**Journal of Applied Informatics and Computing (JAIC)**
Vol.x, No.x, Juli 201x, pp. xx~xx
e-ISSN: 2548-6861 

# ANALISIS DAN PENCEGAHAN MALWARE EMOTET PADA WINDOWS 10 DAN 11

**Azzam Albasith[1], Nur Cahyono Kushardianto[2]**
[1,2]Teknik Informatika, Rekayasa Keamanan Siber, Politeknik Negeri Batam
azzam.albasith3@students.polibatam.ac.id[1], anung@polibatam.ac.id[2]

**Article Info**

***Article history:***

Received …
Revised …
Accepted …

***Keyword:***

*Emotet, Windows 10, Windows 11, Analisis Statis, Analisis Dinamis, Signature Based Detection, ClamAV.*

**ABSTRACT**

Emotet merupakan salah satu malware trojan yang terkenal karena kemampuan penyebaran cepat, pencurian data sensitif, pengumpulan kredensial, serta teknik penghindaran deteksi yang terus berkembang. Penelitian ini bertujuan untuk menganalisis karakteristik dan perilaku Emotet pada sistem operasi Windows 10 dan Windows 11 melalui pendekatan analisis statis dan analisis dinamis. Analisis statis dilakukan untuk mengidentifikasi karakteristik sampel, termasuk struktur file, *toolchain* yang dipakai, dan library yang dipanggil, sedangkan analisis dinamis digunakan untuk mengamati perilaku malware, seperti aktivitas pada *file system* dan komunikasi Command and Control (C2), saat malware dijalankan di lingkungan terisolasi. Selain itu, penelitian ini juga mengevaluasi efektivitas metode *signature-based detection* ClamAV dalam mendeteksi malware lain yang memiliki karakteristik yang serupa dengan Emotet. Hasil penelitian menunjukkan bahwa meskipun tidak terdapat perbedaan signifikan pada dampak dan perilaku Emotet antara Windows 10 dan Windows 11, pemanfaatan signature ClamAV yang dikembangkan mampu mencapai nilai akurasi sebesar 87,5%.



## I. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi telah membawa perubahan yang signifikan terhadap cara manusia menjalani kehidupan sehari-hari. Internet sebagai sarana utama pertukaran data dan informasi telah mempercepat proses komunikasi, transaksi, serta akses terhadap pengetahuan. Di sisi lain, kemajuan sistem operasi turut memberikan kemudahan dalam pengelolaan perangkat computer dan mobile, memungkinkan pengguna untuk menjalankan aplikasi dengan efisien. Sinergi antara internet dan sistem operasi ini menjadikan berbagai aktivitas manusia baik dibidang pendidikan, pekerjaan, maupun sosial semakin praktis, cepat, dan terintegrasi secara digital. Badan pusat statistik (BPS) dari hasil pendataan survei Susenas tahun 2024, terdapat 72,78% penduduk Indonesia telah mengakses internet di tahun 2024 dan 69,21% di tahun 2023[1]. Sementara itu, sistem operasi Windows sebagai salah satu sistem operasi mengalamai perkembangan dari waktu ke waktu. Data periode Oktober 2021 hingga Mei 2025 dari StatCounter menunjukkan penggunaan Windows 10 di Indonesia mendominasi sebesar 73.08%[2]. Meskipun demikian, Microsoft telah mengumumkan bahwa dukungan terhadap Windows 10 akan dihentikan pada 14 Oktober 2025[3], sehingga seluruh perangkat komputer dapat beralih ke Windows 11. Seiring dengan perkembangan teknologi informasi, serangan malware (Malicious Software) dan *cybercrime* juga meningkat. Badan Siber dan Sandi Negara (BSSN) mencatat adanya 3,64 milliar serangan atau anomali trafik di Indonesia sepanjang Januari hingga Juli 2025. 83,68% merupakan serangan berbasis *malware.* Sisanya adalah unauthorized access dan serangan terhadap sistem sebesar 4,32%, serta eksploitasi sistem sebanyak 0,64%[4].

Salah satu malware yang berbahaya adalah Emotet, pertama kali ditemukan sebagai *trojan* perbankan pada tahun 2014, Emotet kemudian berkembang menjadi infrastruktur serangan yang kompleks[5]. Setelah berhasil menginfeksi sistem, malware ini dapat melakukan berbagai aktivitas berbahaya, termasuk pencurian data sensitif, pengumpulan kredensial, serta penyebaran malware lain seperti Trickbot dan Ryuk ransomware. Emotet dirancang untuk untuk sulit dideteksi karena menggunakan teknik *obfuscation* dan

e-ISSN: 2548-6861

modul modular yang terus diperbarui. Selain itu, Emotet memiliki kemampuan *lateral movement*, kemampuan berpindah ke perangkat lain dalam satu jaringan. Penyebarannya umumnya terjadi melalui kampanye *phishing* berskala besar, di mana korban diarahkan untuk membuka lampiran atau tautan berbahaya yang kemudian memicu proses infeksi[6].

Pencegahan serangan berbasis malware dapat dicegah dengan melakukan analisis malware. Melalui analisis malware peneliti dapat memahami bagaimana suatu *malware* bekerja, apa tujuan pembuatnya, dan bagaimana berinteraksi dengan sistem. Analisis malware terbagi menjadi dua pendekatan yaitu analisis statis dan dinamis. Analisis statis merupakan proses memeriksa sebuah file yang mencurigakan tanpa perlu mengeksekusinya. Bertujuan untuk memahami karakteristik file, bagaimana strukturnya, apa fungsi yang mungkin dijalankan, serta apakah file tersebut menunjukkan indikasi aktivitas berbahaya[7]. Sedangkan analisis dinamis adalah teknik yang digunakan untuk mendeteksi aktivitas berbahaya dengan menjalankan malware di dalam lingkungan yang terisolasi. Dengan teknik ini, peneliti dapat mengamati secara langsung bagaimana sebuah malware bereaksi terhadap sistem, seperti perubahan file, modifikasi *registry*, komunikasi jaringan, dan perintah yang dieksekusi[8].

Dengan melakukan analisis malware, *Indicator of Compromise* (IoC) dapat diidentifikasi. *Indicator of Compromise* (IoC) merupakan informasi yang dihasilkan dari aktivitas malware, seperti jenis malware yang digunakan, alamat IP yang digunakan, dan sebagainya. Informasi ini dapat dimanfaatkan oleh tim keamanan untuk menyusun langkah pencegahan terhadap serangan malware seperti pengembangan *signature-based detection* yang berguna untuk memperkuat sistem keamanan dan meningkatkan kesadaran terhadap potensi ancaman baru[9].

