

Detecting HermeticWiper | Splunk

By Splunk Threat Research Team

Published: 2022-03-10 · Archived: 2026-04-10 02:43:28 UTC

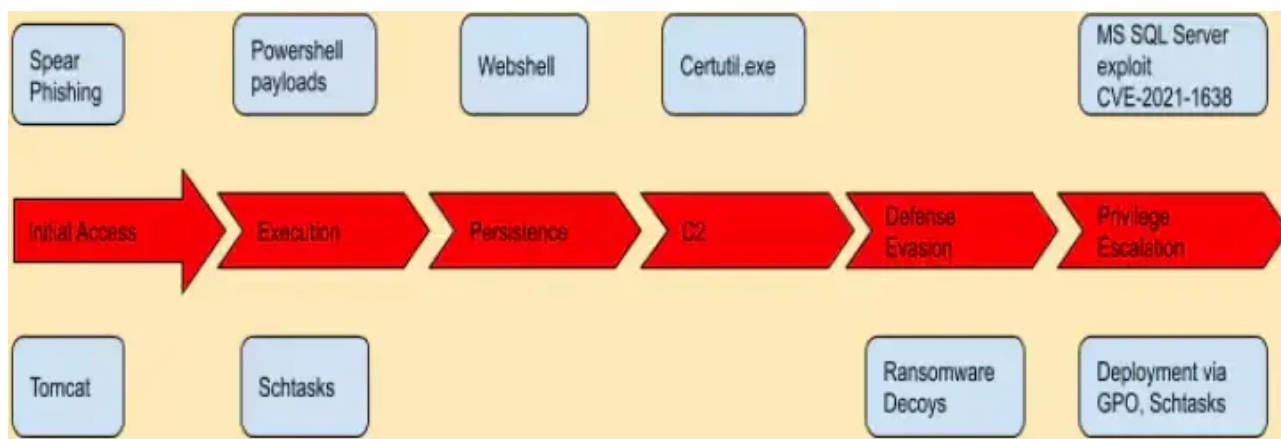
Splunk is committed to using inclusive and unbiased language. This blog post might contain terminology that we no longer use. For more information on our updated terminology and our stance on biased language, please visit [our blog post](#). We appreciate your understanding as we work towards making our community more inclusive for everyone.

As stated in our previous threat advisory [STRT-TA02](#) in regards to destructive software, past historical data suggests that for malicious actors to succeed in long-standing campaigns they must improve and add new ways of making their payloads stealthier, resistant, and damaging. HermeticWiper introduces some unique features, applying destructive actions on compromised hosts. In addition to other commonly known wiper destructive features, [HermeticWiper](#) also presents the following unique behaviors:

- Interacts with the system via signed driver
- Disables crash dump functionality (Anti-Forensic)
- Modifies “GlobalFolderOptions” registry at file permission level (NTFS)
- Checks for FAT (Windows XP) and NTFS (Windows OS newer than XP using NTFS)
- Corrupts (Destroys) MBR and NTFS file system
- Reported to have been deployed via Group Policy Object (Windows Active Directory Group Policy Object)

This payload is another destructive tool in the ongoing campaign which has included DDoS attacks, web defacements, [MDM attacks](#), [Microsoft SQL attacks](#) and now [two known as of yet destructive payloads](#).

STRT has also released a new analytic story covering [HermeticWiper](#) itself. We have collected information about the observed vectors in relation to HermeticWiper according to several security vendors including [Symantec](#), [ESET](#), [Sentinel One](#). The following diagram shows a visual flow of the observed attack vectors per tactic.



As seen above malicious actors are gaining initial access by either compromising publicly exposed services or via spear phishing, following the establishment of persistence and privilege escalation via web shells or the use of schtasks, PowerShell payloads, and finally deploying additional payloads via certutil.exe or Powershell which include genuine wiper payloads and ransomware decoy binaries seeking to distract and delay defense and containment from defenders. Here is a brief breakdown of HermeticWiper features and detections.

HermeticWiper Analysis

Signed driver (hermetic name reference)



Dropping Driver Component Base on Windows Version (XP or above)

This wiper will first adjust its token privileges with “SeShutdownPrivilege” and “SeBackupPrivilege” for later purposes like initiating shutdown or accessing files with high-security descriptor context.

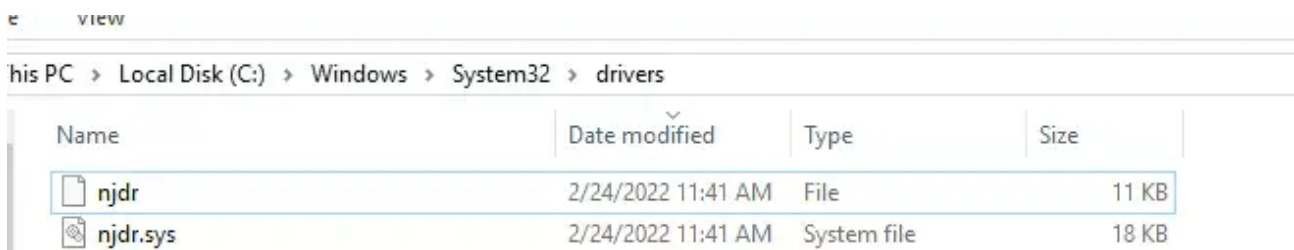
It contains 4 compressed drivers in its RSRCsection. It will drop one of those drivers depending on the Windows version or OS architecture of the compromised host by using VerifyVersionW API. Below is the summary table of the RSRC TYPE ID and the name of its rsrc entry for each driver.

