Analisis Malware AsyncRAT Berdasarkan Perilaku dalam Sistem Operasi Windows 11 Menggunakan Metode Hybrid

Satria Afif Ramdani
Program Studi Teknik Rekayasa Keamanan Siber
Politeknik Negeri Batam
Jl. Ahmad Yani, Batam Centre, Batam 29461, Indonesia
Satria.afif15@gmail.com

Festy Winda Sari
Program Studi Teknik Rekayasa Keamanan Siber
Politeknik Negeri Batam
Jl. Ahmad Yani, Batam Centre, Batam 29461, Indonesia
festy@polibatam.ac.id

Abstrak – Meningkatnya pravelensi malware trojan menjadi ancaman terhadap keamanan informasi karena dapat membahayakan kerahasiaan, integritas, dan ketersediaan. Salah Jenis malware trojan yang masih akif dugunakan dalam berbagai serangan siber adalah Remote Access Trojan (RAT) yang memungkinkan kriminal siber untuk mendapatkan akses kontrol tidak sah dan mengendalikan sistem untuk dieksploitasi. Salah satu varian RAT yang akitif dalam serangan siber modern adalah AsyncRAT. Penelitian ini menganalisis perilaku malware AsyncRAT pada sistem operasi Windows 11 menggunakan pendekatan Hybrid yang mengkorelasikan hasil analisis statis dan dinamis untuk mendapatkan pemahaman secara komprehensif. Analisis malware AsyncRAT dilakukan di lingkungan terisolasi untuk menghindari external call back. Tahapan analisis statis dilakukan tanpa menjalankan sampel malware menggunakan PEStudio sebagai alat untuk mengidentifikasi karakteristik internal file. Analisis dinamis dilakukan dengan menjalankan sample malware di lingkungan terisolasi untuk mengamati perilaku malware selama dieksekusi dengan menggunakan Process Explorer dan Process Monitor. Analisis hybrid mengkorelasikan hasil temuan analisis statis dan dinamis, penelitian menghasilkan profil perilaku AsyncRAT.

Keywords - Malware Analysis, Remote Access Trojan, AsyncRAT, Analisis dinamis, Analisis statis, Analisis Hybrid

I. PENDAHULUAN

Perkembangan teknologi informasi dan pemanfaatan teknologi digital di berbagai sektor meningkatkan pentingnya keamanan sistem informasi. Seiring dengan meningkatnya ketergantungan terhadap teknologi digital, berbagai ancaman siber terus berkembang yang berpotensi mengganggu kerahasiaan, integritas, dan kesediaan informasi. Salah satu ancaman siber yang masih aktif saat ini adalah ancaman *Malicious Software*. *Malicious Software* atau biasa disingkat sebagai *malware*, merujuk pada program atau kode yang sengaja dirancang untuk menyusup, merusak, mengganggu, atau mendapatkan akses tanpa izin ke sistem dan jaringan komputer [1]. *Trojan* merupakan salah satu ancaman *malware* yang banyak digunakan sebagai media infiltrasi ke dalam sistem komputer. Tidak seperti virus pada umumnya, *Trojan* menyamar sebagai program yang sah atau tidak berbahaya, namun secara diam-diam mengandung kode berbahaya. *Trojan* tidak mereplikasi dirinya sendiri secara otomatis, Sebaliknya, Trojan membutuhkan interaksi pengguna seperti mengunduh atau menjalankan *file* yang terinfeksi untuk diinstal pada sistem komputer. Setelah terinstal, *Trojan* dapat membuka *backdoor* bagi penyerang untuk mencuri data, mengambil kendali perangkat, atau merusak sistem. semuanya tanpa sepengetahuan pengguna [2]. Salah satu varian *Trojan* yang menjadi perhatian dalam perkembangan ancaman siber saat ini adalah *Remote Access Trojan (RAT)*. berdasarkan laporan Lanskap Keamanan Siber Indonesia 2024 oleh BSSN dan Laporan Tahunan Kamsiber 2024 oleh KOMDIGI, *Remote Access Trojan* termasuk dalam kategori *malware* yang mendominasi tren ancaman siber dalam beberapa tahun terakhir.