Signature-based detection merupakan teknik yang digunakan untuk mengidentifikasi ancaman seperti malware dengan cara membandingkan pola atau signature yang telah diketahui sebelumnya. Salah satu perangkat lunak yang menerapkan teknik ini adalah ClamAV, yaitu antivirus bersifat open-source yang banyak digunakan untuk mendeteksi berbagai jenis ancaman seperti virus, trojan, dan worm. ClamAV dapat dijalankan pada berbagai sistem operasi, termasuk Windows, macOS, dan Linux, serta memiliki basis data signature yang terus diperbarui secara berkala oleh komunitas keamanan siber. Selain itu, ClamAV juga memungkinkan pengguna untuk menambahkan custom signature sesuai kebutuhan[10], sehingga dapat digunakan untuk penelitian maupun pengujian malware tertentu.

Penelitian ini bertujuan untuk pencegahan malware Emotet menggunakan signature dari ClamAV berdasarkan hasil analisis malware menggunakan pendekatan analisis statis dan analisis dinamis pada Windows 10 dan Windows 11, serta menguji dan mengkaji akurasi *signature* tersebut dalam mendeteksi jenis malware yang berbeda. Penelitian ini diharapkan dapat memberikan wawasan bagi pembaca mengenai karakteristik dan perilaku malware Emotet, serta berkontribusi dalam pengembangan sistem deteksi malware untuk menghadapi ancaman malware di masa mendatang.

## II. TINJAUAN PUSTAKA

Secara umum, analisis malware memiliki banyak parameter yang dapat diamati. Parameter analisis statis dan analisis dinamis dapat berbeda-beda tergantung dengan tujuan analisis yang ingin dicapai.

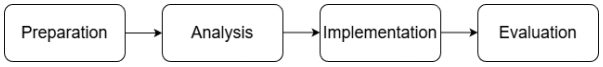
Yunike et.al.[11] melakukan analisis terhadap malware berjenis trojan menggunakan kombinasi pendekatan statis dan dinamis. Pada analisis statis, sampel dipindai melalui VirusTotal untuk memperoleh informasi seperti nilai hash dan ukuran file. Sedangkan pada analisis dinamis, mereka melakukan pengamatan terhadap paket jaringan yang dihasilkan oleh malware. Tujuan dari penelitian tersebut adalah menghasilkan dataset malware berdasarkan standar STIX (Structured Threat Information Expression) versi 2.1.

Indra et.al.[12] meneliti botnet Proteus dengan menerapkan pendekatan gabungan antara analisis statis dan dinamis. Analisis statis dibagi menjadi beberapa bagian yaitu string extract, module dependency, disassembly, dan obfuscated detect. Sedangkan analisis dinamis berfokus pada perubahan registry dan analisa paket data dalam aktivitas jaringan. Keseluruhan analisis bertujuan untuk memahami mekanisme operasional botnet saat menginfeksi korban.

Hafish et. al.[13] menganalisis trojan bernama Aquvaprn.exe dengan pendekatan statis dan dinamis. Analisis statis dilakukan dengan meninjau nilai hash dari sampel malware sedangkan analisis dinamis difokuskan pada pengamatan aktivitas sistem seperti perubahan file system, registry, network communication, serta analisis forensik memori. Hasil dari penelitian tersebut bertujuan untuk menguraikan tahapan serangan serta teknik eksploitasi yang digunakan malware.

## III. METODE PENELITIAN

Penelitian ini menggunakan metode eksperimental dengan Emotet dipilih sebagai objek penelitian karena merupakan salah satu malware trojan yang memiliki tingkat infeksi tinggi serta terus berkembang dalam berbagai varian. Analisis dilakukan menggunakan pendekatan analisis statis dan analisis dinamis pada sistem operasi Windows 10 dan Windows 11. Gambar 1 merupakan tahapan yang akan dilakukan dalam penelitian ini.




*Gambar 1 Tahap Penelitian*

A. Preparation

Pada tahap ini dilakukan studi literatur di berbagai sumber seperti website, jurnal, dan artikel ilmiah yang membahas topik analisis malware di platform Windows, *repository* tempat pengambilan sampel malware, dan alat-alat yang digunakan dalam proses analisis malware beserta fungsinya. Tahap ini bertujuan untuk membangun pemahaman dasar mengenai konsep, teknik, prosedur, dan dasar teori dalam merancang metode analisis malware.

Tahap ini juga mencakup persiapan infrastruktur dalam membangun lingkungan pengujian menggunakan aplikasi virtualisasi VirtualBox untuk menciptakan *Virtual Machine* (VM) yang aman dan terisolasi selama analisis statis dan analisis dinamis. *Virtual machine* dikonfigurasi menggunakan mode jaringan NAT (Network Address Translation) dan *host-only*, dengan fitur *shared folder* antara *host* dan VM dinonaktifkan untuk mencegah potensi penyebaran malware ke sistem *host*. Supaya analisis malware bisa dilakukan secara berulang, *snapshot* digunakan agar VM dapat dikembalikan ke kondisi awal sebelum malware dieksekusi.

B. Analysis

Pada tahap ini, analisis dilakukan dengan pendekatan statis dan pendekatan dinamis pada sistem operasi Windows 10 dan Windows 11. Hal ini bertujuan untuk memahami karakteristik yang dimiliki oleh malware dan perilaku malware ketika berhasil masuk ke dalam sistem. Analisis malware mengacu pada *checklist* yang disediakan dalam *malware analysis Framework* dari FIRST (Forum of Incident Response and Security Teams)[14].

1) Analisis Statis

Tahapan pertama analisis statis adalah identifikasi sampel malware. Proses ini dimulai dengan menghasilkan nilai hash (MD5, SHA1, SHA256) dari setiap sampel malware kemudian mengirimkan dan mencocokkan dengan data malware di Virus total, hal ini bertujuan untuk memastikan apakah sampel malware sudah dikenal oleh komunitas keamanan siber.

Tahap kedua yaitu analisis header sampel malware, bertujuan untuk mencari informasi teknis dari struktur internal sampel. Peneliti dapat mengetahui *toolchain* yang digunakan oleh *developer*, seperti compiler, linker, dan IDE yang digunakan dalam proses pembuatan malware. Selain itu, *timestamp* pada header juga dianalisis untuk melihat waktu kompilasi sampel.

Tahap ketiga yaitu memeriksa *library* yang digunakan oleh sampel malware. Analisis ini dilakukan untuk melihat fungsi-fungsi API yang dipanggil melalui library Windows. Dengan ini, peneliti dapat memahami tujuan dan kemampuan malware tanpa harus mengeksekusinya langsung.

