**Analisis *Malware* Menggunakan Bot Telegram Yang Terintegrasi**

**Dengan *Opensource* Sandbox**

1. **PENDAHULUAN**

Keamanan dunia maya (*cybersecurity*) saat ini bukan lagi sebuah pilihan, melainkan sebuah kebutuhan mendesak yang harus dipenuhi oleh setiap individu dan organisasi. Di zaman yang serba digital seperti sekarang, ancaman seperti *malware, ransomware,* dan serangan siber lainnya bisa datang kapan saja dan dari mana saja. Tidak hanya menyerang sistem, malware juga berpotensi mencuri data sensitif, merusak integritas sistem, hingga menurunkan kredibilitas perusahaan yang diserang. Bayangkan saja, data pelanggan yang sensitif atau informasi penting perusahaan yang jatuh ke tangan yang salah bisa sangat merugikan, baik dari segi finansial maupun reputasi. Di tengah ancaman yang terus berkembang, satu hal yang sangat penting adalah deteksi dini dan penanganan yang cepat. Namun dengan begitu banyaknya jenis malware dan serangan siber yang ada, cara untuk menganalisis dan mendeteksi ancaman ini secara manual sudah jelas tidak lagi cukup. Dibutuhkan sebuah solusi yang dapat mempercepat proses analisis serta memberikan hasil yang lebih terperinci dan terstruktur, agar pihak yang bertanggung jawab dapat segera mengambil langkah pencegahan atau mitigasi.

Salah satu solusi canggih yang kini banyak digunakan oleh perusahaan-perusahaan di seluruh dunia adalah menggunakan analisis malware berbasis *sandbox*. *Platform* seperti *Hybrid-Analysis*, yang menyediakan *sandbox* untuk melakukan analisis *file* secara otomatis, memberikan kemampuan untuk menguji dan mengidentifikasi potensi ancaman (*behaviour file*) dengan lebih efisien. Namun, meskipun canggih, proses ini biasanya memerlukan waktu dan keterampilan teknis untuk mengakses dan menganalisis laporan hasil *sandbox* secara manual. Inilah yang menjadi titik balik bagi kemajuan teknologi bot dan otomatisasi dalam dunia *cybersecurity*. Dengan memanfaatkan teknologi bot telegram, kita bisa membuat solusi yang lebih cepat, praktis, dan mudah digunakan oleh siapa saja—terutama bagi mereka yang mungkin tidak memiliki pengetahuan teknis yang mendalam. Salah satu contohnya adalah penggunaan Bot Telegram yang terintegrasi dengan *API Hybrid-Analysis*.

Bot ini dirancang untuk mempermudah analisis malware dengan memungkinkan pengguna untuk mengirimkan *file* yang mencurigakan langsung melalui Telegram, dan bot ini secara otomatis akan mengirimkan laporan analisis *malware* dalam bentuk yang mudah dipahami. Tak hanya itu, bot ini juga akan menghindari kesulitan-kesulitan teknis yang biasanya ditemukan dalam proses manual. Dengan memanfaatkan API Hybrid-Analysis, yang memberikan laporan mendalam terkait *verdict, hash file*, deteksi antivirus, *behaviour* dan banyak lagi, bot ini mampu menyajikan hasil analisis dalam format yang terstruktur.

1. **TUJUAN DAN RUANG LINGKUP**

Tujuan artikel ini adalah untuk memperkenalkan sistem analisis *malware* otomatis menggunakan Bot Telegram yang terhubung dengan *Hybrid-Analysis API*, serta mengeksplorasi manfaat, tantangan, dan solusi yang diberikan oleh sistem ini dalam meningkatkan respons keamanan di Perusahaan. Ruang lingkup artikel ini meliputi :

1. Penggunaan teknologi API Hybrid-Analysis untuk menganalisis file;
2. Integrasi Telegram Bot sebagai antarmuka yang mudah digunakan untuk mengakses analisis tersebut;
3. Proses dan hasil dari implementasi sistem ini.
4. **METODOLOGI**

Untuk membangun sistem ini, kami mengintegrasikan Hybrid-Analysis API yang memungkinkan analisis malware secara otomatis dengan menggunakan file yang dikirim melalui bot Telegram. Berikut adalah tahapan teknis yang terlibat dalam sistem ini:

1. Penerimaan File dari Pengguna

Pengguna mengirimkan file melalui Telegram Bot, yang kemudian diunduh oleh bot dan disiapkan untuk dianalisis.

