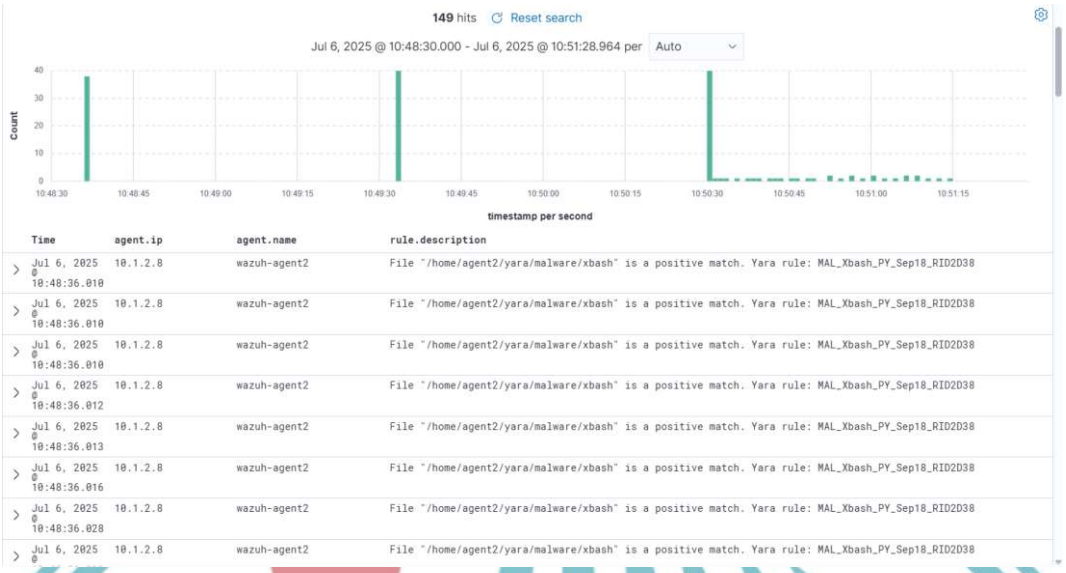
- Webshell_Worse_Linux_Shell_php_RID3323
- Webshell_Worse_Linux_Shell_1_RID320C



Gambar 4.98 Hasil Serangan Malware Kedua pada Wazuh Agent

- Hasil serangan *malware* ketiga

YARA **berhasil** mendeteksi *file* yang ditambahkan (xbash) sebagai *malware* dan cocok dengan *rules* YARA, yaitu MAL_Xbash_PY_Sep18_RID2D38.



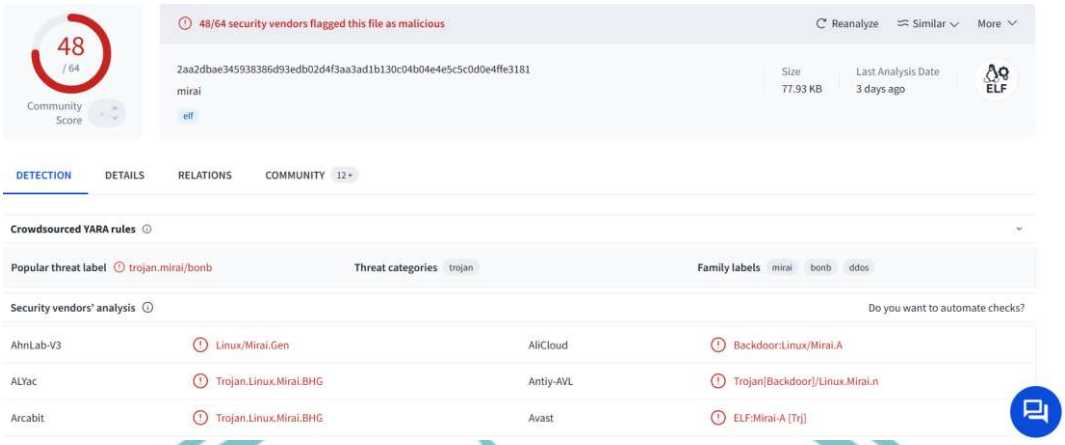
Gambar 4.99 Hasil Serangan Malware Ketiga pada Wazuh Agent