2) Analisis Dinamis

Analisis dinamis dilakukan dengan menjalankan malware pada lingkungan yang terisolasi, tujuannya yaitu memahami perilaku (behavior) dari malware. Proses ini terdiri dari tiga tahap. Tahap pertama yaitu *monitoring* seluruh operasi yang terjadi saat malware pertama kali dijalankan pada sistem korban, seperti akses ke registry, pembuatan file atau folder baru, dan pemanggilan API penting.

Tahap kedua dilanjutkan dengan analisis aktivitas *file system* yang dihasilkan oleh malware. Tahap ini, peneliti mengamati setiap perubahan yang terjadi pada struktur *file system* seperti pembuatan file baru, modifikasi file yang sudah ada, pola penamaan file, atau upaya malware menempatkan dirinya di direktori tertentu.

Tahap ketiga yaitu analisis *network communication*, bertujuan untuk mengetahui aktivitas jaringan yang dilakukan oleh malware selama proses eksekusi. Pada tahap ini diamati apakah malware berupaya mengirim data, jenis data apa yang dikirimkan, kemana data tersebut dikirimkan, dan protool apa yang digunakan dalam proses komunikasi.

C. Implementation

Pada tahap ini, pencegahan Emotet dilakukan dengan memanfaatkan *Indicator of Compromise* (IoC) yang diperoleh dari tahap *analysis*, kemudian memasukkannya ke dalam signature ClamAV. IoC seperti *file hash*, *byte sequences* atau *pattern* tertentu dapat digunakan sebagai acuan untuk membuat aturan signature yang mampu mengenali keberadaan Emotet. Hal ini dilakukan untuk memastikan hasil analisis malware tidak hanya dipahami secara teori, tetapi juga langsung diimplementasikan sebagai langkah perlindungan nyata bagi sistem.

D. Evaluation

Pada tahap ini dilakukan pengujian terhadap signature ClamAV untuk memastikan kemampuan deteksi berjalan sesuai dengan yang diharapkan. Pengujian ini bertujuan untuk menilai tingkat akurasi dan kinerja signature yang dikembangkan menggunakan *confusion matrix*, yaitu alat yang digunakan untuk mengukur kinerja suatu model dengan membandingkan nilai aktual dengan nilai prediksi[15]. Tabel 1 merupakan tabel nilai yang dihasilkan dari *confusion matrix*.

*Tabel 1 Tabel Confusion Matrix*

| | Prediksi Positif | Prediksi Negatif |
|---|---|---|
| Aktual Positif | True Positive (TP) | False Negative (FN) |
| Aktual Negatif | False Positive (FP) | True Negative (TN) |

Salah satu metrik evaluasi yang dihasilkan yaitu *Accuracy* (akurasi), yang dapat menunjukkan berapa banyak prediksi yang benar dari semua prediksi yang dibuat oleh model. *Accuracy* dirumuskan sebagai berikut:

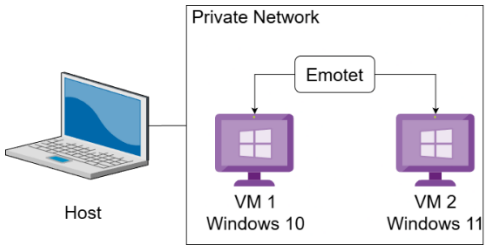
e-ISSN: 2548-6861

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \times 100\%$$

## IV. HASIL DAN PEMBAHASAN

### A. Persiapan Analisis Malware

Dalam penelitian ini, infrastruktur dirancang untuk mendukung proses analisis Emotet secara menyeluruh. Infrastruktur tersebut mencakup perangkat keras dan perangkat lunak yang dibutuhkan untuk menyediakan lingkungan pengujian yang aman dan terisolasi. Lingkungan ini secara khusus disiapkan untuk pelaksanaan analisis statis dan analisis dinamis terhadap sampel Emotet. Gambar 2 menunjukkan topologi lingkungan analisis Emotet, dimana sampel Emotet dianalisis didalam *Virtual Machine* untuk melindungi sistem *host* dari malware.




*Gambar 2 Topologi Lingkungan Analisis*

*Tabel 2 Spesifikasi Host dan VM*

| | **Host** | **VM 1** | **VM 2** |
|---|---|---|---|
| OS | Windows 11 v24H2 | Windows 10 v22H2 | Windows 11 v22H2 |
| RAM | 8 GB | 3 GB | 4 GB |
| Storage | 256 GB | 50 GB | 60 GB |
| Network | Wifi | NAT + Host Only | |

Jumlah sampel Emotet yang digunakan dalam penelitian ini sebanyak lima sampel yang dikumpulkan dari situs tria.ge[16], situs repository yang menyediakan sampel berbagai jenis malware. Jumlah sampel dibatasi sebanyak lima dikarenakan karena keterbatasan informasi yang valid tentang varian Emotet sehingga penelitian difokuskan pada analisis mendalam terhadap pola karakteristik dan perilaku malware. Tabel 3 menunjukkan daftar sampel Emotet yang digunakan dalam analisis.

*Tabel 3 Sampel Malware Emotet*

| **No** | **Sampel Emotet** | **Nilai Hash MD5** |
|---|---|---|
| 1. | 250805-m5halaaq4w | 39da06ed1df732ad36827a0-c6cc66f4c |
| 2. | 250803-edl6gazjv7 | 7ff55186f1361cd22eb0fb2d-1ce89fd0 |
| 3. | 251112-dn49csxqay | 7062ff5c1677971ebb973110-1de7f256 |
| 4. | 250726-2v2zkst1bt | 255dd424089fe839711e09d-220e89d48 |
| 5. | 250822-zjbbdadk9s | 28ac8ddbcb65ea52d2adab0-fd20fe534 |

Tabel 4 merupakan *tools* yang digunakan selama proses analisis statis dan dinamis. Setiap tools memiliki fungsi spesifik yang mendukung identifikasi karakteristik sampel dan pemantauan perilaku,

*Tabel 4 Alat Analisis*

| **Pendekatan Analisis malware** | **Alat** | **Fungsi** |
|---|---|---|
| Analisis Statis | Ghidra v11.4.1 | Memetakan dan membaca isi *file binary* |
| | PEstudio v9.61 | Identifikasi indikator malware didalam file |
| | Virus Total | Sumber data dan *repository* hasil analisis malware |
| Analisis Dinamis | Process Monitor v4.01 | Merekam secara *real time* aktivitas *file system* |
| | System Informer v3.2.25011 | *Monitoring* aktivitas internal malware saat dijalankan |
| | Wireshark v.4.6.1 | *Monitoring* dan analisis *network activity* |