Remote Access Trojan (RAT) adalah program *malware Trojan* yang memungkinkan aktivitas pengawasan tersembunyi atau memperoleh kendali terhadap perangkat korban. *Remote Access Trojan (RAT)* dapat diinstal melalui beberapa teknik yang serupa dengan infeksi *malware* lainnya. Lampiran email yang dirancang khusus, tautan web, paket unduhan, atau *file* .torrent dapat digunakan sebagai mekanisme instalasi malware tersebut [3]. Salah satu jenis *malware Remote Access Trojan (RAT)* adalah *AsyncRAT*. Untuk memahami karakteristik dan perilaku *malware* ini secara menyeluruh, diperlukan teknik analsisi yang mampu mengungkap aktivitas berbahaya yang dilakukan *malware* di dalam sistem. Secara umum analisis malware dapat dilakukan melalui tiga jenis pendekatan yaitu, analisis statis, analisis dinamis, dan analisis *hybrid*.

II. LANDASAN TEORI

A. *Remote Access Trojan (RAT)*

Remote Access Trojan (RAT) merupakan jenis malware yang memungkinkan penyerang mengendalikan sistem korban dari jarak jauh melalui *backdoor* yang dibuat pada perangkat korban. *Malware* ini dapat digunakan untuk memantau aktivitas pengguna, mencuri data sensitif, memanipulasi *file*, serta melakukan komunikasi jaringan tanpa sepengetahuan pengguna. *Remote Access Trojan (RAT)* umumnya menyebar melalui file *executable* (.exe) berbahaya, lampiran email, maupun teknik *social engineering* [4].

B. *Analisis Statis*

Analisis statis merupakan metode analisis *malware* yang dilakukan tanpa mengeksekusi *malware* tersebut pada sistem. Metode ini digunakan untuk mengidentifikasi karakteristik *malware* seperti struktur *file*, konfigurasi, *import function*, maupun *string*

mencurigakan yang terdapat pada *file malware*. Analisis statis bertujuan untuk memperoleh informasi awal mengenai kemampuan dan karakteristik malware [5].

C. Analisis Dinamis

Analisis dinamis merupakan metode analisis *malware* yang dilakukan dengan menjalankan *malware* dalam lingkungan yang terisolasi seperti *virtual machine* atau *sandbox* untuk mengamati perilaku *malware* secara langsung. Melalui analisis dinamis, aktivitas *malware* seperti *process activity*, *registry modification*, perubahan *file*, serta komunikasi jaringan dapat diamati selama *malware* dijalankan. Metode ini digunakan untuk memahami perilaku *runtime malware* dan mengidentifikasi aktivitas berbahaya yang dilakukan *malware* pada sistem [5].

D. Analisis Hybrid

Metode *hybrid* merupakan pendekatan analisis *malware* yang mengkorelasikan hasil analisis statis dan analisis dinamis untuk memperoleh pemahaman yang lebih komprehensif terhadap perilaku *malware*. Analisis dimulai dengan analisis statis untuk mengidentifikasi karakteristik awal *malware* tanpa menjalankan file, kemudian dilanjutkan dengan analisis dinamis untuk mengamati perilaku *malware* saat dijalankan dalam lingkungan terisolasi. Pendekatan hybrid dinilai lebih efektif karena mampu menggabungkan kelebihan dari kedua metode analisis dalam proses identifikasi dan pengamatan perilaku malware [5].

E. PEStudio

PEStudio merupakan *software* yang dirancang khusus untuk analisis statis. *software* ini memeriksa struktur *file* yang dapat dieksekusi, *embedded string*, *import*, *header*, dan *metadata* tanpa mengeksekusi *file malware* itu sendiri. *Software* ini membantu pendekatan yang memastikan keamanan maksimal selama pemeriksaan. [6].

F. Process Monitor

Process Monitor (ProcMon) adalah *software* pemantauan *Windows* yang digunakan selama analisis dinamis untuk mengamati aktivitas sistem secara *real-time* di dalam lingkungan virtual. Alat ini membantu melacak perubahan *file system*, *registry modification*, dan *create process* yang dipicu oleh *file* eksekusi yang mencurigakan oleh malware [7].

