Analisis Malware dengan Metode Hybrid untuk Mengidentifikasi *Indicator of Compromise*

Muhammad Wahyu Syafi'uddin[1], Bana Handaga[2]

[1] Universitas Muhammadiyah Surakarta, Jl. A. Yani, Mendungan, Pabelan, Kec. Kartasura, Kabupaten Sukoharjo, Jawa Tengah, Indonesia

[2] Universitas Muhammadiyah Surakarta, Jl. A. Yani, Mendungan, Pabelan, Kec. Kartasura, Kabupaten Sukoharjo, Jawa Tengah, Indonesia

Email korespondensi: l200210056@student.ums.ac.id

Abstrak. Ancaman siber merupakan konsekuensi logis dari peningkatan aktivitas digital manusia pada zaman ini. Perangkat lunak berbahaya muncul sebagai tantangan nyata yang secara signifikan dan terorganisir mampu melampaui persepsi korban yang sebelumnya hanya menganggap sebagai gangguan kecil yang bersifat tidak serius. Dalam upaya melindungi dunia digital dari ancaman baru yang memiliki dorongan politik dan keuntungan finansial, maka dibutuhkan sebuah proses analisis guna mendapatkan informasi yang lebih cepat dibandingkan dengan penyebaran perangkat lunak berbahaya tersebut. Sesuai anjuran dari NIST, diperlukan adanya integrasi dan penyelarasan manajemen risiko keamanan siber, dengan melakukan pembagian informasi berupa Indicator of Compromises (IoC) yang diperoleh dengan cara melakukan analisis secara statis dan dinamis. Pada penelitian ini, akan dilakukan analisis hibrid kepada lima sampel perangkat lunak berbahaya dengan mengambil informasi yang menjadi indikasi adanya aktivitas perangkat lunak di dalam sebuah sistem. Metode yang cocok untuk penelitian ini yaitu metode eksperimental dengan langkah yang dimulai dari persiapan lingkungan virtual, analisis statis, analisis dinamis, dan dilanjutkan ke tahap data parsing untuk dilakukan penyeragaman dengan standarisasi menggunakan YARA dan SIGMA rules. Berdasarkan hasil penelitian ini, dapat disimpulkan bahwa analisis statis dapat mengambil informasi tanpa perlu adanya konfigurasi lingkungan virtual, sehingga dapat mempercepat alur analisis, serta analisis dinamis dapat diandalkan untuk memvalidasi temuan-temuan yang dihasilkan pada

proses analisis statis dari adanya temuan yang false positive ataupun true negative.

Kata kunci: *Analisis Hybrid; Indicator of Compromises (IoC); Keamanan Siber; Malware; Sigma; Yara*

PENDAHULUAN

Peningkatan konektivitas telah mendorong transformasi bisnis dan kehidupan sehari-hari, tetapi juga membawa konsekuensi logis berupa tantangan yang signifikan dalam bentuk ancaman siber, seperti serangan ransomware, penargetan data sensitif, dan ancaman dari insider [1]. Perangkat lunak berbahaya atau yang biasa disebut malware juga muncul sebagai ancaman yang signifikan dan terorganisir hingga dapat melampaui persepsi sebelumnya yang hanya dianggap sebagai gangguan kecil. Berdasarkan penelitian dari [2], dijelaskan bahwa industri malware, terutama melalui model seperti Malware-as-a-Service (MaaS), telah berkembang menjadi perusahaan bernilai jutaan dolar, yang menghasilkan pendapatan melalui akses yang tidak sah ke sistem komputer dengan tujuan untuk mencuri kredensial dan data sensitif, perilaku ini didorong oleh motif ekonomi yang bertujuan untuk mendapatkan keuntungan finansial hingga dorongan geopolitik. Tren terbaru pada tahun 2024 menunjukkan hasil eskalasi risiko yang cukup mengkhawatirkan, menurut hasil laporan yang diunggah dari website any.run, dikatakan bahwa peningkatan deteksi malware berjenis info stealer naik sebesar 180% dari tahun sebelumnya [3]. Perangkat lunak berbahaya dengan jenis info stealer, atau secara umum disebut stealer memiliki kemampuan untuk mencuri berbagai informasi seperti kredensial, cookie, metadata dari browser, hingga perilaku pengguna [4]. Hasil dari laporan ini tidak hanya mencerminkan kerumitan ancaman keamanan siber modern, tetapi juga menegaskan pentingnya pengembangan strategi yang lebih proaktif dan lebih terkoordinasi untuk menghadapi tantangan tersebut.

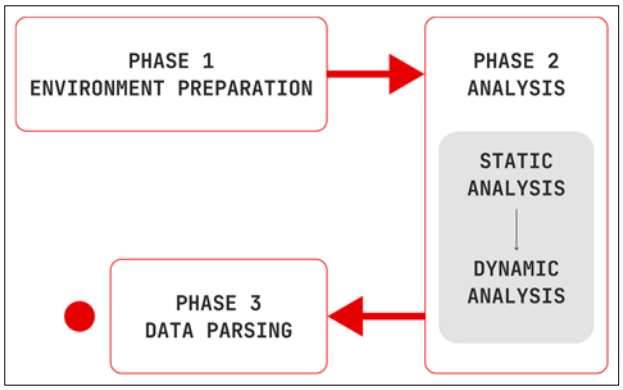
Keterlibatan lintas sektor yang dapat di inisiasi dari pemerintah, pelaku bisnis, hingga masyarakat digital menjadi krusial untuk dapat membentuk kesiagaan secara mandiri dalam menghadapi ancaman keamanan siber serta dapat mendukung inisiatif nasional yang menargetkan peningkatan kapasitas teknologi dan sumber daya manusia. Sebagai upaya untuk mengurangi risiko keamanan siber yang diakibatkan oleh adanya malware yang semakin beragam, harus dilakukan dengan cara yang lebih proaktif, seperti yang dianjurkan pada NIST Framework yaitu, suatu organisasi dapat melakukan integrasi dan penyelarasan manajemen risiko keamanan siber, yang dapat dilakukan dengan melakukan pembagian informasi indicator of compromises (IoC) yang berisi informasi seperti alamat IP, Domain, hingga pola perilaku dari suatu sampel malware [5]. Dengan menerapkan prosedur ini, organisasi yang terkena ancaman keamanan siber

dapat berbagi informasi berupa IoC kepada organisasi lainnya untuk merespons risiko yang diakibatkan oleh malware dengan harapan dapat direspons lebih dini.

Terdapat banyak pendekatan untuk mendapatkan IoC dan tidak sedikit threat actor yang mencoba menyembunyikan proses malware mereka bekerja, sehingga diperlukan sebuah analisis yang mendalam terhadap sampel malware yang ditemukan. Proses analisis dapat dilakukan secara statis dengan mengamati bentuk dan ciri khas suatu malware untuk memperoleh informasi secara cepat. Namun, metode analisis statis sering kali tidak memberikan informasi sepenuhnya karena proses obfuscation yang dilakukan oleh pembuat malware. Agar bisa mendapatkan informasi yang lebih akurat, proses analisis statis dapat dibantu dengan melakukan analisis secara dinamis untuk mempelajari perilaku malware. Pada penelitian ini, proses analisis dinamis dilakukan dengan cara mengamati proses malware berjalan di dalam lingkungan yang terkontrol dengan bantuan Virtual Machine yang sudah disiapkan berbagai alat untuk melakukan monitoring jaringan, system log, dan registry. Dengan menggabungkan dua metode analisis statis dan dinamis atau yang dikenal dengan hybrid analysis, IoC dapat dengan cepat dan akurat untuk ditemukan.

METODE

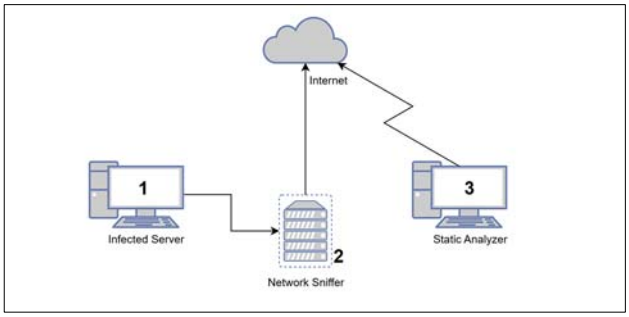
Metode merupakan sebuah cara ilmiah yang disusun secara sistematis dan logis untuk menyelesaikan suatu masalah [6]. Dengan banyaknya variasi malware berjenis stealer yang juga berevolusi semakin canggih [7]. maka metode yang sesuai untuk menyusun penelitian "Analisis Malware Dengan Metode Hybrid Untuk Mengidentifikasi Indicator Of Compromise" yaitu menggunakan metode eksperimental yang dapat menguji dan mengetahui keadaan dari sistem yang sedang dalam proses pengujian yang berbeda-beda [8]. Alur analisis malware yang akan dilakukan pada penelitian sesuai dengan Gambar 1, dimulai dengan persiapan lingkungan virtual, pada fase ini mesin-mesin virtual akan dibuat dan dipersiapkan, kemudian langkah analisis yang memiliki dua tahap yaitu statis dan dinamis, dan langkah terakhir adalah melakukan standarisasi temuan dengan skrip otomatis.



