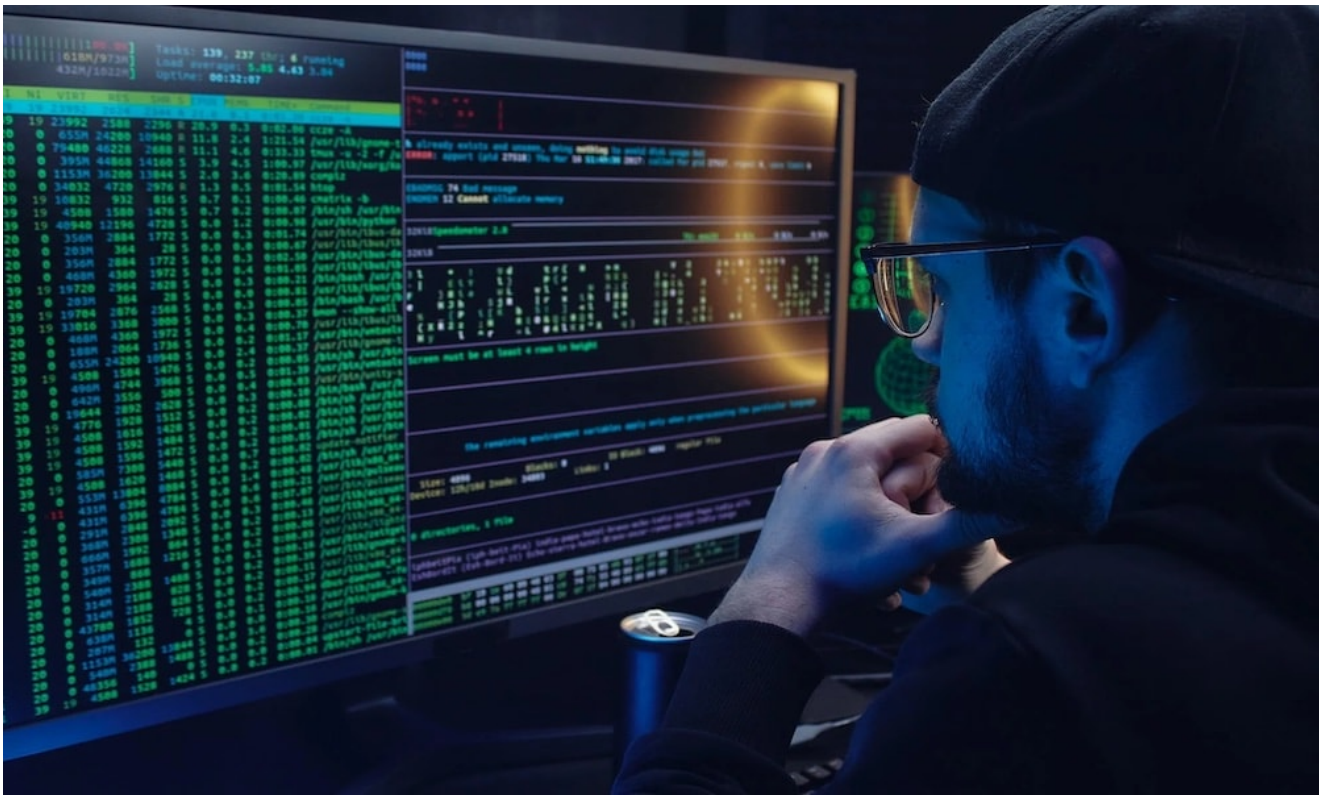
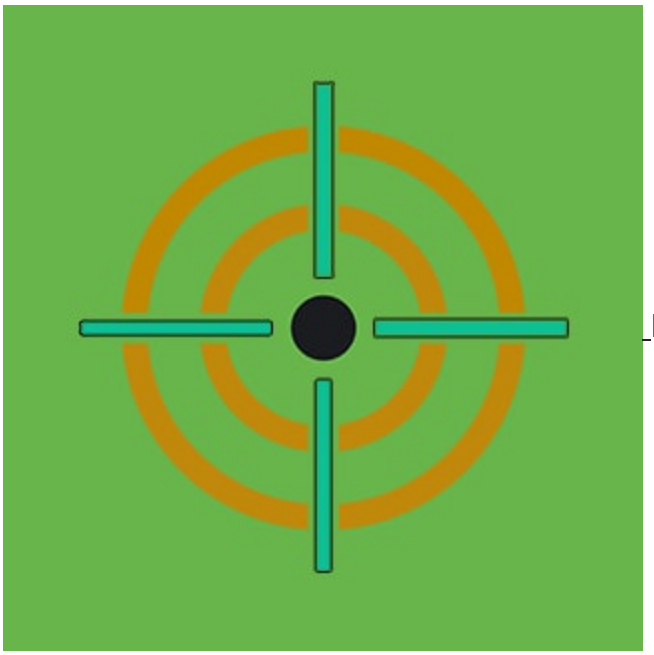
Detecting HermeticWiper