G. Process Explorer

Process Explorer adalah *software* yang memungkinkan untuk mengakses banyak informasi tentang proses yang berjalan di suatu sistem. Dalam analisis dinamis, *software* ini digunakan untuk mengamati aktivitas proses secara *real-time* termasuk pembentukan proses baru [8].

III. METODOLOGI

Penelitian ini menggunakan metode kualitatif yang berfokus pada analisis perilaku *malware AsyncRAT* pada sistem operasi Windows 11 menggunakan pendekatan hybrid yang mengkorelasikan hasil analisis statis dan dinamis. Objek penelitian berupa sampel *in-the-wild AsyncRAT malware* yang diperoleh dari MalwareBazaar dan dijalankan pada lingkungan *virtual* yang terisolasi menggunakan Oracle VM VirtualBox.

A. Langkah Penelitian

Adapun tahapan penelitian adalah sebagai berikut:

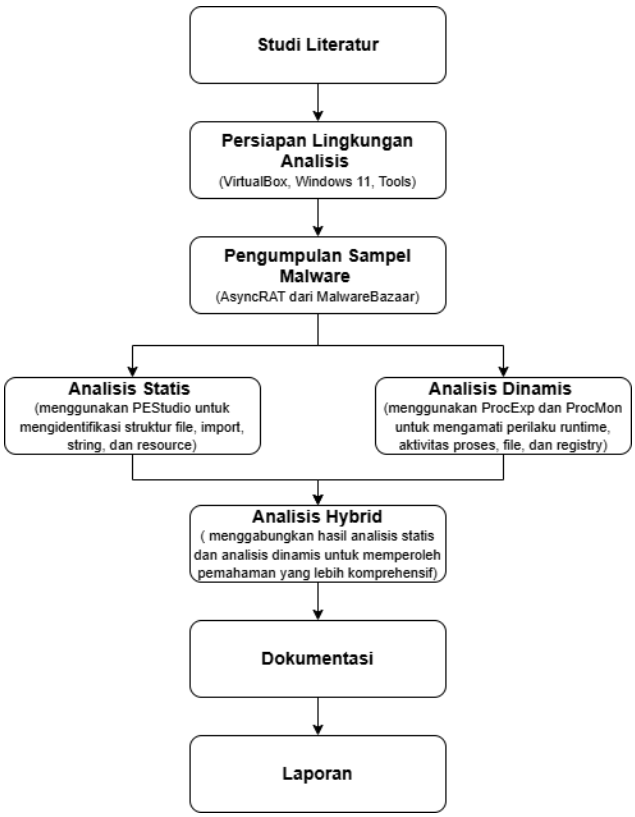

Figure 1 Langkah Penelitian

B. Studi Literatur

Tahapan ini mencakup pemahaman mengenai *malware AsyncRAT*, analisis *malware*, analisis statis, analisis dinamis, analisis *hybrid*, dan *tools* yang akan digunakan selama proses analisis.

C. Persiapan Lingkungan Analisis

Lingkungan analisis dirancang untuk menyediakan lingkungan yang terisolasi agar proses analisis malware dapat dilakukan denga aman tanpa mempengaruhi sistem host.

Table 1 Spesifikasi Lingkungan Analisis

Komponen	Spesifikasi
Hypervisor	Oracle VM VirtualBox
Guest Operation System	Windows 11 64-bit
RAM	4096 MB
Processor	2 vCPU

Storage	80 GB Virtual Disk
Network Config	NAT adapter disconnected during malware execution
Isolation Mechanism	No shared folder, no USB passthrough

D. Pengumpulan Sampel Malware AsyncRAT

In-the-wild malware sample diperoleh dari *platform* MalwareBazaar dengan tag "*AsyncRAT"* dan diunduh di dalam lingkungan yang terisolasi dalam bentuk .zip *file* yang kemudian diekstrak untuk mendapatkan *executable file* untuk digunakan sebagai objek penelitan.

E. Analisis Statis

Tahapan analisis statis dilakukan tanpa menjalankan sample *malware* menggunakan PEStudio. Indikator yang yang akan diamati adalah *entropy, persistence indicator, file system indicator, process activity*, dan *memory/process manipulation*. Hasil analisis digunakan untuk mengidentifikasi karakter dasar sampel.