Gambar 1. Alur Analisis Malware Secara *Hibrid* dengan Tiga Fase

Persiapan Lingkungan Virtual

Pada proses ini, mencakup penyiapan lingkungan virtual untuk analisis malware yang aman menggunakan virtual machine VMware dengan spesifikasi 4 core CPU, 4 GB RAM, dan 64 GB penyimpanan. Lingkungan ini dilengkapi alat pengawasan untuk memantau lalu lintas jaringan, proses sistem operasi Windows, aktivitas registry, serta pembuatan atau penghapusan file oleh malware. Konfigurasi jaringan melibatkan pemasangan sniffer antara mesin virtual dan internet untuk merekam lalu lintas jaringan. Pada Mesin virtual untuk analisis statis tidak memerlukan sniffer karena tidak mengeksekusi malware. Seperti pada Gambar 2, Windows 10 terhubung dengan internet melewati Ubuntu sebagai gateway, model ini dibutuhkan karena dalam penelitian ini harus dapat membaca lalu lintas jaringan yang lewat dan digunakan Ubuntu dengan spesifikasi yang cukup, model lain adalah dengan menggunakan Test Access Point (TAP) yang tidak membutuhkan sumber daya yang tinggi dan dapat diaplikasikan pada perangkat jaringan fisik. Spesifikasi serta perangkat lunak yang dibutuhkan diringkas pada Tabel 1 yang memuat data spesifikasi perangkat keras dan perangkat lunak yang digunakan, serta keterangan penggunaan.

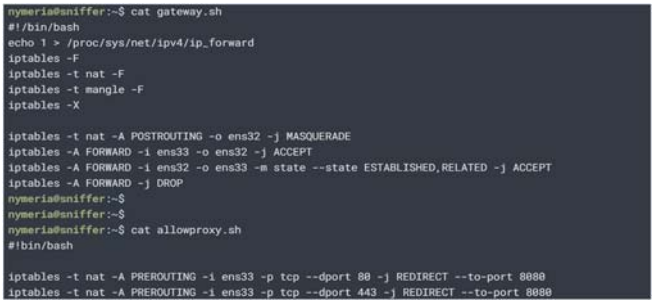


Gambar 2. Topologi Lingkungan Virtual Untuk *Analysis Hybrid*

Tabel 1. Spesifikasi Perangkat Keras dan Perangkat Lunak

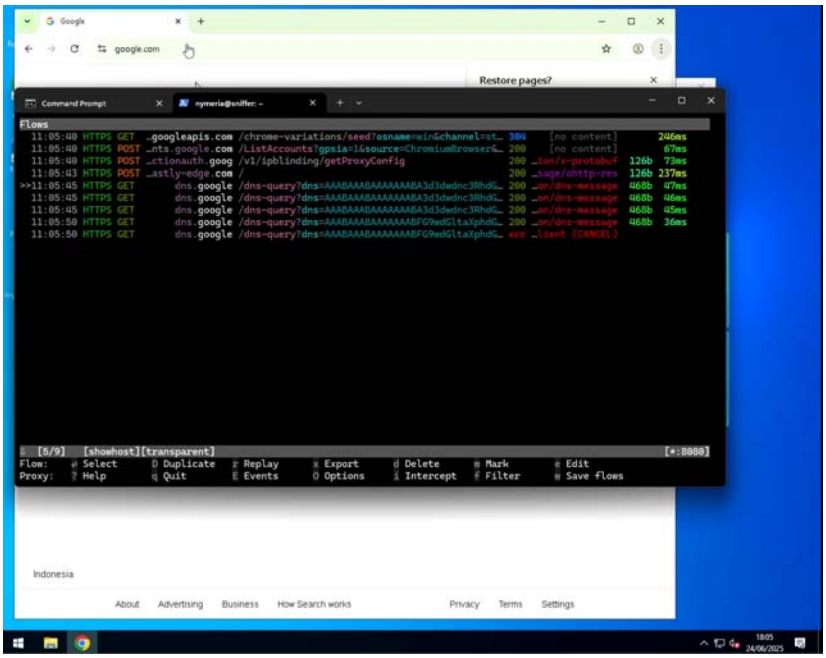
No.	Perangkat Keras	Perangkat Lunak	Keterangan
1	4 Core CPU 4 GB RAM 64 GB Storage	- Windows 10 x64 - Process Monitor - RegShot - Sysmon	Komputer guest yang akan digunakan untuk menjalankan malware
2	2 Core CPU 2 GB RAM 32 GB Storage	- Ubuntu Server 24.04 - Iptables - MITM Proxy - Wireshark	Komputer server yang digunakan untuk menjalankan MITM Proxy dan Wireshark
3	6 Core CPU 8 GB RAM 256 GB Storage	- Windows 11 x64 - Detect it Easy - Ghidra - Oletools	Komputer yang digunakan untuk melakukan malware static analysis

Persiapan lingkungan dimulai pada Host Machine yang menjalankan Windows 11 x64 dengan memasang VMWare Workstation Pro yang akan menjadi perangkat lunak yang mensimulasikan komputer untuk menjalankan Windows 10 dan Sniffer Jaringan. Terdapat dua mesin virtual yang berjalan pada VMWare secara bersamaan, yaitu Windows dan Ubuntu Server, kedua mesin ini akan dihubungkan dengan menggunakan Virtual Network Adapter yang menggunakan metode Host-Only agar terisolasi dari jaringan internet untuk menghindari kebocoran lalu lintas, kemudian interface yang digunakan untuk mengakses internet akan dipasang pada Ubuntu server dan dilakukan proses routing jaringan antara adaptor Host-Only ke NAT milik Ubuntu Server menggunakan iptables, perintah untuk mengkonfigurasi rute menggunakan iptables dapat dilihat pada Gambar 3, yaitu dengan meneruskan seluruh traffic yang masuk dari antarmuka ens33 (host-only) ke antarmuka ens32 yang memiliki akses internet, dan sebaliknya semua status jaringan yang sudah *ESTABLISHED* akan dikembalikan ke ens33 dari ens32 agar koneksi dapat berjalan secara full-duplex.



Gambar 3. Script Otomatis untuk Mengkonfigurasi Iptables Agar Menjadi Gateway dan Proxy

Kemudian perangkat lunak MITM Proxy dibutuhkan untuk merekan lalu lintas http dengan mengkonfigurasi iptables untuk meneruskan semua port default http dan https ke port 8080 dimana merupakan proxy http yang dijalankan oleh MITM Proxy seperti yang terlihat pada Gambar 4. Dengan menggunakan konfigurasi ini, maka seluruh jaringan yang masuk, keluar, dan terenkripsi pada Windows 10 dapat direkam oleh Ubuntu yang bekerja sebagai gateway.



Gambar 4. Lalu Lintas Https yang Dibaca Oleh MITM Proxy Tanpa Adanya Enkripsi

Pada penelitian ini juga dibutuhkan sampel malware yang dapat ditemukan di internet secara acak dan sudah dicatat pada Tabel 2 yang memuat nama malware, nama file yang ditemukan, dan tipe file dari malware. Secara umum, malware akan menyembunyikan dirinya sendiri ke sebuah proses palsu, sehingga terdapat kemungkinan jika nama file dibuat seakan seperti nama file asli. Dalam melakukan analisis, malware akan dijalankan secara bergantian dengan melakukan *rollback* mesin virtual Windows 10 agar hasil infeksi malware pertama tidak akan mempengaruhi proses analisis pada malware kedua dan seterusnya.

Tabel 2. Sampel *malware* berjenis *stealer* yang akan dianalisis

No.	Nama Stealer	Filename	File Type
1	Exela Stealer	Growtopia Proxy.exe	Windows PE (Portable Executable)
2	Lumma Stealer	[Random].exe	Windows PE (Portable Executable)
3	Formbook	Quotation.xls	Microsoft Excel
4	Redline Stealer	0645c8287be23a2f83c68797ed6afb38.docx	Microsoft Word
5	Lumma Stealer	The Wolf Among Us 2 instruction for YouTube partners.pdf	Portable Document Format

Tahap Analisis

Analisis malware adalah proses investigasi sistematis untuk memperoleh informasi komprehensif dari malware, yang menjadi dasar pengembangan Indicator of Compromises (IoC) untuk memitigasi risiko keamanan. Analisis ini juga mendukung perancangan sistem otomasi dengan REST API untuk membantu pencegahan dan penanganan agar lebih efektif. Secara metodologis, analisis malware terbagi menjadi dua pendekatan utama, yaitu analisis statis dan dinamis, yang dapat digabungkan menjadi metode hybrid yang menjalankan kedua teknik analisis.