splunk.com/en_us/blog/security/detecting-hermeticwiper.html

March 10, 2022





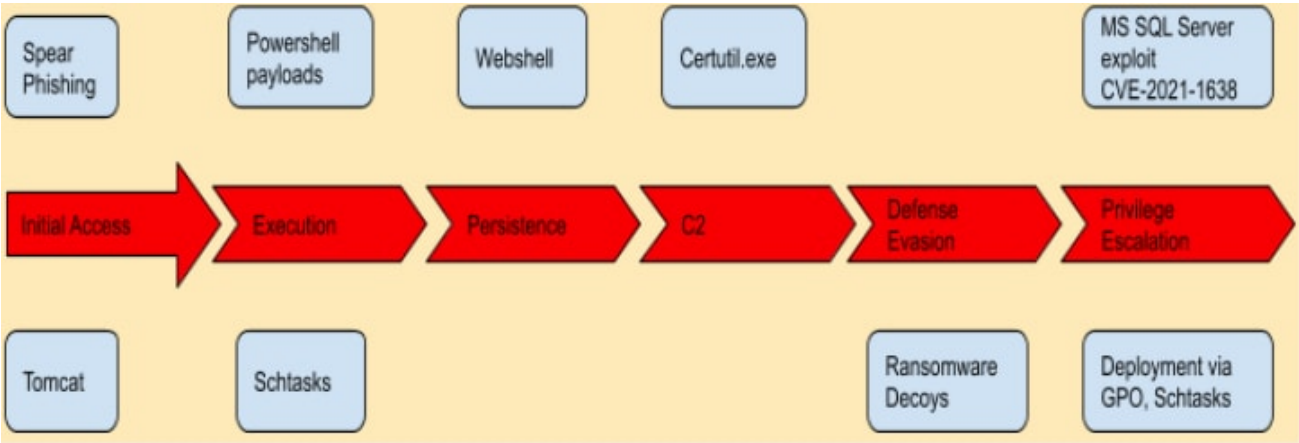
By Splunk Threat Research Team March 10, 2022

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.