### B. Analisis Statis

#### 1) Identifikasi Sampel Emotet

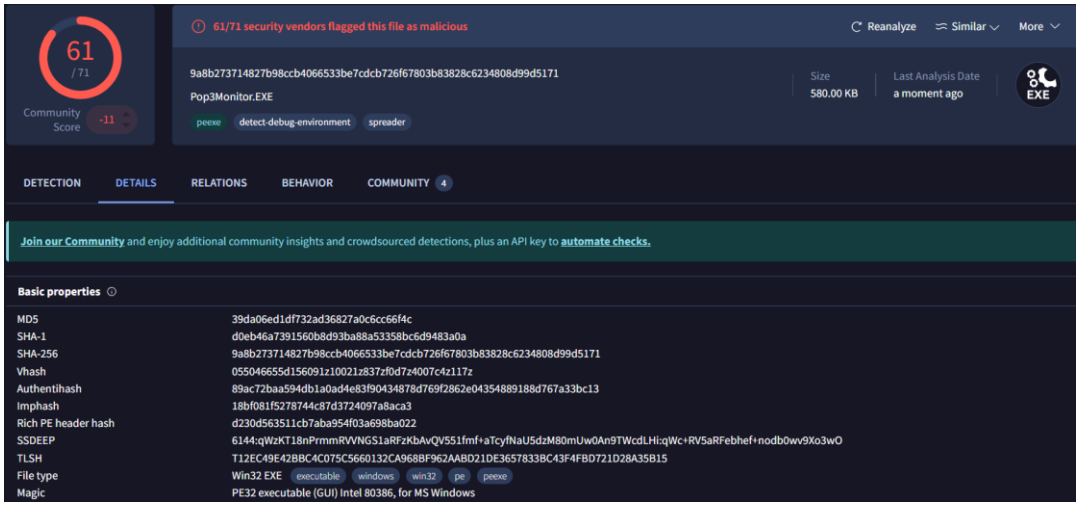
Tahap pertama analisis statis yaitu mengidentifikasi sampel Emotet menggunakan Virus total, yaitu dengan mengirimkan sampel ke platform tersebut untuk melihat sejauh mana sampel telah dikenali oleh komunitas keamanan siber. Hasil identifikasi ditunjukkan oleh tabel 5.

Berdasarkan hasil identifikasi sampel Emotet dari Virus Total, seluruh sampel sudah cukup dikenali oleh komunitas keamanan siber dan vendor antivirus dengan tingkat deteksi yang bervariasi. Jenis sampel terdiri dari file EXE dan *file* berjenis DLL (Dynamic Link Library) pada sampel 251112-dn49csxqay dan 250726-2v2zkst1bt. DLL merupakan jenis file yang isinya kumpulan kode dan fungsi yang dipakai oleh

banyak program di Windows. Dalam praktiknya, Emotet menanamkan kode VBA Macro pada dokumen *office* untuk mengunduh dan mengeksekusi payload berformat DLL pada sistem korban, DLL tersebut kemudian berperan sebagai komponen utama untuk melanjutkan proses infeksi[17]. Gambar 3 merupakan salah satu hasil identifikasi sampel di *platform* virus total

*Tabel 5 Identifikasi Sampel Emotet*

| No | Sampel Emotet | Jumlah Deteksi *Security Vendors* | Jenis File | Nilai Hash SHA256 |
|---|---|---|---|---|
| 1. | 250805-m5halaaq4w | 58/72 | EXE | 9a8b273714827b98ccb4066533be7cdcb72-6f67803b83828c6234808d99d5171 |
| 2. | 250803-edl6gazjv7 | 55/72 | EXE | 737036d3705d1e15f411a3b51ec46fd22d3d-27dfb8a87f0986ea68b43fd0c091 |
| 3. | 251112-dn49csxqay | 53/72 | DLL | e05243ec70891d75bbd33d5ac93a6a4f40adc-d1d0f9e3e6f8a9cc2331b5c11c6 |
| 4. | 250726-2v2zkst1bt | 63/72 | DLL | 44c658ef537581dae5f3953f7865a1dc0b095-30cdb20643c2cf366bb21e57fff |
| 5. | 250822-zjbbdadk9s | 60/69 | EXE | 421e4c9a65a8a002868b831c7bacff3ee9b6-6f2d76b9a642d608902f9549b66a |




*Gambar 3 Identifikasi Sampel*

2) Analisis Header Portable Executable (PE)

Tahap kedua analisis statis yaitu memeriksa struktur *file* tiap sampel melalui header menggunakan PEstudio. Tujuannya adalah untuk menemukan informasi yang dapat mengungkap *toolchain* yang digunakan oleh *developer* dalam membuat malware dan waktu pembuatannya.

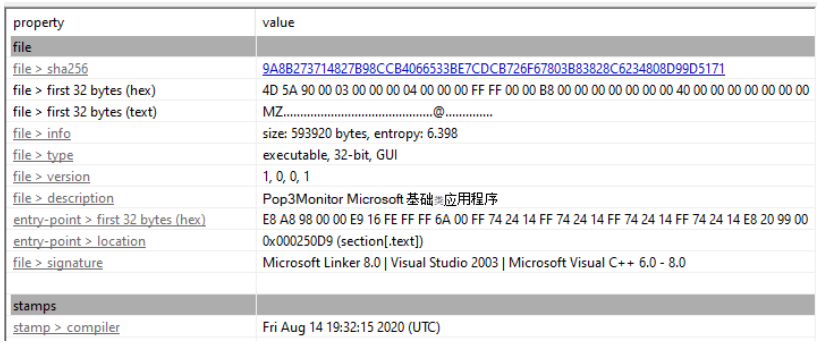
Tabel 6 merupakan rangkuman hasil analisis statis menggunakan PE studio. Beberapa sampel memiliki bagian yang kosong karena struktur PE yang digunakan berbeda-beda. Dari hasil pengamatan, seluruh sampel menggunakan compiler, linker, dan IDE versi lawas. Penggunaan *toolchain* ini menghasilkan *binary* yang lebih sederhana dan kompatibel dengan berbagai versi Windows termasuk windows 10, windows 11 dan windows server. Gambar 4 merupakan hasil analisis header menggunakan PEstudio.

