

**IMPLEMENTASI SENSOR MALTRAIL DAN FAIL2BAN UNTUK
MENDETEKSI DAN MENCEGAH SERANGAN MALWARE PADA
JARINGAN SERVER DISKOMINFO SUMEDANG DENGAN PUSH
NOTIFIKASI**

*Implementation of Maltrail Sensor and Fail2Ban For Detection and Prevention System
with Malware Attack on Server Network at Diskominfo of Sumedang With Push
Notification*

PROYEK AKHIR

**Disusun sebagai salah satu syarat untuk memperoleh Gelar Ahli Madya pada
Program Studi Diploma-3 Teknologi Telekomunikasi Fakultas Ilmu Terapan
Universitas Telkom**

oleh:

**RAMA WIJAYA SHIDDIQ
6705184073**



**D3 TEKNOLOGI TELEKOMUNIKASI
FAKULTAS ILMU TERAPAN
UNIVERSITAS TELKOM
2021**

LEMBAR PENGESAHAN

Proposal Proyek Akhir dengan judul:

**IMPLEMENTASI SENSOR MALTRAIL DAN FAIL2BAN UNTUK MENDETEKSI
DAN MENCEGAH SERANGAN *MALWARE* PADA JARINGAN SERVER
DISKOMINFO SUMEDANG DENGAN PUSH NOTIFIKASI**

*Implementation of Maltrail Sensor and Fail2Ban For Detection and Prevention System
with Malware Attack on Server Network at Diskominfo of Sumedang With Push
Notification*

oleh:

RAMA WIJAYA SHIDDIQ

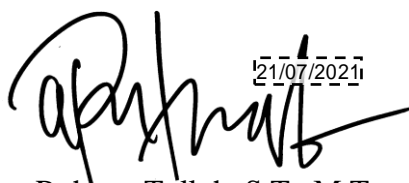
6705184073

Telah diterima dan disahkan sebagai salah satu syarat untuk memperoleh gelar Ahli Madya
pada Program Studi D3 Teknologi Telekomunikasi Universitas Telkom

Bandung, Juli 2021

Menyetujui,

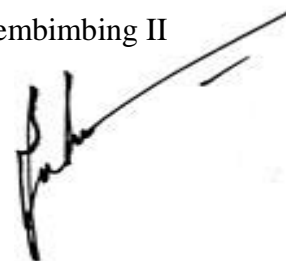
Pembimbing I

A handwritten signature in black ink, appearing to read 'Rohmat Tulloh', with a date stamp '21/07/2021' to its right.

Rohmat Tulloh, S.T., M.T.

NIP. 06830002

Pembimbing II

A handwritten signature in black ink, appearing to read 'Nugraha', with a long horizontal stroke extending to the right.

Nugraha, S.Sos. M.Si.

NIK. 197204122009011001

LEMBAR PERNYATAAN ORISINALITAS

Dengan ini, Saya :

Nama : Rama Wijaya Shiddiq
NIM : 6705184073
Alamat : Lingkungan Selareuma RT.03 RW.07 Kecamatan Sumedang Selatan
Kab. Sumedang
No. Tlp/HP : 082118529153
Email : ramawijayashiddiq7@gmail.com

Menyatakan bahwa Proyek Akhir dengan judul:

**IMPLEMENTASI SENSOR MALTRAIL DAN FAIL2BAN UNTUK MENDETEKSI
DAN MENCEGAH SERANGAN MALWARE PADA JARINGAN SERVER
DISKOMINFO SUMEDANG DENGAN PUSH NOTIFIKASI**

*Implementation of Maltrail Sensor and Fail2Ban For Detection and Prevention System
with Malware Attack on Server Network at Diskominfo of Sumedang With Push
Notification*

Merupakan karya orisinil saya sendiri dan atas pernyataan ini, saya siap menanggung resiko/sanksi yang dijatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap kejujuran akademik atau etika keilmuan dalam karya ini, atau ditemukan bukti yang menunjukkan ketidakaslian karya ini.



Bandung, Juli 2021

Rama Wijaya Shiddiq

6705184073

IDENTITAS BUKU

Nama Penulis	:	Rama Wijaya Shiddiq
Tahun Pengesahan	:	2021
Pembimbing 1	:	Rohmat Tulloh, ST., MT.
Afiliasi Pembimbing 1	:	D3 Teknologi Telekomunikasi Universitas Telkom
Pembimbing 2	:	Nugraha, S.Sos. M.Si
Afiliasi Pembimbing 2	:	Diskominfo
Program Studi	:	D3 Teknologi Telekomunikasi
Fakultas	:	Fakultas Ilmu Terapan
Jenis Buku	:	Laporan Proyek Akhir
Subjek Buku	:	Network Security

ABSTRAK

Dinas Komunikasi dan Informatika Persandian dan Statistik (Diskominfo) Kota Sumedang, merupakan organisasi pelayanan publik yang bertanggung jawab menangani bidang data dan jaringan komunikasi yang menghubungkan semua lembaga pemerintahan seperti kelurahan, kecamatan dan dinas-dinas yang terhubung ke server Diskominfo Sumedang. Tugas server yaitu melayani semua perangkat yang terhubung ke jaringannya, seperti memonitoring seluruh keamanan aktivitas jaringan, perlindungan sistem, data, dan peningkatan kualitas keamanan jaringan. Melihat hal tersebut dibutuhkan sebuah sistem yang dapat mendeteksi dan memblokir malware-malware yang berusaha masuk ke jaringan server Diskominfo Sumedang.

Pada Proyek Akhir ini dirancang suatu sistem implementasi sensor Maltrail (Malware Trail) dan Fail2Ban untuk mendeteksi dan mencegah serangan malware pada jaringan server Diskominfo Sumedang dengan push notifikasi, yang merupakan solusi lain dari permasalahan tersebut. Software yang digunakan untuk melakukan pendeteksian yaitu Maltrail. Cara kerja dari software ini sebagai sensor yang memindai seluruh aktivitas trafik pada jaringan server. Kemudian, software yang digunakan untuk melakukan blocking atau pencegahan dari serangan malware, yaitu Fail2Ban. Sistem tersebut menggunakan bot telegram sebagai push notifikasi jika ada serangan malware ke server.

Dari hasil pengujian serangan malware pada server, terjadi penurunan throughput sebesar 18%, hasil implementasi sistem ini mampu mendeteksi dan memblokir malware trafik pada jaringan, akibatnya tidak mengalami penurunan throughput yang cukup jauh. Kemudian sistem mampu mendeteksi serangan selain malware yaitu scanning port dengan tingkat ancaman 2.7%. Sehingga sistem mampu meminimalisir ancaman serangan yang datang dari internet maupun intranet dan mengurangi kerugian-kerugian yang disebabkan oleh malware pada jaringan server Diskominfo Sumedang.

Kata Kunci: Maltrail, Fail2Ban, *malware*, mendeteksi, mencegah.

ABSTRACT

Diskominfo Sumedang is a public service organization that is responsible for data and communications that connects all government institutions such as sub-districts, and offices that are connected to the Sumedang Diskominfo server. The server's job is to server all devices connected to the network, such as monitoring all network activity security, system protection, data, and improving network security quality. Seeing this, the basic needs of Diskominfo Sumedang, we need a system that can detect and block malware that tries to enter the server network at Diskominfo Sumedang.

In this final, a Maltrail (Malware Trail) and Fail2Ban sensor implementation system is designed to detect and prevent malware attacks on Diskominfo Sumedang network server with push notifications, which is another solution to this problem. The software used to detect malware is Maltrail. The way this software works is with sensors that scan all traffic activities on the server network. Then, the software used to block or prevent malware attacks, namely Fail2Ban. The system uses telegram bots as notifications if there is a malware attack on the server.

From the result of malware attacks on the server, there was a decrease in throughput of 18%, the results of the implementation of this system were able to detect and block malware traffic on the network, as a result it did not experience a significant decrease in throughput. Then the system detects attacks other than malware, namely scanning ports with a threat level of 2.7%. So that the system is able to minimize the threats of attacks coming from internet or intranets and reduce losses caused by malware on the Diskominfo Sumedang network server.

Keywords: *Maltrail, Fail2Ban, Malware, Detection, Prevention.*

KATA PENGANTAR

Assalamualaikum warahmatullahi wabarakatuh.

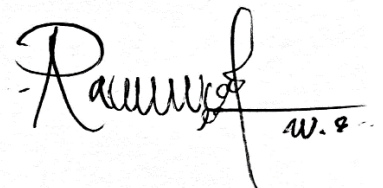
Puji syukur alhamdulillah, penulis persembahkan kehadiran Allah SWT yang senantiasa mencurahkan taufik, hidayah dan inayah-Nya sehingga penulis dapat menyusun Proyek Akhir ini. Sholawat serta salam semoga selalu tercurahkan kepada junjungan Rasulullah SAW, yang akan kita nantikan safa'atnya di hari kiamat nanti.

Proyek Akhir ini dibuat untuk memenuhi syarat kelulusan tahap Ahli Madya pada program studi D3 Teknologi Telekomunikasi Fakultas Ilmu Terapan Universitas Telkom. Judul yang dibahas dalam Proyek Akhir ini adalah **“Implementasi Sensor Maltrail dan Fail2Ban Untuk Mendeteksi dan Mencegah Serangan Malware Pada Jaringan Server Diskominfo Sumedang Dengan Push Notifikasi”**

Penulis menyadari bahwa Proyek Akhir ini masih jauh dari kesempurnaan, untuk itu saran dan kritik yang bertujuan membangun dari pembaca sangat diharapkan demi perbaikan di masa yang akan datang. Dengan segala kerendahan hati, penulis berharap semoga Proyek Akhir ini dapat bermanfaat bagi kita semua.

Wassalamualaikum warahmatullahi wabarakatuh.

Bandung, Juli 2021

A handwritten signature in black ink, appearing to read 'Ranung', followed by a horizontal line and the initials 'w. &'. The signature is written in a cursive, flowing style.

Penulis

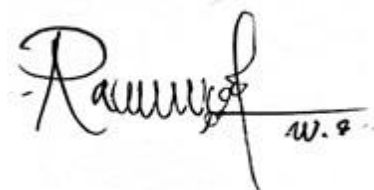
UCAPAN TERIMA KASIH

Dalam mengerjakan Proyek Akhir ini, tentu saja merupakan hal yang tidak mungkin apabila penulis berjalan sendiri tanpa berhubungan dengan pihak – pihak yang telah dengan ikhlas memberikan bimbingan, bantuan, dukungan, dan pengarahan baik dalam bentuk materil maupun moril. Karena itu, penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Allah S.W.T., berkat Rahmat dan Hidayah Nya, penulis diberikan kesehatan dan kelancaran dalam melaksanakan setiap proses demi proses dalam pengerjaan Proyek Akhir ini.
2. Kedua orang tua yang telah memberikan doa dan dukungan yang sangat besar sehingga penulis termotivasi untuk menyelesaikan Proyek Akhir ini.
3. Bapak Rohmat Tulloh, S.T., M.T. selaku pembimbing I yang telah memberikan arahan dan motivasi kepada penulis agar dapat mengerjakan Proyek Akhir ini dengan terencana dan sesuai dengan target.
4. Bapak Nugraha, S.Sos. M.Si, selaku pembimbing II yang telah memberikan dukungan dan bimbingan dalam penyelesaian Proyek Akhir.
5. Bapak Musni selaku pembimbing Lapangan yang telah memberikan dukungan dan bimbingan dalam penyelesaian Proyek Akhir.
6. Seluruh dosen D3 Teknologi Telekomunikasi selaku pengajar dan pendidik bagi penulis, karena berkat bantuan dan ilmu yang diberikan, sehingga penulis dapat menyelesaikan Proyek Akhir tepat waktu.
7. Keluarga dan teman-teman yang selalu memberikan dukungan, semangat, dan semua bantuannya yang tidak bisa dihitung dalam pengerjaan Proyek Akhir ini.

Kesempurnaan hanya milik Allah SWT. Oleh karena itu, penulis memohon maaf sebesar-besarnya apabila masih terdapat kekurangan serta kesalahan dalam penyelesaian Proyek Akhir ini. Semoga dapat bermanfaat bagi semua pihak.

Bandung, Juli 2021



Penulis

vii

DAFTAR ISI

LEMBAR PENGESAHAN	i
LEMBAR PERNYATAAN ORISINALITAS	ii
IDENTITAS BUKU	iii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR.....	vi
UCAPAN TERIMA KASIH	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Tujuan dan Manfaat	2
1.3 Rumusan Masalah.....	3
1.4 Batasan Masalah	3
1.5 Metodologi	3
1.6 Sistematika Penulisan	4
BAB II DASAR TEORI.....	5
2.1 <i>Malware (malicious)</i>	5
2.2 Malware Trail	5
2.2.1 Cara Kerja	6
2.2.2 Fitur.....	6
2.2.3 Database <i>malware</i> pada Maltrail	7
2.3 Fail2Ban	8
2.3.1 Fitur Fail2Ban	8
2.4 OS Debian Server	9
2.5 Python	9
2.6 Pcap	9
2.7 Telegram	9
2.8 Wireshark	10

2.9	Profile Diskominfo Sumedang.....	10
2.9.1	Visi dan Misi Diskominfo Sumedang.....	11
BAB III ANALISIS DAN PERANCANGAN.....		12
3.1	Analisis.....	12
3.1.1	Gambaran Sistem Saat ini	12
3.1.2	Analisis Kebutuhan	13
3.1.3	Kebutuhan Pengguna	14
3.2	Perancangan.....	14
3.2.1	Gambaran Sistem Usulan	14
3.2.2	Topologi Sistem.....	15
3.2.3	Spesifikasi Sistem.....	17
3.2.4	Flowchart Sistem.....	18
3.2.5	Fungsi dan Fitur Sistem	19
BAB IV IMPLEMENTASI DAN PENGUJIAN		28
4.1	Implementasi	28
4.1.1	Rencana Pengerjaan.....	28
4.1.2	Instalasi dan Konfigurasi Maltrail.....	28
4.1.3	Instalasi dan Konfigurasi Fail2Ban.....	29
4.1.4	Tampilan Dashboard Sistem Maltrail	33
4.1.5	Grafik Diagram Sistem Maltrail	34
4.1.6	Rules Pengintegrasian Fail2Ban dengan Telegram	35
4.1.7	Laporan Status Sistem ke Telegram Administrator	36
4.2	Pengujian.....	37
4.2.1	Tujuan Pengujian	37
4.2.2	Skenario Pengujian.....	37
4.3	Hasil Pengujian dan Pembahasan	38
4.3.1	Hasil Uji Coba Situs yang Terindikasi <i>Malware</i>	38
4.3.2	Hasil Banned Alamat IP oleh Fail2Ban	39
4.3.3	Hasil Laporan Blocking oleh Fail2Ban ke Telegram Administator	40
4.3.4	Hasil Tampilan <i>log</i> Rekapitulasi <i>Malware</i>	41
4.3.5	Hasil <i>DDos Attack</i>	42
4.3.6	Hasil <i>Scanning Port</i>	43
4.3.7	Hasil <i>Syn Flooding</i>	44

4.4	Hasil dan Analisis	45
4.4.1	Analisis Kinerja	49
4.4.2	Hasil Analisis Throughput	52
4.4.3	Hasil Pemantauan Serangan pada Server	52
4.4.4	Hasil Uji Optimasi Sistem.....	53
BAB V KESIMPULAN DAN SARAN.....		54
5.1	Kesimpulan.....	54
5.2	Saran	54
DAFTAR PUSTAKA		55

DAFTAR GAMBAR

Gambar 2.1 Database <i>Malware</i> pada Maltrail.....	7
Gambar 2.2 Fitur Fail2Ban.....	8
Gambar 2.3 Struktur Organisasi Diskominfo Sumedang	10
Gambar 3.1 Gambaran Sistem Saat ini.....	12
Gambar 3.2 Gambaran Sistem Usulan	14
Gambar 3.3 Topologi Sistem.....	15
Gambar 3.4 Penerapan Skema Topologi Jaringan Diskominfo Sumedang	16
Gambar 3.5 Flowchart Sistem	18
Gambar 3.6 Use Case Diagram Sistem Maltrail	21
Gambar 4.1 Perintah Instalasi <i>Software</i> Maltrail	28
Gambar 4.2 Konfigurasi Sistem Maltrail.....	29
Gambar 4.3 Instalasi dan Konfigurasi Fail2Ban.....	30
Gambar 4.4 Kode Program Deklarasi <i>regex-script</i> pada Fail2Ban.....	31
Gambar 4.5 Kode Program Deklarasi pelaporan <i>real-time</i> Fail2Ban ke Telegram	32
Gambar 4.6 Kode Program Deklarasi <i>variable konfigurasi</i>	32
Gambar 4.7 Tampilan Dashboard Sistem Maltrail.....	33
Gambar 4.8 Grafik Sistem Maltrail.....	34
Gambar 4.9 Rules Pengintegrasian Fail2Ban ke Telegram.....	35
Gambar 4.10 Laporan Status Sistem Ryzenware ke Telegram Administrator	36
Gambar 4.11 Hasil Uji Coba Lima Situs yang terindikasi <i>Malware</i>	38
Gambar 4.12 Hasil <i>Banned</i> Alamat IP oleh Fail2Ban.....	39
Gambar 4.13 Hasil Laporan Blocking oleh Fail2Ban ke Telegram Administrator	40
Gambar 4.14 Hasil Tampilan <i>Log</i> Rekapitulasi <i>Malware</i>	41
Gambar 4.15 Hasil Serangan <i>DDos Attack</i>	42
Gambar 4.16 Hasil <i>Scanning Port</i>	43
Gambar 4.17 Grafik Tingkat Ancaman <i>Scanning Port</i>	43
Gambar 4.18 Hasil Serangan <i>Syn Flooding</i>	44
Gambar 4.19 Tampilan Wireshark untuk menangkap paket data pada server.....	45
Gambar 4.20 Jumlah Throughput pada saat Trafik Normal	46

Gambar 4.21 Ping <i>Malware</i>	46
Gambar 4.22 Maltrail Mendeteksi Paket Data <i>Malware sinkhole confiker</i>	47
Gambar 4.23 Wireshark Menangkap Paket Enkripsi Berupa <i>Malware</i>	47
Gambar 4.24 Jumlah Throughput pada saat menangkap Paket Data <i>Malware</i>	48
Gambar 4.25 Jumlah Throughput pada saat Fail2Ban memblokir akses <i>Malware</i>	48
Gambar 4.26 Grafik Throughput pada saat Trafik Normal	49
Gambar 4.27 Grafik Throughput pada saat menangkap paket Data <i>Malware</i>	49
Gambar 4.28 Grafik Throughput pada saat Fail2Ban memblokir akses <i>Malware</i>	50
Gambar 4.29 CPU <i>Utilization</i> dengan Trafik Normal.....	51
Gambar 4.30 CPU <i>Utilization</i> dengan serangan <i>Malware</i>	51

DAFTAR TABEL

Tabel 3.1 Perangkat Keras yang di Gunakan.....	17
Tabel 3.2 Perangkat Lunak yang di Gunakan.....	17
Tabel 3.3 Daftar Fungsi pada Sistem yang akan di terapkan.....	19
Tabel 3.4 <i>Use case description login</i>	22
Tabel 3.5 <i>Use case description logout</i>	22
Tabel 3.6 <i>Use case description issues</i>	23
Tabel 3.7 <i>Use case description view documentation</i>	23
Tabel 3.8 <i>Use case description calendar</i>	23
Tabel 3.9 <i>Use case description threats</i>	24
Tabel 3.10 <i>Use case description events</i>	24
Tabel 3.11 <i>Use case description severity</i>	24
Tabel 3.12 <i>Use case description sources</i>	25
Tabel 3.13 <i>Use case description trails</i>	25
Tabel 3.14 <i>Use case description view threats per page</i>	25
Tabel 3.15 <i>Use case description filter search</i>	26
Tabel 3.16 <i>Use case description clear</i>	26
Tabel 3.17 <i>Use case description print</i>	26
Tabel 3.18 <i>Use case description tools</i>	27
Tabel 4.1 Pengujian dengan Sejumlah Domain	37
Tabel 4.2 Metode Penerapan Serangan	37
Tabel 4.3 Hasil Pengukuran Throughput.....	52
Tabel 4.4 Hasil Analisis Pemantauan pada Server	52
Tabel 4.5 Hasil Uji Optimasi Sistem.....	53

BAB I

PENDAHULUAN

1.1 Latar Belakang

Keamanan jaringan merupakan hal yang sangat penting, terutama di era teknologi sekarang ini. Banyak instansi atau organisasi yang tidak sadar dan tidak memperdulikan terkait masalah keamanan. Jika mendapat serangan dan terjadi kerusakan sistem, banyak biaya yang harus dikeluarkan untuk melakukan perbaikan sistem. Untuk itu sudah selayaknya investasi di bidang keamanan jaringan lebih diperhatikan, untuk mencegah kerusakan dari ancaman serangan yang saat ini semakin beragam. Terlebih lagi saat computer server terhubung dengan internet maka serangan pun akan semakin meningkat. Untuk itu perlu dipersiapkan keamanan untuk mengamankan dan meminimalisir ancaman pada jaringan dan server khusus penyedia jasa layanan internet [1].