1. Pengiriman File ke Hybrid-Analysis API

File yang diterima dikirim ke platform Hybrid-Analysis untuk dianalisis dalam lingkungan sandbox yang aman. Proses ini melibatkan pemilihan environment yang sesuai berdasarkan tipe file (misalnya, Windows, Linux, atau Android).

1. Hasil Analisis

Setelah file dikirim untuk dianalisis, bot akan menunggu hasilnya. Waktu tunggu bervariasi tergantung pada ukuran dan kompleksitas file yang dianalisis, yang dapat memakan waktu beberapa menit. Oleh karena itu, sistem ini juga dilengkapi dengan *delaying mechanism* yang memastikan hasil yang lebih lengkap dan akurat.

1. Pengiriman Laporan ke Pengguna

Setelah hasil analisis diperoleh, bot akan mengirimkan laporan dalam dua bentuk: laporan teks yang dapat dibaca di Telegram dan laporan PDF yang lebih terperinci.

1. Keamanan dan Kinerja

Kami juga memperhatikan aspek keamanan data pengguna dan kinerja sistem, memastikan bahwa proses pengiriman dan penerimaan file dilakukan secara aman, dan laporan dihasilkan dalam waktu yang wajar.

1. **HASIL DAN PEMBAHASAN**
2. *Sequence Diagram*

Berikut adalah *use case* dalam bentuk sequence diagram yang diterapkan dalam pengembangan bot telegram ini :

A screenshot of a computer

Description automatically generated

Gambar 1.1 *Sequence Diagram*

1. *Requirement* Pengembangan Bot Analysis Malware

Untuk mengembangkan bot analisis malware ini dibutuhkan beberapa komponen atau kebutuhan sebagai berikut :

* Hybrid-Analysis *API key* dan *API Endpoint*
* Telegram Bot Token
* Python versi 3.13 or higher
* Python Library (os, time, request, fpdf, python-telegram-bot)
* Server dengan koneksi internet (*no need public IP or DNS*) untuk running *script*

1. Tahapan pengembangan

* *Generate* Hybrid-Analysis *API key* dan menentukan *API Endpoint*

Untuk mendapatkan *API key/token* kita perlu mendaftar akun pada website Hybrid-Analysis (<https://www.hybrid-analysis.com/>) . Setelah mendaftarkan akun kita perlu masuk ke menu *profile* 🡪 *API key* 🡪 *Generate API Key*

A screenshot of a computer

Description automatically generated

Gambar 1.2 *Generate Hybrid Analysis API Key*

Setelah mendapatkan *API key* kita perlu mempelajari dokumentasi *Rest API* dari Hybrid-Analysis untuk menentukan *endpoint* yang tepat digunakan integrasi dengan Bot Telegram. Disini kami menggunakan 2 *endpoint* API sebagai berikut :

1. *Sandbox Submission* 🡪 /submit/file

*Endpoint* ini digunakan untuk mengunggah file ke *url* “<https://www.hybrid-analysis.com/api/v2/submit/file>” dengan *method POST*. Terdapat 4 *parameter* yang digunakan dalam request *endpoint* yaitu “*file”* dan “*environment\_id”* yang bersifat *mandatory* serta “*allow\_community\_access”* dan *“comment”* yang bersifat *opsional*.

A screenshot of a computer

Description automatically generated

Gambar 1.3 *Endpoint and Parameter 1*

1. *Sandbox Report*

*Endpoint* ini digunakan untuk mendapatkan *summary report* berdasarkan *submission id* dari file yang diunggah dengan *url “*[*https://www.hybrid-analysis.com/api/v2/report/{id}/summary*](https://www.hybrid-analysis.com/api/v2/report/%7bid%7d/summary)*”* dengan *method GET.* Terdapat 1 *parameter* yang digunakan dalam *request* *endpoint* yaitu *“id”* yang bersifat *mandatory*.