- Hasil serangan *malware* pertama

VirusTotal **berhasil** mendeteksi *file* yang ditambahkan (mirai) sebagai *malware* dan memberikan *output* berupa *link* detail mengenai *malware* tersebut pada *database* VirusTotal.

Gambar 4.100 Hasil Serangan Malware Pertama pada Wazuh Agent

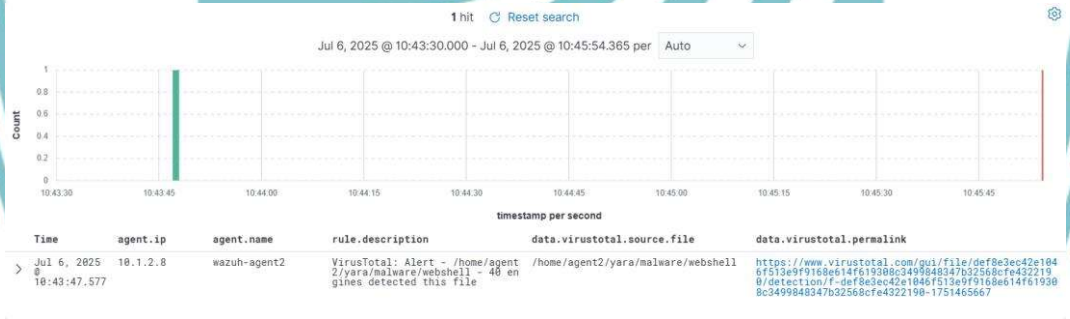
Gambar 4.101 menampilkan hasil dari VirusTotal yang mendeteksi *file* tersebut (mirai) sebagai *malware*. *Output* yang diberikan menjelaskan *file* tersebut tergolong *malicious* pada *database* VirusTotal *threat intelligence*.



Gambar 4.101 Detail Malware Serangan Pertama di VirusTotal

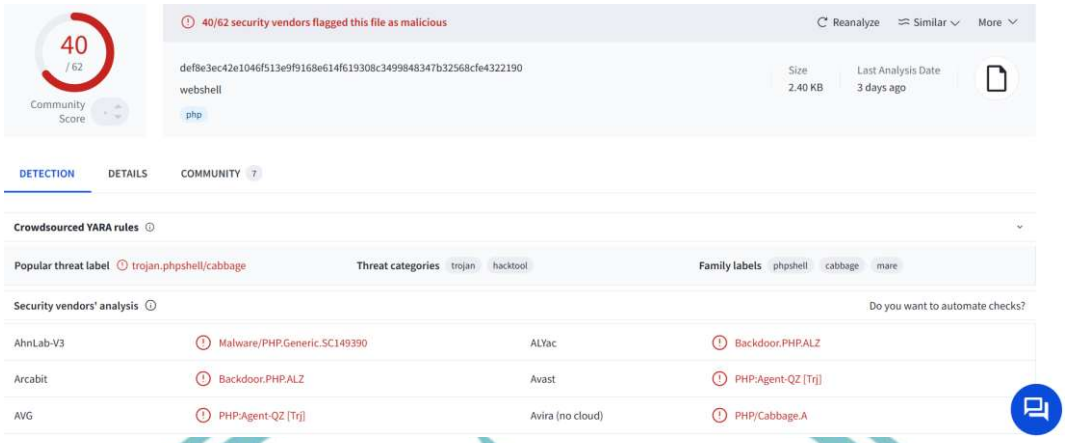
- Hasil serangan *malware* kedua

VirusTotal **berhasil** mendeteksi *file* yang ditambahkan (webshell) sebagai *malware* dan memberikan *output* berupa *link* detail mengenai *malware* tersebut pada *database* VirusTotal.