Salah satu ancaman utama di Internet saat ini yaitu *software* berbahaya yang sering disebut sebagai *malware*. Faktanya, sebagian besar masalah keamanan Internet disebabkan oleh *malware*. *Malware* hadir dalam berbagai bentuk dan variasi, seperti *virus*, *worm*, *botnet*, *rootkit*, *trojan horse*, *spyware* dan program *denial tools* lainnya. Setiap tahun, banyak sistem komputer di seluruh dunia akan rusak akibat *malware*. Baru-baru ini melaporkan bahwa file, sistem, email dan server masing-masing telah terinfeksi oleh virus *Cookie.Weborama*, *Cookie.Rub*, dan *Exploit.Iframe*. Meskipun demikian pada tahun 2019 serangan oleh virus *Ransomware* dan *PowerShell* baru telah meningkat sebesar 118% dan 460% [2].

Perusahaan anti-virus Malwarebytes (2019) merilis laporan tahunan tentang kondisi *malware* diseluruh dunia dalam jurnal “2019 States of Malware”. Laporan tersebut menyatakan bahwa terdapat kurang lebih 750 juta serangan *malware* yang terdeteksi menyerang computer end-user (personal) sepanjang tahun 2017-2018 di seluruh dunia. Kemudian, terdapat kurang lebih 71 juta *malware* yang terdeteksi menyerang pengguna *business-user* (perusahaan/industry/lembaga) sepanjang tahun 2017-2018. Sayangnya, jumlah yang meningkat dan keragaman *malware* membuat teknik keamanan klasik, seperti pemindai *anti-virus* tidak efektif, dan sebagai konsekuensinya, jutaan host di Internet saat ini terinfeksi dengan perangkat lunak berbahaya [3].

Berdasarkan penelitian tersebut, dibutuhkan sistem keamanan jaringan untuk *monitoring* dan pencegahan dari serangan *malware* yang keluar masuk dan melintasi perangkat jaringan di Diskominfo Sumedang. Berdasarkan survei yang dilakukan, perangkat firewall yang berfungsi untuk memblokir serangan yang masuk ke server terkadang tidak bekerja secara maksimal, firewall tersebut justru memblokir jaringan untuk akses aplikasi pegawai, kemudian ada insiden mengenai file-file dan database di server Diskominfo sumedang yang tidak bisa diakses akibat dari malware. Melihat hal tersebut solusi lain sebagai sistem tambahan yang penerapannya di server yaitu dengan sensor Maltrail dan Fail2Ban untuk meminimalisir serangan yang tidak terblokir oleh firewall dan sebagai sistem monitoring aktivitas *malware* trafik pada jaringan server Diskominfo Sumedang.

Beberapa penelitian yang telah dilakukan berkaitan dengan sistem Maltrail sebagai sistem malware monitoring, yaitu [4]-[7].

1.2 Tujuan dan Manfaat

Adapun tujuan dari penulisan Proyek Akhir ini, sebagai berikut.

1. Dapat mendeteksi paket-paket yang masuk melalui jaringan server yang terindikasi dan terdeteksi sebagai malware.
2. Dapat melakukan *blocking* terhadap alamat IP dari sumber *malware*.
3. Dapat melaporkan status sistem dan IP yang di blokir melalui aplikasi Telegram secara *real-time*.
4. Dapat menampilkan hasil laporan pemindaian data *log traffic malware* melalui *browser* secara *realtime*.

Manfaat dari penulisan Proyek Akhir ini, sebagai berikut.

1. Dapat melakukan monitoring dan melakukan *blocking* terhadap paket-paket yang terindikasi dan terdeteksi sebagai *malware* secara otomatis pada jaringan server.
2. Dapat membantu administrator dalam me-monitoring jaringan secara *real-time*.
3. Dapat membantu administrator dalam mendapatkan hasil rekapitulasi pemindaian *log traffic malware*.
4. Dapat membantu sistem anti virus dalam mengidentifikasi jenis-jenis malware, jejak IP, nama domain, alamat URL atau IP.

1.3 Rumusan Masalah

Adapun rumusan masalah dari Proyek Akhir ini, sebagai berikut.

1. Apa saja fungsi dan fitur jaringan yang akan diterapkan pada sistem tersebut?
2. Bagaimana implementasi penggunaan sensor Maltrail dan Fail2Ban dalam mendeteksi dan mencegah serangan *malware* pada jaringan server dengan push notifikasi?
3. Bagaimana hasil dan analisa pengujian dengan sejumlah domain dan hasil pengujian dengan serangan selain *malware*?

1.4 Batasan Masalah

Adapun batasan masalah dari Proyek Akhir ini, sebagai berikut.

1. Server Maltrail dan Fail2Ban harus selalu dalam keadaan *running* dan terhubung dengan internet.
2. Sistem diakses melalui aplikasi *web browser*.
3. Sistem ini hanya bisa merekapitulasi *log traffic malware*.
4. Sistem yang digunakan hanya bersumber dari Github resmi Developer *software* Maltrail.

1.5 Metodologi

Adapun metodologi pada penelitian Proyek Akhir ini, sebagai berikut.

1. Studi Literatur

Studi literatur dilakukan dengan mengumpulkan literatur-literatur dan kajian-kajian yang berkaitan dengan permasalahan yang ada pada penelitian Proyek Akhir ini, baik berupa buku referensi, artikel, maupun *e-journal* yang berhubungan dengan cara kerja Maltrail menggunakan python, Fail2Ban, *OS Debian* dan implementasi sistem.

2. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan dengan mengumpulkan kebutuhan *software* dan cara kerja sistem dalam mengintegrasikan sistem yang akan digunakan dalam proyek akhir.

3. Perencanaan Sistem

Perencanaan sistem dilakukan dengan melakukan analisis sistem yang akan dibangun, dalam hal ini analisis sistem dalam mendeteksi dan mencegah serangan *malware* pada jaringan server menggunakan Maltrail dan Fail2Ban.

4. Implementasi

Langkah selanjutnya adalah implementasi sistem. Implementasi ini termasuk pembuatan, instalasi dan konfigurasi Maltrail, Fail2Ban, Telegram.

5. Pengujian

Pengujian dilakukan setelah pembuatan, instalasi dan konfigurasi *software* berjalan dengan baik. Kemudian dilakukan pengujian dengan beberapa metode serangan.

6. Pembuatan Laporan.

Pada langkah ini semua metode yang telah dilakukan, dibuat dokumentasi dari Proyek Akhir ini.

1.6 Sistematika Penulisan

Dalam penulisan Proyek Akhir terdiri atas lima bab, dengan keterangan sebagai berikut :

BAB I PENDAHULUAN

Pada bab ini berisi latar belakang, rumusan masalah, tujuan dan manfaat, batasan masalah, metodologi penelitian, serta sistematika penulisan.

BAB II DASAR TEORI

Pada bab ini membahas tentang teori pendukung pengerjaan Proyek Akhir, seperti *malware (malicious)*, cara kerja Maltrail dan Fail2Ban dan lain sebagainya.

BAB III ANALISIS DAN PERANCANGAN

Pada bab ini membahas tentang deskripsi Proyek Akhir, alur pengerjaan Proyek Akhir, analisis kebutuhan dan perancangan sistem yang akan diterapkan.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Pada bab ini membahas tentang implementasi sistem dan pengujian.

BAB V PENUTUP

Pada bab ini membahas tentang kesimpulan dari pengerjaan Proyek Akhir dan saran untuk pembaca yang akan mengambil penelitian dengan topik yang sama.

BAB II

DASAR TEORI

2.1 *Malware*

Malware (malicious software) adalah sebuah program yang berisi kode-kode yang ditambahkan, diubah, atau dihapus dari suatu sistem *software* untuk secara sengaja menyebabkan kerusakan atau menumbangkan fungsi sistem tersebut. Meskipun masalah *malware* memiliki sejarah yang panjang, sejumlah serangan *malware* baru-baru ini yang dipublikasikan secara luas dan tren ekonomi tertentu menunjukkan bahwa *malware* cepat menjadi masalah penting bagi industri, pemerintah, dan individu. *Malware* menjadi istilah umum yang mencakup *virus*, *trojan*, *spywares*, dan kode intrusif lainnya yang tersebar luas saat ini. Analisis *malware* adalah proses multi-langkah yang memberikan wawasan tentang struktur dan fungsi *malware*, memfasilitasi pengembangan penangkal racun. Secara teknis, *malicious traffic* ialah suatu kejadian abnormal pada lalu lintas jaringan dan merupakan perbuatan user yang tidak bertanggung jawab tanpa sepengetahuan pengguna komputer yang sah [4].

2.2 *Malware Trail*

Malware Trail (Maltrail) adalah sistem pendeteksi lalu lintas *malware* berbahaya yang menggunakan daftar hitam (*blacklists*) yang *repository*-nya disediakan oleh pihak ketiga yang berisi daftar *malware* berbahaya dan mencurigakan, serta jalur statis atau lokal yang disusun dari berbagai laporan anti- virus dan daftar yang ditentukan pengguna khusus. Maltrail didasarkan pada *traffic sensor–sensor–server–client architecture*. Sensor adalah komponen mandiri yang berjalan pada *monitoring mode* atau di mesin mandiri. Ia memantau paket-paket terdaftar sebagai *blacklist packet* (berupa nama domain, URL, atau alamat IP) yang melewati jaringan tersebut [8].

2.2.1 Cara Kerja

Pada umumnya Arsitektur Maltrail didasarkan pada Sensor> Server> Client [8]

1. *Sensor*

Sensor adalah komponen mandiri yang berjalan pada node pemantau yang bertugas memantau trafik yang lewat untuk jalur yang masuk daftar hitam (URL atau IP) pada jaringan. Sensor akan mengirimkan detail acara ke Server.

2. *Server*

Server merupakan komponen yang menyimpan semua peristiwa yang terjadi dalam periode (24h) dan memberikan data ke client untuk aplikasi web pelaporan. Data dikirim ke Client dalam potongan terkompresi, dan diproses secara berurutan.

3. *Client*

Client berupa web browser (IE, Chrome, Firefox, dll.). Semua peristiwa (yaitu entri log) dalam periode (24h) akan ditransfer client, dan aplikasi web pelaporan yang bertanggung jawab penuh atas bagian presentasi seperti ancaman, kejadian, sumber, dan jejak.

2.2.2 Fitur

Secara umum, Maltrail mempunyai fitur antara lain :

1. Menggunakan banyak daftar hitam publik

Daftar hitam publik yang digunakan berisi jalur mencurigakan seperti (alientvault, autoshun, badips, sblam, dll) dan jejak statis yang dikumpulkan dari berbagai laporan Anti Virus dan daftar yang ditetapkan pengguna khusus.

2. Memiliki jalur statis yang luas

Maltrail memiliki jalur statis yang luas untuk identifikasi (nama domain, URL, atau alamat IP). Sistem maltrail dapat menampilkan informasi berupa DNS dan WHOIS dari RIPE sebagai penyedia informasi.

3. Interface presentasi laporan memakai aplikasi web browser

Ketika data semua detail peristiwa (event) diterima oleh client, maka client akan mempresentasikan laporan data tersebut dengan memakai web browser. Mulai

dari waktu pertama kejadian, waktu kejadian terakhir, protocol yang dipakai, sumber alamat IP, dan alamat IP tujuan..

2.2.3 Database *Malware* pada Maltrail

```

root@Diskominfo:/home/ramawijayashiddiq/maltrail/trails/static/malware# ls
lms0rry.txt      apt_familiarfeeling.txt  clientmeshrat.txt      globeimposter.txt      neshuta.txt          sinkhole_certpl.txt
404.txt          apt_ferociouskitten.txt  clipea.txt             glupteba.txt           nestrat.txt          sinkhole_certtr.txt
44caliber.txt   apt_finfisher.txt        cloudatlas.txt         gobotkr.txt           netbounce.txt        sinkhole_changeip.txt
9002.txt        apt_flame.txt            cloudeye.txt          gobrut.txt            netsupport.txt       sinkhole_checkpoint.txt
a310.txt        apt_fruityarmor.txt      cloudstalker.txt       godlua.txt            netwalker.txt        sinkhole_cirtck.txt
aboc.txt        apt_gallmaker.txt        coabalbot.txt          godzilla.txt          netwire.txt          sinkhole_cnert.txt
absent.txt       apt_gamaredon.txt        cobaltstrike.txt       goldbrute.txt         neutrino.txt         sinkhole_collector.txt
ab.txt          apt_gaza.txt             cobalt.txt            goldenspy.txt         newddosbot.txt       sinkhole_conficker.txt
acbackdoor.txt  apt_glasses.txt          cobint.txt            gomorra.txt          newpos.txt           sinkhole_cryptolocker.txt
acridrain.txt   apt_goldenbird.txt       coderware_ransomware.txt  goomba.txt           nexelogger.txt       sinkhole_dnssinkhole.txt
activeagent.txt apt_goldenrat.txt        collector.txt          gookit.txt           nexus.txt            sinkhole_doombringer.txt
adroze.txt      apt_goldmouse.txt        cometer.txt           grandoreiro.txt       nigelthorn.txt      sinkhole_drweb.txt
advisorbob.txt  apt_gorgon.txt          conflicker.txt        grandt.txt           nionspy.txt         sinkhole_dynadot.txt
adwind.txt      apt_gothicpanda.txt      conti.txt             grandt.txt           nitro.txt            sinkhole_dyre.txt
adylkuzz.txt    apt_greenspot.txt        contopee.txt          gravityrat.txt        nivdort.txt          sinkhole_farsight.txt
adzok.txt       apt_gref.txt             corebot.txt           greamerat.txt        njrat.txt            sinkhole_fbiuzeus.txt
afrodita.txt    apt_greyenergy.txt       cotxrat.txt           grimagent.txt         nofersok.txt         sinkhole_fireeye.txt
agaadex.txt     apt_groundbait.txt      couponarific.txt      grooboor.txt         nombolqu.txt        sinkhole_fitsec.txt
agenttesla.txt  apt_group5.txt           crackonosh.txt        gruntsrager.txt      notrobin.txt         sinkhole_fmord.txt
aguijon.txt     apt_hackingteam.txt     criakl.txt            gtbot.txt            novahttp.txt         sinkhole_fraunhofer.txt
aldibot.txt     apt_hafnium.txt          cridex.txt            guildma.txt          novaloader.txt       sinkhole_gamaredon.txt
alina.txt       apt_hangover.txt         crilock.txt           guloader.txt         novalminer.txt       sinkhole_gameoverzeus.txt
allakore.txt    apt_hermit.txt           crismoonrat.txt       h1n1.txt             novel_bot.txt        sinkhole_georgiatech.txt
almsalocker.txt apt_higaisa.txt          cring.txt             hacked_apkpure.txt   novobot.txt          sinkhole_gladtech.txt
almsalreq.txt   apt_hogfish.txt          cryakl.txt            hacked_codecov.txt   novter.txt           sinkhole_hyas.txt
alpha.txt       apt_icefog.txt           crylocker.txt         hacked_fs.txt        nozelesn.txt         sinkhole_infosecjp.txt
alureon.txt     apt_indigozebra.txt      cryptobot.txt         hacked_healthcheck.txt  nucleartor.txt      sinkhole_kaspersky.txt
amadey.txt      apt_infy.txt             cryptofiled.txt       hacked_mint.txt      nuggetphantom.txt   sinkhole_kryptoslogic.txt
amavaldo.txt    apt_innput.txt           cryptofinite.txt      hacked_monero.txt    nworm.txt            sinkhole_menuspas.txt
amend_miner.txt apt_iron2.txt            cryptocoore.txt       hacked_gnappas.txt   nuget.txt            sinkhole_microsoft.txt
ammyrat.txt     apt_irontiger.txt        cryptodofense.txt     hacking_team.txt     nwt.txt              sinkhole_nolp.txt
anchor.txt       apt_ke3chang.txt         cryptolocker.txt      hamaetot.txt         nymaim.txt           sinkhole_nowdns.txt
android_aecard.txt apt_keyboy.txt          cryptoshield.txt      harnig.txt           nymeria.txt          sinkhole_oceanlotus.txt
android_actionspy.txt apt_kimsuky.txt        cryptoswall.txt       hawkball.txt         obliqueat.txt        sinkhole_opendns.txt
android_addr.txt apt_lazarus.txt         cryptotxxx.txt         hellokitty.txt       octopus.txt          sinkhole_paloalto.txt
android_ahmythrat.txt apt_leafminer.txt      outwall.txt           helolomy.txt         ododoc.txt           sinkhole_rsa.txt
android_alien spy.txt apt_longhorn.txt       cybergaterat.txt     hellokitty.txt       odyssey.txt          sinkhole_scarletshark.txt
android_andichap.txt apt_lotusblossom.txt   cypress.txt           hennsey.txt          oficla.txt           sinkhole_secureworks.txt
android_andorotat.txt apt_luckyat.txt        dionis.txt            hiddenbeer.txt       onkods.txt           sinkhole_shadowserver.txt
android_anubis.txt apt_lyceum.txt          dionis.txt            hiddenteat.txt      optima.txt           sinkhole_sinksns.txt
android_arspam.txt apt_machete.txt         dailyscriptlet.txt    hiddenteat.txt      orcusrat.txt         sinkhole_sobaken.txt
android_asacub.txt apt_magichound.txt
android_backflash.txt

```

Gambar 2.1 Database malware pada maltrail [8]

Gambar 2.1 merupakan tampilan database *malware* pada Maltrail, terdapat 1524 database dengan memanfaatkan daftar repositori dari berbagai perusahaan anti-virus international. Saat Maltrail berstatus *running*, secara otomatis sistem akan memperbarui daftar repositori *malware* yang berasal dari situs-situs anti-virus internasional tersebut. Mekanismenya, Maltrail akan men-download repositori tersebut dengan menggunakan metode RSS (Really Simple Syndication). Teknologi ini digunakan untuk mendapatkan update terbaru dari suatu website secara otomatis.

2.3 Fail2Ban

Fail2Ban merupakan *software* yang dapat memindai *file log* dan melarang alamat IP yang menunjukkan tanda-tanda paket berbahaya, banyaknya kegagalan kata sandi, dan lain-lain. Umumnya, Fail2Ban digunakan untuk menambahkan *firewall rules* untuk menolak alamat IP untuk jumlah waktu tertentu, meskipun tindakan pencegahan lainnya juga dapat dikonfigurasi lebih lanjut (misalnya mengirim email) juga dapat dikonfigurasi. Keluar dari kotak Fail2Ban hadir dengan filter untuk berbagai layanan (apache, kurir, ssh, dll) [9].

2.3.1 Fitur Fail2Ban

```
ramawijayashiddiq@Diskominfo:~
root@Diskominfo:/etc/fail2ban# ls
action.d fail2ban.conf fail2ban.d jail.conf jail.d paths-arch.conf paths-common.conf paths-debian.conf paths-opensuse.conf
root@Diskominfo:/etc/fail2ban# cd action.d
root@Diskominfo:/etc/fail2ban/action.d# ls
abuseipdb.conf      firewallcmd-multiport.conf      iptables-ipset-protocol.conf      nftables-common.conf      sendmail-whois.conf
apf.conf            firewallcmd-new.conf            iptables-multiport.conf           nftables-multiport.conf    sendmail-whois-ipjailmatches.conf
badips.conf         firewallcmd-rich-logging.conf    iptables-multiport-log.conf       nginx-block-map.conf       sendmail-whois-ipmatches.conf
badips.py           firewallcmd-rich-rules.conf      iptables-new.conf                 npf.conf                   sendmail-whois-lines.conf
blocklist_de.conf   helpers-common.conf             iptables-xt_recent-echo.conf      nsupdate.conf              sendmail-whois-matches.conf
bsd-ipfw.conf       hostsdeny.conf                  mail-buffered.conf                osx-afctl.conf             shorewall.conf
cloudflare.conf     ipfilter.conf                   mail.conf                          osx-ipfw.conf              shorewall-ipset-protocol.conf
complains.conf      ipfw.conf                       mail-whois-common.conf             pf.conf                     smtp.py
dshield.conf        iptables-allports.conf          mail-whois.conf                    route.conf                  symbiosis-blacklist-allports.conf
dummy.conf          iptables-common.conf            mail-whois-lines.conf              sendmail-buffered.conf     ufv.conf
firewallcmd-allports.conf iptables.conf                    mynetwatchman.conf                sendmail-common.conf       xarxf-login-attack.conf
firewallcmd-common.conf iptables-ipset-protocol.conf      netcaler.conf                     sendmail.conf
firewallcmd-ipset.conf iptables-ipset-protocol-allports.conf nftables-allports.conf       sendmail-geopj-lines.conf
root@Diskominfo:/etc/fail2ban/action.d# cd ..
root@Diskominfo:/etc/fail2ban# cd filter.d
root@Diskominfo:/etc/fail2ban/filter.d# ls
3proxy.conf      aasp.conf      drupal-auth.conf      ignorecommands      nsd.conf      gmail.conf      squid.conf
apache-auth.conf  asterisk.conf  ejabberd-auth.conf    kerio.conf           openhab.conf  recidive.conf   squirrelmail.conf
apache-badbots.conf botsearch-common.conf exim-common.conf      lighttpd-auth.conf   openwebmail.conf roundcube-auth.conf sshd.conf
apache-botsearch.conf common.conf      exim.conf             mongod-auth.conf     oracleins.conf  screensharingd.conf stunnel.conf
apache-common.conf counter-strike.conf exim-spam.conf         monit.conf           pam-generic.conf  selinux-common.conf suhosin.conf
apache-fakegooglebot.conf courier-auth.conf freeswitch.conf        murmur.conf          perdition.conf    selinux-ssh.conf  tine20.conf
apache-modsecurity.conf courier-smtp.conf froxlor-auth.conf      mysql-auth.conf       phpyadmin-syslog.conf sendmail-auth.conf uwimap-auth.conf
apache-nohome.conf cyrus-imap.conf groupoffice.conf       nagios.conf           php-url-fopen.conf  sendmail-reject.conf vsftpd.conf
apache-noscript.conf directadmin.conf gssftpd.conf           named-refused.conf    portentry.conf      sieve.conf         webmin-auth.conf
apache-overflows.conf domino-smtp.conf guacamole.conf         nginx-botsearch.conf  postfix.conf        postfix.conf       wuoftd.conf
apache-pass.conf  dovecot.conf   haproxy-http-auth.conf nginx-http-auth.conf  profptd.conf         sogo-auth.conf     xinetd-fail.conf
apache-shellshock.conf dropbear.conf   horde.conf             nginx-limit-req.conf  pure-ftpd.conf      solid-pop3d.conf    zoneminder.conf
root@Diskominfo:/etc/fail2ban/filter.d#
```

Gambar 2.2 Fitur Fail2Ban [9]

Gambar 2.2 merupakan tampilan fitur yang terdapat dalam direktori Fail2Ban, terdapat dua direktori yang sangat penting dalam penggunaan Fail2Ban yaitu direktori *action.d* dan *filter.d*. Direktori *action.d* merupakan direktori utama yang digunakan untuk seluruh rules pemblokiran yang akan diterapkan dengan mengeksekusi perintah pencegahan terhadap serangan. Sedangkan direktori *filter.d* digunakan untuk fitur tambahan supaya bisa terintegrasi dengan *tools* lain seperti *regex-script* yaitu *script* pemblokiran *malware*, *sendmail*, *zimbra mail server*, notifikasi telegram *messenger* dan hasil tampilan log pada SQL database.