Analisis Statis

Menurut yang disampaikan oleh [9], Analisis malware secara statis dilakukan tanpa mengeksekusi malware dengan tujuan mendapatkan informasi tanpa harus menunggu malware tereksekusi pada mesin korban. Teknik analisis ini melibatkan proses reverse engineering dan ekstraksi string yang ada dalam struktur biner sebuah aplikasi. Metode ini lebih cepat karena tidak menganalisis sebab dan akibat dari proses eksekusi malware. Hasil dari analisis statis ini dapat berupa kode dari tingkat rendah seperti kode bahasa Assembly hingga kode tingkat tinggi seperti halnya pseudo-code, dan dapat juga ditemukan proses malware melewati pemeriksaan keamanan, seperti mekanisme anti-debugger dan anti-virtual machine yang sering ditemui pada malware untuk menghindari proses identifikasi.

Analisis Dinamis

Proses analisis malware secara dinamis melibatkan eksekusi malware dalam lingkungan terkendali, seperti menggunakan mesin virtual dan jaringan yang terkontrol [10] untuk mencegah kerugian dan kerusakan sistem yang diakibatkan oleh malware Proses ini bertujuan untuk mengamati perilaku malware secara langsung yang mencakup perilaku malware dalam proses mengirimkan program-program berbahaya lainnya atau

biasa disebut dengan malware dropper, ataupun perilaku malware yang menyerang langsung pada target. Analisis ini juga mengidentifikasi aktivitas seperti eskalasi hak akses, penulisan ke registry, atau penjadwalan tugas untuk memastikan adanya persistensi malware yang membuat malware tidak dapat hilang setelah mesin dimatikan dengan menggunakan alat seperti Wireshark, Sysmon, dan RegShot.

Data Parsing

Setelah mendapatkan hasil static analysis dan dynamic analysis, maka langkah terakhir adalah melakukan pemrosesan terhadap data-data tersebut untuk dijadikan sebuah indikator yang menandakan adanya intrusi ke dalam sebuah sistem. Data parsing yang akan dilakukan pada penelitian ini dilakukan secara otomatis dengan melalui API yang sudah disiapkan untuk mengkonversi data mentah yang disebut dengan artifact menjadi ruleset yang sudah terstandarisasi dan siap di implementasikan pada alat-alat deteksi.

Adapun luaran dari IoC ini adalah sebuah ruleset yang akan menerapkan beberapa standar-standar yang sering diaplikasikan pada sebuah Threat Intelligence yaitu menggunakan YARA yang merupakan standarisasi format untuk mengidentifikasi dan mengklasifikasi malware dengan cara membandingkan dengan aturan-aturan atau rules yang akan dibuat pada saat data parsing [11] dengan harapan jika ditemukan sampel malware, akan dapat lebih cepat mendeteksi dengan YARA. Sementara itu, SIGMA Rules juga akan digunakan untuk mengambil behavior hasil dari eksekusi Sysmon yang berupa DNS Query, dan Network Logs.

HASIL

Hasil Analisis

Hasil Analisis Statis

Berdasarkan pada topologi yang dijelaskan pada Gambar 2, proses analisis malware secara statis dilakukan dalam lingkungan yang menjalankan Windows 11 x64 dengan akses ke internet secara langsung. Hal ini dilakukan karena proses analisis statis tidak membutuhkan interaksi dengan malware secara langsung sehingga tidak menimbulkan proses infeksi pada komputer.

Dalam melakukan analisis statis, terdapat alur kerja yang akan dilakukan yaitu dimulai dengan melakukan deteksi awal dengan menggunakan perangkat Detect it Easy untuk mendapatkan informasi detail terkait file executeables dengan mendapatkan library yang digunakan pada file, bahasa pemrograman yang dipakai, serta proteksi-proteksi dasar yang terpasang. Kemudian ditambah dengan melakukan strings analysis dengan program dari sysinternals untuk mengambil text dengan kondisi tertentu dari file executeables. Strings merupakan kata dengan panjang karakter tertentu yang di simpan

pada sekmen .data ataupun .rdata (read-only data) pada sebuah binary, informasi ini dapat berupa nama fungsi, nama variabel, nama library, dan semua yang disimpan secara hardcoded dan tidak diproses oleh compiler. Hasil dari analisis statis ini akan menghasilkan informasi berupa File Signature, OLE dump, Strings, System Calls dan informasi detail terkait proses jalannya sebuah malware yang dapat digunakan sebagai informasi pendukung untuk analisis dinamis.

Pada proses analisis File Signature, digunakan sebuah powershell script untuk mendapatkan signature MD5, SHA1, dan SHA256 secara otomatis. Script ini akan mengambil nama file dari input console, dan kemudian melakukan pengulangan eksekusi untuk mendapatkan hash MD5, SHA1, dan SHA256 yang pada akhirnya akan ditampilkan hasilnya. Pada Tabel 3 dibawah ini merupakan ringkasan hasil dari proses analisis yang mencakup hash dari masing-masing sampel diurutkan berdasarkan nomor sampel terkait.

Tabel 3. Hasil Perhitungan *File Signature*

No.	MD5	SHA1	SHA256
1	a974a9eb22e21d1e2 9305ba98bba0bb6	8c504ce7c215598b12b9 c4c1cf9a99d443105ecd	448404fa5ea865d87c14c782309facd20f241 b12455396b97e7e3f4dce65c2b7
2	3fef586415c1c48517 c6f2f3ca7a9b8d	cf4f14a4fa764500fb7f9a 4717101d2c55ed77a6	17bd201df28bae2fbdf1861221fbffd0ef64f 9b787cc4d8bf2042929137cb9d5
3	97274c5ce0a5f9fa80 e43549e7cd9564	d9459c2a8e1f10b32d37 23062769b703c4c2465f	8665f5fd5f4dd05793ffbdb16fb892d6393e 65ea7f79c6d9716c79817c93845a
4	0645c8287be23a2f83 c68797ed6afb38	66e1723877faffc58975a e88efccd89c5f4aaee4	de9551cb55ec7515ea71b124dfea3ca609fe 0442265d247fb3d744546770ed96
5	35dd2ebdaca625f39 7e0be0705acec0a	b793a963dc4063b2481b 79974906c3e9f9533b3c	e4d9fb58159ee1189b286fc3cabde1bf180e bc46c01e78c6e41656b7a0d00cb2

Proses analisis selanjutnya adalah mendapatkan OLE Data pada setiap sampel, dapat diperhatikan bahwa OLE merupakan struktur penyimpanan data yang diimplementasikan oleh Microsoft pada dokumen-dokumen Office mereka seperti Word, Excel, dan Powerpoint. Pada proses ini, digunakan alat dari https://github.com/decalage2/oletools untuk mendapatkan data OLE. Karena OLE dikhususkan untuk aplikasi Office, tidak semua sampel didapatkan data OLE. Pada Tabel 4 diberikan ringkasan hasil pemeriksaan OLE untuk stealer nomor 3, 4, dan 5 yang

merujuk pada Tabel 2. Pada hasil nomor 1, didapatkan bahwa malware memiliki kemungkinan adanya VBA Macro yang dapat menjalankan script saat dokumen dibuka, hanya saja saat analisis statis ini dibutuhkan password untuk mendapatkan script tersebut, kemudian pada data nomor 2, terdapat VBA Macro juga dengan tingkat risiko tinggi dan juga terdapat script powershell yang akan berjalan saat dokumen dibuka, powershell script tersebut akan mengunduh malware lain dan mengeksekusi malware tersebut. Dan pada data nomor 3, tidak didapatkan indikasi apapun pada struktur data OLE.

Tabel 4. Hasil Pemeriksaan OLE Data untuk *Stealer* dengan tipe Microsoft Office

No.	Output
1	VBA Macro is Present with Risk Medium Empty Macros in ThisWorkbook, Sheet1, Sheet2, Sheet3 Suspicious Hex : { 00 02 08 19 }; { 00 00 00 00 00 0F }; { 00 02 08 20 } FILEPASS record found, file is password protected
2	VBA Macro is Present with **Risk High** (suspicious) VBA Macro at ThisDocument.cls with OLE Stream: 'VBA/ThisDocument' Found powershell payload: cABvAHcAZQByAHMAaABlAGwAbAAuAGUAeABlACAALQBFAHgAZQBjAHU AdABpAG8AbgBQAG8AbABpAGMAeQAgAGIAeQBwAGEAcwBzACAALQBuAG 8AcAByAG8AZgBpAGwAZQAgAC0AdwBpAG4AZABvAHcAcwB0AHkAbABlAC AAaABpAGQAZABlAG4AIAAtAGMAbwBtAG0AYQBuAGQAIAAiAE4AZQB3AC 0ASQB0AGUAbQAgAC0AUABhAHQAaAAgACcAQwA6AFwAXABUAGUAbQB wAFwAXAAnACAALQBJAHQAZQBtAFQAeQBwAGUAIABEAGkAcgBlAGMAd ABvAHIAeQA7AEEAZABkAC0ATQBwAFAAcgBlAGYAZQByAGUAbgBjAGUAIA AtAEUAeABjAGwAdQBzAGkAbwBuAFAAYQB0AGgAIAAnAEMAOgBcAFQAZQ BtAHAAXAAnADsAKABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMA dABlAG0ALgBOAGUAdAAuAFcAZQBiAEMAbABpAGUAbgB0ACkALgBEAG8A dwBuAGwAbwBhAGQARgBpAGwAZQAoACcAaAB0AHQAcABzADoALwAvAG IAaQB0AGIAdQBjAGsAZQB0AC4AbwByAGcALwB4ADkAOAA5ADgAOQAvAD gANgA3ADgANgA3ADgAZgBmAC8AZABvAHcAbgBsAG8AYQBkAHMALwB3A G8AcgBkAC4AegBpAHAAJwAsACcAQwA6AFwAXABUAGUAbQBwAFwAXABO