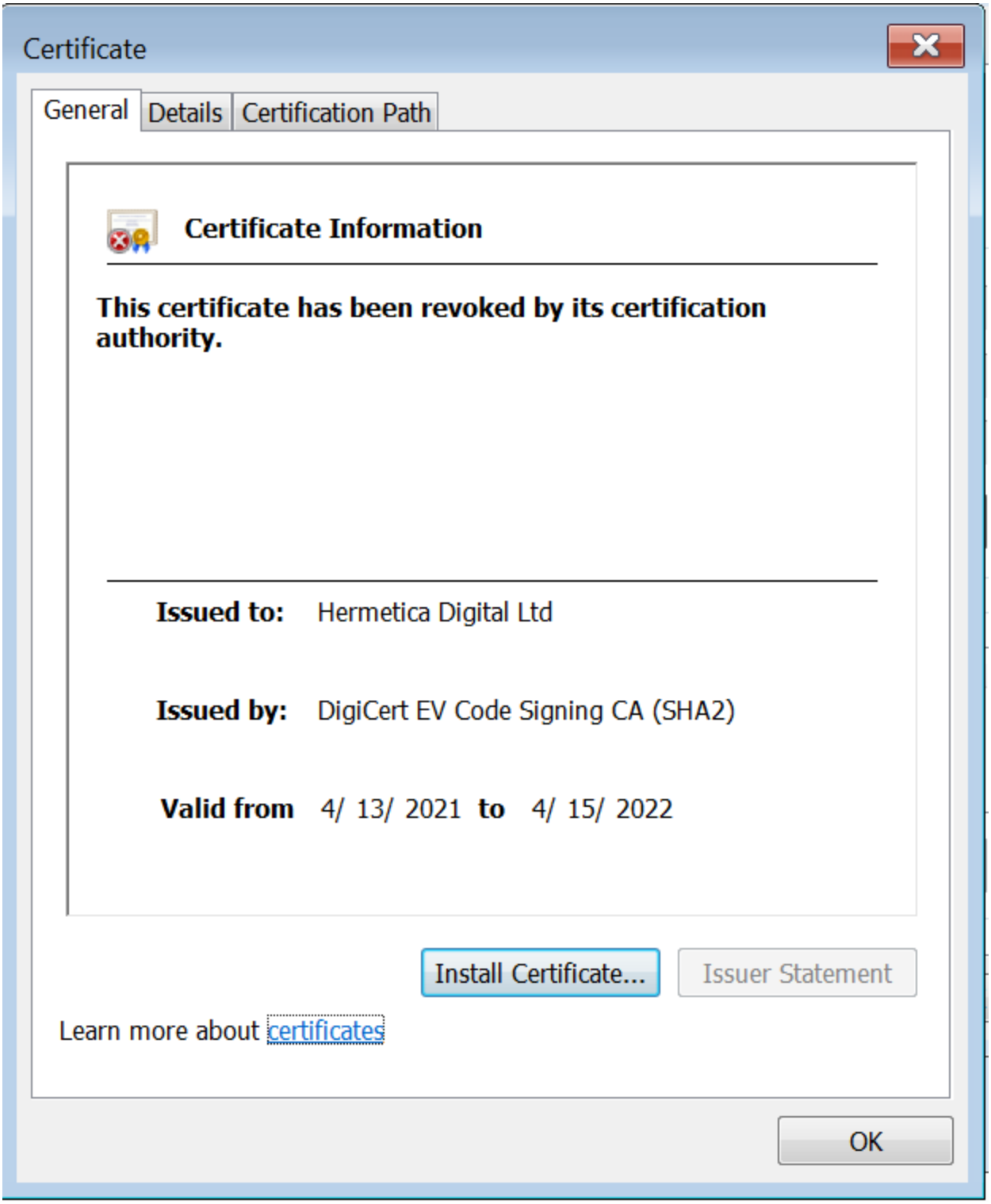


As seen above malicious actors are gaining initial access by either compromising publicly exposed services or via spear phishing, following the establishment of persistence and privilege escalation via web shells or the use of schtasks, PowerShell payloads, and finally deploying additional payloads via 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.