Gambar 4.102 Hasil Serangan Malware Kedua pada Wazuh Agent

Gambar 4.103 menampilkan hasil dari VirusTotal yang mendeteksi *file* tersebut (webshell) sebagai *malware*. *Output* yang diberikan menjelaskan *file* tersebut tergolong *malicious* pada *database* VirusTotal *threat intelligence*.



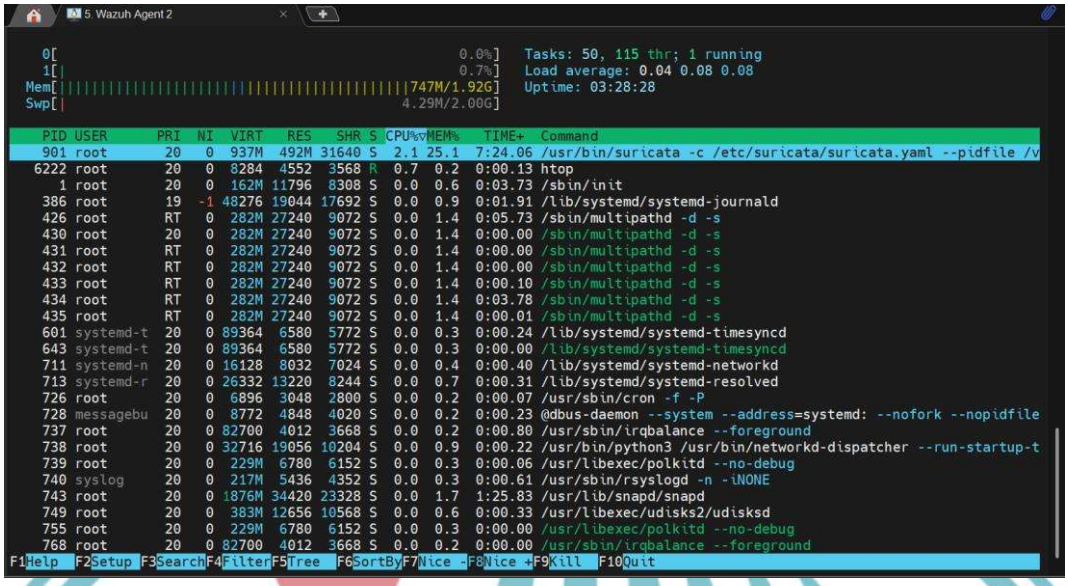
Gambar 4.103 Detail Malware Serangan Kedua di VirusTotal

- Hasil serangan *malware* ketiga

VirusTotal **tidak berhasil** mendeteksi *file* yang ditambahkan (xbash) sebagai *malware* dan tidak dapat memberikan *output* berupa *link* detail mengenai *malware* tersebut pada *database* VirusTotal.

Gambar 4.104 Hasil Serangan Malware Ketiga pada Wazuh Agent

Gambar menampilkan beban sistem yang digunakan oleh FIM, YARA, dan VirusTotal pada Wazuh *agent*. Beban sistem berpacu pada penggunaan CPU dan RAM pada sistem.



Gambar 4.105 Beban Sistem yang Digunakan

4.4.4 Analisis Data / Evaluasi Pengujian

4.4.4.1 Analisis Data Hasil Pengujian Serangan *Port Scanning*

Berikut adalah hasil analisis pengujian serangan *port scanning* yang dilakukan oleh *attacker* (10.1.2.13) menggunakan *tool* Nmap ke Wazuh *agent* (10.1.2.8).

a. Nama virtual mesin : Wazuh *agent*
b. Terdeteksi SIEM : terdeteksi
c. Keberhasilan serangan : berhasil
d. Hasil *port* yang terbuka : 80 (HTTP) dan 2200 (ICI)

