

MALWARE 1 ANALYSIS

RAMNICK FRANCIS P. RAMOS

METADATA

```
[12/02/25]seed@VM:~/.../malware$ file malware1.exe
malware1.exe: PE32 executable (GUI) Intel 80386, for MS Windows
[12/02/25]seed@VM:~/.../malware$ ls -lh malware1.exe
-rw-r--r-- 1 seed seed 242K Nov 23 2021 malware1.exe
[12/02/25]seed@VM:~/.../malware$ r2 -q -c 'iH-TimeDateStamp' malware1.exe
TimeDateStamp = 0x619d19c6
```

malware1.exe is identified as a Windows executable. Based on an inspection of the PE Header, it has been determined to be a PE32 (32-bit) Windows GUI application. This indicates that it operates within a Windows GUI subsystem.

Additionally, the compilation timestamp and header features align with those of a standard C/C++ binary.

CRYPTANALYSIS

```
[12/02/25]seed@VM:~/.../malware$ sha256sum malware1.exe
02e8efb9ef253dd7a61cee319c44bd7e1529d06d038f8fbefdbec58b9a95a144  malware1.exe
[12/02/25]seed@VM:~/.../malware$ md5sum malware1.exe
05abed0280458bb76d84ad0aefbde0db  malware1.exe
[12/02/25]seed@VM:~/.../malware$ shasum malware1.exe
c5a6f2687a8ef6c69fc1c85311f31681d097e2e6  malware1.exe
```

Using Linux hashing commands for sha256, md5sum, and sha1, the cryptographic hashes of the executable were calculated to identify the sample across various systems. These hash values can serve as identification indicators if the malware spreads to other machines. The collected values are displayed above.

SECTION ANALYSIS

PE sections

```
[12/02/25]seed@VM:~/.../malware$ r2 -q -c "iS" malware1.exe
[Sections]
nth paddr      size vaddr          vsize perm name
0 0x000000400 0x29400 0x00401000 0x2a000 -r-x .text
1 0x00029800 0xf400 0x0042b000 0x10000 -r-- .rdata
2 0x00038c00 0x1000 0x0043b000 0x2000 -rw- .data
3 0x00039c00 0x200 0x0043d000 0x1000 -r-- .gfps
4 0x00039e00 0x200 0x0043e000 0x1000 -rw- .tls
5 0x0003a000 0x2600 0x0043f000 0x3000 -r-- .reloc
```

The portable executable inspection will enable us to comprehend the standard set of sections found in Windows-native applications.

By examining the PE, we notice that the .text section holds the most significant data, indicating that the malware's operations may reside here. Additionally, the PE includes instructions in the .tls section, which implies that this malware utilizes multi-threading for background task exfiltration. Finally, the .reloc section contains relocation information, suggesting that this malware relocates data to memory.

STRING ANALYSIS

Domain, URL, IPs

```
[12/02/25]seed@VM:~/.../malware$ strings -n 5 malware1.exe | grep -Ei "http|https|ftp|.com|.net|.cn|.se"
sec"
cnsec.cmc191.net
cnsec.cmc191.net
cnsec.exe
cnsec.exe
CMSC191.COMC
This is cmc malware for use in CMSC191:cNSEC class @ICS-UPLB
Complete Object Locator'
CreateCompatibleBitmap
CreateCompatibleEtc
InternetConnectA
InternetConnectA
InternetSetOptionA
PutFileA
Win32.dll
GetComputerLineA
GetComputerNameW
```

Image/screenshot indicators

```
[12/02/25]seed@VM:~/.../malware$ strings -n 5 malware1.exe | grep -Ei "jpg|png|scr"
screen.jpg
screen.jpg
[Print Screen]
screen.jpg
```

Executable or commands

```
[12/02/25]seed@VM:~/.../malware$ strings -n 5 malware1.exe | grep -Ei "exe|cmd|run|shell"
!This program cannot be run in DOS mode.
\cnssec.exe
cnssec.exe
SOFTWARE\Microsoft\Windows\CurrentVersion\Run
executable format error
InitOnceExecuteOnce
.?AVruntime_error@std@@
```

- cnsec.cmc191.net appears repeatedly in the binary file. This suggests that the malware operates through a networked communication with this certain server.
- screen.jpg suggests that the malware, when it was run before, captures screenshot and stores them. This suggests that the malware captures screenshots and sends it to a certain cmc191.net