*Tabel 6 Analisis PE*

| Sampel Emotet | Jenis File | Compiler | Linker | Time Stamps | IDE |
|---|---|---|---|---|---|
| 250805-m5halaaq4w | EXE 32-bit | Microsoft Visual C++ 6.0-8.0 | Microsoft Linker 8.0 | 14 Agustus 2020 | Visual Studio 2003 |
| 250803-edl6gazjv7 | EXE 32-bit | Microsoft Visual C++ 6.0-8.0 | Microsoft Linker 9.0 | 7 September 2016 | Visual Studio 2005 |

e-ISSN: 2548-6861

| | | | | | |
|---|---|---|---|---|---|
| 251112-dn49csxqay | DLL 64-bit | - | Microsoft Linker 14.31 | 27 April 2022 | Visual Studio 2015 |
| 250726-2v2zkst1bt | DLL 32-bit | - | Microsoft Linker 2.50 | 12 Januari 2021 | - |
| 250822-zjbbdadk9s | EXE 32-bit | Microsoft Visual C++ 6.0-8.0 | Microsoft Linker 10.0 | 18 September 2020 | Visual Studio 2010 |



| property | value |
|---|---|
| file | |
| file > sha256 | 9A8B273714827B98CCB4066533BE7CDCB726F67803B83828C6234808D99D5171 |
| file > first 32 bytes (hex) | 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 |
| file > first 32 bytes (text) | MZ.....@..... |
| file > info | size: 593920 bytes, entropy: 6.398 |
| file > type | executable, 32-bit, GUI |
| file > version | 1, 0, 0, 1 |
| file > description | Pop3Monitor Microsoft基础※应用程序 |
| entry-point > first 32 bytes (hex) | E8 A8 98 00 00 E9 16 FE FF FF 6A 00 FF 74 24 14 FF 74 24 14 FF 74 24 14 FF 74 24 14 E8 20 99 00 |
| entry-point > location | 0x000250D9 (section[.text]) |
| file > signature | Microsoft Linker 8.0 \| Visual Studio 2003 \| Microsoft Visual C++ 6.0 - 8.0 |
| | |
| stamps | |
| stamp > compiler | Fri Aug 14 19:32:15 2020 (UTC) |

*Gambar 4 Analisis Header*

3) Identifikasi Library Sampel

Tahap ketiga analisis statis dilakukan dengan mengidentifikasi library serta fungsi-fungsi API yang dipanggil oleh sampel menggunakan Ghidra. Informasi mengenai library dan API ini menjadi indikator awal untuk memahami kemampuan dan tujuan malware, khususnya dalam berinteraksi dengan sistem. Dengan ini, dapat diperkirakan fungsi utama yang dimiliki sampel tanpa perlu mengeksekusinya. Tabel 7 merupakan hasil identifikasi library dari seluruh sampel.

*Tabel 7 Identifikasi Library Setiap Sampel*

| Sampel Emotet | Library |
|---|---|
| 250805-m5halaaq4w | ADVAPI32.DLL |
| | COMDLG32.DLL |
| | GDI32.DLL |
| | KERNEL32.DLL |
| | OLE32.DLL |
| | OLEACC.DLL |
| | OLEAUT32.DLL |
| | OLEDLG.DLL |
| | SHLWAPI.DLL |
| | USER32.DLL |
| | WINSPOOL.DRV |
| | WS2_32.DLL |
| 250803-edl6gazjv7 | ADVAPI32.DLL |
| | GDI32.DLL |
| | KERNEL32.DLL |
| | OLEACC.DLL |
| | OLEAUT32.DLL |
| | SHLWAPI.DLL |
| | USER32.DLL |
| | WINSPOOL.DRV |
| 251112-dn49csxqay | KERNEL32.DLL |
| | NTDLL.DLL |
| 250726-2v2zkst1bt | ADVAPI32.DLL |
| | GDI32.DLL |
| | KERNEL32.DLL |
| | USER32.DLL |
| 250822-zjbbdadk9s | ADVAPI32.DLL |
| | COMCTL32.DLL |
| | KERNEL32.DLL |
| | PSAPI.DLL |
| | USER32.DLL |
| | VERSION.DLL |

Berdasarkan hasil identifikasi *library* pada tabel 7, terdapat tiga library utama yang memiliki peranan penting agar sampel bisa berjalan di sistem Windows. Ketiga library itu adalah *KERNEL32.dll*, *ADVAPI32.DLL,* dan *USER32.DLL*.

*KERNEL32.DLL* merupakan *library* inti Windows yang menyediakan fungsi-fungsi dasar sistem seperti manajemen memori, operasi input/output, dan eksekusi proses. Fungsi *KERNEL32.DLL* seperti *CreateThread* dapat digunakan untuk menjalankan perintah untuk berkomunikasi dengan server C2 dan fungsi *MapViewOfFile* untuk membaca atau menulis *file* sistem. Selain itu, sampel menggunakan *VirtualAlloc*, *WriteProcessMemory*, dan *CreateRemoteThread* dari *KERNEL32*.dll untuk mengalokasikan memori, menyisipkan kode berbahaya ke proses lain, dan menjalankan instruksi berbahaya tanpa terdeteksi.

*ADVAPI32.DLL* merupakan *library* sistem Windows yang berisi fungsi-fungsi tingkat lanjut terkait keamanan, pengelolaan *registry*, layanan Windows dan hak akses

pengguna. Sampel memanfaatkan *ADVAPI32.dll* karena menyediakan API untuk membaca dan menulis registry guna membuat mekanisme *persistence*, mengatur *privilege* seperti meningkatkan hak akses dengan *LookupPrivilegeValueW* dan *AdjustTokenPrivileges*, serta mengelola Windows Service melalui fungsi seperti *CreateService* atau *StartServiceA* agar sampel bisa berjalan secara permanen di background.

*USER32.DLL* merupakan *library* yang mengatur segala hal terkait antarmuka pengguna, seperti pembuatan dan pengelolaan jendela aplikasi, menangani input keyboard dan mouse, serta memproses pesan atau event yang terjadi saat pengguna berinteraksi dengan sistem. *CreateWindow* dan *AdjustWindowsRect* merupakan fungsi yang dapat dimanfaatkan sampel untuk membuat *fake window* atau *pop-up* palsu. Sampel juga dapat menggunakan fungsi *MessageBox* untuk mengganggu sistem atau menampilkan pesan *error* palsu.

Dari hasil identifikasi tiap sampel Emotet, ketika sampel dijalankan pada sistem Windows, sampel membangun infrastruktur atau *service* melalui *KERNEL32.DLL* dan kemudian berkomunikasi dengan IP C2. Disaat yang sama, untuk menghindari deteksi atau mempertahankan keberadaannya (persistence), sampel memodifikasi registry serta meningkatkan hak akses melalui *ADVAPI32.DLL*. Sementara itu untuk memanipulasi pengguna dan melakukan pencurian data, sampel memanfaatkan fungsi-fungsi yang ada di *USER32.DLL*.