4.4.4.2 Analisis Data Hasil Pengujian Serangan *Denial of Service* (DoS)

Berikut adalah hasil analisis pengujian serangan *Denial of Service* (DoS) yang dilakukan oleh *attacker* (10.1.2.13) menggunakan *tool* Hping3 ke *web server* Apache pada Wazuh *agent* (10.1.2.8).

a. Nama virtual mesin : Wazuh *agent*
b. *Port* yang diserang : 80 (HTTP)
c. Sebelum diblokir *firewall* : *web server* lambat
d. Setelah diblokir *firewall* : *web server* normal

4.4.4.3 Analisis Data Hasil Pengujian Serangan SSH *Brute-force*

Berikut adalah hasil pengujian serangan SSH ***b**rute-force* yang dilakukan oleh *attacker* (10.1.2.13) menggunakan *tool* Hydra ke Wazuh *agent* (10.1.2.8).

a. Nama virtual mesin : Wazuh *agent*
b. Terdeteksi SIEM : terdeteksi
c. *Port* yang diserang : 2200
d. Kombinasi *password username*: tidak berhasil
e. Kombinasi *password* mudah : tidak berhasil
f. Kombinasi *password* sedang : tidak berhasil
g. Kombinasi *password* sulit : berhasil

4.4.4.4 Analisis Data Hasil Pengujian Serangan *Malware*

Berdasarkan hasil pengujian serangan *malware* yang dilakukan dengan cara mengakses langsung ke Wazuh *agent*, kemudian membuat *script* yang tidak mencurigakan namun terdapat *malware* di dalamnya.

a. *File Integrity Monitoring* (FIM)

Berikut ini adalah hasil analisis data hasil pengujian serangan *malware* pada Wazuh *agent* yang dapat dilihat pada Tabel 4.4.

Tabel 4.4 Analisis Data Hasil Pengujian Serangan Malware Menggunakan FIM

FIM	Keberhasilan Unduh *Malware*	Terdeteksi SIEM	Terdeteksi Sebagai *Malware*
Malware 1	Berhasil	Terdeteksi	Tidak Terdeteksi
Malware 2	Berhasil	Terdeteksi	Tidak Terdeteksi
Malware 3	Berhasil	Terdeteksi	Tidak Terdeteksi

1. Jumlah *malware* yang terdeteksi

   = **Tidak ada**

2. Kecepatan deteksi *malware*

   = **Tidak ada**

3. Tingkat kesalahan

   True Positive (TP) = 0

   False Negative (FN) = 3

   False Negative Rate (FNR)

   $= \frac{FN}{(FN+TP)} = \frac{3}{(3+0)} = 1.00 =$ **100%**

4. Jenis *malware* yang terdeteksi

   = **Tidak ada**

5. Beban sistem yang digunakan

   CPU = (*core* 0 + *core* 1) : 2

   = (0.0% + 0.7%) : 2

   = **0.35%**

   RAM = 687 MB dari 1.92 GB

   Konversi GB ke MB = 1.92 GB x 1024 MB = 1966.08 MB

   Persentase = $\frac{747}{1966.08}$ x 100 = **38%**

b. YARA

Berikut ini adalah hasil analisis data hasil pengujian serangan *malware* pada Wazuh *agent* yang dapat dilihat pada Tabel 4.5.