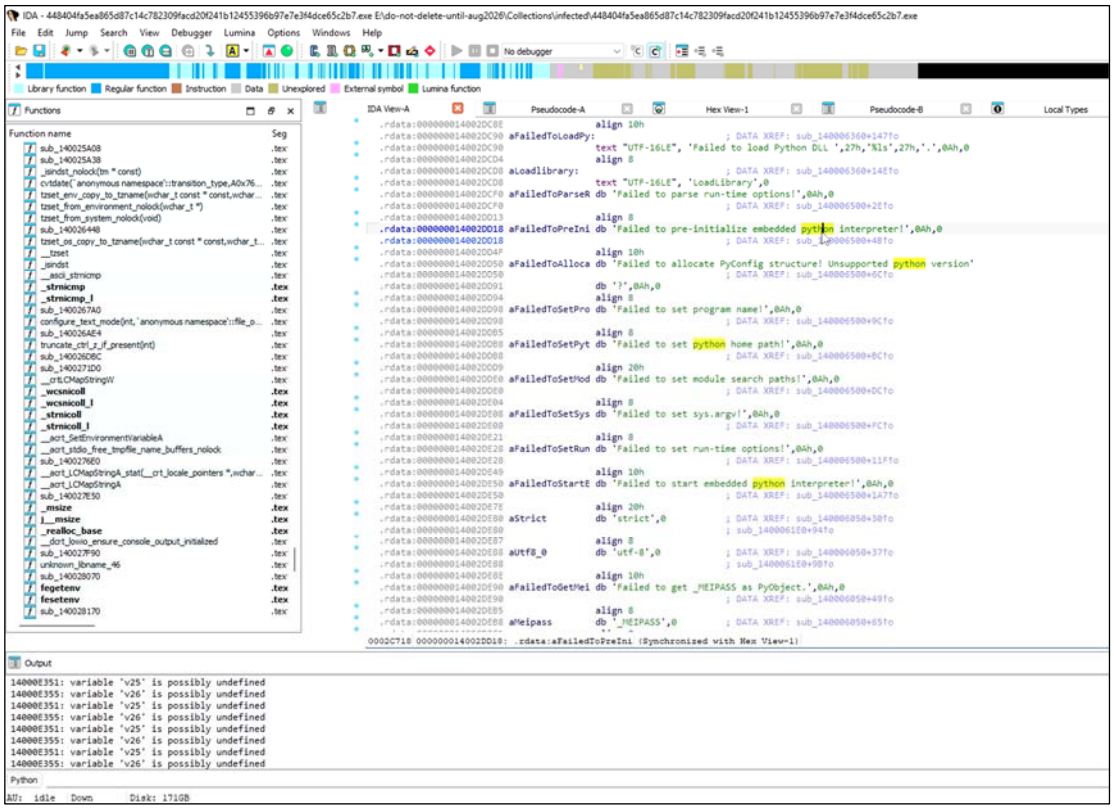
	AGUAdwBmAGkAbABlAC4AegBpAHAAJwApADsARQB4AHAAYQBuAGQALQ BBAHIAYwBoAGkAdgBlACAALQBQAGEAdABoACAAJwBDADoAXABcAFQAZ QBtAHAAXABcAE4AZQB3AGYAaQBsAGUALgB6AGkAcAAnACAALQBEAGUAc wB0AGkAbgBhAHQAaQBvAG4AUABhAHQAaAAgACcAQwA6AFwAXABUAGU AbQBwAFwAXAAnACAALQBGAG8AcgBjAGUAOwBTAHQAYQByAHQALQBQ AHIAbwBjAGUAcwBzACAAYwBtAGQALgBlAHgAZQAgAC0AQQByAGcAdQBt AGUAbgB0AEwAaQBzAHQAIAAnAC8AYwAgAEMAOgBcAFwAVABlAG0AcABc AFwAdwBvAHIAZAAuAGUAeABlACcAIgA= base64 decode: powershell.exe -ExecutionPolicy bypass -noprofile -windowstyle hidden -command "New-Item -Path 'C:\\Temp\\' -ItemType Directory;Add-MpPreference - ExclusionPath 'C:\Temp\';(New-Object System.Net.WebClient).DownloadFile('https://bitbucket.org/x98989/8678678ff/downl oads/word.zip','C:\\Temp\\Newfile.zip');Expand-Archive -Path 'C:\\Temp\\Newfile.zip' -DestinationPath 'C:\\Temp\\' -Force;Start-Process cmd.exe -ArgumentList '/c C:\\Temp\\word.exe'"
3	Tidak ditemukan indikator apapun pada saat pemeriksaan OLE

Analisis strings pada proses ini digunakan limitasi 10 karakter ASCII, sehingga semua karakter yang panjangnya dibawah 10 karakter tidak akan ditampilkan. Data yang diberikan ini dapat mengacu pada Tabel 2 dengan urutan malware nomor 1, 2, dan 5 dengan hasil pada Tabel 5 yang didapatkan strings berupa indikasi adanya kode bahasa python pada nomor 1, adanya konfigurasi anydesk pada data nomor 2, dan adanya http URL ke malware lain pada nomor 3.

Tabel 5. Pemeriksaan Strings yang *Hardcoded* di dalam File dengan minimum 10 karakter

No.	Group Strings
1	Could not get __main__ module's dict. Failed to extract script from archive! Absolute path to script exceeds PYI_PATH_MAX Failed to unmarshal code object for %s

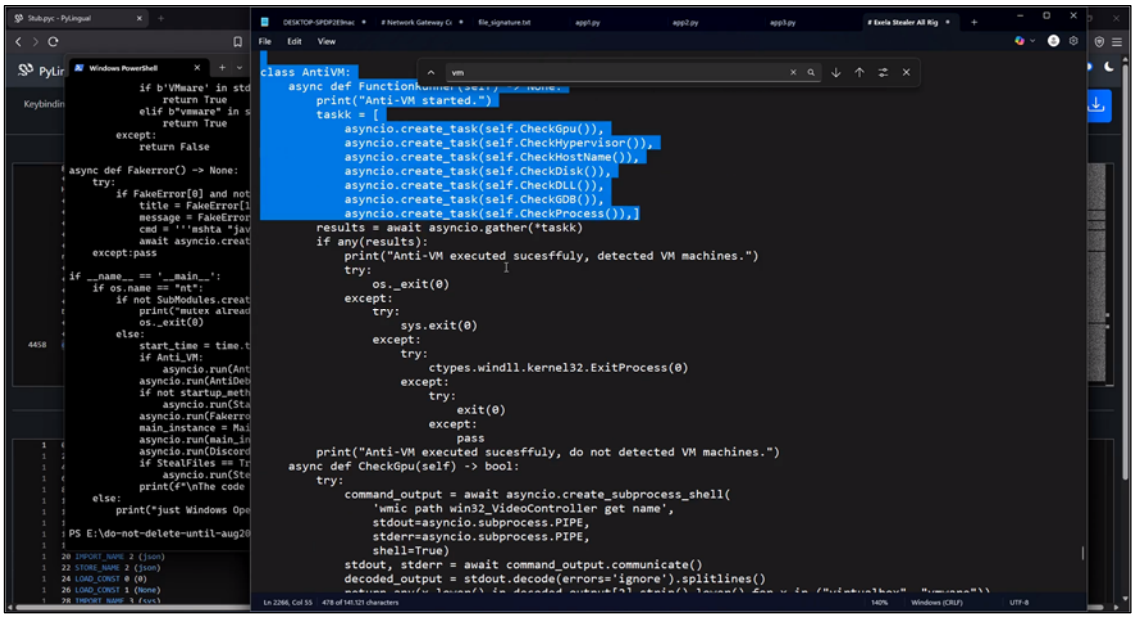
	_pyi_main_co Traceback is disabled via bootloader option. PYINSTALLER_RESET_ENVIRONMENT _PYI_ARCHIVE_FILE _PYI_APPLICATION_HOME_DIR _PYI_PARENT_PROCESS_LEVEL _PYI_SPLASH_IPC Invalid value in _PYI_PARENT_PROCESS_LEVEL: %s PYINSTALLER_STRICT_UNPACK_MODE
2	<assemblyIdentity version="6.3.2.0" processorArchitecture="x86" name="AnyDesk.AnyDesk.AnyDesk" type="win32" /> <description>AnyDesk screen sharing and remote control software.</description> <dependency> <dependentAssembly> <assemblyIdentity type="win32" name="Microsoft.Windows.Common-Controls" version="6.0.0.0" processorArchitecture="x86" publicKeyToken="6595b64144ccf1df" language="*" /> <dpiAware>true/PM</dpiAware> </asmv3:windowsSettings> </asmv3:application>
3	/URI (https://www.dropbox.com/scl/fi/4b0qoxs096pbwp1tlw2oy/The-Wolf-Among-Us-2-promotional-materials-for-YT.rar?rlkey=39ifbdt7v6gsutyvf5gxwspnq&st=xwkdu08h&dl=1) /Type /Action /URI (https://twitter.com/telltalegames?ref_src=twsrc%5Egoogle%7Ctwcamp%5Eserp%7Ctwgr%5Eauthor) /Type /Action /URI (https://telltale.com/) /Type /Action /URI (https://store.epicgames.com/ru/p/the-wolf-among-us-2)