2.4 OS Debian Server

OS Debian server digunakan untuk menjalankan segala macam aktivitas layanan server pada jaringan. Debian mengutamakan kestabilan dan kehandalan meskipun mengorbankan kemudahan dan kemutakhiran program. Debian menggunakan *.deb dalam paket instalasi programnya [10].

2.5 Python

Python adalah bahasa pemrograman yang memungkinkan anda bekerja lebih cepat dan mengintegrasikan sistem anda lebih efektif. Tidak seperti bahasa lain yang susah untuk dibaca dan dipahami, Python lebih menekankan pada keterbacaan kode agar lebih mudah untuk memahami sintaks [4].

2.6 Pcap

Pcap adalah modul ekstensi Python yang berinteraksi dengan libpcap paket capture library. Pcap memungkinkan skrip python untuk menangkap paket pada jaringan. Dalam administrasi jaringan komputer, pcap (*packet capture*) terdiri dari antarmuka pemrograman aplikasi (API) untuk menangkap lalu lintas jaringan. Libpcap menyediakan *packet capture* dan penyaringan, termasuk protokol analisis (*packet sniffers*), jaringan monitor, jaringan sistem deteksi gangguan, trafik Generator dan jaringan penguji [4].

2.7 Telegram

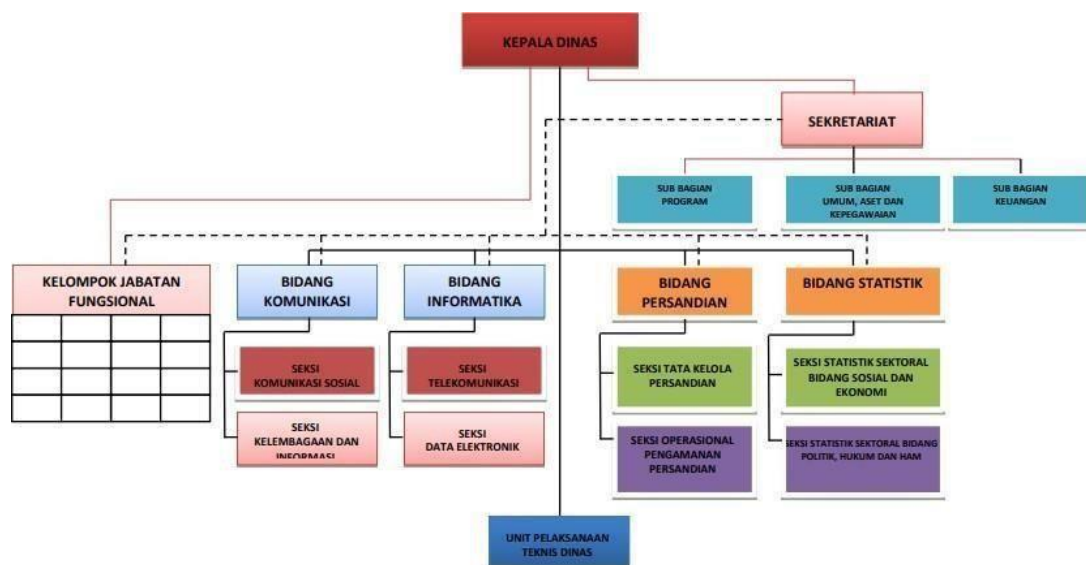
Telegram adalah platform IM (*instant messaging*) yang memungkinkan penggunaanya untuk bertukar pesan, menggunakan berbagai skema komunikasi (yaitu *one-to-one*, *one-to-many*, dan *many-to-many*), serta untuk melakukan panggilan suara, menggunakan berbagai teknik menjaga keamanan/privasi. Telegram mendukung pertukaran data berupa pesan teks (yang isinya teks biasa) dan pesan non-teks (dari jenis apapun, termasuk informasi kontak, koordinat geografis, dan fail jenis apa pun) [11].

2.8 Wireshark

Wireshark adalah tool yang ditujukan untuk penganalisaan paket data jaringan (Kurniawan, 2012). Wireshark disebut juga *Network packet analyzer* yang berfungsi menangkap paket-paket jaringan dan berusaha untuk menampilkan semua informasi sedetail mungkin. Sebenarnya *network packet analyzer* sebagai alat untuk memeriksa apakah sebenarnya terjadi di dalam jaringan baik kabel maupun *wireless*. Dengan adanya wireshark ini semua sangat dimudahkan dalam hal memonitoring dan menganalisa paket yang lewat di jaringan [12].

2.9 Profile Diskominfo Sumedang

Dinas Komunikasi dan Informatika, Persandian dan Statistik Kabupaten Sumedang merupakan unsur Pelaksana Pemerintahan Daerah Kabupaten Sumedang dalam Bidang komunikasi, informatika, persandian dan statistik. Berikut merupakan struktur organisasi Diskominfo Sumedang [13]



Gambar 2.3 Struktur Organisasi Diskominfo Sumedang [13]

Berdasarkan Gambar 2.3 struktur organisasi Diskominfo kota Sumedang dibagi menjadi 4 divisi kerja yaitu Bidang Komunikasi, Bidang Informatika, Bidang Persandian dan Bidang Statistik. Mempunyai tugas melaksanakan urusan pemerintahan yang menjadi kewenangan daerah dalam rangka pelaksanaan sebagian tugas Bupati di bidang komunikasi, informatika, persandian dan statistik sesuai dengan ketentuan dan peraturan perundang-undangan.

2.9.1 Visi dan Misi Diskominfo Sumedang

Secara umum visi dan misi dari Diskominfo Sumedang antara lain :

a. Visi

“Terwujudnya Masyarakat Sumedang yang Sejahtera, Agamis, Maju, Profesional, dan Kreatif (SIMPATI) Pada Tahun 2023” Sejahtera Masyarakatnya, Agamis Akhlaqnya, Maju Daerahnya, Profesional Aparaturnya dan Kreatif Ekonominya”

b. Misi

Misi Diskominfo Sumedang ialah sebagai berikut :

1. Memenuhi kebutuhan dasar secara mudah dan terjangkau untuk kesejahteraan masyarakat.
2. Memperkuat norma agama dalam tatanan kehidupan sosial masyarakat dan pemerintahan.
3. Mengembangkan wilayah ekonomi didukung dengan peningkatan infrastruktur dan daya dukung lingkungan serta penguatan budaya dan kearifan lokal.
4. Menata birokrasi pemerintah yang responsif dan bertanggung jawab secara profesional dalam pelayanan masyarakat; dan
5. Mengembangkan sarana prasarana dan sistem perekonomian yang mendukung kreativitas dan inovasi masyarakat Kabupaten Sumedang.

BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisis

3.1.1 Gambaran Sistem Saat ini

Pada Proyek Akhir ini dilakukan implementasi sensor Maltrail dan Fail2Ban untuk mendeteksi dan mencegah serangan *malware* pada jaringan server Diskominfo Sumedang dengan push notifikasi. Diilustrasikan sistem keamanan ini dilakukan sebagai langkah awal untuk mencegah terjadinya serangan *malware* di dalam jaringan internet yang dapat membahayakan keamanan pada jaringan server Diskominfo Sumedang.



Gambar 3.1 Gambaran sistem saat ini

Gambar 3.1 merupakan gambaran sistem yang dipersiapkan untuk sistem keamanan jaringan di Diskominfo Sumedang. Pada Gambar 3.1, sistem diletakkan untuk pendeteksi paket-paket data yang melewati server. Peran sistem maltrail sebagai sensor yang mengecek apakah terdapat paket *malware* yang melewati trafik pada jaringan tersebut atau tidak. Sedangkan Fail2Ban diletakkan berdampingan dengan maltrail, karena Fail2Ban berperan sebagai *firewall* yang memblokir, mencegah, dan

melarang *malware* yang lewat berdasarkan dari daftar *log* yang telah dicatat dan dideteksi oleh Maltrail.

3.1.2 Analisis kebutuhan

Sistem pendeteksi dan pencegah serangan *malware* memanfaatkan daftar public (*blacklist*) serta jejak statis yang dikumpulkan dari berbagai laporan AV (*Anti Virus*) untuk dijadikan parameter beroperasinya komponen pada sistem yang dibuat.

3.1.2.1 Analisis Kebutuhan Masukan

Pada sistem pendeteksi dan pencegah serangan *malware* ini dibutuhkan masukan sebagai berikut :

- a. Data masukan dari sensor Maltrail dan Fail2Ban yang menjadi parameter beroperasinya sistem monitoring di jaringan server Diskominfo Sumedang yang nantinya akan memberikan data detail peristiwa (*event*) kepada administrator.
- b. Data masukan dari server yang menjadi parameter beroperasinya client yang berperan sebagai presentasi pelaporan semua peristiwa yang terjadi dengan memakai *web browser*.

3.1.2.2 Analisis Kebutuhan Keluaran

Pada sistem ini dibutuhkan keluaran sebagai berikut :

- a. Dapat mendeteksi paket-paket yang terindikasi *malware*.
- b. Dapat mencegah *malware log* dengan alamat IP *blocking*.
- c. Dapat menampilkan *malware log* pada browser.
- d. Dapat menampilkan GUI berupa jejak IP, alamat IP, waktu kejadian, tingkat level berbahaya, dan protocol yang digunakan.
- e. Dapat melaporkan status sistem dan alamat IP *malware* ke aplikasi telegram.
- f. Dapat berjalan dengan baik keintegrasian seluruh komponen software yang digunakan untuk keberhasilan dalam mendeteksi dan mencegah serangan *malware*.

3.1.3 Kebutuhan Pengguna

Adapun kebutuhan pengguna sebagai berikut :

1. Laptop sebagai perangkat akses Maltrail, Fail2Ban dan Telegram.
2. *Web browser* sebagai penghubung untuk hasil detail *event* presentasi laporan *malware* pada Maltrail, mengaktifkan Fail2Ban dan Telegram.
3. Mengunduh paket aplikasi Maltrail <https://github.com/stamparm/maltrail>.

3.2 Perancangan

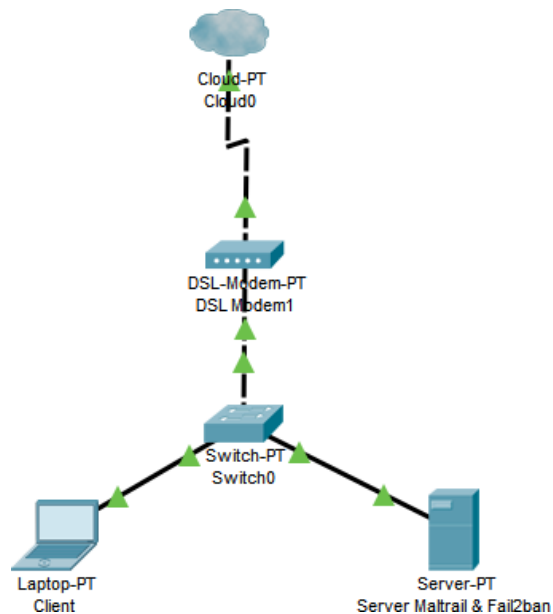
3.2.1 Gambaran Sistem Usulan



Gambar 3.2 Gambaran sistem usulan

Pada Gambar 3.2 menjelaskan sistem usulan dengan melakukan instalasi sensor Maltrail dan Fail2Ban, saat *cloud* menerima *packet request* dan mengirimnya ke client, paket tersebut akan melewati serangkaian tahap pemindaian paket oleh sensor Maltrail berdasarkan database yang tersedia pada *repository-nya*. Jika paket tersebut terindikasi *malware*, Sensor Maltrail akan melakukan pencatatan berupa *malware log*. Selanjutnya, *log* tersebut akan dieksekusi oleh sistem Fail2Ban dalam bentuk *IP blocking* atau pembatasan akses paket oleh sebuah alamat IP di dalam jaringan tersebut. Setelah alamat IP tersebut di blokir, Fail2Ban akan mengirimkan informasi pemblokirannya kepada administrator melalui aplikasi Telegram yang penempatannya di server.

3.2.2 Topologi Sistem



Gambar 3.3 Topologi sistem

Pada Gambar 3.3 tentang topologi sistem, yang menjelaskan cara kerja sistem ini. Ketika trafik melewati sensor Maltrail dan Fail2Ban, maka Maltrail akan mendeteksi trafik yang terindikasi sebagai malware melalui sensor. Kemudian sensor akan mengirimkan data ke server dan server akan menyimpan seluruh peristiwa yang terjadi.

3.2.2.1 Penerapan Skema Topologi Jaringan Diskominfo Sumedang

Pada Gambar 3.4 topologi tersebut dibuat berdasarkan topologi yang telah diterapkan di Diskominfo Sumedang, sekaligus implementasi sistem pendeteksi dan pencegah serangan *malware* yaitu Maltrail dan Fail2Ban.

3.2.3 Spesifikasi Sistem

Berikut ini adalah kebutuhan perangkat keras dan perangkat lunak yang dibutuhkan dalam Proyek Akhir ini.

3.2.3.1 Perangkat Keras

Adapun beberapa perangkat keras yang digunakan di sistem ini, yaitu :

Tabel 3.1 Perangkat Keras Yang di Gunakan

No.	Hardware	Unit	Keterangan
1.	Laptop	1	Penempatan Sensor, server, dan client
2.	Wifi	1	Untuk koneksi internet

3.2.3.2 Perangkat Lunak

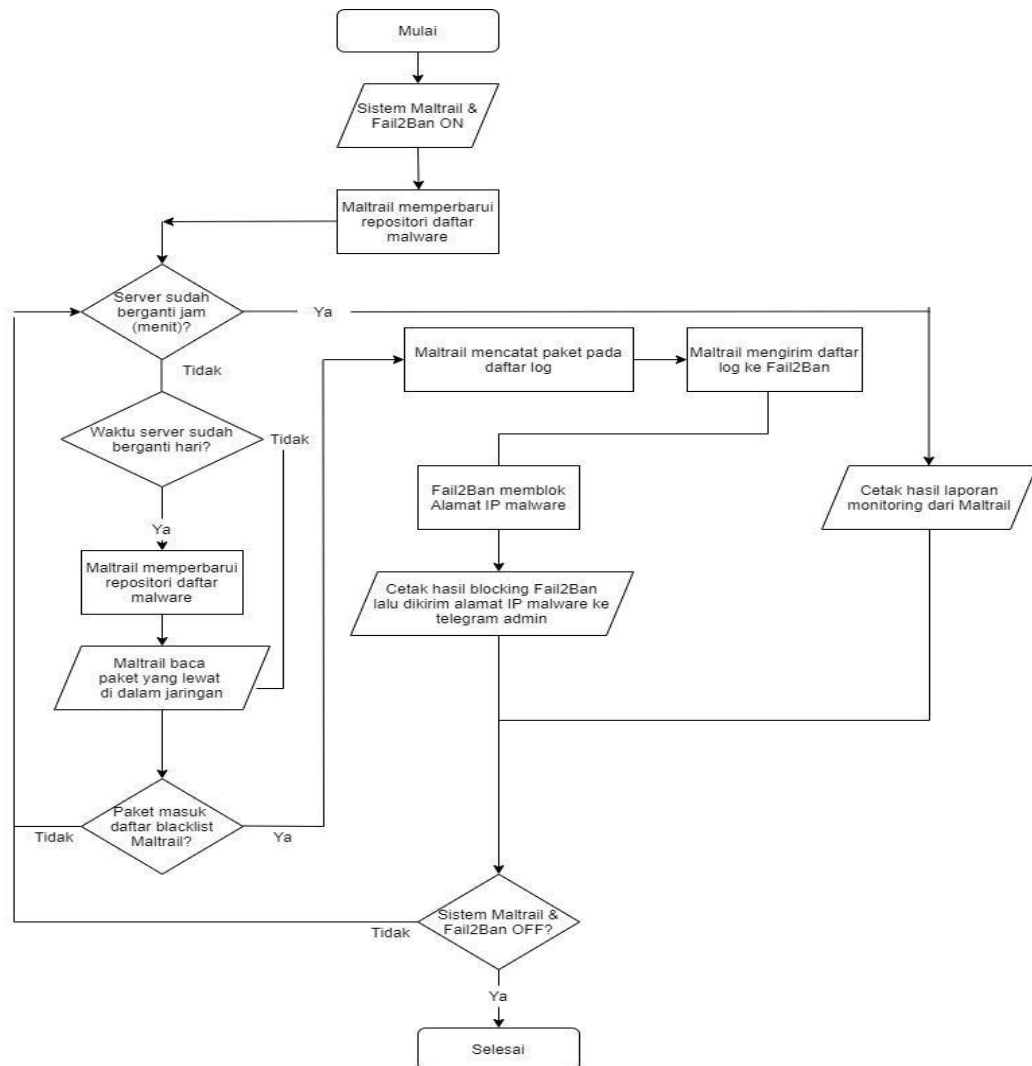
Adapun beberapa perangkat lunak yang digunakan di sistem ini, yaitu :

Tabel 3.2 Perangkat Lunak Yang di Gunakan

No.	Software	Spesifikasi	Keterangan
1.	OS Debian Server	Distro Debian Linux versi 10.8.0	Sensor dan Server
2.	Windows	Windows 10	Client
3.	Python	Python 3.8.2	Bahasa pemrograman yang dipakai
4.	Pcap	Modul ekstensi python	Untuk menangkap paket pada jaringan
5.	Maltrail	Versi 0.17.5	<i>Software</i> sistem pendeteksi <i>malware traffic</i> pada jaringan <i>server</i>
6.	Fail2ban	Versi 0.11.12	<i>Software</i> sistem pencegah dan pelindung server komputer dari serangan <i>brute-force</i> dan serangan malware
7.	Cisco Packet Tracer	Versi 8.0	<i>Software</i> untuk desain topologi jaringan
8.	Telegram	Versi 5.15.0	Aplikasi media sosial untuk pelaporan status keadaan sistem beserta IP yang diblokir

3.2.4 Flowchart Sistem

Berikut ini adalah *flowchart* sistem jaringan yang akan diterapkan dapat dilihat pada Gambar 3.5 untuk proses awal, Sistem Maltrail dan Fail2Ban dijalankan pada server.



Gambar 3.5 *Flowchart Sistem*

Pada gambar 3.5 terdapat sebuah kondisi pelaporan *log report* setiap satu jam sekali. Jika waktu menunjukkan pergantian jam (0 menit), sistem akan mencetak laporan hasil monitoring. Jika tidak, sistem akan melakukan ke tahap selanjutnya, yaitu pergantian hari. Sistem akan melakukan pembaruan repositori. Jika tidak, sistem Maltrail akan melakukan tugas utamanya untuk me-monitoring setiap paket-paket data yang melewati jaringan. Terdapat kondisi, jika sistem Maltrail mendeteksi adanya paket *malware*, sistem Maltrail akan mencatatnya berupa *log*, kemudian mengirimkan log tersebut ke sistem Fail2Ban untuk dieksekusi dengan metode alamat IP *blocking*. Kemudian, alamat IP yang mengandung *malware* tersebut dilaporkan ke administrator melalui Telegram. Selanjutnya, jika server masih aktif, sistem akan melakukan *looping* atau kembali ke tahap awal, yaitu pengecekan jam pada server. Sedangkan, Jika server dalam keadaan nonaktif, sistem secara otomatis akan nonaktif.