Tabel 4.5 Analisis Data Hasil Pengujian Serangan Malware Menggunakan YARA

YARA	Keberhasilan Unduh *Malware*	Terdeteksi SIEM	Terdeteksi Sebagai *Malware*
Malware 1	Berhasil	Terdeteksi	Terdeteksi
Malware 2	Berhasil	Terdeteksi	Terdeteksi
Malware 3	Berhasil	Terdeteksi	Terdeteksi

1. Jumlah *malware* yang terdeteksi

   = **3 (tiga)**

2. Kecepatan deteksi *malware*

   - *Malware* 1

   = waktu serangan – waktu terdeteksi

   = 10:37:56 - 10:37:58

   = **2 detik**

   - *Malware* 2

   = waktu serangan – waktu terdeteksi

   = 10:43:45 - 10:43:46

   = **1 detik**

   - *Malware* 3

   = waktu serangan – waktu terdeteksi

   = 10:47:42 - 10:48:36

   = **54 detik**

3. Tingkat kesalahan

   True Positive (TP) = 3

   False Negative (FN) = 0

   False Negative Rate (FNR)

   $= \frac{FN}{(FN+TP)} = \frac{0}{(0+3)} = 0.00 =$ **0%**

4. Jenis *malware* yang terdeteksi

   - *Malware* 1: **Mirai**
   - *Malware* 2: **Webshell**
   - *Malware* 3: **Xbash**

5. Beban sistem yang digunakan

   CPU = (*core* 0 + *core* 1) : 2

   = (0.0% + 0.7%) : 2

   = **0.35%**

   RAM = 687 MB dari 1.92 GB

   Konversi GB ke MB = 1.92 GB x 1024 MB = 1966.08 MB

   Persentase = $\frac{747}{1966.08}$ x 100 = **38%**

c. VirusTotal

   Berikut ini adalah hasil analisis data pengujian serangan *malware* pada Wazuh *agent* yang dapat dilihat pada Tabel 4.6.

Tabel 4.6 Analisis Data Hasil Pengujian Serangan Malware Menggunakan VirusTotal

VirusTotal	Keberhasilan Unduh *Malware*	Terdeteksi SIEM	Terdeteksi Sebagai *Malware*
Malware 1	Berhasil	Terdeteksi	Terdeteksi
Malware 2	Berhasil	Terdeteksi	Terdeteksi
Malware 3	Berhasil	Terdeteksi	Tidak Terdeteksi

1. Jumlah *malware* yang terdeteksi

   = **2 (dua)**

2. Kecepatan deteksi *malware*

   - *Malware* 1

   = waktu serangan – waktu terdeteksi

   = 10:37:56 - 10:38:00

   = **4 detik**

- *Malware* 2

= waktu serangan – waktu terdeteksi

= 10:43:45 - 10:43:47

= **2 detik**

- *Malware* 3

= **Tidak terdeteksi**

3. Tingkat kesalahan

True Positive (TP) = 2

False Negative (FN) = 1

False Negative Rate (FNR)

$= \frac{FN}{(FN+TP)} = \frac{1}{(1+2)} = 0.33 =$ **33.33%**

4. Jenis *malware* yang terdeteksi
   - *Malware* 1: **Mirai**
   - *Malware* 2: **Webshell**
   - *Malware* 3: **Tidak terdeteksi**
5. Beban sistem yang digunakan

CPU = (*core* 0 + *core* 1) : 2

= (0.0% + 0.7%) : 2

= **0.35%**

RAM = 687 MB dari 1.92 GB

Konversi GB ke MB = 1.92 GB x 1024 MB = 1966.08 MB

Persentase = $\frac{747}{1966.08}$ x 100 = **38%**

Berikut ini adalah hasil analisis data pengujian *malware* pada Wazuh *agent* yang dapat dilihat pada Tabel 4.7.

Tabel 4.7 Analisis Data Hasil Pengujian Serangan Malware pada Wazuh Agent

Parameter	FIM	YARA	VirusTotal
Jumlah *malware* yang terdeteksi	0	3	2
Kecepatan deteksi *malware*	Tidak Ada	*Malware* 1 = 2 detik	*Malware* 1 = 4 detik
		Malware 2 = 1 detik	*Malware* 2 = 2 detik
		Malware 3 = 54 detik	*Malware* 3 = Tidak Terdeteksi
Tingkat kesalahan	100%	0%	33.33%
Jenis *malware* yang terdeteksi	Tidak Ada	*Malware* 1 = Mirai	*Malware* 1: Mirai
		Malware 2 = Webshell	*Malware* 2: Webshell
		Malware 3 = Xbash	*Malware* 3: Tidak Terdeteksi
Beban sistem yang digunakan	CPU: 0.35%	CPU: 0.35%	CPU: 0.35%
	RAM: 38%	RAM: 38%	RAM: 38%