Gambar 5. Proses *decompile malware*

Pada Gambar 5, dilakukan proses review kode sumber menggunakan IDA Pro dengan detail pada window sebelah kiri merupakan nama-nama fungsi yang ada pada file executable, terdapat nama fungsi yang dapat dibaca dan nama fungsi yang diawali dengan sub_ yang artinya fungsi tersebut bukan fungsi yang ada pada sistem tetapi dibuat oleh pemrogram. Kemudian pada windows sebelah kanan merupakan hasil dekompilasi file exe dalam bentuk kode bahasa Assembly. Serta pada bagian bawah merupakan output dari penggunaan IDA yang normalnya digunakan untuk informasi saat debugging. Dari hasil analisis sumber kode tersebut, didapatkan bahwa file exe merupakan hasil kompilasi dari bahasa python yang dijelaskan dalam window sebelah kanan yang terdapat identitas adanya ciri bahasa Python. Karena bahasa python bukan merupakan Compiled Language, sehingga proses reversing dapat dilakukan untuk mendapatkan source code asli yang hanya dilakukan obfuscation saja. Sampai dengan pada Gambar 6, diketahui adanya kelas AntiVM pada kode sumber asli yang melakukan pemeriksaan terhadap lingkungan sistem yang menjalankan malware, sehingga malware tidak akan berjalan jika sistem merupakan lingkungan virtual. Dari data ini, dapat digunakan sebagai langkah memodifikasi lingkungan yang akan digunakan untuk

analisis dinamis, yaitu dengan cara mengubah beberapa value yang terpasang secara default pada vmware menjadi value yang biasanya terdapat pada mesin-mesin asli seperti BIOS.UUID, Hypervisor Name, SSD Name.

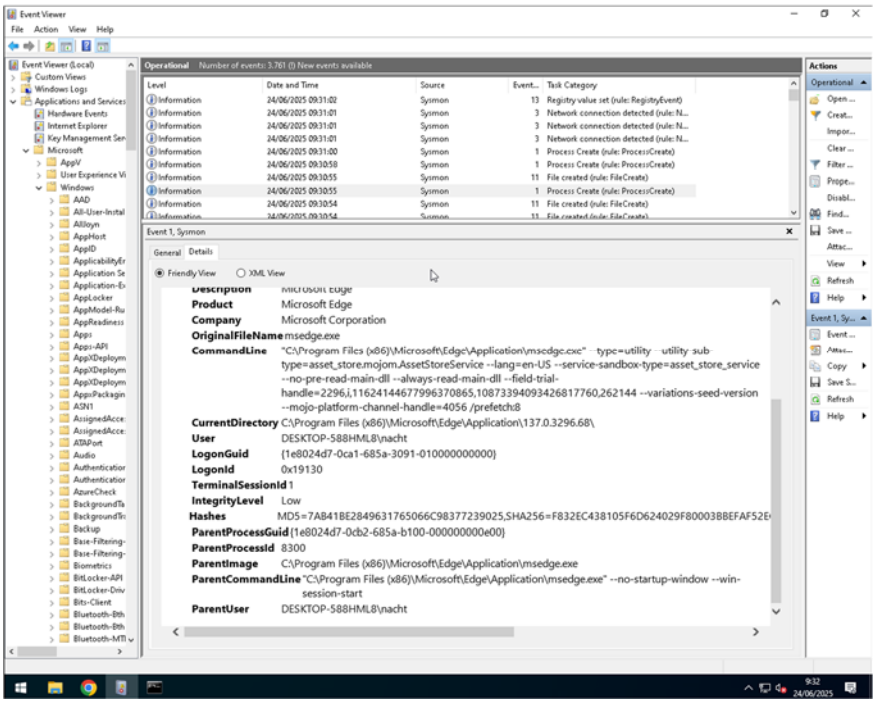


Gambar 6. Mendapatkan adanya fungsi AntiVM

Hasil Analisis Dinamis

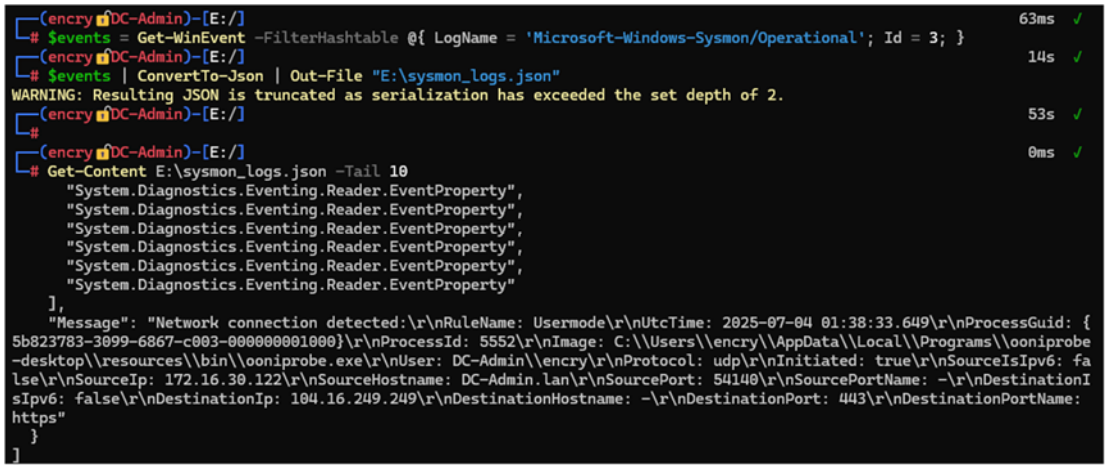
Analisis dinamis dilakukan dengan mengeksekusi lima sampel malware stealer (Tabel 2) dalam lingkungan virtual yang telah disiapkan sesuai dengan yang sudah dijelaskan pada Gambar 2, dan Tabel 1. Pada tahap pertama, sebelum menjalankan malware, Sysmon harus dikonfigurasi terlebih dahulu agar dapat merekam kegiatan-kegiatan yang terjadi pada sistem operasi. Tidak membutuhkan konfigurasi khusus pada Sysmon agar dapat digunakan pada kasus malware analisis, cukup melakukan instalasi dan menjalankan sysmon dengan menggunakan perintah sysmon.exe -accepteula -i sysmonconfig-export.xml.

Sysmon merupakan sebuah layanan sistem yang dapat memantau pembuatan proses, eskalasi proses, akses ke registry, dan DNS Query. Dengan adanya sysmon, didapatkan hasil-hasil berupa sebuah events dengan identitas khusus untuk membantu mendeteksi adanya aktivitas ilegal yang kemudian dapat diolah menjadi SIGMA Rules secara langsung.