3.2.5 Fungsi dan fitur sistem

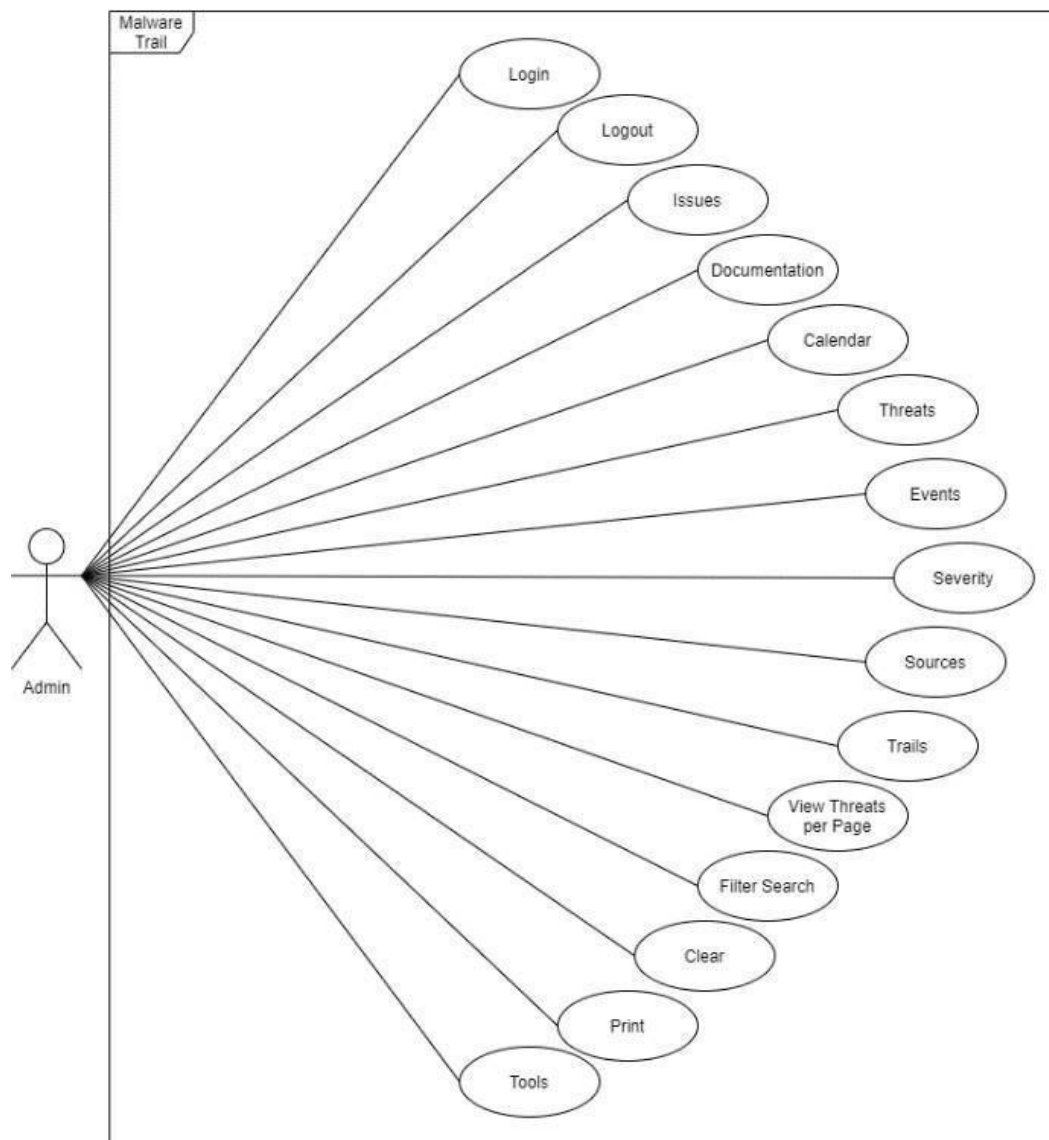
Berikut ini adalah fungsi dan fitur utama yang ada pada sistem ini sebagai pemantauan dan pemblokiran paket-paket data yang melewati trafik pada jaringan oleh Maltrail dan Fail2Ban. Fungsi dan fitur yang akan diterapkan dapat dilihat pada Tabel berikut [8]

Tabel 3.3 Daftar fungsi pada sistem yang akan diterapkan [8]

No.	Nama Fungsi	Deskripsi
1.	<i>Login</i>	Pengguna dapat melakukan <i>login</i> untuk mendapatkan hak akses dalam menggunakan fitur-fitur lainnya
2.	<i>Logout</i>	Pengguna dapat melakukan <i>logout</i> untuk menghentikan seluruh hak akses
3.	<i>Issues</i>	Pengguna dapat mengakses halaman <i>issues</i> atau persoalan-persoalan mengenai <i>software</i> ini di situs Github Maltrail
4.	<i>Documentation</i>	Pengguna dapat mengakses halaman terkait informasi <i>software</i> Maltrail di situs Github Maltrail
5.	<i>Calendar</i>	Pengguna dapat memilih data yang akan ditampilkan pada tabel berdasarkan tanggal

6.	<i>Threats</i>	Sistem akan menampilkan grafik lingkaran dari daftar klasterisasi ancaman <i>malware</i> yang telah dideteksi oleh sensor Maltrail
7.	<i>Events</i>	Sistem dapat menampilkan grafik garis dari daftar klasterisasi jumlah kejadian <i>malware</i> yang telah dideteksi oleh sensor Maltrail
8.	<i>Severity</i>	Sistem dapat menampilkan grafik lingkaran dari daftar klasterisasi tingkat ancaman <i>malware</i> yang telah dideteksi oleh sensor Maltrail
9.	<i>Sources</i>	Sistem dapat menampilkan grafik <i>bar</i> dari daftar klasterisasi sumber-sumber alamat IP <i>malware</i> yang telah dideteksi oleh sensor Maltrail
10.	<i>Trails</i>	Sistem dapat menampilkan grafik lingkaran dari daftar klasterisasi sumber-sumber alamat IP atau DNS <i>malware</i> yang telah dideteksi oleh sensor Maltrail
11.	<i>View Threats per Page</i>	Sistem dapat menyortir jumlah <i>threats</i> yang ditampilkan per satu halaman dashboard
12.	<i>Filter Search</i>	Sistem dapat mencari data dengan menyeleksi berdasarkan kata kunci
13.	<i>Clear</i>	Sistem dapat menghapus kata kunci yang telah diketik
14.	<i>Print</i>	Sistem dapat mencetak keseluruhan data <i>threats</i> dengan format fail berbentuk PDF
15.	<i>Tools</i>	Sistem dapat mengedit IP <i>aliases</i> atau menghapus seluruh <i>threats log</i> pada <i>local storage</i> Maltrail

Setelah semua fungsi dan fitur diterapkan dan telah dideskripsikan, langkah selanjutnya yaitu membuat *use case diagram* (Gambar 3.6). *Use case diagram* dibuat untuk menggambarkan tingkah laku sistem yang akan dibuat, serta mengetahui hubungan antar *use cases*, actor dan sistem.



Gambar 3.6 *Use case diagram* sistem Maltrail [8]

Pada Gambar 3.6 diilustrasikan dengan *user* yang berperan sebagai administrator yaitu memiliki hak akses untuk menjalankan fungsi-fungsi sistem Maltrail yang telah dideskripsikan sebelumnya. Secara garis besar, pengguna dapat melakukan pemantuan sistem. Pengguna juga dapat menambahkan akun dengan memasukkan *username* dan *password* baru pada sistem Maltrail di CLI (*command line interface*). Setelah *use case diagram* selesai dibuat, tahap selanjutnya yaitu mendeskripsikan setiap *uses cases* dengan cara membuat *use case description*. *Use case description* setiap *use cases* dapat dilihat pada Tabel sebagai berikut :

Tabel 3.4 *Use case description login [8]*

<i>Use Case</i>	<i>Login</i>
<i>Actor</i>	Administrator
<i>Objective</i>	Melakukan login agar dapat mengakses semua fitur yang ada pada sistem
<i>Precondition</i>	<ol style="list-style-type: none"> 1. Pengguna belum melakukan <i>login</i> 2. Pengguna sudah memiliki akun untuk login
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Pengguna memasukkan <i>username</i> dan <i>password</i> 2. Pengguna menekan tombol <i>login</i> 3. Sistem melakukan validasi <i>username</i> dan <i>password</i>
<i>Alternative Flow</i>	Jika <i>username</i> atau <i>password</i> tidak valid, akan muncul <i>Alert</i>
<i>Postcondition</i>	Pengguna masuk ke dalam sistem serta dapat menggunakan fitur-fitur yang ada

Tabel 3.5 *Use case description logout [8]*

<i>Use Case</i>	<i>Logout</i>
<i>Actor</i>	Administrator
<i>Objective</i>	Melakukan <i>logout</i> untuk keluar dari sistem
<i>Precondition</i>	Pengguna telah melakukan <i>login</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Pengguna menekan <i>button logout</i> Sistem mencabut hak akses pengguna untuk menggunakan fitur-fitur sistem
<i>Alternative Flow</i>	-
<i>Postcondition</i>	Pengguna keluar dari sistem

Tabel 3.6 *Use case description issues [8]*

<i>Use Case</i>	<i>issues</i>
<i>Actor</i>	Administrator
<i>Objective</i>	Mengakses situs linimasa <i>help</i> atau bantuan dari <i>developer</i> sistem Maltrail
<i>Precondition</i>	<ol style="list-style-type: none"> 1. Pengguna telah melakukan <i>login</i> 2. Pengguna membutuhkan bantuan terkait <i>bug</i> pada sistem
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Pengguna menekan <i>button issues</i> 2. Sistem akan mengalihkan laman ke situs https://github.com/stamparm/maltrail/issues
<i>Alternative Flow</i>	-
<i>Postcondition</i>	Pengguna mendapat bantuan dari <i>developer</i>

Tabel 3.7 *Use case description view documentation [8]*

<i>Use Case</i>	<i>Documentation</i>
<i>Actor</i>	Administrator
<i>Objective</i>	Mengakses halaman terkait informasi <i>software</i> Maltrail di situs Github Maltrail
<i>Precondition</i>	Pengguna telah melakukan <i>login</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Pengguna menekan <i>button documentation</i> 2. Sistem akan mengalihkan laman ke situs https://github.com/stamparm/maltrail
<i>Alternative Flow</i>	-
<i>Postcondition</i>	Pengguna membuka situs Maltrail di GitHub

Tabel 3.8 *Use case description calendar [8]*

<i>Use Case</i>	<i>Calendar</i>
<i>Actor</i>	Administrator
<i>Objective</i>	Memilih data yang akan ditampilkan pada tabel berdasarkan tanggal yang dipilih
<i>Precondition</i>	Pengguna telah melakukan <i>login</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Pengguna menekan tombol ikon kalender 2. Pengguna memilih tanggal yang diinginkan 3. Sistem menampilkan <i>tabel threats</i> berdasarkan tanggal yang dipilih
<i>Alternative Flow</i>	-
<i>Postcondition</i>	Pengguna dapat melihat data yang ditampilkan berdasarkan tanggal yang dipilih

Tabel 3.9 *Use case description threats [8]*

<i>Use Case</i>	<i>Threats</i>
<i>Actor</i>	Administrator
<i>Objective</i>	Menampilkan grafik dari daftar klasterisasi ancaman <i>malware</i> yang telah dideteksi oleh sensor Maltrail
<i>Precondition</i>	Pengguna telah melakukan <i>login</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>threats</i> 2. Sistem menampilkan informasi <i>threats</i> dalam bentuk grafik lingkaran
<i>Alternative Flow</i>	-
<i>Postcondition</i>	Pengguna dapat melihat grafik lingkaran dari daftar klasterisasi ancaman <i>malware</i> yang telah dideteksi

Tabel 3.10 *Use case description events [8]*

<i>Use Case</i>	<i>Events</i>
<i>Actor</i>	Administrator
<i>Objective</i>	Menampilkan grafik garis dari daftar klasterisasi jumlah kejadian <i>malware</i> yang telah dideteksi oleh sensor Maltrail
<i>Precondition</i>	Pengguna telah melakukan <i>login</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>events</i> 2. Sistem menampilkan informasi <i>events</i> dalam bentuk grafik garis
<i>Alternative Flow</i>	-
<i>Postcondition</i>	Pengguna dapat melihat grafik garis dari daftar klasterisasi ancaman <i>malware</i> yang telah dideteksi

Tabel 3.11 *Use case description severity [8]*

<i>Use Case</i>	<i>Severity</i>
<i>Actor</i>	Administrator
<i>Objective</i>	Menampilkan grafik lingkaran dari daftar klasterisasi tingkat ancaman <i>malware</i> yang telah dideteksi oleh sensor Maltrail
<i>Precondition</i>	Pengguna telah melakukan <i>login</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>severity</i> 2. Sistem menampilkan informasi <i>threats</i> dalam bentuk grafik garis
<i>Alternative Flow</i>	-
<i>Postcondition</i>	Pengguna dapat melihat grafik lingkaran dari daftar klasterisasi ancaman <i>malware</i> yang telah dideteksi

Tabel 3.12 *Use case description sources [8]*

<i>Use Case</i>	<i>Sources</i>
<i>Actor</i>	Administrator
<i>Objective</i>	Menampilkan grafik <i>bar</i> dari daftar klasterisasi sumber-sumber alamat IP <i>malware</i> yang telah dideteksi oleh sensor Maltrail
<i>Precondition</i>	Pengguna telah melakukan <i>login</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>sources</i> 2. Sistem menampilkan informasi <i>threats</i> dalam bentuk grafik <i>bar</i>
<i>Alternative Flow</i>	-
<i>Postcondition</i>	Pengguna dapat melihat grafik lingkaran dari daftar klasterisasi ancaman <i>malware</i> yang telah dideteksi

Tabel 3.13 *Use case description trails [8]*

<i>Use Case</i>	<i>Trails</i>
<i>Actor</i>	Administrator
<i>Objective</i>	Menampilkan grafik lingkaran dari daftar klasterisasi sumber-sumber alamat IP atau DNS <i>malware</i> yang telah dideteksi oleh sensor Maltrail
<i>Precondition</i>	Pengguna telah melakukan <i>login</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>trails</i> 2. Sistem menampilkan informasi <i>threats</i> dalam bentuk grafik lingkaran
<i>Alternative Flow</i>	-
<i>Postcondition</i>	Pengguna dapat melihat grafik lingkaran dari daftar klasterisasi ancaman <i>malware</i> yang telah dideteksi

Tabel 3.14 *Use case description view threats per page [8]*

<i>Use Case</i>	<i>View Threats per Page</i>
<i>Actor</i>	Administrator
<i>Objective</i>	Menyortir jumlah <i>threats</i> yang ditampilkan per satu halaman <i>dashboard</i>
<i>Precondition</i>	Pengguna telah melakukan <i>login</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>View per Page</i> 2. Pengguna menentukan jumlah <i>threats</i> yang ditampilkan antara 10, 25, 50 atau 100 baris per halaman
<i>Alternative Flow</i>	Jika jumlah <i>threats</i> kurang dari yang pengaturan ditentukan, sistem akan menampilkan <i>threats</i> dengan jumlah <i>threats</i> seadanya
<i>Postcondition</i>	Sistem berhasil menampilkan jumlah <i>threats</i> sesuai dengan halaman yang ditentukan

Tabel 3.15 *Use case description filter search [8]*

<i>Use Case</i>	<i>Filter Search</i>
<i>Actor</i>	Administrator
<i>Objective</i>	Mencari data <i>threats</i> dengan menyeleksi berdasarkan kata kunci
<i>Precondition</i>	<ol style="list-style-type: none"> 1. Pengguna telah melakukan <i>login</i> 2. Sistem telah mendata sejumlah <i>threats</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Pengguna memasukkan kata kunci <i>threats</i> yang akan dicari pada <i>filter search</i> 2. Pengguna menekan tombol <i>search</i> 3. Sistem akan menampilkan data <i>threats</i> yang dicari
<i>Alternative Flow</i>	Jika data yang dicari tidak ada, tabel akan kosong
<i>Postcondition</i>	Sistem berhasil menampilkan data <i>threats</i> yang dicari

Tabel 3.16 *Use case description clear [8]*

<i>Use Case</i>	<i>Clear</i>
<i>Actor</i>	Administrator
<i>Objective</i>	Menghapus kata kunci <i>threats</i> yang telah diketik
<i>Precondition</i>	<ol style="list-style-type: none"> 1. Pengguna telah melakukan <i>login</i> 2. Pengguna telah mengetik kata kunci yang dicari pada <i>filter search</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>clear</i> 2. Sistem menghapus kata kunci yang terdapat pada <i>filter search</i>
<i>Alternative Flow</i>	-
<i>Postcondition</i>	Kata kunci berhasil dihapus oleh sistem

Tabel 3.17 *Use case description print [8]*

<i>Use Case</i>	<i>Print</i>
<i>Actor</i>	Administrator
<i>Objective</i>	Mencetak keseluruhan data <i>threats</i> dengan format fail berbentuk PDF
<i>Precondition</i>	Pengguna telah melakukan <i>login</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>print</i> 2. Sistem menampilkan tabel yang akan di-<i>print</i> 3. Pengguna menekan tombol CTRL-P pada <i>keyboard</i> untuk <i>print</i> laman <i>browser</i> 4. Pengguna memilih opsi untuk menyimpan dokumen dalam bentuk .PDF Sistem mengonversi dan menyimpan data pada tabel <i>threats</i> menjadi dokumen dalam bentuk PDF
<i>Alternative Flow</i>	Jika data yang ingin dicetak tidak ada, maka dokumen yang disimpan akan berbentuk kosong atau <i>blank space</i>
<i>Postcondition</i>	Sistem berhasil menyimpan data tabel <i>threats</i> dalam bentuk PDF

Tabel 3.18 *Use case description tools [8]*

<i>Use Case</i>	<i>Tools</i>
<i>Actor</i>	Administrator
<i>Objective</i>	Mengedit IP <i>aliases</i> atau menghapus seluruh <i>threats log</i> pada <i>local storage</i> sistem Maltrail
<i>Precondition</i>	Pengguna telah melakukan <i>login</i>
<i>Main Flow</i>	1. Pengguna menekan tombol <i>tools</i> Sistem menamplkan opsi IP <i>aliases</i> atau <i>clear local storage</i> terhadap sistem Maltrail
<i>Alternative Flow</i>	-
<i>Postcondition</i>	Pengguna dapat mengedit IP <i>aliases</i> atau menghapus seluruh <i>threats log</i> pada <i>local storage</i> sistem Maltrail

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

4.1.1 Rencana Pengerjaan

Pada tahap ini akan dilakukan implementasi yang merupakan tahap instalasi dan konfigurasi terhadap perangkat-perangkat sistem keamanan jaringan sesuai dengan spesifikasi sistem yang akan diterapkan pada jaringan server Diskominfo Sumedang.

4.1.2 Instalasi dan konfigurasi Maltrail

Pada tahap instalasi dan konfigurasi maltrail, dilakukan berdasarkan proses sistem perancangan yang telah dilakukan. Terdapat dua sistem yang dibuat, yaitu sistem Maltrail untuk monitoring serta pencatatan *malware* log yang terdeteksi, dan sistem Fail2Ban untuk mengeksekusi berupa IP *blocking* terhadap *malware* tersebut. Kode program yang diterapkan dimulai dengan melakukan instalasi *software* Maltrail pada server Debian 10.

```
sudo apt-get install git python-pcap python-setuptools  
sendemail  
  
git clone https://github.com/stamparm/maltrail.git
```

Gambar 4.1 Perintah instalasi *software* Maltrail

Pada Gambar 4.1 terdapat perintah-perintah yang dilakukan untuk penginstalan *software* Maltrail pada server. Paket pertama yang diinstal yaitu *python-pcap* dan *python-setuptools*. Kemudian, paket utama *software* Maltrail di-*clone* atau diunduh dari situs resmi Maltrail di GitHub. *Software* Maltrail diletakkan pada direktori maltrail, secara otomatis akan memudahkan administrator untuk menjalankan Maltrail.

```
sudo nano maltrail.conf

#di dalam file, diubah baris

admin: RANDOM_STRING_OF_CHARACTERS changeme!
#menjadi

echo -n 'PASSWORDNYA' | sha256sum | cut -d '"' -f 1
#dimana USERNAME adalah nama pengguna yang akan
ditambahkan

sudo python sensor.py server.py
```

Gambar 4.2 Konfigurasi sistem Maltrail

Pada Gambar 4.2 USERNAME adalah nama pengguna yang akan ditambahkan, serta RANDOM_STRING_OF_CHARACTERS adalah *string* yang disalin dari *output* dari perintah *echo*. Kode *string* didapat dari hasil *hash code generate* kata 'PASSWORDNYA' dengan panjang *digest*-nya sebesar 256 *bit*. Setelah itu, Maltrail dijalankan, sehingga administrator dapat masuk ke sistem dengan menggunakan *username* dan *password* yang telah didaftarkan.