Then it will generate random characters based on the current process ID of its running process. Once the wiper parses the needed rsrc entry, and has a filename, It will locate the C:\windows\system32\Drivers folder to drop its driver component.

The driver extracted from the rsrc section of this wiper is in LZW compressed (SZDD file format). The screenshot below shows how it uses LZ API to decompress that to retrieve the actual driver binary file.

```
v20 = LZOpenFileW(lpstrFileName, &ReOpenBuf, 2u);
if ( v20 >= 0 )
{
    PathAddExtensionW(pszDest, L".sys");
    hStyle = LZOpenFileW(lpstrFileName, &lpReOpenBuf_1, 0x1002u); // OF_CREATE | OF_READWRITE
    lpBuffer = hStyle;
    if ( hStyle >= 0 )
    {
        v22 = LZCopy(v20, hStyle); // mw_SZDDDecompression
        LZClose(v20);
        LZClose(lpBuffer);
        if ( v22 > 0 )
        {
            v23 = lpstrFileName;
            if ( v35 )
                v23 = StrStrIW(lpstrFileName, L"System32");
        }
    }
}
```

Interestingly during analysis, we found out that it drops both the compressed driver (<4 char random name> without file extension) and also the actual driver (<4 char random name> with .sys file extension) in C:\windows\system32\Drivers. Then it will delete the compressed version afterwards.



Disable Crash Dump

It also has some features where it disables the generation of crash dumps of the compromised host that serve as anti-forensic techniques. This is done by modifying a registry as shown in the screenshot below:

```
phkResult = 0;
if ( !RegOpenKeyW(HKEY_LOCAL_MACHINE, L"SYSTEM\\CurrentControlSet\\Control\\CrashControl", &phkResult) )
{
    *lpstrSystemDirPath = 0;
    RegSetValueExW(phkResult, L"CrashDumpEnabled", 0, 4u, lpstrSystemDirPath, 4u);
    RegCloseKey(phkResult);
}
```

Loading The Driver

The way it loads its driver component is by creating a service entry for that file. First It will adjust its token privilege with "SeLoadDriverPrivilege". If the service related to its driver does not exist it will just create and start a new service for it using CreateServiceW() and StartServiceW() API. If it already exists but is not active, it will modify the service config of that kernel driver to DEMAND_START to start the service. Below is the code, how it uses ChangeServiceConfigW() API to change the status of its driver if it is not active. This driver is a legitimate

component of the EaseUS Partition Master application. This file was leveraged by this wiper to interact and retrieve storage device information for its destructive purposes.

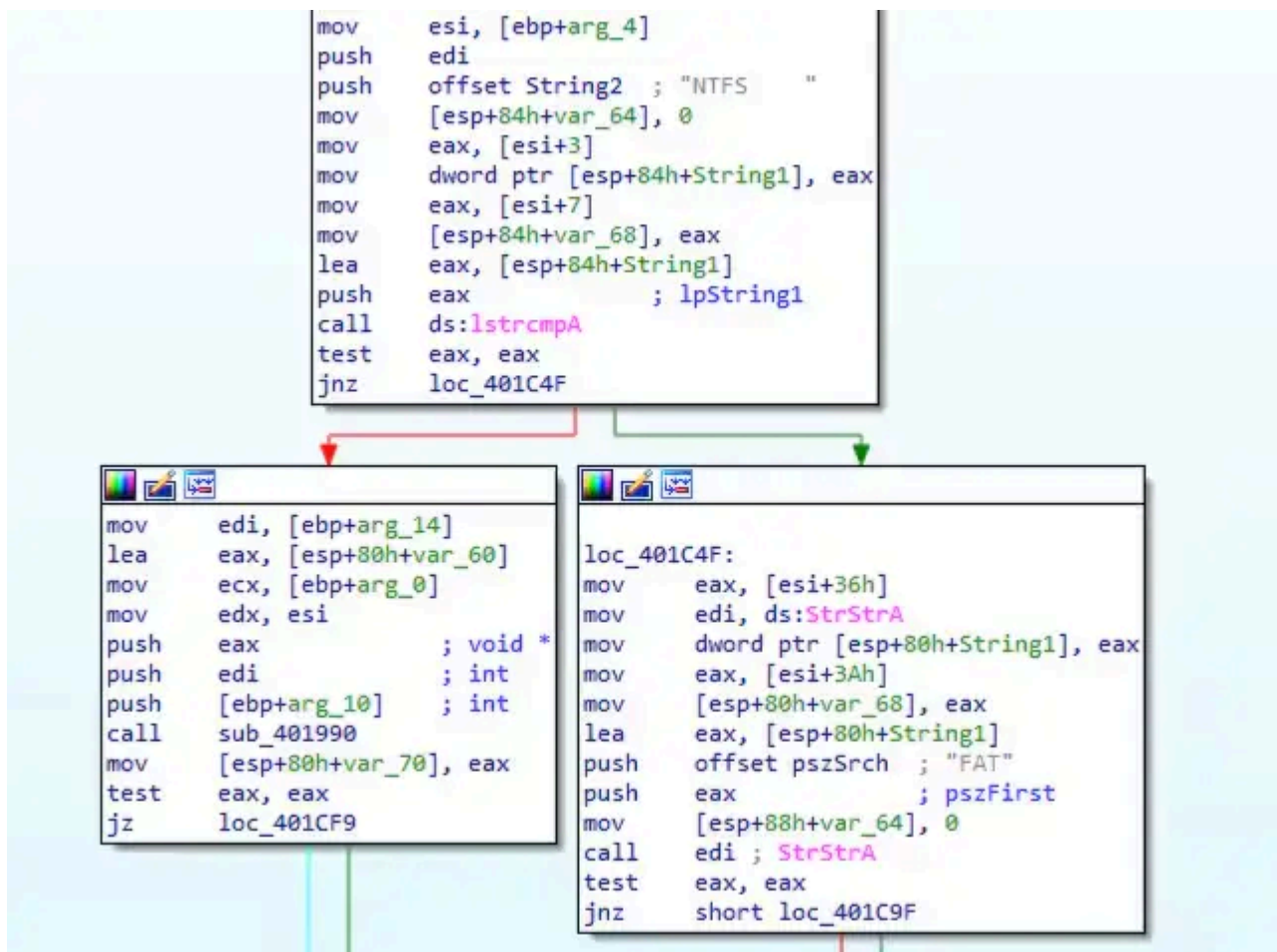
```
CurrentProcess = GetCurrentProcess();
if ( OpenProcessToken(CurrentProcess, 0x28u, &TokenHandle) )
{
    LookupPrivilegeValueW(0, L"SeLoadDriverPrivilege", &TOKEN_PRIVILEGES->Privileges[0].Luid);
    TOKEN_PRIVILEGES->PrivilegeCount = 1;
    TOKEN_PRIVILEGES->Privileges[0].Attributes = SE_PRIVILEGE_ENABLED;
    hSCManager = AdjustTokenPrivileges(TokenHandle, 0, TOKEN_PRIVILEGES, 0, 0, 0);
}
GetLastError();
v6 = GetProcessHeap();
HeapFree(v6, 0, TOKEN_PRIVILEGES);
if ( hSCManager )
{
    if ( lpBinaryPathName )
    {
        hSCManager_1 = OpenSCManagerW(0, L"ServicesActive", 3u);
        hSCManager = hSCManager_1;
        if ( !hSCManager_1 )
        {
            SetLastError = GetLastError();
            SetLastError(LastError);
            return 0;
        }
        ServiceW = OpenServiceW(hSCManager_1, lpServiceName, 0x16u);
        if ( ServiceW )
        {
            memset(&ServiceStatus, 0, sizeof(ServiceStatus));
            if ( QueryServiceStatus(ServiceW, &ServiceStatus) )
            {
                started = ServiceStatus.dwCurrentState == SERVICE_RUNNING;
            }
            else if ( !ChangeServiceConfigW(
                ServiceW,
                SERVICE_KERNEL_DRIVER,
                SERVICE_DEMAND_START,
                SERVICE_ERROR_NORMAL,
                lpBinaryPathName,
                0,
                0,
                0,
                0,
                0,
                0) )
            {
                v15 = ServiceW;
                wrap_CloseHandle = CloseServiceHandle;
                ErrorCode = GetLastError();
                CloseServiceHandle(v15);
                goto LABEL_13;
            }
        }
    }
}
```