F. Analisis Dinamis

Tahapan analisis dinamis, sampel *malware* dijalankan di dalam lingkungan terisolasi menggunakan menggunakan *virtual machine* mensimulasikan sistem Windows 11. Analisis dilakukan menggunakan software *Process Monitor* dan *Process Explorer* untuk mengamati aktivitas sampel di dalam sistem.

G. Analisis Hybrid

Pada tahapan analisis *hybrid*, hasil temuan pada tahapan analisis statis dan dinamis dikorelasikan untuk mendapatkan pemahaman perilaku sampel *malware* secara komprehensif.

H. Dokumentasi

Tahapan dokumentasi dilakukan dengan mengumpulkan dan mengorganisir hasil temuan analisis statis dan dinamis.

I. Laporan

Hasil temuan yang telah didokumentasikan akan disusun dalam bentuk laporan terstruktur.

IV. ANALISIS DAN PEMBAHASAN

Sampel yang dikumpulkan berupa sampel *malware AsyncRAT* yang diperoleh dari *platform* MalwareBazaar. Rangkuman informasi mengenai sample tersebut adalah sebagai berikut:

Figure 2 Sampel 1 AsyncRAT

Komponen	Detail
File Name	3ba2080a5a4791a7a8c5ea42ac40826bfca758f3abea4da90d3f22fbc50c2d60.exe
SHA256	3ba2080a5a4791a7a8c5ea42ac40826bfca758f3abea4da90d3f22fbc50c2d60
File Size	209 KB
Source	MalwareBazaar

Figure 3 Sampel 2 AsyncRAT

Komponen	Detail
File Name	3758bbe67d3bb290e49a4317881084eb64e1e4228af281382059d4ad0e5c11c1.exe
SHA256	3758bbe67d3bb290e49a4317881084eb64e1e4228af28138205 9d4ad0e5c11c1.exe
File Size	783 KB
Source	MalwareBazaar

A. Analisis Statis Terhadap Sampel

Analisis statis dilakukan di dalam lingkungan terisolasi virtual dengan tanpa menjalankan sampel *malware*. Indikator yang diamanti diantaranya yaitu *Entropy*, indikator *persistence*, indikator *file system*, *process activity*, dan *memory/process manipulation* [9][10]. Kedua file *executable (.exe)* sampel dimasukkan ke dalam PEStudio dan diperoleh hasil sebagai berikut:

Figure 4 Temuan Analisis Statis PEStudio Sampel 1

Indikator	Evidence
Entropy	6.573
Persistence Indicator	Inkonklusif
File System Indicator	`CopyFileA`, `CreateDirectoryA`, `DeleteFileA`
Process Activity	`CreateProcessA`, `CreatePipe`
Memory/Process Manipulation	`AdjustTokenPrivileges`, `CallNextHookEx`, `GetKeyState`, `GetClipboardData`

Figure 5 Temuan Analisis Statis PEStudio Sampel 2

Indikator	Evidence
Entropy	6.417
Persistence Indicator	Inkonklusif

File System Indicator	`CopyFileW`, `CreateDirectoryW`, `FindFirstFileExW`, `FindNextFileW`, `WriteFile`
Process Activity	`ShellExecuteW`
Memory/Process Manipulation	`VirtualProtect`

Nilai *entropy* yang diperoleh 6.573 pada sample 1 dan 6.417 pada sampel 2 menunjukan tingkat keacakan (*randomness*) distribusi *byte* yang relatif tinggi. Nilai tersebut mengindikasikan adanya teknik *packing*, *compression*, atau *encryption* untuk menyamarkan kode berbahaya [11].

Analisis untuk indikator presistensi, bagian *strings* pada PEStudio tetap berada dalam status "*waiting*" dan tidak menampilkan hasil dan identifikasi mekanisme presistensi menghasilkan hasil inkonklusif. Analisis kemudian dilanjutkan pada indikator *file system* untuk mengidentifikasi kemampuan sampel berinteraksi dan memanipulasi *system file*.