C. Analisis Dinamis

Analisis dinamis dilakukan dengan menjalankan sampel pada lingkungan terisolasi yang telah disiapkan. Fokus analisis dinamis yaitu memahami aktivitas atau *behavior* sampel Emotet melalui monitoring *file system*, mengidentifikasi kemampuan *malware* dijalankan didalam sistem dan *monitoring* komunikasi *network*.

*Tabel 8 Analisis Dinamis*

| Sampel Emotet | File System Monitoring | Aktivitas Malware | Komunikasi Network |
|---|---|---|---|
| 250805-m5halaaq4w | Terjadi perbedaan penamaan *child process*. Pada Windows 10, ketika sampel dijalankan, ia mengeksekusi *child process* bernama "taskschd.exe", sedangkan pada Windows 11 sampel menghasilkan child dengan nama "fsutil.exe" | Dalam memulai eksekusinya, sampel membangun *persistence* dengan membuat file di direktori C:\Windows\SysWOW64. Kemudian sampel melakukan *RegOpenKey* pada registry **HKLM\SOFTWARE\Policies\Microsoft\Windows** bertujuan mengevaluasi konfigurasi Windows Defender, *firewall*, *system rectriction*, atau Group Policy lain yang dapat menghalangi aktivitas berbahaya | Hasil *monitoring network* menunjukkan bahwa beberapa IP mencoba untuk terhubung dengan sistem. dimulai dari IP 174.100.27.229:80, kemudian diikuti oleh IP 209.126.6.222:8080 dan 85.105.140.135:443. *Monitoring network* juga menunjukkan tidak ada data yang dikirim selama proses berlangsung |
| 250803-edl6gazjv7 | Ketika sampel dijalankan di Windows 10 dan Windows 11 sampel menghasilkan nama *child process* yang berbeda. *Child process* pada Windows 10 bernama "gpapi.exe" sedangkan pada Windows 11 menghasilkan "rasgcw.exe". Tidak ada indikasi sampel menggunakan aplikasi *legitimate* dalam hal *file system activity*. | Setelah dijalankan sampel menanamkan *stealthy persistence* di beberapa lokasi *registry* seperti **HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Explorer\User\Files\NameSpace\DelegateFolders** yang mengakibatkan malware berjalan tanpa *scheduled task* dan di lokasi **HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Policies\NonEnum** yang mengakibatkan malware dapat menyembunyikan drive tertentu atau folder tertentu | Terjadi komunikasi dari sistem Windows 10 dan Windows 11 menuju server C2. Upaya pertama ditujukan pada IP 2.45.176.233 pada port 80, namun tidak mendapatkan respon. Sampel kemudian mencoba menghubungi IP lain, yaitu 172.103.204.12 pada port 8080 dan kembali gagal lagi. Setelah itu, terlihat komunikasi ketiga menuju IP 98.103.204.12 pada *port* 443. Upaya percobaan komunikasi ini dilakukan berkali-kali tanpa ada data yang dikirim selama proses eksekusi |

e-ISSN: 2548-6861

| | | | |
|---|---|---|---|
| 251112-dn49csxqay | Saat sampel dieksekusi, sampel melakukan *drop files* dengan nama mencurigakan seperti "aachvgzpxquhn.smm" di direktori C:\Users\user\AppData\Local\ | Sampel dijalankan melalui command-line menggunakan regsvr32.exe, kemudian membangun infrastrukturnya dengan melakukan operasi *CreateFile*, *GetSecurityFile*, dan *CloseFile* ke beberapa API di direktori C:\Windows\System32. Selain itu, sampel juga operasi *RegOpenKey* pada registry **HKLM\System\CurrentControlSet\Control\Session Manager** | Terlihat adanya upaya komunikasi dari berbagai IP C2 yang mencoba terhubung ke sistem, mulai dari 176.31.73.90:443, 45.76.159.214:8080, 138.197.147.101:443, dan seterusnya. Namun dari seluruh komunikasi tersebut tidak terdeteksi adanya data yang dikirim |
| 250726-2v2zkst1bt | Setelah dijalankan, sampel membangun infrastruktur dan menyimpan payload utamanya di dalam direktori sistem dengan menggunakan nama aneh dan file tidak wajar seperti C:\Windows\SysWOW64\Oefaetaavbc\gmuallywyd.nzo | setelah dijalankan menggunakan Rundll32.exe, sampel membuat file di direktori C:\Windows\SysWOW64, kemudian melakukan *RegQueryKey* dan *RegOpenKey* pada registry **HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options** yang merupakan mekanisme Windows untuk mengatur eksekusi program. Bertujuan untuk menanamkan persistence dan menonaktifkan aplikasi tertentu | Terlihat adanya percobaan koneksi dari berbagai IP lain ke sistem Windows 10 dan 11, IP tersebut beragam diantaranya 93.146.143.191:80, 206.189.232.2:8080, 37.36.149.248:8080 dan beberapa IP lainnya. Dari seluruh koneksi tersebut tidak terdeteksi adanya data yang dikirim |
| 250822-zjbbdadk9s | Di Windows 11, saat sampel dijalankan menghasilkan *child process* bernama "Windows.ManagementWorkplace.Workplace.Setting.exe" yang terlihat *legitimate*. Sementara itu pada Windows 10, sampel justru menghasilkan *child process* yang berbeda yaitu "KBDCZ2.exe" | Saat pertama kali dijalankan, sampel melakukan operasi *RegOpenKey*, *ReqQueryValue*, *RegCloseKey* pada lokasi registry **HKLM\System\CurrentControlSet\Services\bam\UserSetting**. Akses ini dilakukan untuk memeriksa dan mengubah aktivitas program yang berjalan. Setelah itu sampel melanjutkan ke tahap berikutnya dengan menyimpan payload utama di direktori AppData\local | Beberapa IP terlihat mencoba melakukan koneksi dengan ke sistem Windows, namun upaya tersebut tidak berhasil karena sistem berada di lingkungan yang terisolasi. IP yang terpantau yaitu 71.72.196.159:80, 174.4513.118:80, 94.23.237.171:443, dan masih banyak lainnya |

Tabel 8 menunjukkan bahwa semua sampel yang dianalisis menggunakan pendekatan analisis dinamis memiliki kemampuan yang beragam namun menerapkan konsep teknis yang sama. Dari sisi *file system activity*, 4 dari 5 menyimpan payload utamanya di direktori C:\Users\user\Appdata\Local, yang merupakan direktori sistem tersembunyi sehingga dapat mengurangi kemungkinan terdeteksi oleh pengguna. Selain itu, keseluruhan sampel menggunakan mekanisme *random filename generation*, dimana sampel menghasilkan *child process* dengan nama acak yang berbeda saat dijalankan pada Windows 10 dan Windows 11. Perbedaan nama ini menunjukkan *payload* yang dijalankan secara dinamis dan sulit diprediksi.