Corrupting Boot Sectors

The wiper starts to enumerate all possible physical devices connected to the compromised host (range 0-100 device). Below is the code how it enumerates all the devices and retrieves partition information of each device using DeviceIoControl() API. The function named “mw_GetDeviceNumberAndGeometry” is the function it uses to check if the physical device is “FILE_DEVICE_DISK” type or not.

```
push    edi
push    ecx
push    offset pszFmt    ; "\\.\PhysicalDrive%"
xorps   xmm0, xmm0
mov     [ebp+var_1C], edx
lea    eax, [ebp+pszDest]
mov     [ebp+var_10], 0
push    104h            ; cchDest
xor     esi, esi
movq   [ebp+var_24], xmm0
xor     edi, edi
mov     [ebp+BytesReturned], esi
push    eax            ; pszDest
movups [ebp+var_44], xmm0
mov     [ebp+var_18], edi
movups xmmword ptr [ebp+dwBytes], xmm0
call   ds:wnsprintfw
add    esp, 10h
lea    eax, [ebp+var_50]
lea    edx, [ebp+var_44]
lea    ecx, [ebp+pszDest] ; lpFileName
push   eax            ; int
call   mw_GetDeviceNumberAndGeometry
mov    ebx, eax
cmp    ebx, 0FFFFFFFh
jz     loc_401F73
test   ebx, ebx
jz     loc_401FA8
mov    edi, 24C0h
push   edi            ; dwBytes
push   8              ; dwFlags
call   ds:GetProcessHeap
push   eax            ; hHeap
call   ds:HeapAlloc
push   0              ; lpOverlapped
mov    esi, eax
lea    eax, [ebp+BytesReturned]
push   eax            ; lpBytesReturned
push   edi            ; nOutBufferSize
push   esi            ; lpOutBuffer
push   0              ; nInBufferSize
push   0              ; lpInBuffer
push   IOCTL_DISK_GET_DRIVE_LAYOUT_EX ; dwIoControlCode
push   ebx            ; hDevice
call   ds:DeviceIoControl
call   ds:GetLastError
```

It also checks what File System type is present at Device, if it is either “NTFS” OR “FAT”. This checking will help the wiper to enumerate all of its partitions to corrupt all possible boot records on it. It also looks for known NTFS files like \$Bitmap, \$LogFile, \$DATA, and many more to be overwritten as part of its file destruction payload.



Below is the code of the Volume Boot Record partition before and after the infection of Hermetic wiper to the compromised host.