Hasil analisis menunjukan kedua sampel memiliki indikator interaksi dan manipulasi *system file*. Pada sampel 1, ditemukan API `CopyFileA`, `CreateDirectoryA`, dan `DeleteFileA` yang menunjukkan kemampuan sampel untuk menyalin *file*, membuat direktori, dan menghapus *file*. Sampel 2 memiliki API `CopyFileW`, `CreateDirectoryW`, `FindFirstFileExW`, `FindNextFileW`, dan `WriteFile`, yang mengindikasikan kemampuan untuk menyalin *file*, membuat direktori, melakukan pencarian atau enumerasi *file* dan direktori, serta menulis data. Temuan ini menunjukan kedua sampel memiliki kemampuan untuk berinteraksi dan memanipulasi *file system* pada sistem korban.

Temuan untuk indikator *memory/process manipulation* pada sampel 1, ditemukan API `AdjustTokenPrivileges` berkaitan dengan perubahan hak akses, `CallNextHookEx` digunakan dalam mekanisme *hooking* untuk meneruskan informasi ke fungsi *hook* berikutnya, `GetKeyState` digunakan untuk memperoleh status tombol pada *keyboard*, dan `GetClipboardData` memungkinkan sampel untuk mengakses data yang tersimpan pada *clipboard*. Pada sampel 2, ditemukan API `VirtualProtect` yang memungkinkan sampel mengubah izin akses pada memori. Dalam konteks malware, fungsi ini dapat digunakan untuk mendukung aktivitas seperti eksekusi kode di memori atau code injection.

Hasil analisis statis ini memberikan gambaran awal mengenai karakteristik dan potensi kedua sampel *AsyncRAT*.

B. Analisis Dinamis Terhadap Sampel

Analisis dinamis dilakukan di dalam lingkungan terisolasi virtual dengan menjalankan sampel malware. Dengan menggunakan *Process Monitor* dan *Process Explorer* untuk mengamati aktivitas sample selama berjalan di dalam sistem diperoleh hasil sebagai berikut:

Pada sampel 1, saat dijalankan, *User Account Control* (UAC) menampilkan permintaan untuk memberikan hak *administrator* kepada `Synaptics.exe`, dengan *publisher* yang ditampilkan *unknwon*. Selanjutnya, melalui *Process Explorer*, `Synaptics.exe` teramati muncul dan berjalan dalam sistem. Sebelum sampel dijalankan. `Synaptics.exe` tidak terlihat di *Process Explorer*.