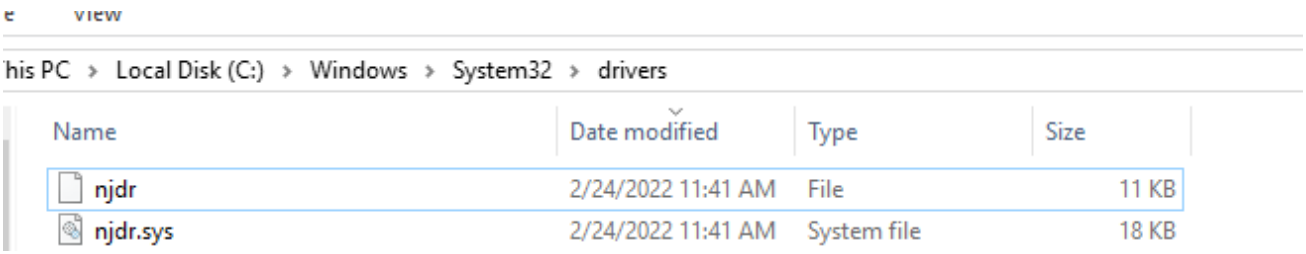
RSRC TYPE ID	RSRC NAME	Description
RCDATA	DRV_X64	Driver for x64 bit architecture
RCDATA	DRV_X32	Driver for x32 bit architecture
RCDATA	DRV_XP_X64	Driver for lower version OS (e.g XP) x64 bit architecture
RCDATA	DRV_XP_X64	Driver for lower version OS (e.g XP) x32 bit architecture

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_SZDDecompression
    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:

```
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 )
      {
        LastError = 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%u"
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, 0FFFFFFFFh
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.

```
mov     esi, [ebp+arg_4]
push    edi
push    offset String2  ; "NTFS    "
mov     [esp+84h+var_64], 0
mov     eax, [esi+3]
mov     dword ptr [esp+84h+String1], eax
mov     eax, [esi+7]
mov     [esp+84h+var_68], eax
lea     eax, [esp+84h+String1]
push    eax             ; lpString1
call    ds:lstrcmpA
test    eax, eax
jnz     loc_401C4F
```

```
mov     edi, [ebp+arg_14]
lea     eax, [esp+80h+var_60]
mov     ecx, [ebp+arg_0]
mov     edx, esi
push    eax             ; void *
push    edi             ; int
push    [ebp+arg_10]    ; int
call    sub_401990
mov     [esp+80h+var_70], eax
test    eax, eax
jz      loc_401CF9
```

```
loc_401C4F:
mov     eax, [esi+36h]
mov     edi, ds:StrStrA
mov     dword ptr [esp+80h+String1], eax
mov     eax, [esi+3Ah]
mov     [esp+80h+var_68], eax
lea     eax, [esp+80h+String1]
push    offset pszSrch  ; "FAT"
push    eax             ; pszFirst
mov     [esp+88h+var_64], 0
call    edi ; StrStrA
test    eax, eax
jnz     short loc_401C9F
```

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

```
C:\Users\Publi[redacted]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 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 : F6 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 EB 33 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 0D 0A 44 69 | ..............Di
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 AA | ..............U.
0x00000200 : 00

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


C:\Users\Publi[redacted]hermeticwiper.exe

C:\Users\Publi[redacted]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 33 | ....S04.H..W >*3
0x00000030 : EF 92 FD 0D C1 44 BD FF 13 C7 C8 5B 03 18 72 41 | .....D.....[..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 26 | .....R......'..&
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-..7O.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-l4
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 F8 | )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 3F 07 AD 8E 24 CC 07 74 48 | .......?...$..tH
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...)=....).8O
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..!......}.l..
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
.text:00403B40 ; DWORD __stdcall mw_InitiateSystemShutdown(LPVOID lpThreadParameter)
.text:00403B40 mw_InitiateSystemShutdown proc near     ; DATA XREF: start+36F↓o
.text:00403B40
.text:00403B40 lpThreadParameter= dword ptr  8
.text:00403B40
.text:00403B40                 push    ebp
.text:00403B41                 mov     ebp, esp
.text:00403B43                 mov     eax, [ebp+lpThreadParameter]
.text:00403B46                 push    dword ptr [eax] ; dwMilliseconds
.text:00403B48                 call    ds:Sleep
.text:00403B4E                 push    80020003h       ; dwReason
.text:00403B53                 push    1               ; bRebootAfterShutdown
.text:00403B55                 push    1               ; bForceAppsClosed
.text:00403B57                 push    0               ; dwTimeout
.text:00403B59                 push    0               ; lpMessage
.text:00403B5B                 push    0               ; lpMachineName
.text:00403B5D                 call    ds:InitiateSystemShutdownExW
.text:00403B63                 test    eax, eax
.text:00403B65                 jz      short loc_403B71
.text:00403B67                 call    ds:GetLastError
.text:00403B6D                 pop     ebp
.text:00403B6E                 retn    4
.text:00403B71 ; ---------------------------------------------------------------------------
```

Other Behaviors

1. Check the C:\Windows\SYSVOL 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:wsprintfW
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
retn    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.