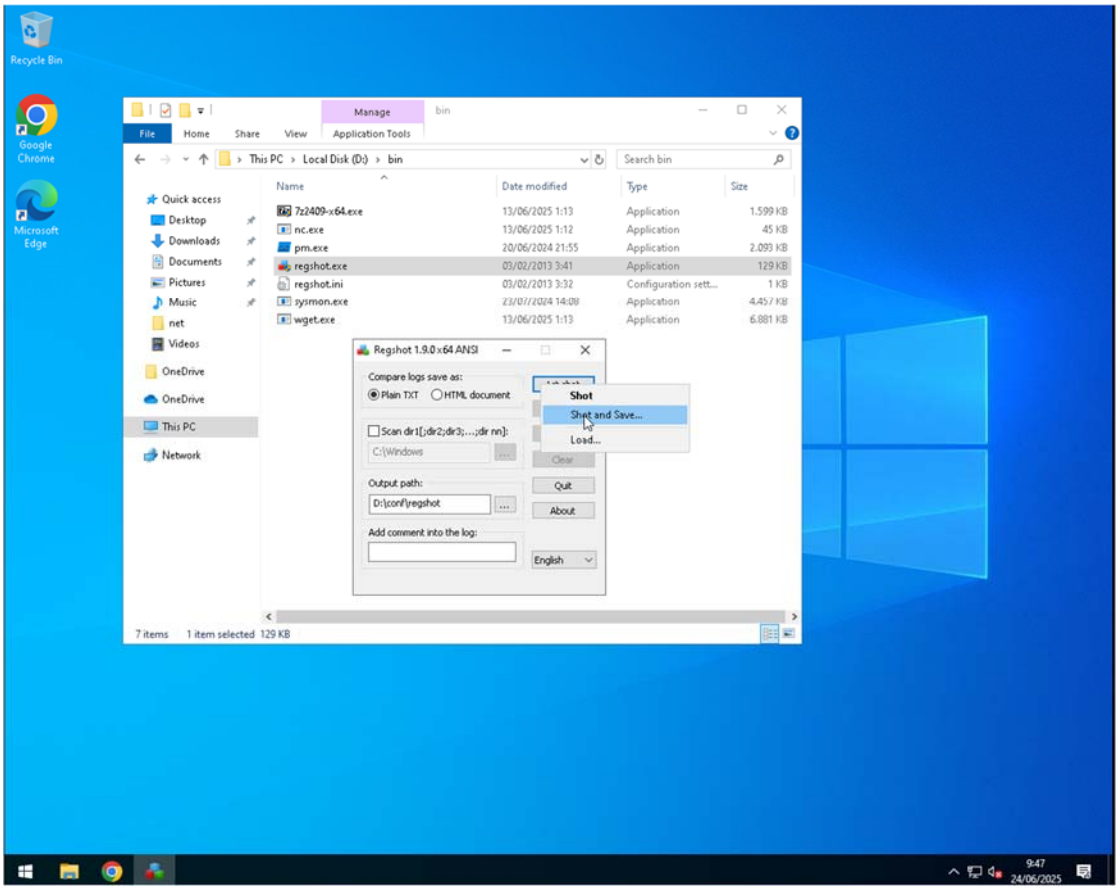
Gambar 7. Log Sysmon pada Windows Event Viewer

Sysmon berjalan pada level service yang dapat dimulai pada kernel space sebelum sistem memasuki user space. Sysmon akan lebih unggul untuk mendeteksi proses persistensi sebuah malware karena sysmon berjalan lebih cepat sebelum malware dieksekusi. Cara melihat log pada sysmon ini dapat dengan memanfaatkan Windows Event Viewer seperti pada Gambar 7 yang menjelaskan detail event yang didapatkan sysmon pada window bagian bawah, list dari seluruh event yang ada pada window bagian atas, dan seluruh event selain sysmon yang ada pada windows di bagian kiri. Tetapi cara yang lebih cepat adalah dengan menggunakan Windows Powershell untuk mengambil log dengan event-event tertentu tanpa harus melihat semua event yang terjadi dengan mengeksekusi perintah yang ditampilkan pada gambar 8, dimana kita mengumpulkan hasil log dengan id 3 yang merupakan Log Jaringan, detail dari variabel event id dapat dilihat pada situs web resmi dari microsoft https://learn.microsoft.com/id-id/sysinternals/downloads/sysmon. Setelah menyimpan log pada variable events, selanjutnya akan dikonversi menjadi JSON dan disimpan ke dalam file.



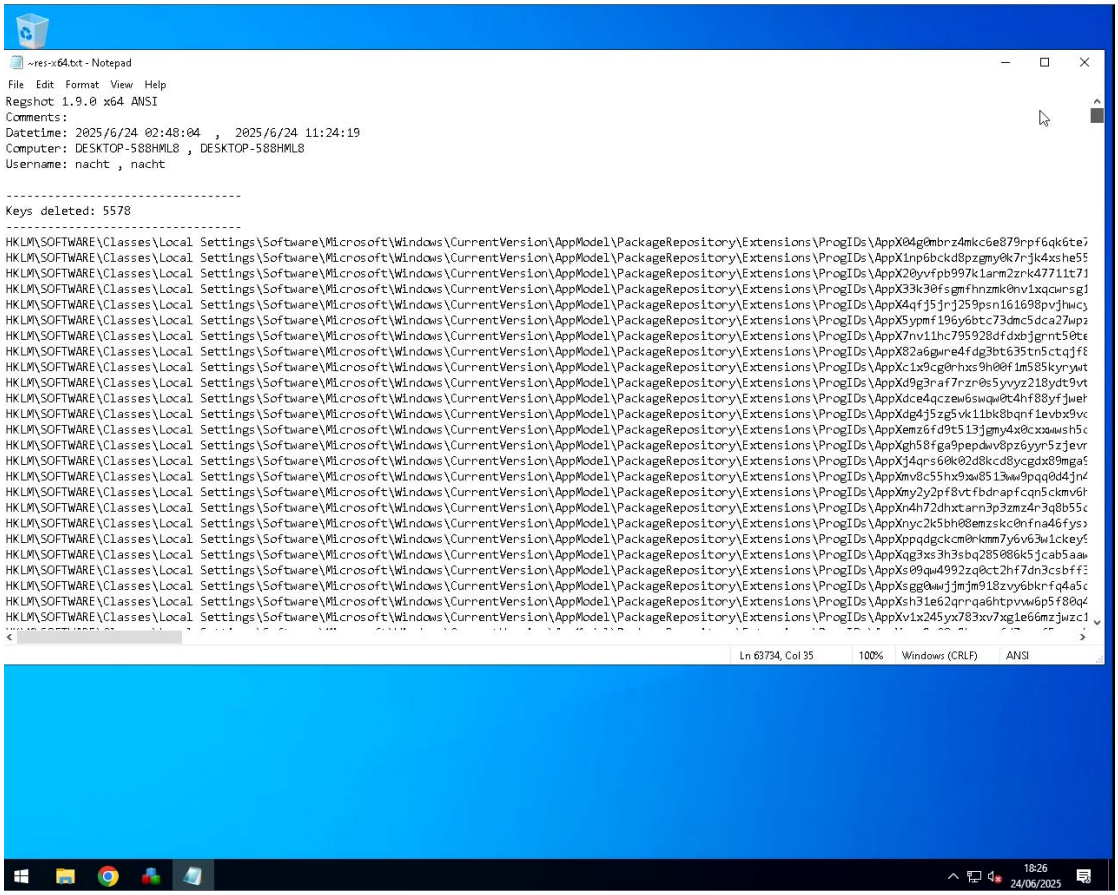
Gambar 8. Mengambil Log Sysmon dengan Perintah Powershell

Windows Registry merupakan database hirarkis pusat yang digunakan dalam sistem operasi untuk menyimpan informasi konfigurasi yang dibutuhkan oleh kernel, program, services. Windows Registry sering menjadi target utama malware karena perannya yang sentral dalam konfigurasi sistem. Malware mengeksploitasi Registry untuk berbagai tujuan jahat seperti melakukan taktik persistent yang membuat malware akan terus berjalan yaitu dengan membuat process scheduler yang mengeksekusi malware setiap sistem dinyalakan, menonaktifkan sistem pertahanan seperti Windows Defender, menyembunyikan aktifitas malware, hingga dapat dilakukan sebagai alat untuk menaikkan level pengguna menjadi lebih tinggi. Dengan membandingkan registry ini, kita dapat memperoleh perilaku malware saat di eksekusi dan mengetahui apa yang dilakukan malware pada saat runtime. Karena jumlah registry pada windows cukup banyak, menganalisis secara manual tidak mungkin dilakukan, sehingga dibutuhkan sebuah alat untuk mengambil data terkini (capture) sebelum menjalankan malware seperti pada Gambar 9. Kemudian jika proses malware sudah selesai, dapat melakukan pengambilan data kedua yang kemudian dibandingkan antara data pertama dengan data kedua dan didapat hasil perbedaan Registry.



Gambar 9. Mengaktifkan Regshot Sebelum menjalankan Malware

Sebelum dapat merekam perbedaan yang terjadi pada Windows Registry, regshot diharuskan mengambil data terkini pada registry, yaitu dengan mengeksekusi capture maka akan didapatkan registry terkini dan disimpan dalam sebuah file data dengan ekstensi .hive.