4.1.3 Instalasi dan konfigurasi Fail2Ban

Pada Gambar 4.3 *software* Fail2Ban diinstal di server. Fail jail ini diterapkan agar *software* Fail2Ban dengan *software* Maltrail dapat saling terintegrasi. Selanjutnya, yaitu membuat dan mengonfigurasi fail baru bernama *jail.local* agar konfigurasi tambahan ini dapat menimpa konfigurasi *default* yang ada di *jail.conf*.

```

apt-get install fail2ban
nano /etc/fail2ban/jail.local

#tambahkan script
[Maltrail]
enabled = true
filter = maltrail
logpath = /var/log/maltrail/*.log
maxretry = 1
bantime = 180
action = iptables-allports[name=Maltrail, protocol=all]
        telegram

```

Gambar 4.3 Instalasi dan konfigurasi Fail2Ban

Pada Gambar 4.3 teknik ini digunakan untuk menghindari dari masalah error pada sistem terhadap penggabungan konfigurasi saat *compiling*. Keterangan *jail script* (*rule* atau peraturan) pada Fail2Ban. Berikut keterangan dari *jail local.script* yaitu sebagai berikut :

Tabel 4.1 Keterangan dari *jail.local script* [9]

No.	Nama Fungsi	Deskripsi
1.	Maltrail	Penggabungan <i>jail script</i> (<i>rule</i>) baru
2.	Filter	Nama filter yang akan digunakan oleh <i>jail script</i> (<i>rule</i>) untuk mendeteksi kecocokan paket. Setiap pengecekan yang berhasil dideteksi dengan filter ini, akan menambah <i>max-retry</i> di dalam <i>jail</i> tersebut
3.	Logpath	Alamat ke fail <i>log</i> yang disediakan sebagai <i>object filtering</i>
4.	Maxretry	Jumlah paket kecocokan (nilai penghitung) yang memicu <i>blocking</i> pada alamat IP
5.	Bantime	Durasi (detik) alamat IP yang akan diblokir
6.	Action	Suatu aksi yang mendefinisikan satu atau beberapa perintah yang dieksekusi

```

nano /etc/fail2ban/filter.d/maltrail.conf
#tambahan script

[Definition]
failregex =

    (.*) debian <HOST> \d+ 10\.10\.10\.1.*(andromeda)
    (.*) debian <HOST> \d+ 10\.10\.10\.1.*(suspicious)
    (.*) debian <HOST> \d+ 10\.10\.10\.1.*(attacker)
    (.*) debian <HOST> \d+ 10\.10\.10\.1.*(scanner)
    (.*) debian <HOST> \d+ 10\.10\.10\.1.*(reputation)
    (.*) debian <HOST> \d+ 10\.10\.10\.1.*(phishing)
    (.*) debian <HOST> \d+ 10\.10\.10\.1.*(spammer)
    (.*) debian <HOST> \d+ 10\.10\.10\.1.*(proxy)
    (.*) debian <HOST> \d+ 10\.10\.10\.1.*(user agent)
    (.*) debian <HOST> \d+ 10\.10\.10\.1.*(user agent)
    (.*) debian <HOST> \d+ 10\.10\.10\.1.*(port scanning)
    (.*) debian <HOST> \d+ 10\.10\.10\.1.*(conficker)
    (.*) debian <HOST> \d+ 10\.10\.10\.1.*(domain)
    (.*) (<HOST>(?:[0-9]{1,3}\.){3}[0-9]{1,3}) \- ((?:[0-9]{1,3}\.){3}[0-9]{1,3}(|,))* \- (.*?)

ignoreregex =

```

Gambar 4.4 Kode program deklarasi *regex-script* pada Fail2Ban

Gambar 4.4 merupakan tampilan *regex script* kategori-kategori *packet log* yang akan diterapkan dan di-*banned* oleh sistem Fail2Ban. Jika *packet log* tersebut memiliki kecocokan kategori yang sama serta *max-retry* yang melebihi dari batas yang telah ditentukan, sistem akan otomatis memblokir alamat IP tersebut.

```

nano /etc/fail2ban/action.d/telegram.conf
[Definition]

actionstart =
/etc/fail2ban/scripts/fail2ban-telegram.sh start
actionstop =
/etc/fail2ban/scripts/fail2ban-telegram.sh stop
actionban =
/etc/fail2ban/scripts/fail2ban-telegram.sh ban <ip>
actionunban =
/etc/fail2ban/scripts/fail2ban-telegram.sh unban <ip>
[Init]
init = 123

```

Gambar 4.5 Kode program deklarasi pelaporan *real-time* Fail2Ban ke Telegram

Gambar 4.5 terdapat empat perintah yang menjadi pelaporan secara *real-time* ke Telegram *bot* Ryzenware. Untuk fungsi *start*, jika status sistem Ryzenware *running*, perintah *start* akan melaporkan ke Telegram *bot* dengan pesan bahwa sistem telah aktif. Jika status Ryzenware *stop*, perintah *stop* akan melaporkan ke Telegram *bot* dengan pesan bahwa sistem telah nonaktif. Jika status sistem Ryzenware telah mem-*banned* sebuah alamat IP, perintah *ban* akan melaporkan ke Telegram *bot* dengan pesan bahwa sistem telah mem-*banned* sebuah alamat IP. Kemudian, jika status sistem Ryzenware telah meng-*unbanned* sebuah alamat IP, perintah *unban* akan melaporkan ke Telegram *bot* dengan pesan bahwa sistem telah meng-*unbanned* sebuah alamat IP.

```

# Send notification
function send_msg {

    apiToken=1134726865:AAEzE74DHXTuBQunJw0HVBOHC8kGAOyk2DE
    chatId=(869498795 -1001420319743)
    url="https://api.telegram.org/bot$apiToken/sendMessage"
    for room in ${chatId[@]}; do
        curl -s -X POST $url -d chat_id=$room -d text="$1" done
    exit
}

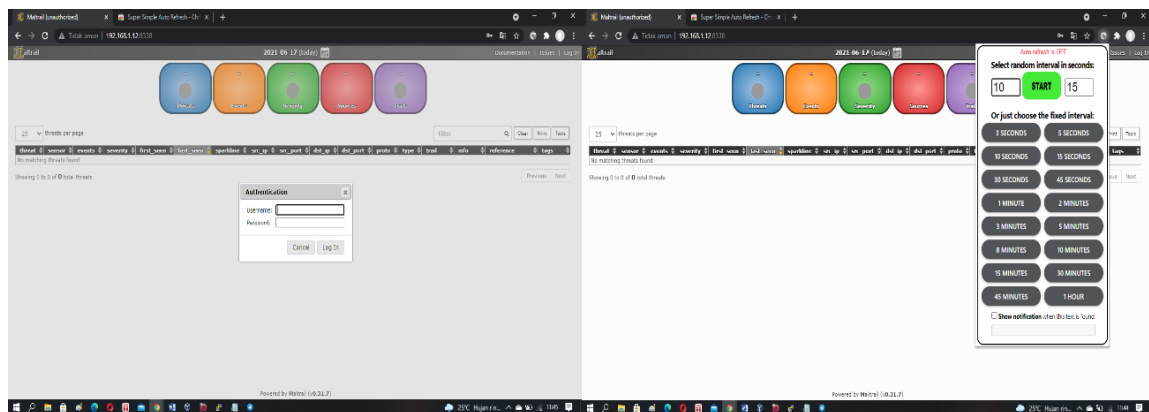
```

Gambar 4.6 Kode program deklarasi *variable konfigurasi*

Gambar 4.6 untuk mengintegrasikan sistem Ryzenware dengan notifikasi di aplikasi Telegram, diperlukan sebuah fungsi untuk koneksi Application Programming Interface (API). Pada Gambar 4.6, sistem ini dihubungkan dengan beberapa *stakeholder*, yaitu API Token, Chat ID, *url* untuk pengiriman pesan beserta isi pesannya..

4.1.4 Tampilan Dashboard Sistem Maltrail

Berikut merupakan tampilan awal sistem Maltrail berupa halaman login. Administrator harus memasukkan *username* dan *password* untuk mendapatkan akses fitur-fitur yang ada pada sistem. Gambar 4.7a merupakan menunjukkan halaman *login* dan Gambar 4.7b menunjukkan opsi *browser extension auto-refresh Google Chrome*.



(a)

(b)

Gambar 4.7 Tampilan Dashboard Sistem Maltrail, (a) halaman *login*, (b) *extension auto-refresh*

Pada Gambar 4.7 setelah melakukan *login*, administrator akan masuk ke halaman *dashboard sistem Maltrail*. Administrator dapat me-monitoring paket-paket *malware* yang telah dideteksi oleh *software* ini pada halaman *dashboard*. Selain itu, administrator dapat melihat sejumlah informasi *malware*, seperti sumber dan tujuan alamat IP penyerang, sumber dan tujuan port yang dilewati, tanggal dan waktu kejadian, jumlah kejadian penyebaran *malware* yang dilakukan oleh penyerang, protokol yang digunakan oleh penyerang, identitas *malware*, level ancaman *malware*, dan referensi *malware database*.

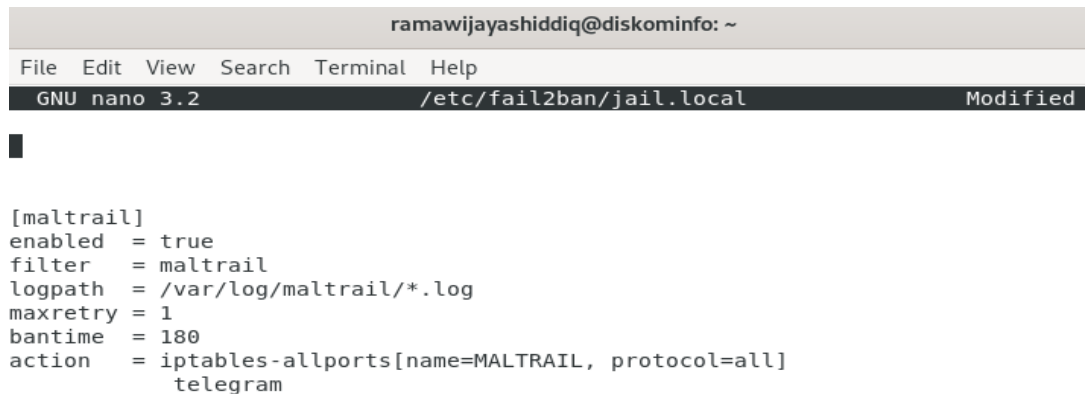
4.1.5 Grafik Diagram Sistem Maltrail



Gambar 4.8 Grafik Sistem Maltrail, (a) grafik *threats*, (b) grafik *events*, (c) grafik *severity* (d) grafik *sources*, (e) grafik *trails*

Pada Gambar 4.8 untuk melihat informasi detail *malware* yang berhasil dideteksi, administrator dapat menekan menu-menu yang ada pada *dashboard* dengan menampilkan grafik yang berisi *threats* (tingkat ancaman), *events* (jumlah kejadian), *severity* (tingkat kesulitan), *sources* (sumber *malware*), dan *trails* (Jejak malware melewati alamat IP). Selain itu juga ada fitur untuk mencetak tabel data *malware* dan melihat hasil rekapitulasi *log malware*.

4.1.6 Rules Pengintegrasian Fail2Ban dengan Telegram



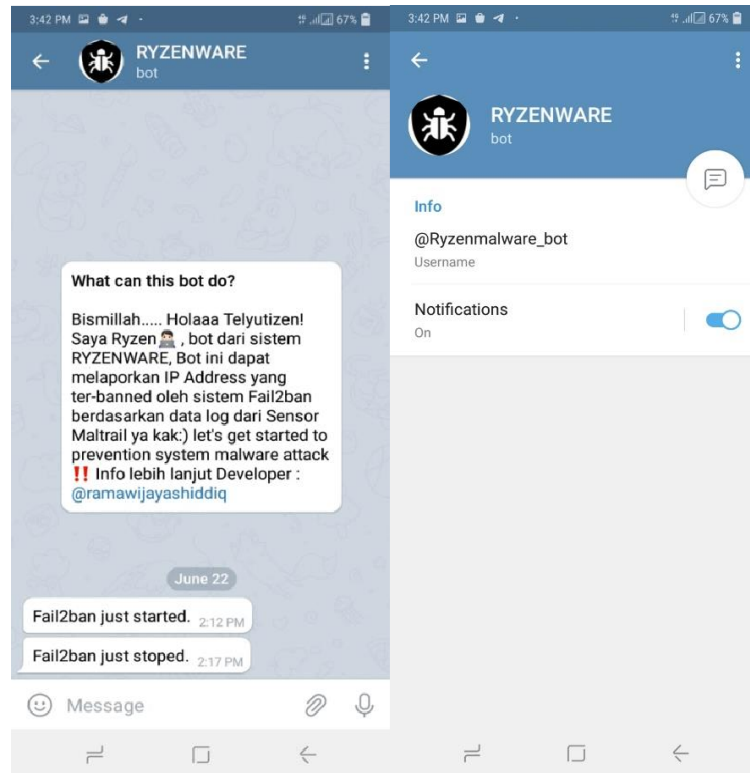
```
ramawijayashiddiq@diskominfo: ~
File Edit View Search Terminal Help
GNU nano 3.2 /etc/fail2ban/jail.local Modified

[maltrail]
enabled = true
filter = maltrail
logpath = /var/log/maltrail/*.log
maxretry = 1
bantime = 180
action = iptables-allports[name=MALTRAIL, protocol=all]
        telegram
```

Gambar 4.9 Rules pengintegrasian Fail2Ban dengan Telegram

Pada Gambar 4.9 menampilkan sistem Fail2Ban telah berjalan pada server Debian. Pada sistem ini terdapat rule “jail” yang berarti dapat membatasi akses alamat IP berdasarkan *log* yang dikirim oleh Maltrail. Pada pengujian ini, *rule-rule* Fail2Ban yang diterapkan, yaitu dengan *maxretry=1*, artinya sebanyak satu kali jumlah *events* transferisasi data dengan situs *malware*. Selain itu, diterapkan *bantime=180* yang artinya lama waktu di-*banned* sebuah alamat IP yang bertransferisasi data dengan situs *malware* selama 180 detik. Jika sudah melewati 180 detik, alamat IP tersebut akan di-*unbanned* oleh Fail2Ban, tergantung kebutuhan.

4.1.7 Laporan Status Sistem ke Telegram Administrator



Gambar 4.10 Laporan status sistem Ryzenware ke Telegram administrator

Pada Gambar 4.10 merupakan tampilan *field-chat bot* Ryzenware pada Telegram yang berfungsi untuk mengabarkan kondisi keadaan sistem apakah sedang aktif atau nonaktif. Jika status sistem nonaktif, server akan melaporkan ke Telegram administrator dengan *statement* “Fail2Ban just stoped”. Sedangkan, jika status sistem aktif, server akan melaporkan ke Telegram administrator dengan *statement* “Fail2Ban just started”.

4.2 Pengujian

4.2.1 Tujuan Pengujian

Pada tahap ini akan dilakukan pengujian agar implementasi yang telah dilakukan sesuai dengan tujuan yang diharapkan. Pengujian dilakukan berupa pengujian penggunaan Maltrail dan Fail2Ban sebagai sistem *malware traffic monitoring* dan pencegah serangan terhadap aktivitas server menggunakan client.

4.2.2 Skenario Pengujian

Pengujian tersebut dilakukan dengan melakukan akses *browsing* ke beberapa situs atau alamat IP yang diidentifikasi sebagai *malware* berdasarkan sumber data perusahaan-perusahaan anti-virus. Kemudian dilakukan pengujian dari serangan lain ke server untuk mengetahui apakah sensor Maltrail dan Fail2Ban bisa mendeteksi dan mencegah atau tidak. Hasilnya akan ditampilkan melalui situs web Maltrail berupa tabel data dan grafik. Berikut tabel dari sejumlah sampel *domain* dan pengujian serangan selain *malware* akan di uji pada tabel 4.1 dan tabel 4.2.

Tabel 4.1 Pengujian dengan sejumlah domain

No.	Domain	Alamat IP	Alamat DNS	Sampel Domain
1	facebook	202.124.205.117	Facebook.com	Legal (non-malware)
2	hhgg3	23.105.122.40	hhgg3.com	Illegal (malware)
3	morphed	63.251.235.82	morphed.ru	Illegal (malware)
4	trololo	192.165.67.186	trololo.cu.cc	Illegal (malware)
5	fqbtpchkp	216.218.185.162	fqbtpchkp.org	Illegal (malware)
6	-	136.161.101.53	-	Illegal (malware)

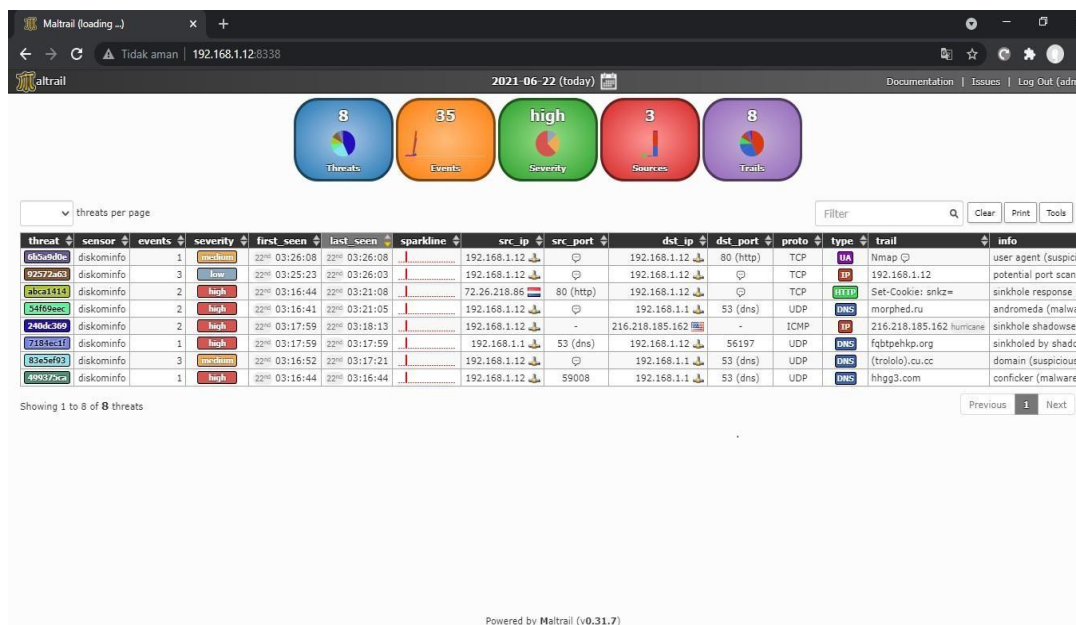
Tabel 4.2 Metode penerapan serangan

Nama Komponen	Metode
Pengujian <i>DDos attack</i>	<i>Blackbox</i>
<i>Scanning Port</i>	<i>Blackbox</i>
<i>Syn flooding</i>	<i>BlackBox</i>

Mengacu pada tabel 4.1 dan 4.2, pengujian *malware* dilakukan dengan melakukan akses *browsing* terhadap domain-domain tersebut. Hasilnya, domain *morphed.ru*, *trololo.cu.cc*, *hhgg3.com*, *fqbtpchkp.org* telah diduga dan dideteksi sebagai *malware*. Kemudian, untuk situs *facebook.com* diduga aman dan tidak terdeteksi sebagai *malware*. Kemudian untuk pengujian *DDos attack*, *Scanning port* dan *Syn flooding* dilakukan untuk mengetahui apakah sistem tersebut bisa mendeteksi dan memblokir serangan selain *malware* atau tidak dengan memaksimalkan sistem supaya mencegah aktivitas mencurigakan yang masuk ke server. Sebagaimana dengan hasil laporan Maltrail pada Gambar 4.11.

4.3 Hasil Pengujian dan Pembahasan

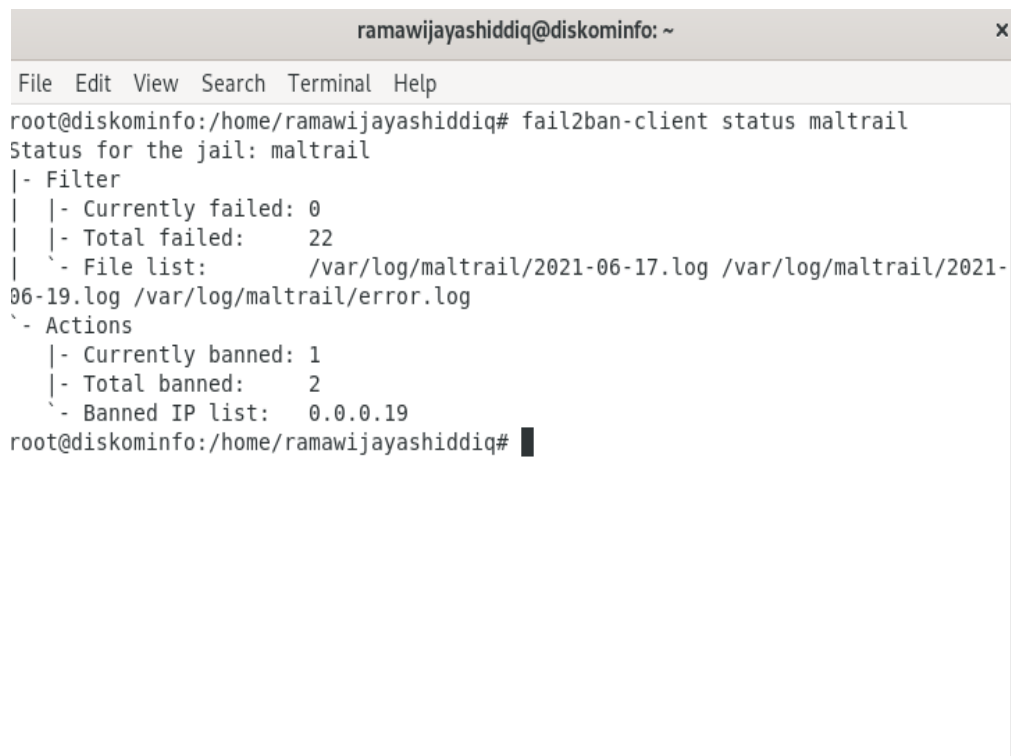
4.3.1 Hasil uji coba situs yang terindikasi *malware*



Gambar 4.11 Hasil uji coba situs yang terindikasi *malware*

Berdasarkan Gambar 4.11 client mencoba mengakses salah satu *domain* pada tabel 4.1, yaitu *fqbtpchkp.org* dengan *access request* secara terus-menerus. Domain tersebut diakses secara *flooding* hingga 35 kali *events* (kejadian), sehingga kejadian tersebut melebihi batas *access limit* atau batas *maxretry* yang ditetapkan pada *rule* Fail2Ban, yaitu sebanyak 1 kali *events*.