```
0000000500FC6    mov    [rsp+78h+a.len+10h], 2
0000000500FCF    mov    rdx, cs:main_ContactInfo.str
0000000500FD6    mov    rbx, cs:main_ContactInfo.len
0000000500FDD    mov    [rsp+78h+a.str+20h], rdx
0000000500FE2    mov    [rsp+78h+a.len+20h], rbx
0000000500FE7    lea    rdx, aEncryptedjb_0 ; ".encryptedJB"
0000000500FEE    mov    [rsp+78h+a.str+30h], rdx
0000000500FF3    mov    [rsp+78h+a.len+30h], 0Dh
0000000500FFC    call   runtime_concatstring4
0000000501001    mov    rax, [rsp+78h+newpath.str]
```

Name	Date modified	Type	Size
a7c75911-9ad8-11ec-8edf-02cc6ec2a76e.exe	3/3/2022 8:58 AM	Application	3,218 KB
a7c75911-9ad8-11ec-8f91-02cc6ec2a76e.exe	3/3/2022 8:58 AM	Application	3,218 KB
a7ce6aac-9ad8-11ec-9163-02cc6ec2a76e.exe	3/3/2022 8:58 AM	Application	3,218 KB
partyticket.exe.[vote2024forjb@protonmail.com].encryptedJB	3/3/2022 8:58 AM	ENCRYPTEDJB File	3,218 KB

```
000000050102E    mov    [rsp+78h+a.len], rax ; newpath
0000000501033    mov    [rsp+78h+a.str+10h], rcx
0000000501038    call   os_Rename
```

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.

/C:/Users/Administrator/Desktop/re

file:///C:/Users/Administrator/Desktop/read_me.html

"The only thing that we learn from new elections is we learned nothing from the old!"

Thank you for your vote! All your files, documents, photoes, videos, databases etc. have been successfully encrypted!

Now your computer has a special ID: **a7b068db-9ad8-11ec-8c94-02cc6ec2a76e**

Do not try to decrypt then by yourself - it's impossible!

It's just a business and we care only about getting benefits. The only way to get your files back is to contact us and get further instuctions.

To prove that we have a decryptor send us any encrypted file (less than 650 kbytes) and we'll send you it back being decrypted. This is our guarantee.

NOTE: *Do not send file with sensitive content. In the email write us your computer's special ID (mentioned above).*

So if you want to get your files back contact us:

1) vote2024forjb@protonmail.com

2) stephanie.jones2024@protonmail.com - if we dont't answer you during 3 days

---

Have a nice day!

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

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

__C__projects_403forBiden_wHiteHousE_baggageGatherings	.text
__C__projects_403forBiden_wHiteHousE_lookUp	.text
__C__projects_403forBiden_wHiteHousE_primaryElectionProcess	.text
__C__projects_403forBiden_wHiteHousE_GoodOffice1	.text
__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 dest process_name process_guid file_name file_path file_create_time
user

  | `security_content_ctime(firstTime)`

  | `security_content_ctime(lastTime)`
```

```
| 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.