IMPORT ANALYSIS

NETWORK-RELATED IMPORTS (WININET)

KEYLOGGING/INPUT HOOKS

```
[12/02/25]seed@VM:~/.../malware$ r2 -q -c "ii-WININET" malware1.exe
1 0x0042b1bc NONE FUNC WININET.dll InternetOpenA
2 0x0042b1c0 NONE FUNC WININET.dll FtpPutFileA
3 0x0042b1c4 NONE FUNC WININET.dll InternetConnectA
4 0x0042b1c8 NONE FUNC WININET.dll InternetSetOptionA
[12/02/25]seed@VM:~/.../malware$ r2 -q -c "ii-USER32" malware1.exe
1 0x0042b184 NONE FUNC USER32.dll GetMessageA
2 0x0042b188 NONE FUNC USER32.dll TranslateMessage
3 0x0042b18c NONE FUNC USER32.dll DispatchMessageA
4 0x0042b190 NONE FUNC USER32.dll GetKeyState
5 0x0042b194 NONE FUNC USER32.dll GetAsyncKeyState
6 0x0042b198 NONE FUNC USER32.dll GetDC
7 0x0042b19c NONE FUNC USER32.dll ReleaseDC
8 0x0042b1a0 NONE FUNC USER32.dll GetClientRect
9 0x0042b1a4 NONE FUNC USER32.dll MessageBoxA
10 0x0042b1a8 NONE FUNC USER32.dll GetDesktopWindow
11 0x0042b1ac NONE FUNC USER32.dll SetWindowsHookExA
12 0x0042b1b0 NONE FUNC USER32.dll CallNextHookEx
13 0x0042b1b4 NONE FUNC USER32.dll UnhookWindowsHookEx
```

SCREENSHOTTING REGISTRY/PERSISTENCE

```
[12/02/25]seed@VM:~/.../malware$ r2 -q -c "ii-GDI32" malware1.exe
1 0x0042b010 NONE FUNC GDI32.dll BitBlt
2 0x0042b014 NONE FUNC GDI32.dll CreateCompatibleBitmap
3 0x0042b018 NONE FUNC GDI32.dll CreateCompatibleDC
4 0x0042b01c NONE FUNC GDI32.dll DeleteObject
5 0x0042b020 NONE FUNC GDI32.dll SelectObject
[12/02/25]seed@VM:~/.../malware$ r2 -q -c "ii-gdiplus" malware1.exe
1 0x0042b100 NONE FUNC gdiplus.dll GdiAlloc
2 0x0042b104 NONE FUNC gdiplus.dll GdiFree
3 0x0042b108 NONE FUNC gdiplus.dll GdiPlusStartup
4 0x0042b110 NONE FUNC gdiplus.dll GdiPlusShutdown
5 0x0042b1e0 NONE FUNC gdiplus.dll GdiCloneImage
6 0x0042b1e4 NONE FUNC gdiplus.dll GdiDisposeImage
7 0x0042b1e8 NONE FUNC gdiplus.dll GdiSaveImageToFile
8 0x0042b1ec NONE FUNC gdiplus.dll GdiCreateBitmapFromHBITMAP
9 0x0042b1f0 NONE FUNC gdiplus.dll GdiGetImageEncodersSize
10 0x0042b1f4 NONE FUNC gdiplus.dll GdiGetImageEncoders
[12/02/25]seed@VM:~/.../malware$ r2 -q -c "ii-ADVAPI32" malware1.exe
1 0x0042b000 NONE FUNC ADVAPI32.dll RegOpenKeyExA
2 0x0042b004 NONE FUNC ADVAPI32.dll RegSetValueExA
3 0x0042b008 NONE FUNC ADVAPI32.dll RegCloseKey
```

- The presence of network communication imports (such as InternetOpenA, FtpPutFileA, etc.) indicates that this malware establishes remote communication. The FTP file upload implies that files are being transmitted.
- The usage of BitBlt and other imports suggests that the malware is capable of capturing screenshots.
- The function GetAsyncKeyState indicates that the malware monitors activity while it is in operation.

VIRUS TOTAL ANALYSIS



SUMMARY OF FINDINGS

This malware is a Windows executable that masquerades as a Windows GUI, operating asynchronously in the background. In reality, its true function is to capture screenshots, primarily for surveillance purposes, and send them to a server.