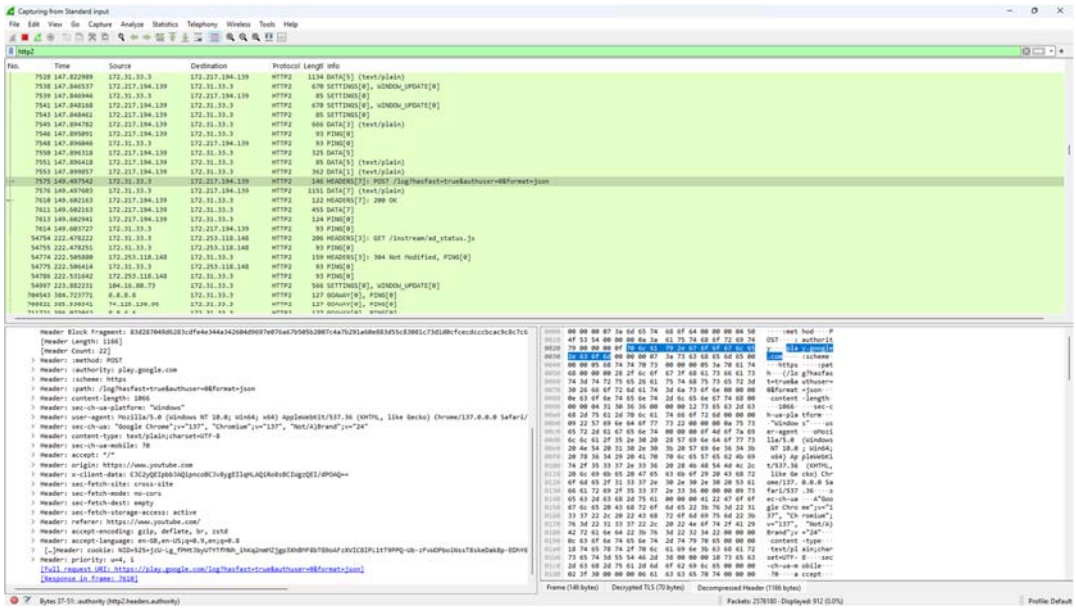


Gambar 10. Perbandingan Regshot

Jika proses analisis sudah berakhir, dapat diambil kembali registry dan dibandingkan dengan registry sebelumnya seperti pada Gambar 10 yang memperlihatkan data Registry yang tidak ada pada proses rekaman pertama, namun ada pada proses rekaman kedua yaitu terdapat registry yang dihapus. Hal ini dapat menjadi indikasi bahwa Registry tersebuh dihasikan dari menjalankan malware. Tetapi, tidak semua perubahan registry merupakan aktivitas yang ilegal, karena sistem operasi juga menuliskan ke dalam registry terhadap data-data yang harus diubah. Sehingga dari perbedaan itu dapat dianalisis lebih lanjut untuk di klasifikasikan antara perubahan registry yang legitimate oleh sistem, dan perubahan registry yang di indikasikan dilakukan secara ilegal.

Aktivitas jaringan merupakan salah satu kunci untuk membuat IoC, dikarenakan traffic jaringan merupakan identitas yang unik dari organisasi pembuat malware. Jika identitas ini sudah didapatkan, maka bad actors diharuskan untuk mengubah cara mereka menyebarkan malware. Dalam praktiknya, melakukan sniffing tidaklah mudah dikarenakan adanya proteksi TLS (Transport Layer Security) yang mengenkripsi lalu lintas jaringan antara malware dan C2 milik penyerang. Tetapi, dengan menggunakan

wireshark dan SSLKEYLOGFILE, maka beberapa traffic jaringan bisa di dekripsi dan dibaca secara plaintext. Pada Gambar 11 merupakan ssebuah log jaringan pada wireshark dimana window atas yang berwarna adalah list dari lalu lintas jaringan itu sendiri. Pada window bagian kanan adalah paket yang dipilih, terdapat beberapa opsi mulai dari raw data dimana paket di enkripsi dan versi paket yang sudah di dekrip. Kemudian pada window kiri adalah detail dari protokol yang sedang dipilih yaitu HTTP2. Dari gambar tersebut dapat dipastikan bahwa domain yang menggunakan protokol https (secure) bahkan melewati protokol yang relatif baru seperti HTTP versi 2 dapat dibaca secara apa adanya, karena proses enkripsi menggunakan TLS berhasil di dekripsi menjadi data utuh yang dapat dibaca manusia.



Gambar 11. Melihat detail paket jaringan HTTP2 tanpa TLS

Menggunakan metode ini, analisis jaringan akan dipermudah untuk mencari lalu lintas yang digunakan oleh penyerang. Yaitu dengan mencocokkan waktu eksekusi dan log jaringan yang direkam oleh wireshark. Dalam melakukan analisis ini, diharapkan tidak membuka program apapun yang menyebabkan adanya noise pada lalu lintas di wireshark. Hasil dari analisis ini, dapat dilihat pada Gambar 12 yang merupakan log http yang digunakan oleh penyerang untuk mengambil data-data pribadi. Terdapat dua segmen pada detail http sesuai pada dasar struktur http yaitu Request dan Response, pada teks berwarna merah merupakan request http dari komputer korban keluar, dan teks berwarna biru merupakan respon yang dikirimkan dari server ke komputer korban. Dalam log tersebut terdapat key dan value yang diambil oleh penyerang seperti session website, file-file dengan ekstensi .wallet yang biasanya digunakan untuk backup dompet kripto, alamat IP, serta alamat Domain penyerang yang sudah dirangkum pada Tabel 6

beserta alamat ip versi 4 dan negara yang menyediakan alamat ip tersebut. Jika diperhatikan, alamat tersebut merupakan alamat dari aplikasi yang aman, dan hal ini dimanfaatkan penyerang untuk menjalankan server mereka melalui aplikasi-aplikasi ini.

Gambar 12. Proses Stealer Mengirimkan Data ke Domain Penyerang

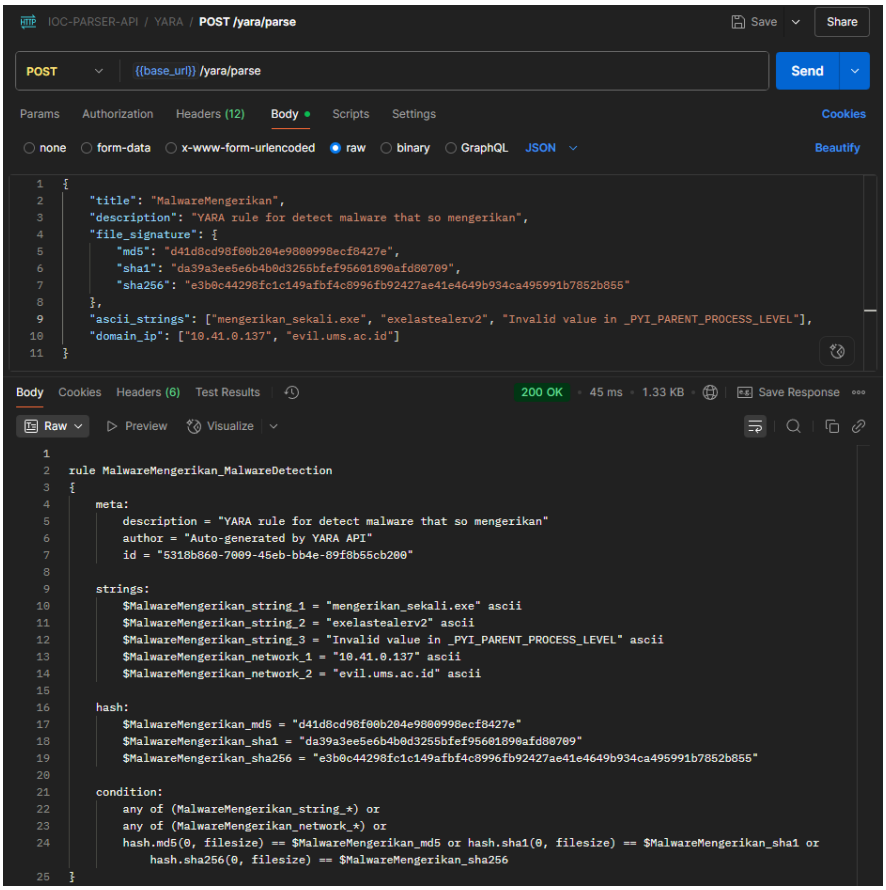
Tabel 6. Tiga Domain dan alamat IP yang di hubungi pada Malware nomor 1

No.	Domain	IPv4	Negara
1	discord.com	162.159.138.232	California, US
	ip-api.com	208.95.112.1	North Carolina, US
	store1.gofile.io	45.112.123.227	Île-de-France, FR

Hasil Data Parsing

Setelah mendapatkan data-data dari hasil analisis statis dan dinamis, langkah terakhir adalah menyatukan informasi tersebut menjadi satu standar agar dapat digunakan untuk memvalidasi adanya file malware pada sistem, standar yang sering ditemui pada industri yaitu adalah YARA, yaitu sebuah alat yang berfungsi untuk membuat deskripsi atau aturan (rules) berdasarkan pola tekstual atau biner dari suatu keluarga malware [12]. Aturan yang telah dibuat kemudian dapat digunakan untuk memindai sistem atau file lain untuk mendeteksi dan mengklasifikasikan sampel malware yang cocok dengan pola tersebut. Dikembangkan oleh Victor M. Alvarez dari VirusTotal, YARA telah menjadi standar de facto di kalangan peneliti keamanan dan analis malware untuk membuat signature yang fleksibel dan kuat.