A screenshot of a login form

Description automatically generated

Gambar 1.4 *Endpoint and Parameter 2*

* Pembuatan Bot Telegram

Untuk membuat Bot Telegram kita perlu melakukan *request* pembuatan bot ke *BotFather* (Bot official telegram) dengan *username “@BotFather”*. Berikut Langkah-langkah melakukan *request* pembuatan *Bot* :

1. Mulai dengan memberikan perintah */start* pada BotFahter untuk mendapatkan opsi perintah / menu dalam Bot

A screenshot of a chat

Description automatically generated

Gambar 1.5 *Request Generate Bot 1*

1. Setelah itu jalankan perintah */newbot* dan masukkan nama serta username dari Bot (harus berakhiran dengan kata bot). dengan perintah ini BotFather akan memberikan kita detail dari Bot termasuk Token HTTP API untuk berkomunikasi dengan Bot :

A screenshot of a chat

Description automatically generated

Gambar 1.6 *Request Generate Bot 2*

* Pengembangan *Sourcecode* Berdasarkan *Sequence Diagram*

Pada tahap ini pengembangan *sourcecode* dilakukan dengan bahasa pemograman *python*. Alasan menggunakan bahasa pemograman python adalah karena python lebih fleksibel dan mempunyai banyak library yang dapat di *import* untuk komunikasi backend dengan telegram Bot maupun Hybrid-Analysis API. Pengembangan *sourcecode* juga harus berdasarkan *sequence diagram* yang sudah dibuat sebelumnya. Pendekatan ini akan memudahkan untuk pembuatan fungsi-fungsi yang ada dalam *sourcecode*. Beberapa fungsi pada pengembangan *sourcecode* untuk saat ini adalah sebagai berikut :

1. Konfigurasi API Key Hybrid-Analysis dan Token Bot Telegram
2. Pembuatan opsi *environment* untuk user Bot berdasarkan opsi yang tersedia di *Hybrid-Analysis*
3. Fungsi untuk mengunggah *file* dari telegram ke *Hybrid Analysis API*
4. Fungsi untuk mendapatkan report dari file yang telah diunggah
5. Fungsi “*Delaying”* untuk menunggu hasil analisis pada sandbox selesai dilakukan. Waktu tunggu dari hasil analisis dapat bervariasi tergantung file yang ingin dianalisis
6. Fungsi parsing JSON (*Output Hybrid Analysis API*)ke dalam output yang bisa dibaca di Telegram
7. Fungsi untuk membuat laporan dalam format PDF
8. Fungsi *Handler* untuk menangani perintah dari user Bot Telegram (start, opsi menu, opsi *environment,* upload file.
9. Fungsi *main* untuk menjalankan bot

Versi lengkap dari *sourcecode* diatas nantinya akan dapat diakses di github dengan username “rajaculaemas”. Versi saat ini masih bersifat *dev* dan perlu pengembangan dari sisi fitur, output maupun efisiensi *sourcecode*.

* Menjalankan dan Uji Coba Bot

Setelah semua kebutuhan telah terpenuhi, Langkah selanjutnya adalah melakukan uji coba bot telegram dengan tahapan sebagai berikut :

1. Menyiapkan *Server*

Penyiapan server yang telah ter-*install environment* yang dibutuhkan seperti python dan *library* yang dibutuhkan. Pada percobaan ini kami menggunakan server Debian versi 12.5 dengan kernal Linux 6.1.0-21-amd64.

1. Menjalankan *Sourcecode*

Untuk menjalankan *sourcecode* kita bisa melakukan secara langsung dengan perintah “sudo python3 <nama *sourcecode*>”. Akan tetapi untuk efisiensi penggunaan session pada OS disini kami akan menjalankan *sourcecode* *as a service*. Metode ini memerlukan pembuatan *file service* dalam folder /etc/systemd/system/, melakukan *reload daemon*, *enable file service* dan *running service* seperti pada gambar dibawah ini :

A screen shot of a computer

Description automatically generated

Gambar 1.7 Pembuatan *file service*

1. Uji Coba Bot Telegram

Uji coba dilakukan dengan mengakses Bot telegram dengan username “t.me/EasySanboxBot” yang sudah di *request* ke BotFather pada tahapan sebelumnya. Berikut ini adalah tampilan awal dari Bot Ketika diakses melalui *chat* telegram :

A green and yellow background with many objects

Description automatically generated

Gambar 1.8 Tampilan Awal Bot

Kita bisa menekan tombol “START” untuk inisiasi Bot. kemudian Bot akan menampilkan menu “Sandbox Analysis” dan menu lain yang nantinya perlu dikembangkan seperti pada gambar berikut :