Berdasarkan hasil analisis data pengujian yang didapatkan, *tools* yang paling baik kinerjanya pada SIEM Wazuh adalah **YARA**. Hal tersebut dikarenakan YARA dapat mendeteksi semua *malware* dari pengujian serangan *malware* yang telah dilakukan.

BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat diambil kesimpulan sebagai berikut:

a. Rancang bangun SIEM Wazuh untuk mendeteksi serangan (*port scanning*, *DoS*, *brute-force*, dan *malware*) berjalan dengan baik, mulai dari instalasi Wazuh *server,* Wazuh *indexer*, Wazuh *dashboard*, Wazuh *agent,* Suricata, *web server* Apache, dan konfigurasi *firewall*.
b. Penerapan FIM, YARA, dan VirusTotal ke SIEM Wazuh berjalan dengan baik. Penerapan FIM dengan cara menambahkan direktori yang ingin dimonitor pada *file* konfigurasi OSSEC. Penerapan YARA dengan cara melakukan instalasi dan konfigurasi YARA, serta menambahkan *rules* dari YARA. Penerapan VirusTotal dengan cara menambahkan API *key* dari VirusTotal pada *file* konfigurasi OSSEC.
c. Hasil analisisnya adalah FIM tidak berhasil mendeteksi semua *malware* karena hanya mendeteksi berdasarkan integritas *file* saja, YARA berhasil mendeteksi semua *malware* karena mendeteksi berdasarkan *rules,* dan VirusTotal hanya berhasil mendeteksi dua dari tiga *malware* saja karena bergantung pada *database*. Oleh karena itu, *tools* yang paling baik dalam mendeteksi *malware* adalah YARA.

5.2 Saran

Saran yang dapat diterapkan untuk melakukan penelitian lebih lanjut sebagai berikut:

a. Melakukan analisis kinerja SIEM lain dalam mendeteksi serangan, seperti IBM QRadar, Elastic, Splunk, dan lain-lain untuk mengukur kinerja SIEM yang paling baik.

b. Menerapkan *tools* lain dalam mendeteksi *malware*, seperti MISP, Kaspersky, Hybrid Analysis, dan lain-lain. Selain itu, menerapkan *tools Endpoint Detection and Response* (EDR) untuk pemblokiran *malware*, seperti CrowdStrike, Microsoft Defender, SentinelOne, dan lain-lain.
c. Melakukan monitoring *server* dengan sistem operasi yang berbeda, seperti CentOS, Debian, Red Hat Enterprise Linux (RHEL), dan lain-lain.
d. Melakukan eksperimen menggunakan perangkat nyata dalam ruang lingkup global dan berada pada jaringan yang berbeda.
e. Menambahkan pengujian serangan-serangan lain, seperti berpacu pada MITRE ATT&CK dan OWASP Top 10.
f. Menambahkan pemblokiran *malware* dengan parameter seberapa cepat dan tepat *malware* tersebut diblokir.
g. Menambahkan analisis kinerja hasil pengujian serangan-serangan lain, seperti berpacu pada MITRE ATT&CK dan OWASP Top 10.

DAFTAR PUSTAKA

Adirosa, G., 2021. ANALISIS KINERJA SECURITY INFORMATION AND EVENT MANAGEMENT (SIEM) IBM QRADAR COMMUNITY EDITION DALAM MENDETEKSI ANCAMAN DAN SERANGAN SIBER PADA SERVER.