C:\Users\Publ... MBR-REWIND.exe -rv

+-----[DUMPHEX]-----+

OFFSET	: 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f	
0x00000000	: EB 58 90 4D 53 44 4F 53 35 2E 30 00 02 04 DE 19	.X.MSDOS5.0.....
0x00000010	: 02 00 00 00 00 00 F8 00 00 3F 00 FF 00 00 08 00 00?.....
0x00000020	: 00 40 06 00 11 03 00 00 00 00 00 00 02 00 00 00	.@.....
0x00000030	: 01 00 06 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000040	: 80 00 29 7C FC 80 C8 4E 4F 20 4E 41 4D 45 20 20	..) ...NO NAME
0x00000050	: 20 20 46 41 54 33 32 20 20 20 33 C9 8E D1 BC F4	FAT32 3.....
0x00000060	: 7B 8E C1 8E D9 BD 00 7C 88 56 40 88 4E 02 8A 56	{..... .V@.N..V
0x00000070	: 40 B4 41 BB AA 55 CD 13 72 10 81 FB 55 AA 75 0A	@.A..U..r...U.u.
0x00000080	: 00 E8 C1 01 74 05 FE 46 02 EB 2D 8A 56 40 B4 08 CD	...t..F...-V@...
0x00000090	: 13 73 05 B9 FF FF 8A F1 66 0F B6 C6 40 66 0F B6	.s.....f...@f..
0x000000a0	: D1 80 E2 3F F7 E2 86 CD C0 ED 06 41 66 0F B7 C9	...?.....Af...
0x000000b0	: 66 F7 E1 66 89 46 F8 83 7E 16 00 75 39 83 7E 2A	f..f.F..~..u9.~*
0x000000c0	: 00 77 33 66 8B 46 1C 66 83 C0 0C BB 00 80 B9 01	.w3f.F.f.....
0x000000d0	: 00 E8 2C 00 E9 A8 03 A1 F8 7D 80 C4 7C 8B F0 AC	..,.....} ...
0x000000e0	: 84 C0 74 17 3C FF 74 09 B4 0E BB 07 00 CD 10 EB	..t.<.t.....
0x000000f0	: EE A1 FA 7D EB E4 A1 7D 80 EB DF 98 CD 16 CD 19	...}...}.....
0x00000100	: 66 60 80 7E 02 00 0F 84 20 00 66 6A 00 66 50 06	f'~.... .fj.fP.
0x00000110	: 53 66 68 10 00 01 00 B4 42 8A 56 40 8B F4 CD 13	Sfh....B.V@....
0x00000120	: 66 58 66 58 66 58 66 58 66 58 66 3B 46 F8 72 03	fXfXfXfX.3f;F.r.
0x00000130	: F9 EB 2A 66 33 D2 66 0F B7 4E 18 66 F7 F1 FE C2	..*f3.f..N.f....
0x00000140	: 8A CA 66 8B D0 66 C1 EA 10 F7 76 1A 86 D6 8A 56	..f..f....v....V
0x00000150	: 40 8A E8 C0 E4 06 0A CC B8 01 02 CD 13 66 61 0F	@.....fa.
0x00000160	: 82 74 FF 81 C3 00 02 66 40 49 75 94 C3 42 4F 4F	.t....f@Iu..BOO
0x00000170	: 54 4D 47 52 20 20 20 20 00 00 00 00 00 00 00 00	TMGR
0x00000180	: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000190	: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000001a0	: 00 00 00 00 00 00 00 00 00 00 00 00 00 0A 44 69Di
0x000001b0	: 73 6B 20 65 72 72 6F 72 FF 0D 0A 50 72 65 73 73	sk error...Press
0x000001c0	: 20 61 6E 79 20 6B 65 79 20 74 6F 20 72 65 73 74	any key to rest
0x000001d0	: 61 72 74 0D 0A 00 00 00 00 00 00 00 00 00 00 00	art.....
0x000001e0	: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000001f0	: 00 00 00 00 00 00 00 00 AC 01 B9 01 00 00 55 AAU.
0x00000200	: 00	