A screenshot of a chat

Description automatically generated

Gambar 1.9 Menu Utama Bot

Setelah itu kita pilih menu “Sandbox Analysis” untuk memulai analisis file yang kita inginkan. Setelah itu Bot akan menampilkan opsi *environment* yang bis akita pilih. Disini kita perlu memilih *environment* yang tepat sesuai dengan ekstensi file yang ingin kita analisis. Misal jika kita ingin menganalisis *file* dengan ekstensi .exe maka kita perlu memilih *environment windows,* atau jika ekstensi *file* adalah .apk maka kita perlu memilih *environment android*. Kesalahan pada pemilihan *environment* dapat menyebabkan hasil analisis menjadi tidak akurat. Bot ini juga bisa menerima file dengan ekstensi .zip atau .rar. kita dapat terlebih dahulu mengkompresi file yang ingin kita analisis dalam format .zip atau .rar.

A screenshot of a chat

Description automatically generated

Gambar 1.10 Menu *Environment*

Pada percobaan kali ini kami sudah menyiapkan suatu *file* bernama LaZagne.exe. *File* ini biasanya digunakan untuk melakukan *dumping* kredensial pada suatu *victim machine* dengan OS Windows. Karena ekstensi file adalah .exe maka kami memilih windows untuk environment nya.

A screenshot of a chat

Description automatically generated

Gambar 1.11 Proses *Upload File*

Setelah mengunggah *file* yang ingin kita analisis kita perlu menunggu sampai analisis selesai dilakukan oleh *Hybrid-Analysis*. Waktu tunggu ini bervariasi tergantung dari ukuran, *signature* dan *behaviour* dari *file* tersebut. Kemudian setelah analisis selesai dilakukan Bot akan menampilkan ringkasan laporan dalam bentuk *text* dan PDF serta link lengkap laporan jika ingin melihat versi lengkap dari laporan tersebut.

Screens screenshot of a chat

Description automatically generated

Gambar 1.12 Output Report

Dari hasil analisis diketahui bahwa *file* LaZagne.exe adalah *file* berbahaya dengan dengan *threat score 100.* Selain itu *behaviour* saat file dijalankan di sandbox juga menginsikasikan melakukan proses yang *suspicious* seperti Inter-Process Communication (T1559) → Malware berkomunikasi antar proses untuk menjalankan perintah tersembunyi, Native API Execution (T1106) → Memanfaatkan API Windows untuk menghindari deteksi berbasis skrip, Shared Modules (T1129) → Menggunakan pustaka dan modul yang ada dalam sistem untuk menjalankan kode berbahaya, Component Object Model (T1559.001) → Mengeksploitasi mekanisme Windows COM untuk mengeksekusi kode tanpa izin pengguna dan lain sebagainya. Dengan output yang begitu sederhana maka pengguna akan dengan mudah dapat menyimpulkan apakah *file* tersebut berbahaya atau tidak.

1. **KESIMPULAN DAN SARAN**

Implementasi Telegram Bot yang terintegrasi dengan Hybrid-Analysis API memberikan solusi yang efisien dan mudah diakses untuk analisis malware di Perusahaan atau pengguna internet secara umum. Dengan sistem ini, kami berhasil mengurangi waktu yang diperlukan untuk mendeteksi ancaman, memberikan laporan yang lebih cepat, dan memudahkan tim keamanan untuk menangani file yang mencurigakan dengan lebih efektif. Untuk saat ini masih terdapat kekurangan dalam Bot ini. Di masa depan, sistem ini dapat diperluas dengan berbagai fitur tambahan, seperti integrasi dengan sistem keamanan perusahaan lainnya, dan mungkin juga menggabungkan teknik-teknik deteksi malware berbasis AI untuk analisis yang lebih mendalam.

1. **REFERENSI**
2. *Hybrid-Analysis API Documentation :* [*https://www.hybrid-analysis.com/docs/api/v2*](https://www.hybrid-analysis.com/docs/api/v2)
3. *Telegram Bot API documentation :* [*https://core.telegram.org/api*](https://core.telegram.org/api)
4. *MITRE ATT&CK Framework :* [*https://attack.mitre.org/*](https://attack.mitre.org/)