```
`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 Computer Image Device ProcessGuid ProcessId EventDescription EventCode

  | `security_content_ctime(firstTime)`

  | `security_content_ctime(lastTime)`
```

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
```

```
| 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 /NOUACCHECK	c.exe		"C:\Temp\c.exe"	{414E8EDF-CABB-6218-F103-000000003702}	HKLM\System\CurrentControlSet\Control\CrashControl\CrashDumpEnabled	CrashDumpEnabled	0x00000000

Windows Modify Show Compress Color And Info Tip Registry

| 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, _time [| 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

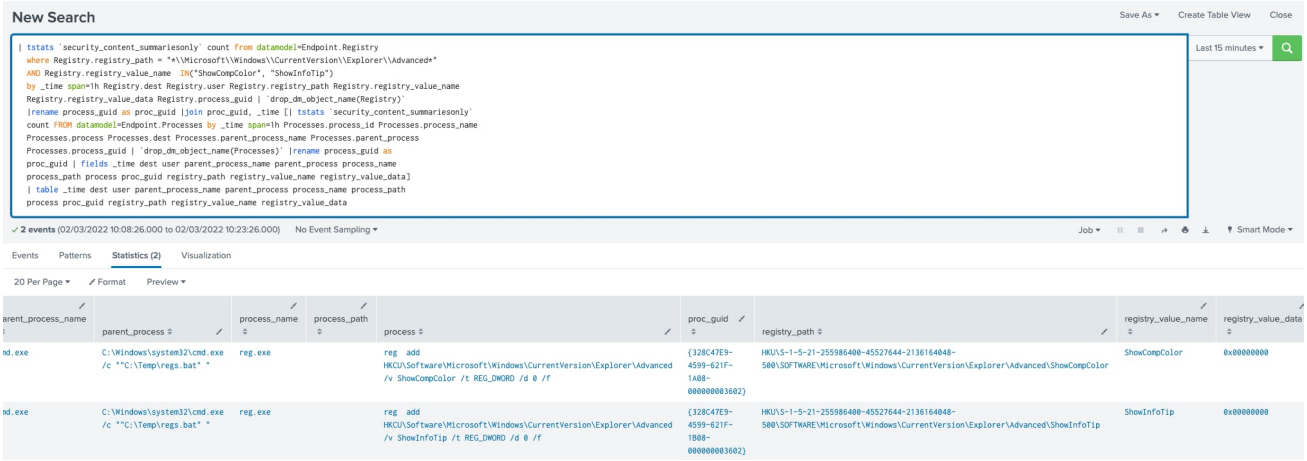
proc_guid | fields _time dest user parent_process_name parent_process process_name

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

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



Name	Technique ID	Tactic	Description
CMD Carry Out String Command Parameter	T1059.003	Execution	The following analytic identifies command-line arguments where cmd.exe /c is used to execute a program
Executable File Written in Administrative SMB Share	T1021.002	Lateral Movement	The following analytic identifies executable files (.exe or .dll) being written to Windows administrative SMB shares (Admin$, IPC$, C$)
Regsvr32 Silent and Install Param Dll Loading	T1218.010	Defense Evasion	This analytic is to detect a loading of dll using regsvr32 application with silent parameter and dllinstall execution.
Executables Or Script Creation In Suspicious Path	T1036	Execution	This analytic will identify suspicious executable or scripts (known file extensions) in list of suspicious file paths in Windows.

Suspicious Process File Path	T1543	Persistence, Privilege Escalation	The following analytic will detect a suspicious process running in a file path where a process is not commonly seen and is most commonly used by malicious software.
Impacket Lateral Movement Commandline Parameters	T1021 T1021.002 T1021.003 T1047 T1543.003	Lateral Movement Execution Persistence, Privilege Escalation	This analytic looks for the presence of suspicious commandline parameters typically present when using Impacket tools.
RunDLL Loading DLL By Ordinal	T1218 T1218.011	Defense Evasion	The following analytic identifies rundll32.exe loading an export function by ordinal value.
WevtUtil Usage To Clear Logs	T1070.001	Defense Evasion	The wevtutil.exe application is the windows event log utility. This searches for wevtutil.exe with parameters for clearing the application, security, setup, powershell, sysmon, or system event logs.
Windows Raw Access To Disk Volume Partition(New)	T1561.002	Impact	This analytic is to look for suspicious raw access read to device disk partitions of the host machine.
Windows Modify Show Compress Color And Info Tip Registry(New)	T1112	Defense Evasion	This analytic is to look for suspicious registry modification related to file compression color and information tips.