+-----[DUMPHEX]-----+

C:\Users\Publ...hermeticwiper.exe

C:\Users\Publ... MBR-REWIND.exe -rv

+-----[DUMPHEX]-----+

OFFSET	: 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f	
0x00000000	: F2 16 50 2D 56 EF 75 73 FA 98 12 E7 3B 55 C5 81	..P-V.us....;U..
0x00000010	: 77 0E FD 01 AE BA FF 28 00 5F 06 13 C1 36 70 5A	w.....(._...6pZ
0x00000020	: EE 12 B4 BD 53 30 34 B5 48 F5 A1 57 20 3E 2A 33S04.H..W >*3
0x00000030	: EF 92 FD 0D C1 44 BD FF 13 C7 C8 5B 03 18 72 41D.....[.rA
0x00000040	: 5D B9 F7 EA 51 52 83 2A ED A6 A9 FE CB BB 18 9E]...QR.*.....
0x00000050	: E2 F7 88 8A 01 52 8A 92 99 14 85 A1 27 E7 03 26R.....'..&
0x00000060	: 90 FA B4 74 C6 F5 C3 D8 C8 73 FD 77 E9 A3 FD C5	...t.....s.w....
0x00000070	: 5C AA 46 DB 7B 87 F2 65 A3 16 6E 12 9E 4B 4C 55	\.F.{.e..n..KLU
0x00000080	: CB 64 99 8E 49 0D FB BC 94 9D 18 77 90 C4 D7 F1	.d..I.....w....
0x00000090	: EA 57 09 05 88 85 63 0A A5 46 F0 A9 54 47 AA 31	.W....c..F..TG.1
0x000000a0	: 58 90 85 68 34 58 35 0F 57 9C 2C 5F AA 0B 77 08	X..h4X5.W.,_..w.
0x000000b0	: 19 77 D4 41 D6 9B 50 BF 04 42 7E B9 9B F7 36 5D	.w.A..P..B~...6]
0x000000c0	: 05 49 0E 50 2D EA C5 37 4F DA 62 BA 72 B7 32 74	.I.P--70.b.r.2t
0x000000d0	: A6 C9 43 91 A4 ED 44 A7 64 53 8C 40 D9 E9 12 8A	..C...D.dS@....
0x000000e0	: 4C C4 1F 16 D4 6F 72 64 69 ED B2 2D 14 0D E1 37	L....ordi...-7
0x000000f0	: BC BC 53 A5 31 30 89 BA 0F 6F E4 25 3C C8 48 FB	..S.10...o.%<.H.
0x00000100	: AA FF 02 7C 2B 1A 28 02 8E 4A 97 E9 77 2D 6C 34	... +.(.J..w-14
0x00000110	: A8 16 8C 20 77 8E 69 C6 5D 61 F0 C9 D7 1D 2A 4F	... w.i.]a....*O.
0x00000120	: 29 71 EA E8 73 12 28 D9 AF 39 9D EC 22 4F 98 F6)q..s.(.9..*O.
0x00000130	: A7 28 61 6B BF 17 22 5E 3E 13 07 6D 9A 41 75 8C	.(ak..")^>..m.Au.

```

0x00000140 : 12 CB E4 AA D0 C5 C7 5F 07 AD 8E 24 CC 07 74 48 | .....7...$.LH
0x00000150 : 39 73 18 17 54 CA C8 DA 5E 2F 50 3B 0D 8D 26 35 | 9s..T...^/P;..&5
0x00000160 : C9 AD 1B 38 4B 3B 93 62 F9 B1 AD 6A 89 B2 DC EC | ...8K;.b...j....
0x00000170 : B6 B8 23 F5 0C AB 96 87 15 C4 9A 07 18 7C 14 E7 | ..#.....|..
0x00000180 : CB D1 F0 8E 47 B8 9A 80 53 19 B6 AA 6E D2 43 34 | ...G...S...n.C4
0x00000190 : C6 01 9F C0 30 F9 AE 8A 6E 43 C3 28 A6 78 5C 55 | ...0...nC.(.x\U
0x000001a0 : BF 50 1F CA BF A2 77 C5 46 69 E3 4C E3 2C 33 77 | .P....w.Fi.L.,3w
0x000001b0 : 56 6E 45 50 21 92 76 BA 22 67 B3 F0 51 38 26 6E | VnEP!.v."g..Q8&n
0x000001c0 : F6 29 66 F4 C4 1C 29 3D 81 B8 A6 C7 29 E7 38 4F | .)f...)=....).80
0x000001d0 : 65 4F F3 E7 75 95 2D 53 8B F3 28 2B E1 8D 72 F8 | eO..u.-S..(+..r.
0x000001e0 : 03 68 DE A8 21 AC B2 F7 B1 7F 07 7D CE 6C 02 FB | .h..!.....}.1..
0x000001f0 : B3 D2 52 CD 7F EA C7 BA 36 2C 93 89 EE 80 FB CE | ..R.....6,.....
0x00000200 : 00

+-----[DUMPHEX]-----+

```

Other Registry Modification

It also has a thread that will modify certain GlobalFolderOptions registry related to showing compressed files and information tips.

```

phkResult = 0;
if ( !RegOpenKeyW(HKEY_USERS, Name, &phkResult) )
{
    hKey = 0;
    if ( !RegOpenKeyW(phkResult, L"Software\\Microsoft\\Windows\\CurrentVersion\\Explorer\\Advanced", &hKey) )
    {
        *Data = 0;
        RegSetValueExW(hKey, L"ShowCompColor", 0, 4u, Data, 4u);
        RegSetValueExW(hKey, L"ShowInfoTip", 0, 4u, Data, 4u);
        RegCloseKey(hKey);
    }
    RegCloseKey(phkResult);
}

```

Trigger Shutdown

Another thread of this malware is responsible for shutting down the compromised host to make the corruption of boot sectors take effect.

```

.text:00403B40 ; DWORD __stdcall mw_InitiateSystemShutdown(LPVOID lpThreadParameter)
.text:00403B40 mw_InitiateSystemShutdown proc near ; DATA XREF: start+36F4o
.text:00403B44
.text:00403B48 lpThreadParameter= dword ptr 8
.text:00403B4C
.text:00403B4D
.text:00403B4E
.text:00403B4F
.text:00403B50
.text:00403B51 push ebp
.text:00403B52 mov ebp, esp
.text:00403B53 mov eax, [ebp+lpThreadParameter]
.text:00403B54 push dword ptr [eax] ; dwMilliseconds
.text:00403B55 call ds:Sleep
.text:00403B56 push 80020003h ; dwReason
.text:00403B57 push 1 ; bRebootAfterShutdown
.text:00403B58 push 1 ; bForceAppsClosed
.text:00403B59 push 0 ; dwTimeout
.text:00403B5A push 0 ; lpMessage
.text:00403B5B push 0 ; lpMachineName
.text:00403B5C call ds:InitiateSystemShutdownExW
.text:00403B5D test eax, eax
.text:00403B5E jz short loc_403B71
.text:00403B5F call ds:GetLastError
.text:00403B60 pop ebp
.text:00403B61 retn 4
.text:00403B62
.text:00403B63
.text:00403B64
.text:00403B65
.text:00403B66
.text:00403B67
.text:00403B68