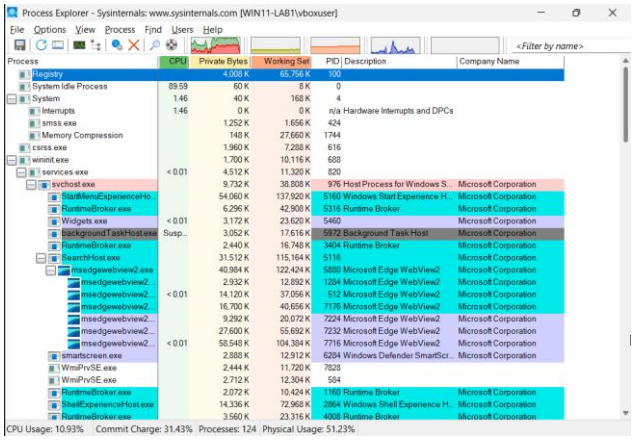

Figure 6 Process Explorer Sebelum Sampel Malware Dijalankan

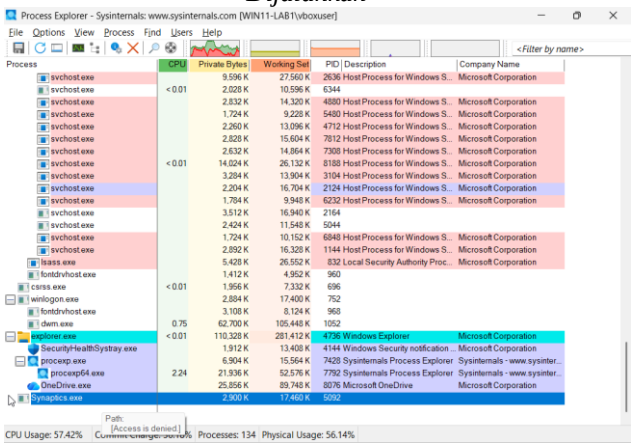

Figure 7 Process Explorer Sesudah Sampel Malware 1 Dijalankan

Hasil pemantauan menggunakan *process monitor* menunjukan sampel 1 melakukan beberapa operasi registri, seperti `RegOpenKey` dan `RegQueryValue`, serta mengakses berbagai berkas sistem Windows dan DLL. Namun, tidak terdapat modifikasi registri, pembuatan berkas, atau aktivitas terkait persistensi yang teramati selama pemantauan. Temuan dinamis ini selaras dengan temuan pada analisis statis yang mengidentifikasi API `AdjustTokenPrivileges`, yang berkaitan dengan pengaturan proses hak akses (*privilege*).

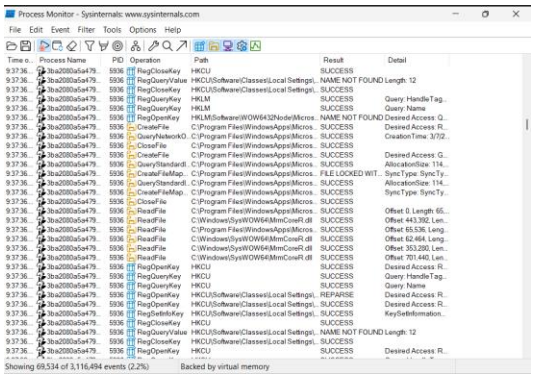

Figure 8 Capture Process Monitor

Pada analisis dinamis untuk sampel 2, proses eksekusi tidak dapat dilakukan secara normal karena file memiliki ekstensi `.exe` teridentifikasi oleh PEStudio saat aanalisis statis sebagai *Dynamic Link Library* (DLL). Ketidaksesuaian tersebut menyebabkan sampel 2 tidak dapat dijalankan seperti *file executable* pada umumnya. Kondisi menunjukan extensi *file* tidak selalu merepresentasikan *file* sebenarnya dan termasuk dalam teknik *file masquerading* yang digunakan penyerang untuk menyamarkan file berbahaya [12].