4.3.2 Hasil banned alamat IP oleh Fail2Ban



```
ramawijayashiddiq@diskominfo: ~
File Edit View Search Terminal Help
root@diskominfo:/home/ramawijayashiddiq# fail2ban-client status maltrail
Status for the jail: maltrail
|- Filter
| |- Currently failed: 0
| |- Total failed: 22
| `-- File list: /var/log/maltrail/2021-06-17.log /var/log/maltrail/2021-
06-19.log /var/log/maltrail/error.log
`- Actions
   |- Currently banned: 1
   |- Total banned: 2
   `-- Banned IP list: 0.0.0.19
root@diskominfo:/home/ramawijayashiddiq#
```

Gambar 4.12 Hasil banned alamat IP oleh Fail2Ban

Gambar 4.12 merupakan hasil pengecekan status dari *rule* Fail2Ban terhadap Maltrail. Berdasarkan Gambar 4.12, terdeteksi bahwa alamat IP client berusaha untuk melakukan akses dan transferisasi dengan *malware domain* lebih dari batas akses *maxretry* yang telah ditentukan pada *rule* Fail2Ban, sehingga alamat IP client, yaitu 0.0.0.19 di-*banned* oleh Fail2Ban selama waktu *bantime* yang diatur pada *rule* Fail2Ban, yaitu selama 180 detik. Sistem Fail2Ban ini tidak secara langsung mem-*banned* IP client yang melakukan transaksi *malware*, tetapi menunggu terlebih dahulu *log malware* dari Maltrail. Kemudian jika client mengakses lebih dari batas *maxretry*, maka IP tersebut otomatis ter-banned.

4.3.3 Hasil laporan blocking oleh Fail2Ban ke Telegram Administrator



Gambar 4.13 Hasil laporan *blocking* oleh Fail2Ban ke Telegram administrator

Gambar 4.13 merupakan hasil laporan *blocking* yang dilakukan oleh Fail2Ban ke Telegram administrator. Berdasarkan Gambar 4.13, Laporan *ban* tersebut dikirimkan dengan waktu *banning* selama 180 detik atau 3 menit. Hasil *banned* tersebut juga dilaporkan via Bot Telegram administrator secara *real-time*. Dan jika waktu *banned* terhadap alamat IP tersebut sudah mencapai 180 detik, maka Fail2Ban secara otomatis akan meng-*unbanned*-nya.

4.3.4 Hasil tampilan *log* rekapitulasi *malware*

```
ramawijayashiddiq@diskominfo: ~  
"2021-06-19 09:07:13.333804" diskominfo 192.168.1.1 53 192.168.1.11 41024 UDP DN  
S (fgbtpehkp).org "sinkholed by shadowserver (malware)" (heuristic)  
"2021-06-19 09:07:17.504260" diskominfo 192.168.1.1 53 192.168.1.11 41024 UDP DN  
S (fgbtpehkp).org "sinkholed by shadowserver (malware)" (heuristic)  
"2021-06-19 09:07:18.111043" diskominfo 192.168.1.11 - 216.218.185.162 - ICMP IP  
216.218.185.162 "sinkhole shadowserver (malware)" (static)  
"2021-06-19 09:07:19.111525" diskominfo 192.168.1.11 - 216.218.185.162 - ICMP IP  
216.218.185.162 "sinkhole shadowserver (malware)" (static)  
"2021-06-19 09:07:20.112628" diskominfo 192.168.1.11 - 216.218.185.162 - ICMP IP  
216.218.185.162 "sinkhole shadowserver (malware)" (static)  
"2021-06-19 09:07:21.114400" diskominfo 192.168.1.11 - 216.218.185.162 - ICMP IP  
216.218.185.162 "sinkhole shadowserver (malware)" (static)  
"2021-06-19 09:07:22.116638" diskominfo 192.168.1.11 - 216.218.185.162 - ICMP IP  
216.218.185.162 "sinkhole shadowserver (malware)" (static)  
"2021-06-19 09:07:23.118593" diskominfo 192.168.1.11 - 216.218.185.162 - ICMP IP  
216.218.185.162 "sinkhole shadowserver (malware)" (static)  
"2021-06-19 09:07:24.119818" diskominfo 192.168.1.11 - 216.218.185.162 - ICMP IP  
216.218.185.162 "sinkhole shadowserver (malware)" (static)  
"2021-06-19 09:07:25.122215" diskominfo 192.168.1.11 - 216.218.185.162 - ICMP IP  
216.218.185.162 "sinkhole shadowserver (malware)" (static)  
"2021-06-19 09:07:26.122827" diskominfo 192.168.1.11 - 216.218.185.162 - ICMP IP  
216.218.185.162 "sinkhole shadowserver (malware)" (static)  
"2021-06-19 09:07:50.761500" diskominfo 192.168.1.1 53 192.168.1.11 58464 UDP DN  
S (fgbtpehkp).org "sinkholed by shadowserver (malware)" (heuristic)  
"2021-06-19 09:07:55.011250" diskominfo 192.168.1.11 - 216.218.185.162 - ICMP IP  
216.218.185.162 "sinkhole shadowserver (malware)" (static)  
"2021-06-19 09:07:56.019305" diskominfo 192.168.1.11 - 216.218.185.162 - ICMP IP  
216.218.185.162 "sinkhole shadowserver (malware)" (static)  
"2021-06-19 09:07:57.019432" diskominfo 192.168.1.11 - 216.218.185.162 - ICMP IP  
216.218.185.162 "sinkhole shadowserver (malware)" (static)  
"2021-06-19 09:12:21.446202" diskominfo 192.168.1.1 53 192.168.1.11 41931 UDP DN  
S (fgbtpehkp).org "sinkholed by shadowserver (malware)" (heuristic)  
"2021-06-19 09:12:25.850313" diskominfo 192.168.1.1 53 192.168.1.11 41931 UDP DN  
S (fgbtpehkp).org "sinkholed by shadowserver (malware)" (heuristic)  
"2021-06-19 09:12:26.206281" diskominfo 192.168.1.11 - 216.218.185.162 - ICMP IP  
216.218.185.162 "sinkhole shadowserver (malware)" (static)  
"2021-06-19 09:12:27.267356" diskominfo 192.168.1.11 - 216.218.185.162 - ICMP IP  
216.218.185.162 "sinkhole shadowserver (malware)" (static)  
"2021-06-19 09:12:28.268754" diskominfo 192.168.1.11 - 216.218.185.162 - ICMP IP  
216.218.185.162 "sinkhole shadowserver (malware)" (static)  
"2021-06-19 09:12:29.271161" diskominfo 192.168.1.11 - 216.218.185.162 - ICMP IP  
216.218.185.162 "sinkhole shadowserver (malware)" (static)  
"2021-06-19 09:12:30.273218" diskominfo 192.168.1.11 - 216.218.185.162 - ICMP IP  
216.218.185.162 "sinkhole shadowserver (malware)" (static)
```

Gambar 4.14 Tampilan *log* rekapitulasi *malware*

Gambar 4.14 merupakan hasil tampilan *log* rekapitulasi *malware*, setelah semua data aktivitas *malware* pada jaringan server tercatat pada *log* Maltrail. Log tersebut bisa dikonversi ke dalam bentuk file.csv, dengan adanya *log malware* yang berisi tentang identitas *malware* secara detail, maka akan mempermudah dan membantu administrator untuk mengetahui seberapa besar intensitas *malware-malware* yang melewati trafik pada jaringan server Diskominfo Sumedang.

4.3.5 DDos attack

DDos attack (Distributed Denial of service) merupakan jenis serangan terhadap sebuah komputer atau server di dalam jaringan internet dengan cara menghabiskan sumber (resource).

```
ramawijayas@cyberhacking: ~  
  
      (.) < (.) > (.) =  
      \__ \__ \__ Ready To Send  
  
=====
```

Created By: TheTechHacker

```
=====
```

If You Use too much bytes
You're Internet might get a bit slow

```
=====
```

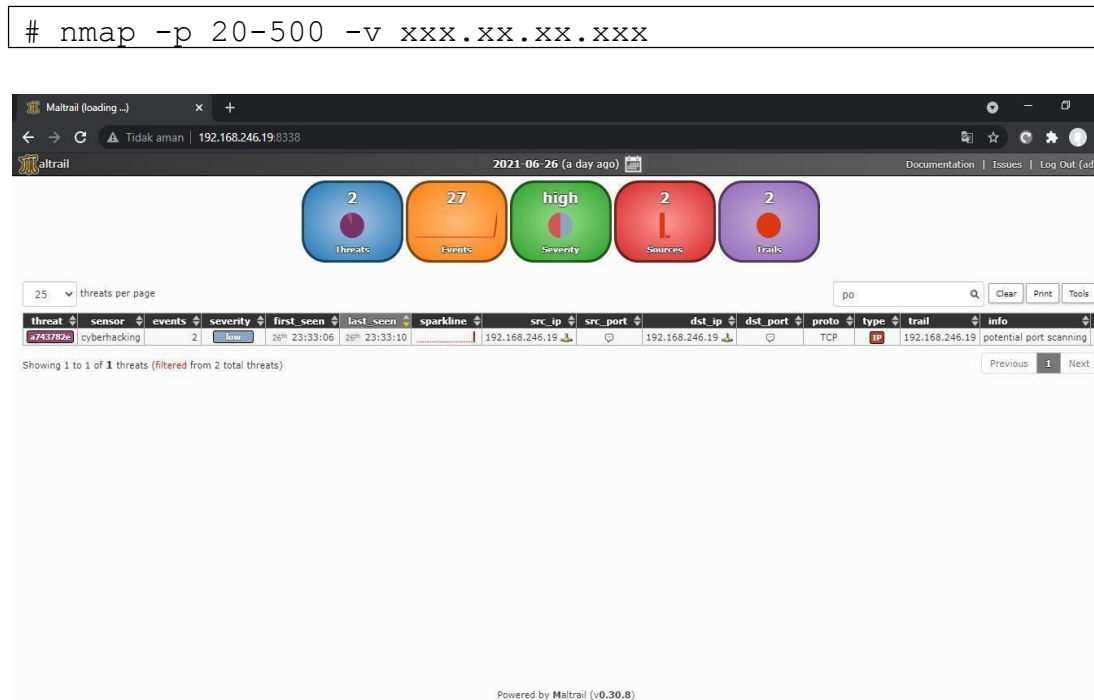
bytes> 100000
IP> 192.168.246.19
port> 8338

Gambar 4.15 Hasil Serangan *DDos attack*

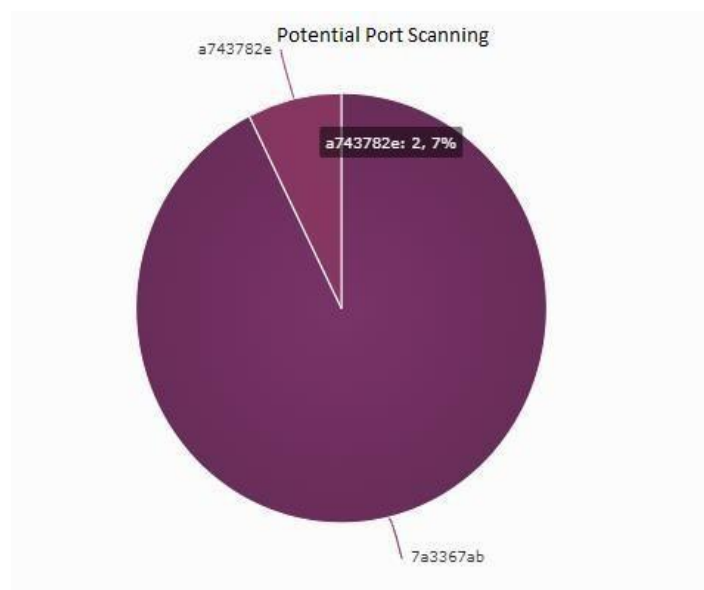
Hasil dari serangan *DDos* berhasil dilakukan ke server dengan mengirimkan 10000 bytes. Pengujian ini dilakukan guna untuk menghabiskan semua bandwidth yang tersedia antara target dengan jaringan internet.

4.3.6 Scanning Port

Scanning port merupakan suatu aktivitas untuk mencari informasi Port-Port yang terbuka dan dipakai dalam sebuah server. Pengujian ini dilakukan dengan cara memasukkan perintah nmap pada terminal.



Gambar 4.16 Hasil *scanning port*

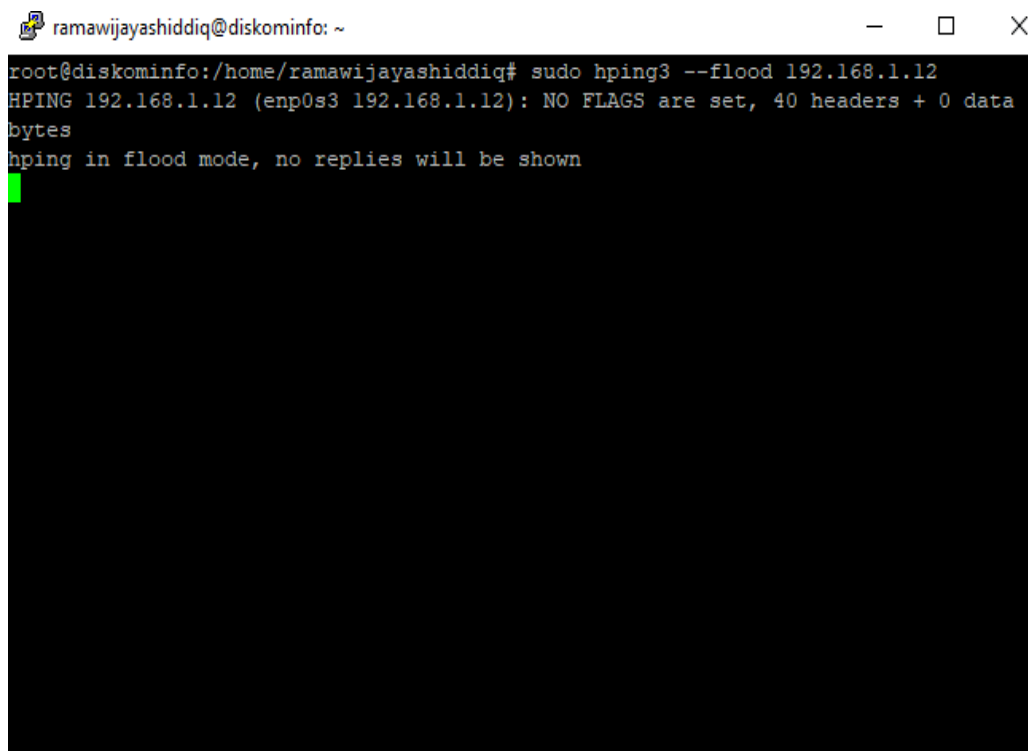


Gambar 4.17 Grafik tingkat ancaman *scanning port*

Pada Gambar 4.16 dan Gambar 4.17 hasil dari *scanning port* berhasil terdeteksi oleh sensor Maltrail dengan intensitas serangan 2.7%, namun tidak berhasil diblokir oleh Fail2Ban karena tingkat ancaman yang rendah.

4.3.7 *Syn Flooding*

Serangan *Syn Flood DDos* merupakan suatu aktivitas penyerangan yang mengeksploitasi proses *three way handshake* pada koneksi TCP yang memanfaatkan *Hyping*.

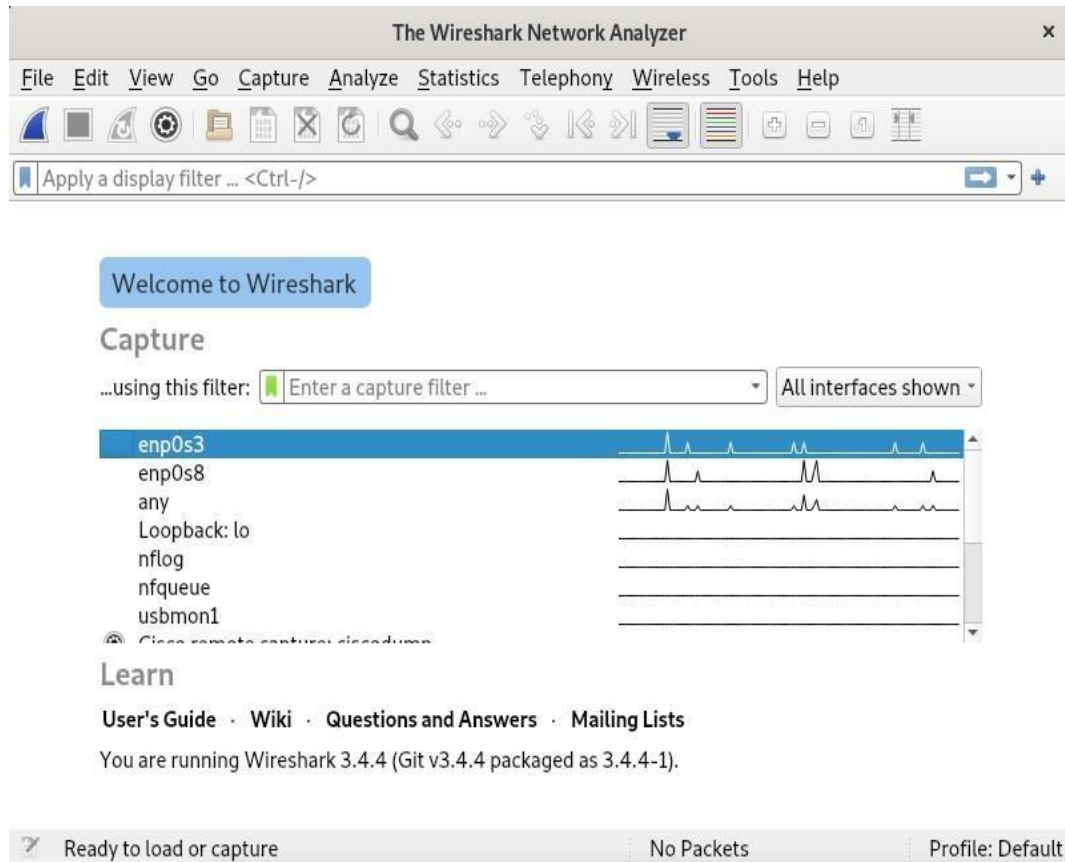
A screenshot of a terminal window with a black background and white text. The window title is 'ramawijayashiddiq@diskominfo: ~'. The user is root at diskominfo. The command executed is 'sudo hping3 --flood 192.168.1.12'. The output shows 'HPING 192.168.1.12 (enp0s3 192.168.1.12): NO FLAGS are set, 40 headers + 0 data bytes' and 'hping in flood mode, no replies will be shown'. A green cursor is visible on the line following the output.

```
ramawijayashiddiq@diskominfo: ~  
root@diskominfo:/home/ramawijayashiddiq# sudo hping3 --flood 192.168.1.12  
HPING 192.168.1.12 (enp0s3 192.168.1.12): NO FLAGS are set, 40 headers + 0 data  
bytes  
hping in flood mode, no replies will be shown
```

Gambar 4.18 Hasil serangan *Syn Flooding*

Pada Gambar 4.18 hasil dari serangan tersebut berhasil dilakukan pada server guna untuk mengkonsumsi sumber daya dari server sehingga server tidak bisa melayani lalu lintas yang memang benar benar sah.

4.4 Hasil dan Analisis



Gambar 4.19 Tampilan Wireshark untuk menangkap paket data pada server

Berdasarkan Gambar 4.19 pengukuran hasil dan analisis dari pengujian malware dilakukan dengan menggunakan Wireshark untuk menganalisa kinerja jaringan. Sensor Maltrail dan Fail2Ban merupakan sistem *malware* monitoring yang berfokus terhadap lalu lintas trafik paket-paket data yang melewati jaringan. Pengukuran ini perlu dilakukan untuk mengetahui intensitas trafik pada saat menangkap paket-paket data yang teriindikasi malware.