```

Other Behaviors

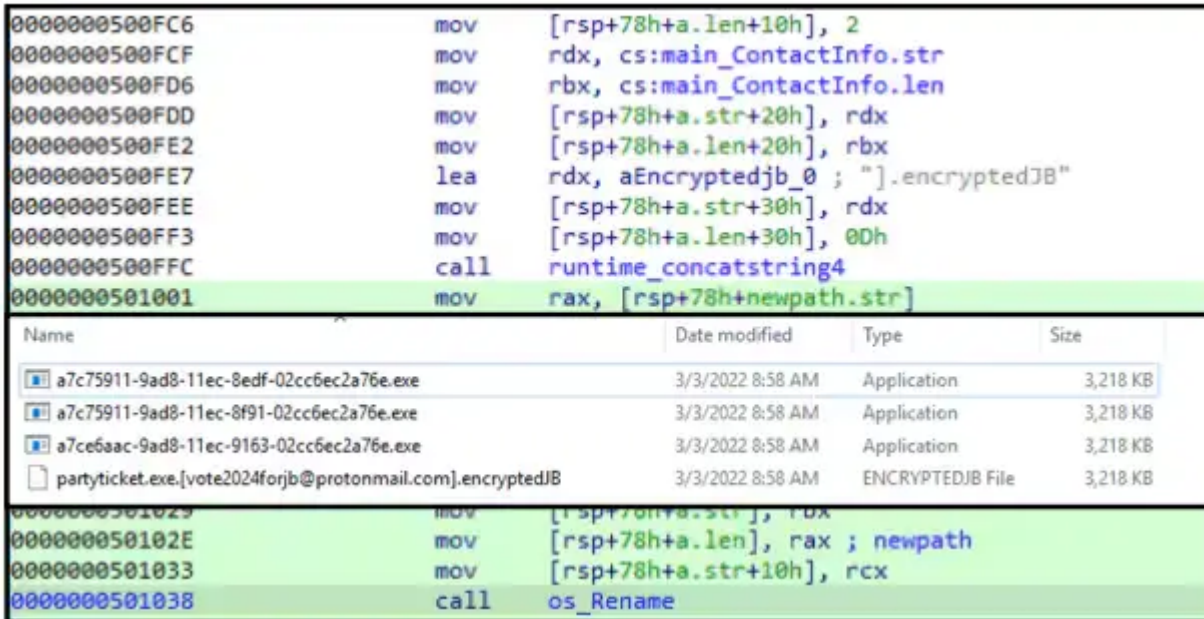
1. Check the C:\Windows\SYSDVOL attribute using GetFileAttributeW() API. If the API returns an invalid handle(possible return if the folder path does not exist) or if it is a folder path it will continue the execution if not exit the process.
2. Disables the VSS service which is related to volume shadow copy service to disable creation of backup copies.

It also has a function that can dismount or lock a disk volume.

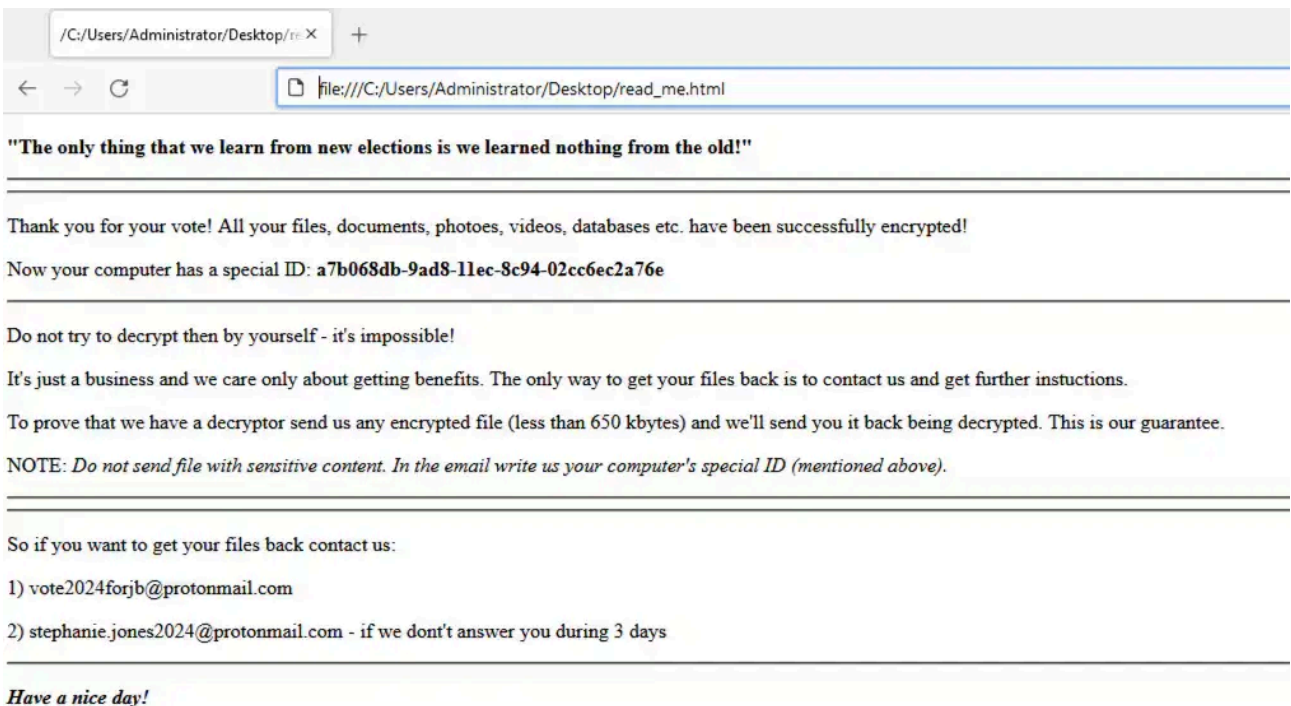
```
push    ebp
mov     ebp, esp
sub     esp, 20Ch
push    edi
push    [ebp+arg_0]
lea    eax, [ebp+FileName]
mov     [ebp+BytesReturned], 0
push    offset asc_4051F0 ; "\\\\.\\\\"
push    offset aS2s      ; "%s%.2s"
push    eax               ; LPWSTR
call   ds:wprintfW
add    esp, 10h
lea    eax, [ebp+FileName]
push    0                 ; hTemplateFile
push    0                 ; dwFlagsAndAttributes
push    3                 ; dwCreationDisposition
push    0                 ; lpSecurityAttributes
push    3                 ; dwShareMode
push    80100000h        ; dwDesiredAccess
push    eax               ; lpFileName
call   ds:CreateFileW
push    0                 ; lpOverlapped
mov    edi, eax
lea    eax, [ebp+BytesReturned]
push    eax               ; lpBytesReturned
push    0                 ; nOutBufferSize
push    0                 ; lpOutBuffer
push    0                 ; nInBufferSize
push    0                 ; lpInBuffer
push    FSCTL_LOCK_VOLUME ; dwIoControlCode
push    edi               ; hDevice
call   ds:DeviceIoControl
push    0                 ; lpOverlapped
lea    eax, [ebp+BytesReturned]
push    eax               ; lpBytesReturned
push    0                 ; nOutBufferSize
push    0                 ; lpOutBuffer
push    0                 ; nInBufferSize
push    0                 ; lpInBuffer
push    FSCTL_DISMOUNT_VOLUME ; dwIoControlCode
push    edi               ; hDevice
call   ds:DeviceIoControl
xor    eax, eax
pop    edi
mov    esp, ebp
pop    ebp
ret    8
```