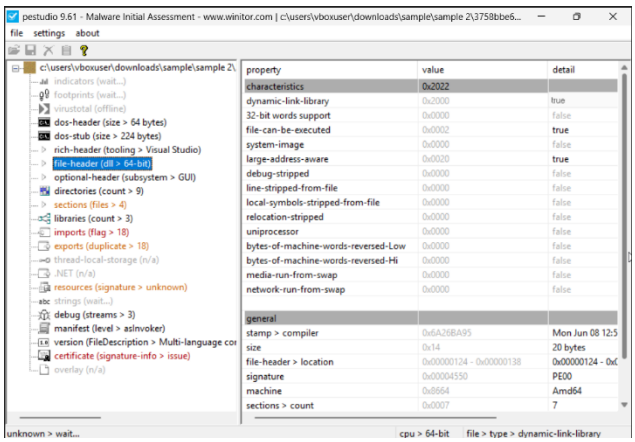

Figure 9 Indikator Sampel 2 di PEStudio menunjukan ekstensi file sebagai DLL

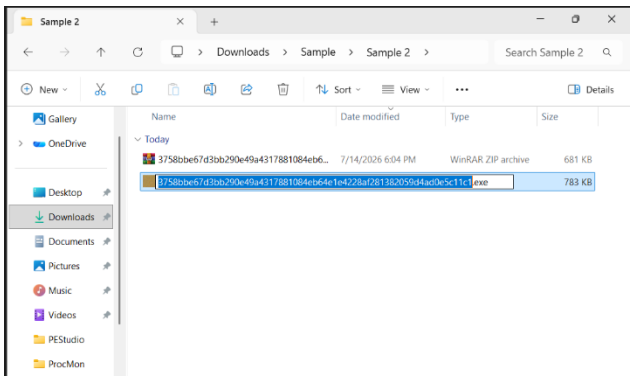

Figure 10 Ekstensi file sampel 2 .exe

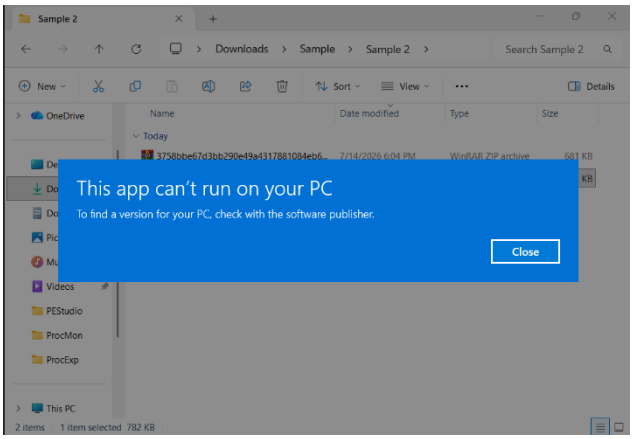

Figure 11 Sampel 2 tidak bisa dijalankan seperti file .exe pada umumnya

Meskipun analisis dinamis tidak dapat dilakukan pada Sampel 2, analisis statis tetap memberikan informasi mengenai karakteristik dan potensi kemampuan sampel, termasuk indikator yang berkaitan dengan manipulasi *file system* dan *memory*. Temuan ini menunjukkan pentingnya analisis statis sebagai tahap awal dalam pendekatan *hybrid*, khususnya dalam mengidentifikasi karakteristik internal file sebelum melakukan analisis perilaku. Dalam kasus Sampel 2, analisis statis membantu mengungkap bahwa file yang tampak sebagai *executable* berdasarkan ekstensi `.exe` sebenarnya memiliki karakteristik sebagai DLL 64-bit.

C. Analisis Hybrid

Hasil analisis *hybrid* pada sampel 1 menunjukan adanya korelasi antara beberapa temuan analisis statis dan dinamis. Pada analisis statis, ditemukan API `AdjustTokenPrivileges` yang berkaitan dengan pengaturan proses hak akses (*privilege*). Temuan ini didukung oleh hasil analisis dinamis, di mana saat sampel 1 dijalankan, sistem menampilkan permintaan UAC untuk memberikan hak administrator kepada `Synaptics.exe.` Selain itu, analisis statis mengidentifikasi beberapa API yang berkaitan dengan *file system*, seperti `CopyFileA`, `CreateDirectoryA`, dan `DeleteFileA`, sedangkan analisis dinamis menggunakan *Process Monitor* menunjukkan adanya interaksi dengan *file system* melalui operasi `CreateFile` dan `ReadFile`. Hal ini menunjukkan adanya kesesuaian antara kemampuan yang teridentifikasi secara statis dan aktivitas yang diamati selama eksekusi saat analisis dinamis.

Analisis dinamis juga menunjukkan adanya aktivitas terhadap registri melalui operasi seperti `RegOpenKey` dan `RegQueryValue`, serta akses terhadap *file system* Windows dan DLL. Namun, tidak ditemukan aktivitas modifikasi registri, pembuatan *file*, atau aktivitas terkait persistensi. Sementara itu, beberapa API yang teridentifikasi pada analisis statis, seperti `GetClipboardData`, `GetKeyState`, dan `CallNextHookEx`, belum terlihat saat analisis dinamis. Secara keseluruhan, hasil analisis *hybrid* menunjukkan bahwa sebagian kemampuan yang teridentifikasi melalui analisis statis memiliki korelasi dengan perilaku yang

diamati secara dinamis, sementara beberapa kemampuan lainnya belum teramati selama sampel dieksekusi.

Pada Sampel 2, hasil analisis *hybrid* menunjukkan bahwa analisis statis memberikan informasi yang tidak dapat diperoleh melalui analisis dinamis. Analisis statis mengidentifikasi beberapa API yang berkaitan dengan interaksi dan manipulasi *file system*, seperti `CopyFileW`, `CreateDirectoryW`, `FindFirstFileExW`, `FindNextFileW`, dan `WriteFile`, serta API `VirtualProtect` yang berkaitan dengan manipulasi proteksi memori. Namun, ketika dilakukan analisis dinamis, Sampel 2 tidak dapat dijalankan secara normal karena *file* yang menggunakan ekstensi `.exe` teridentifikasi sebagai *Dynamic Link Library* (DLL). Oleh karena itu, perilaku *runtime* dari API yang teridentifikasi secara statis tidak dapat diamati.