Windows Disable Memory Crash Dump(New)	T1485	Impact	The following analytic identifies a process that is attempting to disable the ability on Windows to generate a memory crash dump.

Windows File Without Extension In Critical Folder (New)	T1485	Persistence, Privilege Escalation	This analytic is to look for suspicious file creation in the critical folder like "System32\Drivers" folder without file extension.
Windows Raw Access To Master Boot Record Drive(New)	T1561.002	Impact	This analytic is to look for suspicious raw access read to drive where the master boot record is placed.

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.

Attack Vectors	Tactic	TTP	Splunk Coverage
Microsoft SQL Server CVE-2021-1636	Privilege Escalation	T1068	Windows Privilege Escalation
Webshell	Persistence	T1505	W3WP Spawning Shell
Tomcat	Initial Access	T1190	Linux Java Spawning Shell
Use of certutil.exe	Command & Control	T1105	Ingress Tool Transfer
Use Schtasks to execute payloads	Execution, Persistence, Privilege Escalation	T1053	Windows Persistence Techniques

Powershell payload execution	Execution	T1059.001	Malicious Powershell
Deployment via GPO	Defense Evasion, Privilege Escalation	T1484	Windows Privilege Escalation
Ransomware Decoys HermeticRansom/ PartyTicket	Defense Evasion	T1027	Ransomware Ransomware Investigate & Contain Ransomware Cloud
Spearphishing	Initial Access	T1566.002	Spearphishing attachments Suspicious Emails

HermeticWiper Analytic Story is available in ESCU release v3.36.0

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

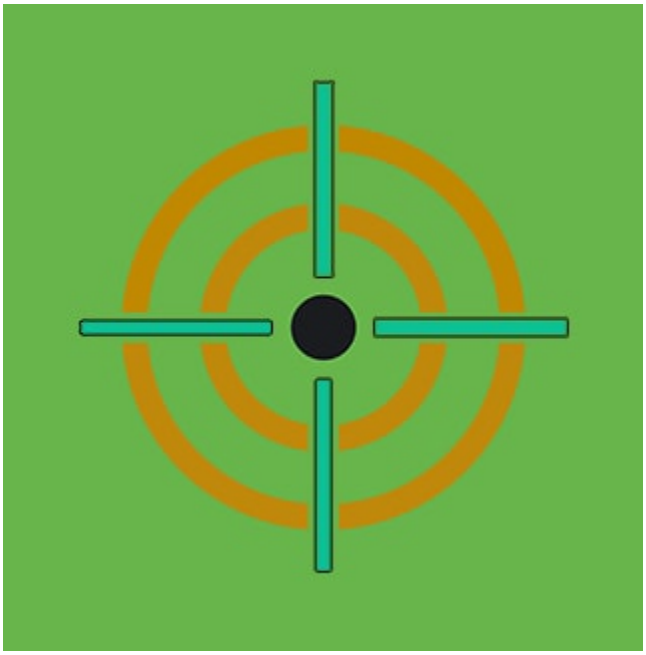
Learn More

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



Posted by

Splunk Threat Research Team

The Splunk Threat Research Team is an active part of a customer's overall defense strategy by enhancing Splunk security offerings with verified research and security content such as use cases, detection searches, and playbooks. We help security teams around the globe strengthen operations by providing tactical guidance and insights to detect, investigate and respond against the latest threats. The Splunk Threat Research Team focuses on understanding how threats, actors, and vulnerabilities work, and the team replicates attacks which are stored as datasets in the Attack Data repository.

Our goal is to provide security teams with research they can leverage in their day to day operations and to become the industry standard for SIEM detections. We are a team of industry-recognized experts who are encouraged to improve the security industry by sharing our work with the community via conference talks, open-sourcing projects, and writing white papers or blogs. You will also find us presenting our research at conferences such as Defcon, Blackhat, RSA, and many more.

Read more Splunk Security Content.