PartyTicket Analysis

During [eset](#) analysis in this incident, they found another binary where they named it as “Hermetic Ransom”. This is a Golang compiled ransomware binary where it tries to encrypt files in the compromised host. Below is the screenshot of its code snippet where it renames the encrypted files with “.encryptedJB” file extension.



It will also drop a ransomware note in the desktop named as “read_me.html” to inform the user that their machine is compromised and encrypted.



Aside from its encryption features, this binary uses strings to its code function name that reference US President Biden.

f	main_selfElect	.text
f	main_subscribeNewPartyMember	.text
f	main_randomiseDuration	.text
f	main_highWay60	.text
f	main_voteFor403	.text

f	__C__projects_403forBiden_wHiteHousE_baggageGatherings	.text
f	__C__projects_403forBiden_wHiteHousE_lookUp	.text
f	__C__projects_403forBiden_wHiteHousE_primaryElectionProcess	.text
f	__C__projects_403forBiden_wHiteHousE_GoodOffice1	.text
f	__C__projects_403forBiden_wHiteHousE_init	.text

Detections

The following detections are focused specifically on HermeticWiper, Splunk STRT has a significant number of analytic stories that cover Ransomware which should also be considered when detecting and hunting for these types of threats.

Windows File Without Extension In Critical Folder

This analytic is to look for suspicious file creation in the critical folder like "System32\Drivers" folder without file extension.

```

| tstats `security_content_summariesonly` count FROM datamodel=Endpoint.Filesystem where Filesystem.file_path IN ("*\System32\drivers\*", "*\syswow64\drivers\*")
by _time span=5m Filesystem.dest Filesystem.user
Filesystem.file_name Filesystem.file_path Filesystem.process_guid Filesystem.file_create_time
| `drop_dm_object_name(Filesystem)`
| rex field="file_name" "\.(?<extension>[^\.]*)$"
| where isnull(extension)
| join process_guid
[] tstats `security_content_summariesonly` count FROM datamodel=Endpoint.Processes
by _time span=5m Processes.process_name Processes.dest Processes.process_guid
Processes.user
| `drop_dm_object_name(Processes)`
| stats count min(_time) as firstTime max(_time)
as lastTime by process_name process_guid file_name file_path file_create_time user dest
| `security_content_ctime(firstTime)`
| `security_content_ctime(lastTime)`
    
```

5 events (before 03/03/2022 10:32:43.000) No Event Sampling

Events Patterns **Statistics (1)** Visualization

20 Per Page Format Preview

process_name	process_guid	file_name	file_path	file_create_time	user
c.exe	{414E8EDF-CABB-6218-F103-000000003702}	fndr	C:\Windows\System32\drivers\fndr	2022-02-25 12:25:31.890	Administrator

Windows Raw Access To Master Boot Record Drive

This analytic is to look for suspicious raw access read to the device where the master boot record is placed.