Secara keseluruhan, hasil analisis *hybrid* menunjukkan bahwa analisis statis dan dinamis memiliki peran yang saling melengkapi dalam mengidentifikasi karakteristik dan perilaku *malware*.

D. Kesimpulan

Berdasarkan hasil analisis terhadap dua sampel *AsyncRAT*, pendekatan analisis *hybrid* yang menggabungkan analisis statis dan dinamis memberikan gambaran yang lebih komprehensif mengenai karakteristik dan perilaku *malware*. Analisis statis berhasil mengidentifikasi berbagai indikator dan potensi kemampuan malware, termasuk interaksi dengan *file system*, manipulasi memori, pengaturan hak akses, serta karakteristik internal file.

Secara keseluruhan, hasil penelitian menunjukkan bahwa analisis statis dan dinamis memiliki peran yang saling melengkapi dalam proses analisis *malware*. Penggabungan kedua pendekatan tersebut memungkinkan identifikasi karakteristik internal sekaligus pengamatan terhadap perilaku *malware* selama dijalankan di dalam sistem, sehingga dapat memberikan pemahaman yang lebih menyeluruh terhadap sampel yang dianalisis.

V. DAFTAR PUSTAKA

[1] Parulekar, W. R., Salvi, H. U., Jikamade, K. M., & Rajendran, S. (2023). Comprehensive Overview of The Techniques, Challenges and Future Prospects of Dynamic *Malware* Analysis.

[2] "Understanding Trojan Viruses: The Hidden Threat in Cyberspace," Telkom University Master of Science in Informatics (accessed May 8, 2026).

[3] *Malware*bytes, "What Is a RAT (Remote Access Trojan)?" *Malware*bytes Blog (Accessed: May 8, 2026.)

[4] Kara, İ., & Aydos, M. (2019). The ghost in the system: technical analysis of remote access trojan. International Journal on Information Technologies & Security, 11(1), 73-84.

[5] [5] BELEA, A. R. (2023, May). Methods for detecting malware using static, dynamic and hybrid analysis. In Proceedings of the International Conference on Cybersecurity and Cybercrime-2023 (pp. 258-265). Asociatia Romana pentru Asigurarea Securitatii Informatiei.

[6] [7] PEStudio, "PEStudio – Free Windows EXE Analyzer & Malware Checker." PEStudio. Available: https://pestudio.org/ (Accessed: Jun. 30, 2026.)

[7] GeeksforGeeks, "Malware Analysis Using Process Monitor (Procmon)." GeeksforGeeks. Available: https://www.geeksforgeeks.org/ethical-hacking/malware-analysis-using-process-monitor-procmon/ (Accessed: Jun. 30, 2026).

[8] Nas Bench, "Hunting Malware with Windows Sysinternals Process Explorer." Medium. Available: https://nasbench.medium.com/hunting-malware-with-windows-sysinternals-process-explorer-2baec974bec9 (Accessed: Jun. 30, 2026.).

[9] Andronache, M. M., Vulpe, A., & Burileanu, C. (2025). Integrated Analysis of Malicious Software: Insights from Static and Dynamic Perspectives. Journal of Cybersecurity and Privacy, 5(4), 98.

[10] Maghanaki, M., Keramati, S., Chen, F. F., & Shahin, M. (2026). Systematic evaluation of machine learning and deep learning models for IoT malware detection across ransomware, rootkit, spyware, trojan, botnet, worm, virus, and keylogger. Sensors, 26(6), 1750.

[11] Emre, "Hunting the Unseen: How Malware Uses Entropy." Medium. Available: https://emreozr.medium.com/hunting-the-unseen-how-malware-uses-entropy-aec6f5535435 (Accessed: Jul. 12, 2026.).

[12] D. Neemani and D. Antonov, "How Attackers Masquerade and Abuse Digital Signatures in DLL Side-Loading." Fortinet Community. Available: https://community.fortinet.com/blogs-103/how-attackers-masquerade-and-abuse-digital-signatures-in-dll-side-loading-218688 (Accessed: Jul. 12, 2026.).