Kemampuan *persistence* yang dimiliki setiap sampel tergolong sangat *stealthy* pada saat pertama kali berinteraksi dengan sistem. 3 dari 5 sampel melakukan *CreateFile, Read File* pada direktori penting Windows seperti C:\Windows\SysWOW64 dan C:\Windows\System32. Khusus sampel berjenis DLL, mekanisme *persistence* dilakukan dengan memanfaatkan komponen bawaan Windows seperti regsvr32.exe dan rundll32.exe, yang secara *default* dianggap sebagai proses *legitimate* oleh sistem.

Dalam aspek *network communication*, seluruh sampel menggunakan teknik *beaconing*. *Beaconing* adalah mekanisme dimana malware secara berkala mengirimkan paket kecil ke server C2 sebagai tanda bahwa perangkat yang terinfeksi sudah aktif. Hal ini terlihat dari tidak adanya data dikirim selama koneksi berlangsung dan adanya pergantian IP C2 secara terus-menerus sebagai mekanisme *fallback* ketika server tujuan tidak merespon.

D. Implementasi Pencegahan Malware

Upaya mitigasi untuk mencegah serangan malware dilakukan dengan menerapkan teknik *signature-based*

*detection*. Pada tahap ini, *signature* yang dikembangkan berdasarkan *Indicator of Compromise* (IoC) dari hasil analisis malware kemudian diubah ke dalam format *signature* ClamAV. Signature tersebut kemudian diimplementasikan kedalam database ClamAV sehingga sistem mampu mengenali pola dari sampel Emotet.

Tabel 9 merupakan komponen-komponen yang menjadi dasar penyusunan signature dan diperoleh dari hasil analisis terhadap lima sampel Emotet. Setiap komponen yang teridentifikasi kemudian dijadikan acuan dalam proses pembuatan signature. Komponen inilah yang memastikan signature dapat mengenali sampel Emotet dengan lebih akurat.

*Tabel 9 List Komponen Signature*

| Kategori | Contoh |
|---|---|
| Library Based Pattern | Kernel32.dll, Advapi32.dll, dan User32.dll |
| API Based Pattern | CreateThread, LoadLibrary, GetCommandLine, CreateFile, ReadFile, GetModuleHandle, dan VirtualAlloc |

Gambar 5 merupakan signature clamAV yang berhasil dikembangkan sebagai hasil dari proses analisis terhadap sampel Emotet. Signature ini berfungsi sebagai pola deteksi yang dapat digunakan clamAV untuk mengenali malware lain dengan karakteristik yang serupa.

```
rule Trojan_Detector
{
    meta:
        description = "Signature deteksi trojan"
        author = "Azzam Albasith"

    strings:
        $library1 = "kernel32.dll" nocase ascii
        $library2 = "advapi32.dll" nocase ascii
        $library3 = "user32.dll" nocase ascii
        $api1 = "CreateThread" ascii
        $api2 = "LoadLibrary" ascii
        $api3 = "GetCommandLine" ascii
        $api4 = "CreateFile" ascii
        $api5 = "ReadFile" ascii
        $api6 = "GetModuleHandle" ascii
        $api7 = "VirtualAlloc" ascii
    condition:
        2 of ($library*) or 4 of ($api*)
}
```

*Gambar 5 Signature ClamAV*

E. Evaluasi Pencegahan Malware

Pada tahap ini, signature ClamAV yang telah dikembangkan masuk ke tahap pengujian untuk mengukur tingkat akurasinya dalam mendeteksi malware. Proses pengujian *signature* dilakukan menggunakan *confusion matrix*. Ilustrasi tabel *confusion matrix* ditunjukkan oleh tabel 10.

*Tabel 10 Confusion Matrix Deteksi Malware*

| | File Malware | File Benign |
|---|---|---|
| Prediksi Positif | True Positive (TP) | False Positive (FP) |
| Prediksi Negatif | False Negative (FN) | True Negative (TN) |

True Positive = Signature benar mendeteksi malware pada file yang memang malware
False Positive = Signature salah mendeteksi malware pada file yang sebenarnya benign
False Negative = Signature gagal mendeteksi malware pada file yang memang malware
True Negative = Signature benar tidak mendeteksi malware pada file yang sebenarnya benign

Proses evaluasi signature ClamAV dilakukan dengan menggunakan 200 file yang terdiri dari 100 file malware dengan varian yang berbeda dan 100 file benign yang diperoleh dari repositori dataset di Github[18]. Nilai akurasi yang diperoleh signature clamAV sebagai berikut:

TP = 78 TN = 97
FN = 22 FP= 3

$$Accuracy = \frac{(78 + 97)}{(78 + 97 + 3 + 22)} \times 100\% = 87{,}5\%$$

Hasil pengujian signature menunjukkan bahwa signature ClamAV berhasil mengidentifikasi 78 dari 100 sampel malware secara benar sebagai malware (*True Positive*) dan mengenali 97 dari 100 *file benign* sebagai file tidak berbahaya (*True Negative*). Berdasarkan hasil tersebut, tingkat akurasi deteksi yang diperoleh mencapai 87,5%, yang menunjukkan bahwa signature yang dikembangkan memiliki kemampuan deteksi yang cukup baik dalam membedakan file berbahaya dan file normal.

## V. KESIMPULAN

Berdasarkan hasil analisis terhadap lima sampel Emotet, tidak ditemukan perbedaan yang signifikan terkait dampak serangan antara Windows 10 dan Windows 11. Kedua sistem operasi menunjukkan respons dan pola yang serupa khususnya pada proses-proses yang terbentuk selama aktivitas malware berlangsung.

Karakteristik dari masing-masing sampel mulai dari nilai hash, informasi *toolchain*, dan *timestamps* yang terekam di dalam header file berhasil diidentifikasi dengan jelas melalui analisis statis. Melalui pendekatan ini, struktur internal setiap sampel dapat dipetakan dengan

e-ISSN: 2548-6861

lebih rinci, termasuk *compiler* atau *linker* yang digunakan, indikasi umur pembuatan file, serta pola *build* yang serupa antar sampel. Selain itu, library dan API yang dijalankan oleh malware juga dapat terlihat sehingga memberikan gambaran awal mengenai fungsi apa saja yang kemungkinan akan dieksekusi saat sampel aktif.