Wireshark · Capture File Properties · enp0s3

Details

Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit
enp0s3	0 (0.0%)	none	Ethernet	262144 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	246	246 (100.0%)	246 (100.0%)
Time span, s	55.971	55.971	55.971
Average pps	4.4	4.4	4.4
Average packet size, B	256	256	256
Bytes	62957	62957 (100.0%)	62957 (100.0%)
Average bytes/s	1,124	1,124	1,124
Average bits/s	8,998	8,998	8,998

Capture file comments

Help

Refresh

Copy To Clipboard

Close

Save Comments

Gambar 4.20 Jumlah throughput pada saat trafik normal

Gambar diatas merupakan tampilan hasil dari pengukuran jumlah throughput pada server dengan trafik normal.

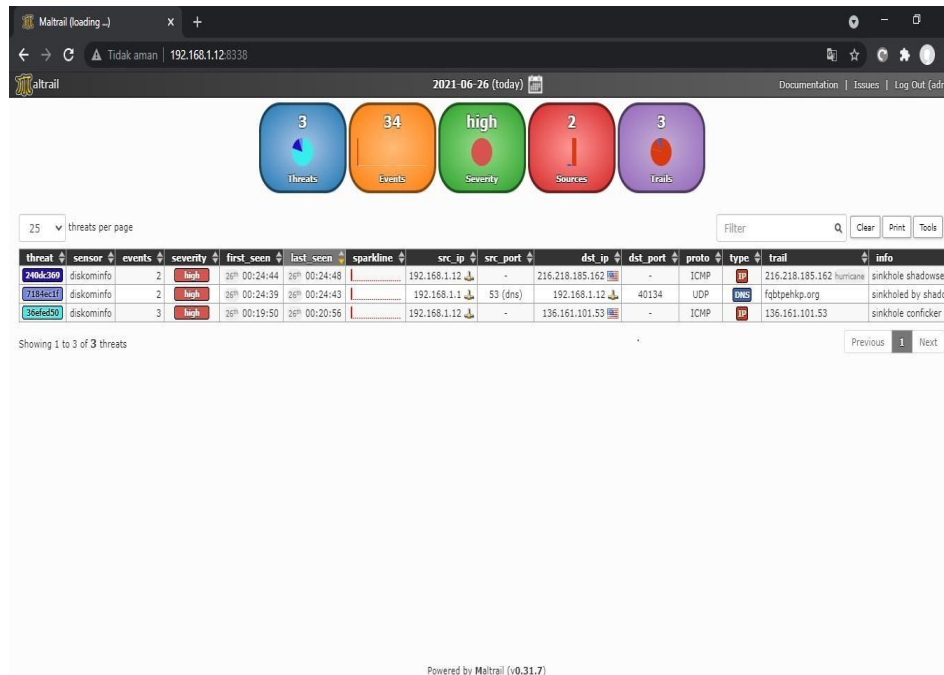
```

ramawijayashiddiq@diskominfo: ~
File Edit View Search Terminal Help
root@diskominfo:/home/ramawijayashiddiq# ping 136.161.101.53
PING 136.161.101.53 (136.161.101.53) 56(84) bytes of data.
64 bytes from 136.161.101.53: icmp_seq=1 ttl=50 time=256 ms
^C
--- 136.161.101.53 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 256.177/256.177/256.177/0.000 ms
root@diskominfo:/home/ramawijayashiddiq# ping 62.210.217.207
PING 62.210.217.207 (62.210.217.207) 56(84) bytes of data.
64 bytes from 62.210.217.207: icmp_seq=1 ttl=56 time=173 ms
64 bytes from 62.210.217.207: icmp_seq=2 ttl=56 time=176 ms
64 bytes from 62.210.217.207: icmp_seq=3 ttl=56 time=174 ms
^C
--- 62.210.217.207 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 5ms
rtt min/avg/max/mdev = 172.871/174.175/176.036/1.433 ms
root@diskominfo:/home/ramawijayashiddiq#

```

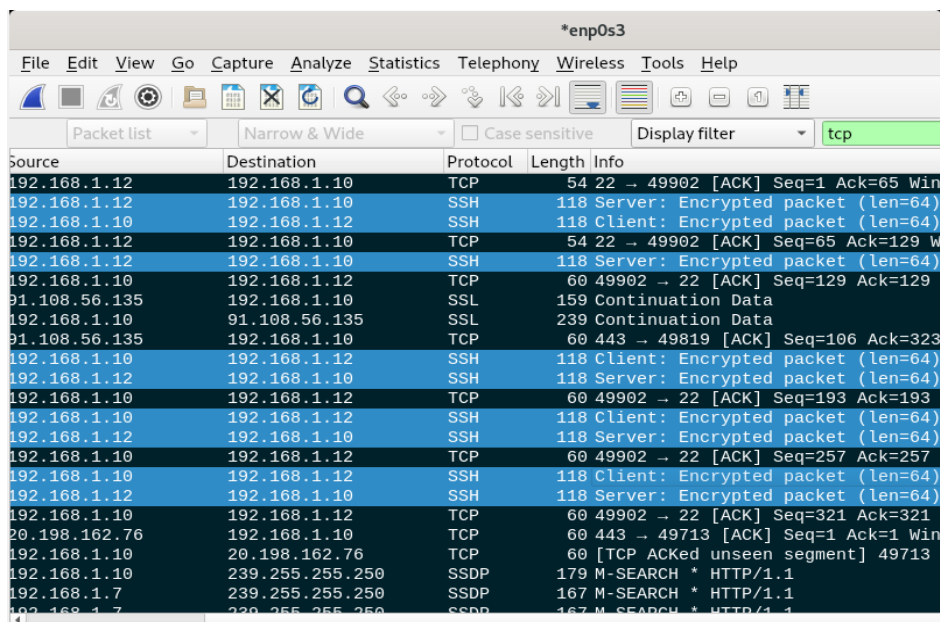
Gambar 4.21 Ping *malware*

Gambar diatas merupakan tampilan hasil ping *malware* sinkhole conficker menggunakan perintah ping -c 136.161.101.53



Gambar 4.22 Maltrail mendeteksi paket data *malware* sinkhole confiker

Gambar diatas merupakan tampilan hasil pendeteksian *malware* pada Sensor Maltrail setelah di lakukan perintah ping.



Gambar 4.23 Wireshark menangkap paket data enkripsi berupa *malware*

Gambar diatas merupakan tampilan Wireshark yang berhasil menangkap paket data yang terindikasi *malware* setelah dilakukan pengujian ping.

Wireshark - Capture File Properties - enp0s3				
Details				
Interfaces				
Interface	Dropped packets	Capture filter	Link type	Packet size limit
enp0s3	0 (0.0%)	none	Ethernet	262144 bytes
Statistics				
Measurement	Captured	Displayed	Marked	
Packets	440	440 (100.0%)	440 (100.0%)	
Time span, s	75.519	75.519	75.519	
Average pps	5.8	5.8	5.8	
Average packet size, B	159	159	159	
Bytes	69753	69753 (100.0%)	69753 (100.0%)	
Average bytes/s	923	923	923	
Average bits/s	7,389	7,389	7,389	
Capture file comments				
<div> <div>Help</div> <div>Refresh</div> <div>Copy To Clipboard</div> <div>Close</div> <div>Save Comments</div> </div>				

Gambar 4.24 Jumlah throughput pada saat menangkap paket data *malware*

Setelah berhasil menangkap paket data *malware*, lalu lintas trafik server pada Wireshark mengalami penurunan throughput menjadi 7.389 bits/s.

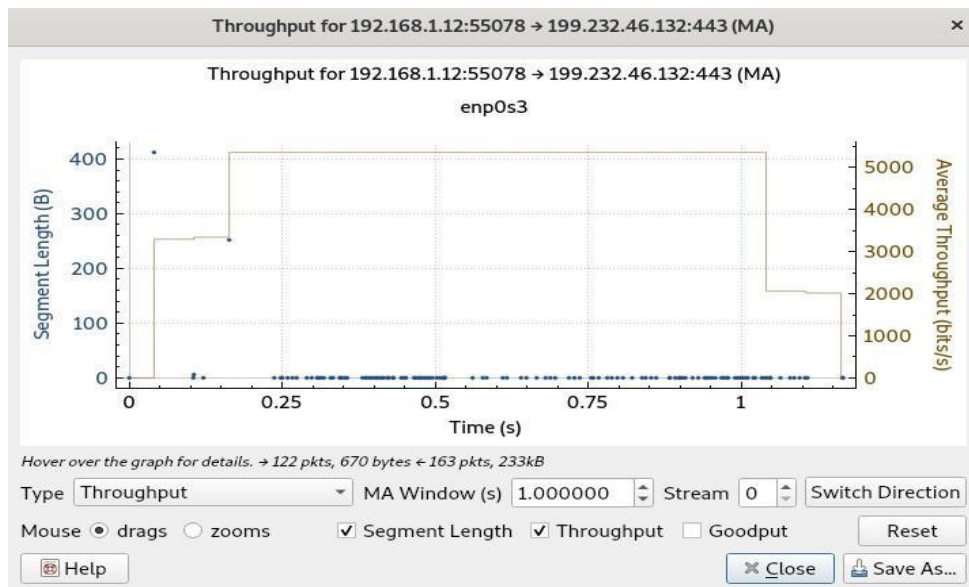
Wireshark - Capture File Properties - enp0s3				
Details				
Interfaces				
Interface	Dropped packets	Capture filter	Link type	Packet size limit
enp0s3	0 (0.0%)	none	Ethernet	262144 bytes
Statistics				
Measurement	Captured	Displayed	Marked	
Packets	475	313 (65.9%)	313 (65.9%)	
Time span, s	52.723	51.886	51.886	
Average pps	9.0	6.0	6.0	
Average packet size, B	166	163	163	
Bytes	78812	50872 (64.5%)	50872 (64.5%)	
Average bytes/s	1,494	980	980	
Average bits/s	11k	7,843	7,843	
Capture file comments				
<div> <div>Help</div> <div>Refresh</div> <div>Copy To Clipboard</div> <div>Close</div> <div>Save Comments</div> </div>				

Gambar 4.25 Jumlah throughput pada saat Fail2Ban memblokir akses *malware*

Kemudian setelah mengalami penurunan throughput, Fail2Ban secara otomatis memblokir akses dari paket data *malware* tersebut dan throughput kembali normal sehingga mengalami kenaikan menjadi 11.000 bits/s.

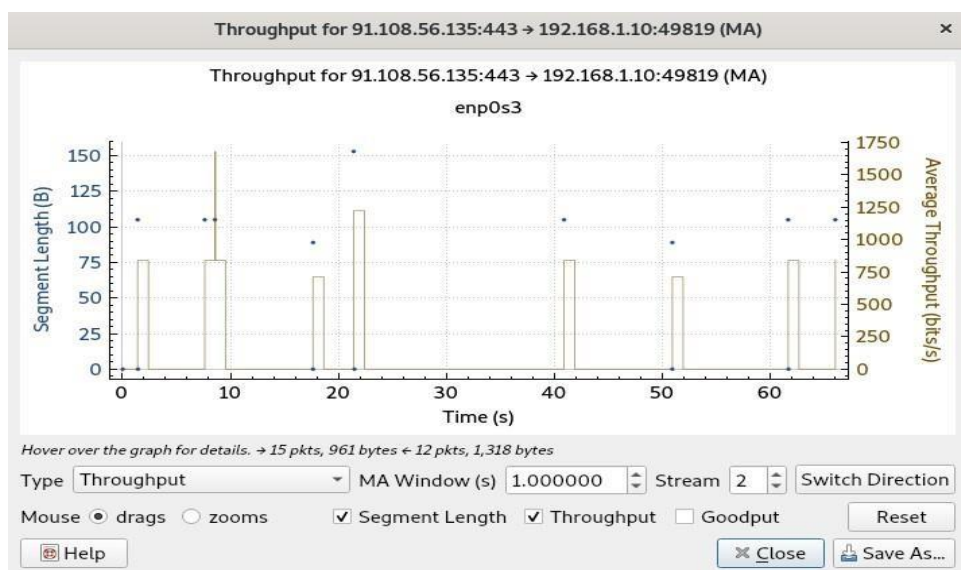
4.4.1 Analisis Kinerja

Throughput jaringan adalah tingkat keberhasilan pengiriman pesan melalui saluran komunikasi. Throughput biasanya diukur dalam bit per detik (bit/s atau bps), dan terkadang dalam paket data per detik (p/s atau pps) atau paket data per slot waktu. $\text{Throughput} = (\text{RWIN}/\text{RTT})$ dimana RWIN adalah TCP Receive Window dan RTT adalah waktu pulang pergi untuk jalur tersebut.



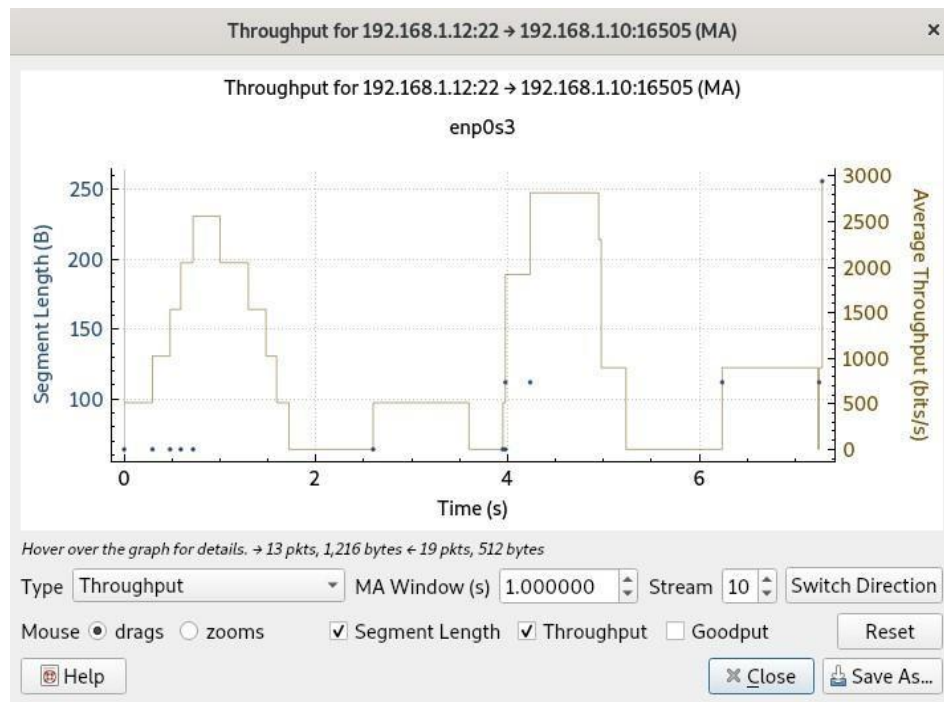
Gambar 4.26 Grafik throughput pada saat trafik normal

Gambar diatas merupakan tampilan grafik throughput pada server dengan lalu lintas normal.



Gambar 4.27 Grafik throughput pada saat menangkap paket data *malware*

Gambar diatas menunjukkan bahwa throughput jaringan berkurang ketika ada ancaman paket data *malware* didalamnya. Hal ini menyebabkan konsumsi bandwidth yang tinggi dan kemacetan jaringan.



Gambar 4.28 Grafik throughput pada saat Fail2Ban memblokir akses *malware*

Gambar diatas menunjukkan peningkatan throughput, karena menerapkan sistem Fail2Ban untuk memblokir lalu lintas berbahaya atau paket data yang teriindikasi sebagai *malware*. Sehingga lalu lintas trafik pada server kembali normal.

ramawijayashiddiq@diskominfo: ~

File

Edit

View

Search

Terminal

Help

top - 03:57:19 up 55 min, 4 users, load average: 0.44, 0.21, 0.26

Tasks: 163 total, 1 running, 162 sleeping, 0 stopped, 0 zombie

%Cpu(s): 0.7 us, 1.7 sy, 4.2 ni, 0.0 id, 93.4 wa, 0.0 hi, 0.0 si, 0.0 st

MiB Mem : 987.4 total, 138.1 free, 531.1 used, 318.2 buff/cache

MiB Swap: 975.0 total, 354.5 free, 620.5 used. 313.3 avail Mem

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
4881	ramawij+	39	19	761192	37812	24224	S	6.0	3.7	0:00.26	tracker+
895	ramawij+	20	0	2548292	159136	44860	S	0.7	15.7	3:15.38	gnome-s+
4869	root	20	0	12140	3652	3212	R	0.7	0.4	0:00.26	top
156	root	0	-20	0	0	0	I	0.3	0.0	0:01.67	kworker+
861	ramawij+	20	0	8692	2380	1432	S	0.3	0.2	0:00.37	dbus-da+
903	ramawij+	20	0	238728	2196	1792	S	0.3	0.2	0:00.06	gvfsd
1100	ramawij+	20	0	443680	9204	4640	S	0.3	0.9	0:00.41	tracker+
1663	root	20	0	567820	39836	4656	S	0.3	3.9	0:30.75	python
4851	root	20	0	398092	20560	9852	S	0.3	2.0	0:00.51	fail2ba+
1	root	20	0	164520	5412	3684	S	0.0	0.5	0:04.66	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par+
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker+
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_perc+
9	root	20	0	0	0	0	S	0.0	0.0	0:00.54	ksoftir+
10	root	20	0	0	0	0	I	0.0	0.0	0:01.27	rcu_sch+

Gambar 4.29 CPU *utilization* server dengan trafik normal

Gambar diatas merupakan CPU *utilization* dengan trafik normal pada server.

ramawijayashiddiq@diskominfo: ~

File Edit View Search Terminal Help

top - 04:23:09 up 1:21, 6 users, load average: 2.95, 1.40, 0.63

Tasks: 184 total, 4 running, 180 sleeping, 0 stopped, 0 zombie

%Cpu(s): 82.6 us, 9.1 sy, 0.0 ni, 0.0 id, 8.4 wa, 0.0 hi, 0.0 si, 0.0 st

MiB Mem : 987.4 total, 70.4 free, 713.0 used, 204.0 buff/cache

MiB Swap: 975.0 total, 141.3 free, 833.7 used. 110.5 avail Mem

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
895	ramawij+	20	0	2550316	103200	27512	R	51.5	10.2	3:55.97	gnome-s+
5068	ramawij+	20	0	2791320	208444	96144	S	11.6	20.6	0:19.13	firefox+
5369	ramawij+	20	0	2613096	193308	107996	R	11.2	19.1	0:03.89	Web Con+
5347	ramawij+	20	0	2589404	170144	93748	R	10.6	16.8	0:03.61	Web Con+
31	root	20	0	0	0	0	S	3.6	0.0	0:04.83	kswapd0
156	root	0	-20	0	0	0	I	0.7	0.0	0:02.08	kworker+
909	ramawij+	20	0	197484	27676	8520	S	0.7	2.7	0:06.72	Xwayland
4830	ramawij+	20	0	369884	29912	19852	S	0.7	3.0	0:03.47	gnome-t+
1	root	20	0	164400	3620	1964	S	0.3	0.4	0:04.87	systemd
10	root	20	0	0	0	0	I	0.3	0.0	0:01.55	rcu_sch+
1663	root	20	0	567820	27180	4036	S	0.3	2.7	0:34.02	python
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par+
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker+
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_perc+
9	root	20	0	0	0	0	S	0.0	0.0	0:00.63	ksoftir+

Gambar 4.30 CPU *utilization* server dengan serangan *malware*

Gambar diatas menunjukkan peningkatan CPU *utilization* pada server yaitu 51.5%, karena telah menangkap paket data *malware*. CPU *utilization* akan terus bertambah hingga 100% jika ada serangan *malware* dengan tingkat ancaman yang sangat tinggi.

4.4.2 Analisis Throughput

Pada pengujian ini, bertujuan untuk mengetahui dampak malware terhadap trafik jaringan terutama kecepatan trafik setelah terinfeksi malware. Untuk mengukur throughput dalam analisis ini data yang diambil adalah jumlah rata-rata Kbit/s.

Tabel 4.3 Hasil pengukuran throughput

<i>Sample</i>	<i>Size (Bytes)</i>	<i>Throughput (Kbits/s)</i>
<i>Normal Traffic</i>	62957	8998
<i>Malware Infected Traffic</i>	69753	7389

Tabel 4.3 menggambarkan throughput trafik normal mencapai 8998000 bit/s atau 8,9 Mbps. Sedangkan trafik *malware* memiliki throughput 7389 Kbits/s. Selain itu, data yang berhasil direkam pada lalu lintas normal adalah 62957 Bytes. Sedangkan data pada trafik malware memiliki ukuran yang lebih besar yaitu 69753 Bytes, hal ini akan menyebabkan terjadinya kemacetan terhadap lalu lintas trafik jaringan.

$$\text{Penurunan throughput} = \frac{(8998-7389)}{8998} \times 100\% = 18\%$$

4.4.3 Hasil Analisis Pemantauan Serangan pada Server

Tabel 4.4 Hasil analisis pemantauan dan pengamanan serangan pada server

Nama Komponen	Hasil Pengujian		
	Dideteksi Maltrail	Diblokir Fail2Ban	Notifikasi Terkirim
<i>DDos attack</i>	Tidak	Tidak	Tidak
<i>Scanning Port</i>	Ya	Tidak	Tidak
<i>Syn Flooding</i>	Tidak	Tidak	Tidak

Tabel 4.4 menunjukkan rangkuman hasil pemantauan dan pengamanan serangan pada server untuk 3 tipe serangan. Dari ketiga serangan Sistem Maltrail hanya berhasil mendeteksi *scanning port* dan Fail2Ban tidak memblokir karena tingkat serangan yang cukup rendah.