New Search

```
'sysmon' EventCode=9 Device = \\Device\\Harddisk0\\DR0 NOT (Image IN("*\\Windows\\System32\\*", "*\\Windows\\SysWOW64\\\\"*))
| stats count min(_time) as firstTime max(_time) as lastTime by Image Device ProcessGuid ProcessId EventDescription EventCode Computer
| `security_content_ctime(firstTime)`
| `security_content_ctime(lastTime)`
```

✓ 1 event (before 25/02/2022 14:02:46.000) No Event Sampling ▾

Events Patterns **Statistics (1)** Visualization

20 Per Page ▾ Format Preview ▾

Image	Device	ProcessGuid	ProcessId	EventDescription	EventCode
C:\Temp\c.exe	\\Device\\Harddisk0\\DR0	{414E8EDF-CABB-6218-F103-000000003702}	6068	RawAccessRead	9

Windows Disable Memory Crash Dump

The following analytic identifies a process that is attempting to disable the ability on Windows to generate a memory crash dump.

```
| tstats `security_content_summariesonly` count FROM datamodel=Endpoint.Registry
WHERE (Registry.registry_path="*\\CurrentControlSet\\Control\\CrashControl\\CrashDumpEnabled") AND Registry.registry_value_data="0x00000000" BY _time span=1h Registry.dest Registry.user
Registry.registry_path Registry.registry_value_name Registry.registry_value_data
Registry.process_guid Registry.registry_key_name | `drop_dm_object_name(Registry)`
| join process_guid [| tstats `security_content_summariesonly`
count FROM datamodel=Endpoint.Processes BY _time span=1h Processes.process_id Processes.process_name
Processes.process Processes.dest Processes.parent_process_name Processes.parent_process
Processes.process_guid | `drop_dm_object_name(Processes)` | fields _time dest user parent_process_name parent_process process_name
process_path process process_guid registry_path registry_value_name registry_value_data
registry_key_name ] | table _time dest user parent_process_name parent_process process_name
process_path process process_guid registry_path registry_value_name registry_value_data
registry_key_name
```

✓ 1 event (before 2/28/22 2:26:37:000 PM) No Event Sampling ▾ Job ▾

Events Patterns **Statistics (1)** Visualization

20 Per Page ▾ Format Preview ▾

parent_process_name	parent_process	process_name	process_path	process	process_guid	registry_path	registry_value_name	registry_value_data
explorer.exe	C:\Windows\Explorer.EXE	c.exe		"C:\Temp\c.exe"	{414E8EDF-CABB-6218-F103-000000003702}	HKLMSystem\CurrentControlSet\Control\CrashControl\CrashDumpEnabled	CrashDumpEnabled	0x00000000

Windows Modify Show Compress Color And Info Tip Registry

This analytic is to look for suspicious registry modification related to file compression color and information tips.

New Search

```
| tstats `security_content_summariesonly` count FROM datamodel=Endpoint.Registry
WHERE Registry.registry_path = "*\\Microsoft\\Windows\\CurrentVersion\\Explorer\\Advanced*"
AND Registry.registry_value_name IN("ShowCompColor", "ShowInfoTip")
BY _time span=1h Registry.dest Registry.user Registry.registry_path Registry.registry_value_name
Registry.registry_value_data Registry.process_guid | `drop_dm_object_name(Registry)`
| rename process_guid as proc_guid | join proc_guid [| tstats `security_content_summariesonly`
count FROM datamodel=Endpoint.Processes BY _time span=1h Processes.process_id Processes.process_name
Processes.process Processes.dest Processes.parent_process_name Processes.parent_process
Processes.process_guid | `drop_dm_object_name(Processes)` | rename process_guid as
proc_guid | fields _time dest user parent_process_name parent_process process_name
process_path process proc_guid registry_path registry_value_name registry_value_data]
| table _time dest user parent_process_name parent_process process_name process_path
process proc_guid registry_path registry_value_name registry_value_data
```

✓ 2 events (02/03/2022 10:08:26.000 to 02/03/2022 10:23:26.000) No Event Sampling ▾ Job ▾ Smart Mode ▾

Events Patterns **Statistics (2)** Visualization

20 Per Page ▾ Format Preview ▾

parent_process_name	parent_process	process_name	process_path	process	process_guid	registry_path	registry_value_name	registry_value_data
cmd.exe	C:\Windows\system32\cmd.exe	reg.exe		reg add HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced /v ShowCompColor /t REG_DWORD /d 0 /f	{328C47E9-4599-621F-1A88-000000003602}	HKU\S-1-5-21-255986408-45527644-2136164048-5001\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced\ShowCompColor	ShowCompColor	0x00000000
cmd.exe	C:\Windows\system32\cmd.exe	reg.exe		reg add HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced /v ShowInfoTip /t REG_DWORD /d 0 /f	{328C47E9-4599-621F-1A88-000000003602}	HKU\S-1-5-21-255986408-45527644-2136164048-5001\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced\ShowInfoTip	ShowInfoTip	0x00000000

Mitigation

Many of these exploits can be prevented by following CISA guides for [preparation](#) and [hardening of systems, applications, and networks](#), including [MDM attacks](#) as well. There is also a [free HermeticRansom/PartyTicket decryptor](#) by AVAST and [CrowdStrike](#). The following table shows Splunk coverage of the aforementioned attack vectors in this ongoing campaign.

HermeticWiper Analytic Story is available in [ESCU release v3.36.0](#)

Also available from Splunk SOAR for automated response against these threats:

- Ransomware Investigate & Contain
- [Suspicious Email Domain Enrichment](#)
- [Crowdstrike Malware Triage](#)

Learn More

You can find the latest content about security analytic stories on [research.splunk.com](#). For a full list of security content, check out the [release notes](#) on [Splunk Docs](#).

Contributors

We would like to thank the following for their contributions to this post.

- Teoderick Contreras
- Rod Soto
- Jose Hernandez
- Patrick Barreiss
- Lou Stella
- Mauricio Velazco
- Michael Haag
- Bhavin Patel
- Eric McGinnis

Source: https://www.splunk.com/en_us/blog/security/detecting-hermeticwiper.html