Perilaku setiap sampel juga berhasil dipahami melalui analisis dinamis yang memantau aktivitas saat malware benar-benar dijalankan. Observasi ini mencakup perubahan file system seperti pembuatan file baru atau modifikasi direktori, aktivitas *registry* seperti pembuatan *key persistence* atau pengaturan konfigurasi tambahan, serta upaya komunikasi jaringan yang mengarah ke server *command and control* (C2). Informasi perilaku ini memberikan gambaran nyata mengenai bagaimana Emotet beroperasi di lingkungan sistem, termasuk pola serangannya, teknik penyamaran, serta upaya mempertahankan keberadaannya di dalam perangkat yang terinfeksi.

Tindakan preventif dalam mencegah malware lain juga berhasil dilakukan melalui penerapan mekanisme *signature-based detection* menggunakan ClamAV. Pendekatan ini diuji dengan metode *confusion matrix* untuk mengukur performa deteksi secara kuantitatif. Berdasarkan hasil pengujian terhadap 200 file yang terdiri dari file benign dan malware, signature yang dikembangkan mampu mencapai nilai akurasi sebesar 87,5%. Hasil ini menunjukkan bahwa ClamAV tidak hanya mampu dalam mendeteksi malware Emotet, tetapi juga memiliki kemampuan untuk mengenali malware lain yang memiliki karakteristik yang serupa. Dengan demikian, mekanisme *signature-based detection* yang diterapkan dapat dijadikan sebagai salah satu solusi dalam menghadapi ancaman malware di lingkungan sistem operasi Windows. Meskipun begitu, *signature-based detection* memiliki memiliki keterbatasan dalam mendeteksi malware jenis zero-day karena pola atau signature dari malware tersebut belum tersedia dalam basis data. Oleh karena itu, metode ini kurang efektif untuk menghadapi malware dengan pola baru. Untuk mengatasi hal tersebut, diperlukan pendekatan tambahan seperti *behavior-based detection*, *heuristic analysis*, atau *machine learning* yang mampu mendeteksi anomali berdasarkan perilaku, bukan hanya berdasarkan pola yang sudah dikenal. Penelitian selanjutnya diharapkan dapat mengembangkan sistem deteksi dengan mengombinasikan pendekatan tambahan tersebut.

## DAFTAR PUSTAKA


[1] Badan Pusat Statistik, “Statistik Telekomunikasi Indonesia 2024.” [Online]. Available: https://www.bps.go.id/id/publication/2025/08/29/beaa2be400eda6ce6c636ef8/statistik-telekomunikasi-indonesia-2024.html

[2] Stat Counter Global Stats, “Desktop Windows Version Market Share Indonesia.” [Online]. Available: https://gs.statcounter.com/windows-version-market-share/desktop/indonesia/#monthly-202110-202505-bar

[3] Microsoft Support, “Windows 10 support has ended on October 14, 2025.” [Online]. Available: https://support.microsoft.com/en-gb/windows/windows-10-support-has-ended-on-october-14-2025-2ca8b313-1946-43d3-b55c-2b95b107f281

[4] D. Dhanya and Z. Wuragil, “BSSN Catat 3,64 Miliar Serangan Siber di Indonesia Setengah Tahun Ini.” [Online]. Available: https://www.tempo.co/digital/bssn-catat-3-64-miliar-serangan-siber-di-indonesia-setengah-tahun-ini-2056396

[5] Europol, “World’s most dangerous malware EMOTET disrupted through global action.” [Online]. Available: https://www.europol.europa.eu/media-press/newsroom/news/world%E2%80%99s-most-dangerous-malware-emotet-disrupted-through-global-action

[6] U.S. Department of Health and Human Services, “Emotet Malware: The Enduring and Persistent Threat to the Health Sector.” [Online]. Available: https://www.hhs.gov/sites/default/files/emotet-the-enduring-and-persistent-threat-to-the-hph-tlpclear.pdf

[7] Geeksforgeeks, “Static Malware Analysis.” Accessed: Dec. 02, 2025. [Online]. Available: https://www.geeksforgeeks.org/ethical-hacking/static-malware-analysis/

[8] GeeksforGeeks, “Dynamic Malware Analysis.” [Online]. Available: https://www.geeksforgeeks.org/ethical-hacking/dynamic-malware-analysis/

[9] CloudFlare, “What are indicators of compromise (IoC)?” [Online]. Available: https://www.cloudflare.com/learning/security/what-are-indicators-of-compromise/

[10] ClamAV, “Creating signatures for ClamAV.” [Online]. Available: https://docs.clamav.net/manual/Signatures.html

[11] Y. D. Puji Rahayu and Nanang Trianto, “Analisis Malware Menggunakan Metode Analisis Statis dan Dinamis untuk Pembuatan IOC Berdasarkan STIX Versi 2.1,” *Info Kripto*, vol. 15, no. 3, pp. 105–111, Nov. 2021, doi: 10.56706/ik.v15i3.30.

[12] I. Gunawan and A. Ferriyan, “Analisis Malware Botnet Proteus Pendekatan Static dan Dinamic,”

*SIMETRIS*, vol. 15, no. 1, pp. 12–17, Jul. 2021, doi: 10.51901/simetris.v15i01.172.

[13] H. N. Aditya, N. Widiyasono, and A. Rahmatulloh, "Analisis Malware Aquvaprn.exe Untuk Investigasi Sistem Operasi Dengan Metode Memory Forensics," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 10, no. 2, pp. 161–172, Aug. 2024, doi: 10.28932/jutisi.v10i2.6562.

[14] FIRST (Forum of Incident Response and Security Teams), "Malware Analysis Framework." [Online]. Available: https://www.first.org/global/sigs/malware/ma-framework/

[15] Geeksforgeeks, "Understanding The Confusion Matrix in Machine Learning", [Online]. Available: https://www.geeksforgeeks.org/machine-learning/confusion-matrix-machine-learning/

[16] Recorded Future Triage, "Malware Sample." [Online]. Available: https://tria.ge/s?q=family%3Aemotet

[17] Hornet Security, "What is Emotet? How Can I Protect Myself." [Online]. Available: https://www.hornetsecurity.com/en/knowledge-base/emotet/

[18] G.-A. Losif, "DikeDataset", [Online]. Available: https://github.com/iosifache/DikeDataset?tab=readme-ov-file