4.4.4 Hasil Uji Optimasi Sistem

Tabel 4.5 Uji Optimasi Sistem

No.	Tujuan sistem	Keterangan
1	Mendeteksi paket-paket yang terindikasi <i>malware</i>	Tercapai
2	Mencegah <i>malware log</i> dengan alamat IP <i>blocking</i>	Tercapai
3	Menampilkan <i>malware log</i> pada <i>browser</i>	Tercapai
4	Melaporkan status sistem dan alamat IP <i>malware</i> ke aplikasi Telegram	Tercapai

Pada tabel 4.5 merupakan hasil dari uji optimasi pada sistem yang dilakukan untuk mengetahui konsep yang telah dirancang sesuai dengan pembuatan awal telah berhasil diterapkan sesuai dengan konsep pada tahap-tahap sebelumnya.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil perancangan, pengujian dan analisa yang telah dilakukan maka dapat diambil beberapa kesimpulan sebagai berikut :

1. Berdasarkan hasil pengujian fungsionalitas terhadap fitur-fitur pada sistem yang diterapkan, seperti menampilkan grafik, log *malware*, cetak hasil data monitoring, untuk tingkat keberhasilan sistem Maltrail dan Fail2Ban telah berhasil mendeteksi dan memblokir serangan malware. Dapat disimpulkan bahwa semua fungsi 100% berjalan dengan baik sebagaimana semestinya.
2. Dari hasil pengujian kategori serangan selain *malware*, didapatkan hasil pengujian *DDos attack* 0%, *Syn Flooding* 0%, dan sistem Maltrail berhasil mendeteksi *scanning port* dengan tingkat ancaman 2.7%. Software Maltrail saat ini belum mampu mendeteksi serangan selain *malware* seperti *DDos attack* dan *Syn flooding*.
3. Berdasarkan hasil dan analisis pengukuran terhadap intensitas malware trafik pada jaringan server, terjadi penurunan throughput sebesar 18%. Dari hasil tersebut bahwa dampak malware pada intensitas trafik normal tidak mengalami penurunan nilai throughput yang cukup jauh karena telah berhasil menerapkan sistem Fail2Ban untuk memblokir akses dari malware tersebut, sehingga sistem akan meningkatkan keamanan dan throughput jaringan pada server.

5.2 Saran

Berdasarkan hasil perancangan Proyek Akhir ini, dapat disampaikan beberapa saran untuk pengembangan selanjutnya yaitu :

1. Menambahkan fitur penambahan daftar kategori *malware jail* fail2Ban secara otomatis yang bersumber dari repositori anti-virus international
2. Sistem dapat dikembangkan dengan kemampuan menyampaikan informasi dan memblokir akses dari serangan selain *malware* yaitu R-Wall.

DAFTAR PUSTAKA

- [1] Riki Triansyah, Dian Novianto. 2017. Prototype Keamanan Jaringan Menggunakan Teknik *Demilitarized Zone* (DMZ) Dengan Sistem Operasi Linux.
- [2] Mariwan Ahmed Hama Saeed. 2020. Malware in computer systems: Problems and Solutions. Vol. 9, No. 1, 2020, Pp. 1-8.
- [3] Kujawa A, Wendy Z, Jovi U, Jerome S, William T, Pieter A, Chris B. 2019. *2019 State of Malware*. California (US): Malwarebytes Corporation. 6—7.
- [4] Hudzaifah, Anang S, Devie RS. 2018. Membangun Sistem Monitoring Malicious Traffic di Jaringan dengan Maltrail. Bandung (ID): Telkom University. Vol 4 No.3: 2018.
- [5] Parita Chandrakant Parekh, Prof. Jayshree Upadhyay. 2018. Detecting and Blocking Encrypted Anonymous Traffic using Deep Packet Inspection. Vol-4 Issue-2 2020.
- [6] Sudahrshan N, P.Dass. 2019. Malicious Traffic Detection System using Publicly Available Blacklist's. Volume-8 Issue-6S, August 2019.
- [7] Adib Fakhri Muhtadi, Ahmad Almaarif. 2020. Analysis of Malware Impact on Network Traffic using Behavior-based Detection Technique. Vol. 1, No.1, April 2020, pp. 17-25.
- [8] Stampar M. 2016. Malicious Traffic Detection System. Github. [diakses 10 Februari 2021]. Tersedia pada: <https://github.com/stamparm/maltrail>.
- [9] Kurniawan I, Ferry Mulyanto, Fuad Nandiasa. 2016. Sistem Pencegah Serangan Bruteforce pada Ubuntu *Server* dengan menggunakan Fail2Ban. Bandung (ID)
- [10] Riki Andri Yusda, 2018. Rancang Bangun Jaringan Client Server Berbasis Linux Debian 6.0.
- [11] Anglano C, Massimo C, Marco G. 2017. Forensic Analysis of Telegram Messenger on Android Smartphones". Alessandria (IT): DiSIT-Computer Science Institute, Università del Piemonte Orientale. Vol 23: 31—49. <https://doi.org/10.1016/j.diin.2017.09.002>.
- [12] Kurniawan, A. (2012). Network Forensic. Yogyakarta: Andi Offset.

- [13] Diskominfo, “arti lambang kominfo,” Indonesian, 10 agustus 2017. [Online].
[Diakses 2021].

LAMPIRAN

LAMPIRAN 1

Lampiran 1 Konfigurasi Maltrail pada sistem Fail2Ban

```
#/etc/fail2ban/jail.local
```

```
[maltrail]
```

```
enabled          = true
```

```
filter           = maltrail
```

```
logpath          = /var/log/maltrail/*.log
```

```
maxretry         = 1
```

```
bantime          = 180
```

```
action           = iptables-allports[name=MALTRAIL, protocol=all] telegram
```

Lampiran 2 Jail Filter Fail2Ban untuk sistem Maltrail

```
#/etc/fail2ban/filter.d/maltrail.conf # Fail2Ban
```

```
filter for maltrail
```

```
[Definition]
```

```
failregex=(.*) debian <HOST> \d+ 10\.\10\.\10\.\1 .*(andromeda)
            (.*) debian <HOST> \d+ 10\.\10\.\10\.\1 .*(suspicious)
            (.*) debian <HOST> \d+ 10\.\10\.\10\.\1 .*(attacker)
            (.*) debian <HOST> \d+ 10\.\10\.\10\.\1 .*(scanner)
            (.*) debian <HOST> \d+ 10\.\10\.\10\.\1 .*(reputation)
            (.*) debian <HOST> \d+ 10\.\10\.\10\.\1 .*(phishing)
            (.*) debian <HOST> \d+ 10\.\10\.\10\.\1 .*(spammer)
            (.*) debian <HOST> \d+ 10\.\10\.\10\.\1 .*(proxy)
            (.*) debian <HOST> \d+ 10\.\10\.\10\.\1 .*(user agent)
            (.*) debian <HOST> \d+ 10\.\10\.\10\.\1 .*(port scanning)
            (.*) debian <HOST> \d+ 10\.\10\.\10\.\1 .*(conficker)
            (.*) debian <HOST> \d+ 10\.\10\.\10\.\1 .*(domain)
            (.*) debian <HOST> \d+ 10\.\10\.\10\.\1 .*(malware)
            (.*) (<HOST>(?:[0-9]{1,3}\.){3}[0-9]{1,3}) \-
            ((?:[09]{1,3}\.){3}[0-9]{1,3}(|,))* \- (.*)
```

```
ignoreregex =
```


Lampiran 3 Pengintegrasian Sistem Fail2Ban dengan Aplikasi

Telegram #/etc/fail2ban/scripts/fail2ban-telegram.sh #!/bin/bash

Display usage information

function show_usage {

 echo "Usage: \$0 action <ip>"

 echo "Where action start, stop, ban, unban" echo "and
 IP is optional passed to ban, unban" exit

 }

Mail text

#mailban =

"/"

Send notification

function send_msg {

 apiToken=1134726865:AAEzE74DHXTuBQunJw0HVBOHC8kGAOyk2
 DE chatId=(869498795 -1001420319743)

 url="https://api.telegram.org/bot\$apiToken/sendMessage" for room in
 \${chatId[@]}; do

 curl -s -X POST \$url -d chat_id=\$room -d text="\$1" done

 exit

 }

Check for script arguments if

[\$# -lt 1]

then

 show_u

sage fi

Take action depending on argument if

["\$1" = 'start']

then

 msg='[SISTEM+ONLINE]%0A%0ASistem+Ryzenware+baru+saja+online+ya

.'

 send_msg \$msg

```

elif ["$1" = 'stop'] then

    msg='[SISTEM+OFFLINE]%0A%0ASistem+Ryzenware+baru+saja+offline+ ya.'

    send_msg $msg

elif ["$1" = 'ban']
then

    msg=$( [ "$2" != " " ] && echo "Sistem+Ryzenware+baru+saja+mem-
banned+IP+Address+$2+!" )

    send_msg $msg

elif ["$1" = 'unban']
then

    msg=$( [ "$2" != " " ] && echo "Sistem+Ryzenware+baru+saja+me-
unbanned+IP+Address+$2+!" || echo "Sistem+Fail2ban+baru+aja+nge-
unbanned+IP+Address+ip.") send_msg $msg

else

    show_u
sage fi

```

Lampiran 4 Pembuatan Log Fail2Ban

```

# !/bin/bash

# log maltrail otomatis saat startup vm

touch /home/kominfo/maltrail/logredirect/empty.txt cat
/home/kominfo/maltrail/logredirect/empty.txt >>

/var/log/maltrail/$(date +"%Y-%m-%d").log

rm /home/kominfo/maltrail/logredirect/empty.txt

```

Lampiran 5 Perintah mengaktifkan sensor sistem Maltrail

```

# /home/kominfo/sysreset/sensorpy.sh #
#!/bin/bash

# Menjalankan sensor Maltrail

python /home/kominfo/maltrail/sensor.py

```

Lampiran 6 Perintah mengaktifkan *server* sistem Maltrail

```
# /home/kominfo/sysreset/serverpy.sh #  
!/bin/bash
```

```
# Menjalankan server Maltrail
```

```
python /home/kominfo/maltrail/server.py
```

Lampiran 7 Perintah mengaktifkan sistem Fail2Ban

```
# /home/kominfo/sysreset/fail2ban.sh #  
!/bin/bash
```

```
# Menjalankan sistem Fail2ban
```

```
systemctl restart fail2ban fail2ban-  
client start maltrail
```

Lampiran 8 Perintah konfigurasi sistem Maltrail

```
# [Server]
```

```
# Listen address of (reporting) HTTP server
```

```
# HTTP_ADDRESS  
0.0.0.0 #  
HTTP_ADDRESS ::
```

```
# HTTP_ADDRESS fe80::12c3:7bff:fe6d:cf9b%eno1  
HTTP_ADDRESS 10.10.10.1
```

```
# Listen port of (reporting) HTTP server
```

```
HTTP_PORT  
8338 # Use  
SSL/TLS
```

```
USE_SSL false
```

```
# SSL/TLS (private/cert) PEM file (e.g. openssl req -new -x509 -keyout  
server.pem -out server.pem -days 1023 -nodes)
```

```
# SSL_PEM misc/server.pem
```

```
# User entries
```

```
(username:sha256(password):UID:filter_netmask(s))
```

Note(s): sha256(password) can be generated on Linux with: echo -n 'password'
| sha256sum | cut -d " " -f 1

UID >= 1000 have only rights to display results (Note: this moment only
functionality implemented at the client side)

filter_netmask(s) is/are used to filter results USERS

Diskominfo:e81d47bc1914daacdf3670959ae7f749fd47976471fc68ed
00041f6150c80b6

```
ramanujayashiddiq@Diskominfo:~$
Maltrail (version) #v0.55.6 [https://github.com/stamparm/maltrail]

[*] starting @ 01:06:16 /2021-07-12/

[!] using configuration file '/home/ramanujayashiddiq/maltrail/maltrail.conf'
[!] using '/var/log/maltrail/' for log storage
[!] using '/sock/.maltrail/trails.csv' for trail storage
[!] updating trails (this might take a while)...
[!] 'https://data.netlab.360.com/feeds/dga/bigvictor.txt'
[!] 'https://data.netlab.360.com/feeds/dga/chloed.txt'
[!] 'https://data.netlab.360.com/feeds/dga/conficker.txt'
[!] 'https://data.netlab.360.com/feeds/dga/cryptolocker.txt'
[!] 'https://data.netlab.360.com/feeds/dga/gameover.txt'
[!] 'https://data.netlab.360.com/feeds/dga/locky.txt'
[!] 'https://data.netlab.360.com/feeds/dga/nequus.txt'
[!] 'https://data.netlab.360.com/feeds/dga/supobox.txt'
[!] 'https://data.netlab.360.com/feeds/dga/ctfees.txt'
[!] 'https://data.netlab.360.com/feeds/dga/virut.txt'
[!] 'https://www.abuseipdb.com/statistics'
[!] 'https://cybercrime-tracker.net/coam.php'
[!] 'https://www.badips.com/get/list/any/2?age=7d'
[!] 'https://raw.githubusercontent.com/firehol/blocklist-ipsets/master/bitcoin_nodes_id.ipset'
[!] 'https://raw.githubusercontent.com/stamparm/blackbook/master/blackbook.csv'
[!] 'https://lists.blocklist.de/lists/all.txt'
[!] 'https://raw.githubusercontent.com/firehol/blocklist-ipsets/master/botscout_id.ipset'
[!] 'https://danger.rules.sk/projects/brute-force-blocker/blist.php'
[!] 'https://raw.githubusercontent.com/fox-it/cobaltstrike-extraneous-space/master/cobaltstrike-servers.csv'
[!] 'https://www.crsit.com/xxvblit.txt.php'
[!] 'https://cybercrime-tracker.net/all.php'
[!] 'https://dataplane.org/cr.txt'
[!] 'https://iplists.firehol.org/files/dshid_top_1000.ipset'
[!] 'https://rules.emergingthreats.net/open/suricata/rules/botoc.rules'
[!] 'https://rules.emergingthreats.net/open/suricata/rules/compromised-ips.txt'
[!] 'https://rules.emergingthreats.net/open/suricata/rules/emerging-dns.rules'
[!] 'https://cybercrime-tracker.net/cpupdate.php'
[!] 'https://feedstacker.abuse.ch/downloads/ipblocklist_recommended.txt'
[!] 'https://iplists.firehol.org/files/gpf_gomica.ipset'
[!] 'https://blocklist.greenwan.co/greenanow.txt'
[!] 'https://ignoise.now.im/blacklist.txt'
[!] 'https://xtrixintel.com/feeds/xtip_malicious_domains.txt'
[!] 'https://xtrixintel.com/feeds/xtip_malicious_ips.txt'
[!] 'https://malnode.now/h1/hosts'
[!] 'https://www.malwaredomainlist.com/hostlist/hosts.txt'

[*] progress: 34/43 (79%)
```

```
ramanujayashiddiq@Diskominfo:~$
GNU nano 3.2 mass_scanner.txt

Copyright (c) 2014-2021 Maltrail developers (https://github.com/stamparm/maltrail/)
See the file 'LICENSE' for copying permission

#####
# Note: to automatically block IPs from this list (with ipset): #
#####
# sudo su
# apt-get -qq install iptables ipset &&
# ipset -q flush mass_scanners &&
# ipset -q create mass_scanners hash:net &&
# for ip in $(curl https://raw.githubusercontent.com/stamparm/maltrail/master/trails/static/mass_scanner.txt 2>/dev/null | grep -v "[]" | cut -d " " -f 1); do ipset add mass_scanners $ip; done
# iptables -I INPUT -m set --match-set mass_scanners src -j DROP
# iptables -nvL
#####

129.82.138.12 # pinger1a.netsec.colostate.edu
129.82.138.31 # pinger1.netsec.colostate.edu
129.82.138.32 # pinger2.netsec.colostate.edu
129.82.138.33 # pinger3.netsec.colostate.edu
129.82.138.34 # pinger4.netsec.colostate.edu
129.82.138.44 # pinger6.netsec.colostate.edu

128.9.168.98 # pinger-w3.ant.isi.edu
203.178.148.18 # pinger-j1.ant.isi.edu
203.178.148.19 # pinger-j2.ant.isi.edu

169.229.3.89 # researchacan-qv.eecs.berkeley.edu
169.229.3.90 # researchacan0.eecs.berkeley.edu
169.229.3.91 # researchacan1.eecs.berkeley.edu
169.229.3.92 # researchacan2.eecs.berkeley.edu
169.229.3.93 # researchacan3.eecs.berkeley.edu
169.229.3.94 # researchacan4.eecs.berkeley.edu

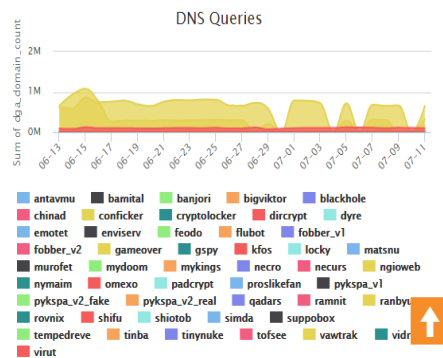
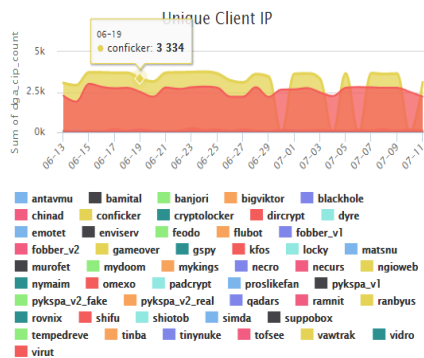
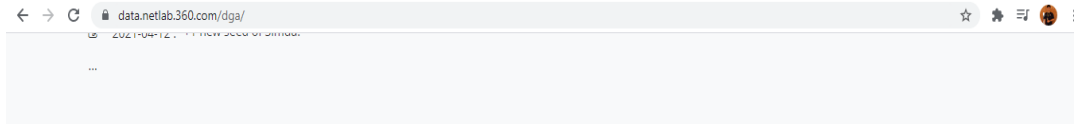
71.6.216.32 # labs.rapid7.com
71.6.216.33 # labs.rapid7.com
71.6.216.34 # labs.rapid7.com
71.6.216.35 # labs.rapid7.com
71.6.216.36 # labs.rapid7.com
71.6.216.37 # labs.rapid7.com
```

```
ramawijayashiddiq@Diskominfo: ~
GNU nano 3.2
README.md

**Maltrail** is a malicious traffic detection system, utilizing publicly available (black)lists containing malicious and/or generally suspicious trails, along with sta
! [Reporting tool] (https://i.imgur.com/Sd9eqoa.png)

The following (black)lists (i.e. feeds) are being utilized:
...
360bigviktork, 360chinad, 360conficker, 360cryptolocker, 360gameover,
360locky, 360mexors, 360suppobox, 360tofssee, 360virut, abuseipdb, alienvault,
atmos, badips, bitcoinnodes, blackbook, blocklist, botscout,
bruteforceblocker, ciarmy, cobaltstrike, cruzit, cybercrimetracker,
dataplane, dshieldip, emergingthreatsbot, emergingthreatscip,
emergingthreatsdns, feodoctrackerip, gpfcomics, greensnow, ipnoise,
kriskinteldns, kriskintelip, malcode, malwaredomainlistdns, malwaredomains,
maxmind, minerchk, myip, openphish, palevotracker, policeman, pony,
proxylists, proxysrs, proxyspy, ransomwaretrackerdns, ransomwaretrackerip,
ransomwaretrackerurl, rtproxies, rugers, sblam, socksproxy, ssbl,
ssllproxies, talosintelligence, torproject, trickbot, turris, urlhaus,
viriback, vxvault, zeustrackermonitor, zeustrackerurl, etc.

As of static entries, the trails for the following malicious entities (e.g. malware C&Cs or sinkholes) have been manually included (from various AV reports and persona
...
lms0rry, 404, 9002, aboc, absent, ab, aedbackdoor, acridrain, activeagent,
adrozok, advisorbot, adwin, adylkuzz, adzok, afrodita, agaalex, agenttesla,
aldbot, alina, allakore, almalocker, almashreg, alpha, alureon, amadey,
amavaldo, amend_miner, ammyrat, android_aecard, android_actionspy,
android_adrd, android_shmythrat, android_allenspy, android_andichap,
android_androrat, android_anubis, android_arspam, android_asacub,
android_backflash, android_bankbot, android_bankun, android_basbanke,
android_basebridge, android_besytia, android_blackrock, android_boxer,
android_buhsam, android_buyspaser, android_calbar, android_callerspy,
android_camsanner, android_cerberus, android_chuli, android_circle,
android_claco, android_clickfraud, android_cometbot, android_cookiethief,
android_coolreaper, android_copypat, android_counterclank, android_cyberwurx,
android_darkshades, android_dendroid, android_dougalek, android_droidjack,
android_droidkungfu, android_emesoluty, android_eventbot, android_ewalls,
android_ewind, android_exodus, android_exprespam, android_fakeapp,
```



Activate Windows