Alfandi, M., 2022. Analisa Security Information and Event Management (SIEM) Menggunakan Elastic Stack SIEM dan Splunk.

Azzah Shafiyyah, G. F. N. R. A. P., 2024. IMPLEMENTASI WAZUH MENGGUNAKAN METODE PPDIOO DI SISTEM KEAMANAN JARINGAN PSDKU UNIVERSITAS LAMPUNG WAYKANAN SEBAGAI DETEKSI DAN RESPON SERANGAN SIBER. *JITET (Jurnal Informatika dan Teknik Elektro Terapan),* 12(2), pp. 1-13.

Badan Siber dan Sandi Negara, 2024. *Laporan Tahunan Layanan Honeynet BSSN,* s.l.: s.n.

Husnul Khotimah, F. B. R. S. K. I. B. K. W., 2022. Implementasi Security Information And Event Management (SIEM) Pada Aplikasi Sms Center Pemerintah Daerah Provinsi Nusa Tenggara Barat. *JBegaTI,* September, 3(2), p. 7.

Lintang, D., 2020. MONITORING AKTIVITAS USER PADA SYSTEM DENGAN MENGGUNAKAN EFK (ELASTICSEARCH, FLUENTD, KIBANA) STACK.

Muhammad Rijal Kamal, M. A. S., 2021. Deteksi Anomali dengan Security Information and Event Management (SIEM) Splunk pada Jaringan UII. *Automata,* 2(2), pp. 1-6.

Nur Rohman Rosyid, B. B. M. B. P. A. F. R. L. S., 2023. Deteksi Malware pada Jaringan Lokal Berbasis Honeypot dan Yara. *JURNAL SISTEMASI,* 12(1), pp. 186-193.

Nurul Qomariah, E. I. A. M. A. A., 2023. Analisis Malware Hummingbad Dan Copycat Pada Android Menggunakan Metode Hybrid. *Jurnal Cyber Security dan Forensic Digital,* 6(2), pp. 39-47.

Oky Dwi Prasetyo, P. H. T. A. B., 2023. Uji Kinerja Host-Based Intrution Detection System WAZUH terhadap Serangan Brute Force dan Dos. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* Juni, 7(6), pp. 2686-2692.

Prakoso, R. D., 2017. IMPLEMENTASI DAN PERBANDINGAN PERFORMA PROXMOX DALAM VIRTUALISASI DENGAN TIGA VIRTUAL SERVER. *JURNAL MANAJEMEN INFORMATIKA,* 8(1), pp. 1-8.

Stefan Stanković, S. G. R. P., 2022. A Review of Wazuh Tool Capabilities for Detecting Attacks Based on Log Analysis.

Thorarensen, C., 2021. A Performance Analysis of Intrusion Detection with Snort and Security Information Management. *DiVA.*

Yuriansyah Ilhamdi, Y. N. K., 2017. ANALISIS MALWARE PADA SISTEM OPERASI WINDOWS. *JUSTINDO (Jurnal Sistem dan Teknologi Informasi Indonesia),* 2(1), pp. 1-12.

DAFTAR RIWAYAT HIDUP

Robby Akbar Abdullah

Lahir di salah satu kota Indonesia pada tahun 2003. Penulis merupakan anak kedua dari dua bersaudara. Saat ini penulis berdomisili di daerah Jakarta Selatan. Penulis menyelesaikan pendidikan Sekolah Dasar (SD) pada tahun 2015 dan pendidikan Sekolah Menengah Pertama (SMP) pada tahun 2018. Kemudian, penulis menyelesaikan pendidikan Sekolah Menengah Kejuruan (SMK) dengan jurusan Teknik Komputer dan Jaringan (TKJ) pada tahun 2021. Lalu, penulis menyelesaikan pendidikan Diploma Empat (D4) di Politeknik Negeri Jakarta dengan program studi Teknik Multimedia dan Jaringan (TMJ) pada tahun 2025.