Selain menggunakan YARA, terdapat standarisasi lain yaitu SIGMA yang berfokus pada deteksi berbasis konten file, SIGMA dirancang sebagai standar terbuka untuk mendeskripsikan peristiwa pada log file untuk sistem SIEM. Aturan Sigma dapat dikonversi secara otomatis ke berbagai bahasa kueri, sebuah fitur yang krusial untuk lingkungan keamanan heterogen dan menjadi subjek riset dalam optimalisasi aturan deteksi [13]. Pendekatan ini memungkinkan analis untuk menulis satu metode deteksi dan membagikannya secara luas, menyatukan cara komunitas keamanan siber mendeteksi ancaman berbasis log. Luaran yang diharapkan pada tahap ini adalah sebuah ruleset atau aturan dalam bentuk YARA dan SIGMA yang dapat dibuat dengan melakukan otomasi input hasil analisis menggunakan API seperti pada Gambar 13 yang mengubah input mentah tersebut hingga menjadi rule yang siap digunakan.



Gambar 13. Proses parsing data menjadi informasi IOC menggunakan REST API

YARA

Peran YARA sebagai standar industri dalam identifikasi malware berbasis signature menjadi landasan pada bagian penelitian ini. Penelitian ini mengusulkan pengembangan sebuah sistem otomatisasi yang bertujuan untuk melakukan penerjemahan artefak yang dihasilkan dari proses analisis statis dan dinamis menjadi sebuah aturan YARA yang terstruktur dan fungsional.

Mekanisme ini dirancang untuk meningkatkan efisiensi dan efektivitas tim keamanan siber dengan mereduksi waktu yang dibutuhkan untuk pembuatan signature secara manual, serta memastikan konsistensi dan skalabilitas dalam menghadapi ancaman baru.

SIGMA

Sejalan dengan pengembangan pada YARA, penelitian ini juga berfokus pada ranah deteksi berbasis perilaku (behavioral detection) dengan memanfaatkan standar Sigma. Tujuan dari tahap ini adalah untuk merancang dan mengimplementasikan sebuah skrip otomatis yang memiliki kapabilitas untuk mengonversi data telemetri sistem yang bersumber dari System Monitor (Sysmon) menjadi aturan deteksi Sigma yang bersifat generik dan portabel. Dengan adanya otomasi ini, diharapkan proses perumusan kueri threat hunting dapat dieksekusi secara lebih cepat dan efisien.

PEMBAHASAN

Proses analisis malware dengan metode statis dan dinamis berhasil dilakukan untuk mendapatkan Indicator of Compromises (IoC). Analisis statis memungkinkan pengambilan informasi seperti hash, string, dan metadata tanpa mengeksekusi malware sehingga dapat mengurangi risiko kerusakan sistem. Namun, teknik obfuskasi seperti file padding dapat mengubah hash, menyebabkan potensi temuan false positive, seperti pada sampel Exela dan Lumma Stealer.

Analisis dinamis mengatasi kelemahan tersebut dengan merekam perilaku aktual malware, seperti koneksi ke server C2 (Command and Control), modifikasi registry, dan eskalasi hak akses dalam lingkungan virtual terkontrol. Penggunaan alat seperti Sysmon, Wireshark, dan RegShot memastikan data perilaku terekam akurat. Konfigurasi lingkungan virtual, seperti Ubuntu Server sebagai gateway dan MITM Proxy untuk mendekripsi lalu lintas HTTPS, mendukung analisis jaringan yang optimal tanpa adanya enkripsi.

Automasi konversi data mentah menjadi ruleset YARA dan SIGMA melalui REST API meningkatkan efisiensi dan skalabilitas deteksi, memungkinkan integrasi dengan sistem seperti Snort dan Suricata. Pendekatan ini lebih komprehensif dibandingkan metode yang hanya menggunakan analisis statis, karena menggabungkan kecepatan statis dengan ketahanan dinamis terhadap teknik evasi.

Keterbatasan utama adalah ketergantungan pada lingkungan virtual, yang dapat dideteksi oleh malware dengan fungsi AntiVM, memerlukan modifikasi manual seperti BIOS UUID. Analisis jaringan juga terhambat oleh enkripsi TLS pada beberapa sampel, meskipun sebagian dapat didekripsi. Penelitian ini berkontribusi dengan sistem otomatisasi untuk deteksi malware yang cepat dan terstandarisasi, membuka peluang integrasi dengan kecerdasan buatan untuk analisis yang lebih adaptif di masa depan.

KESIMPULAN

Penelitian ini menunjukkan bahwa metode hybrid yang menggabungkan analisis statis dan dinamis, efektif digunakan untuk mengidentifikasi Indicator of Compromise (IoC) pada malware jenis stealer. Proses analisis dimulai dengan persiapan lingkungan virtual menggunakan VMware untuk menghasilkan lingkungan analisis yang aman. Analisis statis mengekstrak IoC awal seperti hash (MD5, SHA1, SHA256), string, VBA macro, dan domain yang berkemungkinan false positive melalui perangkat bawaan Windows dan menggunakan Powershell script. Analisis dinamis memantau perilaku malware, seperti koneksi ke server Command and Control (C2), modifikasi registry, dan lalu lintas jaringan menggunakan Sysmon, Wireshark, dan RegShot untuk memvalidasi domain yang digunakan oleh bad actors dari kemungkinan false positive dan kemungkinan true negative pada saat proses analisis statis. Data hasil analisis diolah secara otomatis melalui API untuk menghasilkan ruleset YARA dan SIGMA yang dapat diintegrasikan ke sistem deteksi.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada organisasi VX-Underground yang telah membantu memberikan data yang digunakan dalam penelitian. Selain itu, penulis mengucapkan terima kasih kepada Dosen Pembimbing yang selalu memberikan arahan dan saran sehingga penelitian ini dapat diselesaikan. Penulis juga mengucapkan terimakasih kepada seluruh pendukung yang senantiasa memberikan dukungan secara materi maupun non-materi.

DAFTAR PUSTAKA

[1] C. Rahmawati, "Tantangan Dan Ancaman Keamanan Siber Indonesia Di Era Revolusi Industri 4.0," *Seminar Nasional Sains Teknologi dan Inovasi Indonesia (SENASTINDO AAU)*, vol. 1, no. 1, pp. 299–306, 2019.

[2] C. Patsakis, D. Arroyo, and F. Casino, "The Malware as a Service ecosystem," May 2024, [Online]. Available: http://arxiv.org/abs/2405.04109

[3] "Malware Trends Overview Report: 2024 - ANY.RUN's Cybersecurity Blog." Accessed: May 19, 2025. [Online]. Available: https://any.run/cybersecurity-blog/malware-trends-2024/

[4] V. Rijn, H. W. J. Van Rijn, and H. W. J. Van Rijn, "An In-depth Analysis of the AZORult Infostealer Malware Capabilities."

[5] A. Mahn, J. Marron, S. Quinn, and D. Topper, "Getting started with the NIST Cybersecurity Framework :," Aug. 2021. doi: 10.6028/NIST.SP.1271.

[6] A. Y. and, Drs. A. M. M. S. Kusuma, "Penguatan Karakter Demokrasi Dan Peduli Sosial(Studi Kasus Pada Kegiatan Karang Sepuh Manunggal Roso di Dusun Punthuk Desa Bulu Kecamatan Pilangkenceng Kabupaten Madiun)," 2023.

[7] Kaspersky, "The evolving threat landscape of infostealers: trends, statistics, and mitigation strategies," 2025. Accessed: Jun. 16, 2025. [Online]. Available: https://content.kaspersky-labs.com/se/media/en/enterprise-security/data-stealer-storm-2025.pdf

[8] M. Alhif Ikhsan and M. T. , Ph. D. Ir. Bana Handaga, "PENERAPAN KEAMANAN SERVER MENGGUNAKAN SECURITY INFORMATION EVENT AND MANAGEMENT PADA SISTEM OPERASI UBUNTU SERVER," 2023. Accessed: May 22, 2025. [Online]. Available: http://eprints.ums.ac.id/id/eprint/117720

[9] T. Quertier and G. Barrué, "Use of Multi-CNNs for Section Analysis in Static Malware Detection," Feb. 2024, [Online]. Available: http://arxiv.org/abs/2402.04102

[10] D. Trizna, L. Demetrio, B. Biggio, and F. Roli, "Nebula: Self-Attention for Dynamic Malware Analysis," Oct. 2024, doi: 10.1109/TIFS.2024.3409083.

[11] V. Patil, N. K. M, P. S. M, and A. Singh, "Effectively Writing YARA Rules to Detect Malware," *Int J Res Appl Sci Eng Technol*, vol. 13, no. 1, pp. 1265–1273, Jan. 2025, doi: 10.22214/ijraset.2025.66535.

[12] Victor M. Alvarez, "YARA Documentation. VirusTotal," YARA Documentation. VirusTotal.

[13] A. Shukla, A. Gandhi, Y. Elovici, and A. Shabtai, "RuleGenie: SIEM Detection Rule Set Optimization," 2025.