ThreatMon

SwiftSlicer Wiper Malware Analysis Report

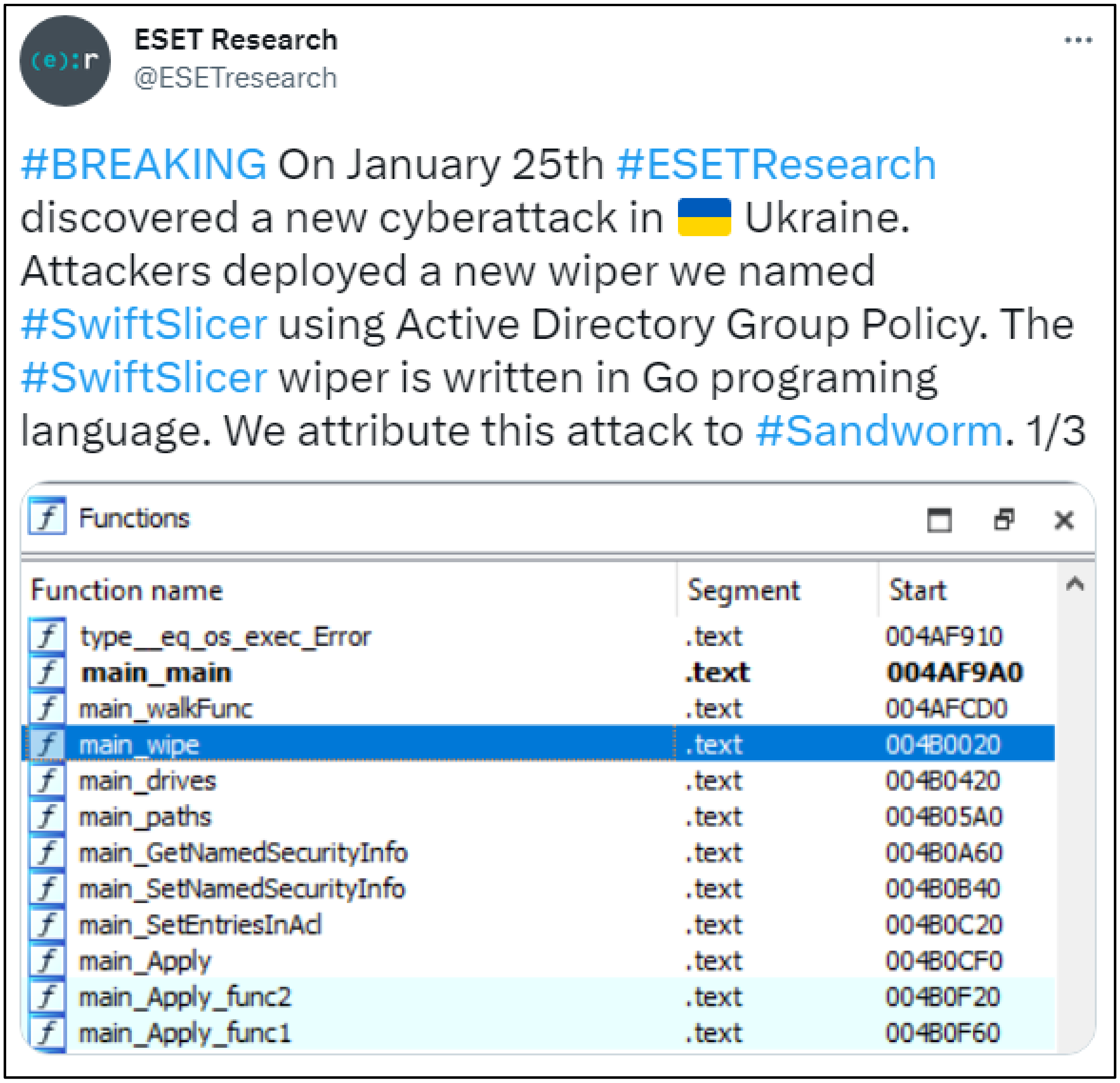
New Destructive Wiper Malware Targets Ukraine

@threatmon

@MonThreat

Executive Summary of SwiftSlicer Wiper

ESET, has uncovered a new wiper attack in Ukraine that has been attributed to the notorious Sandworm APT group. The malicious software, referred to as SwiftSlicer, was discovered on the network of a targeted organization on January 25th. The deployment of the malware was carried out through the use of Group Policy, indicating that the attackers had gained access and control over the victim's Active Directory environment. This discovery highlights the need for organizations to be vigilant in protecting their networks against advanced persistent threats.



ESET Research
@ESETresearch

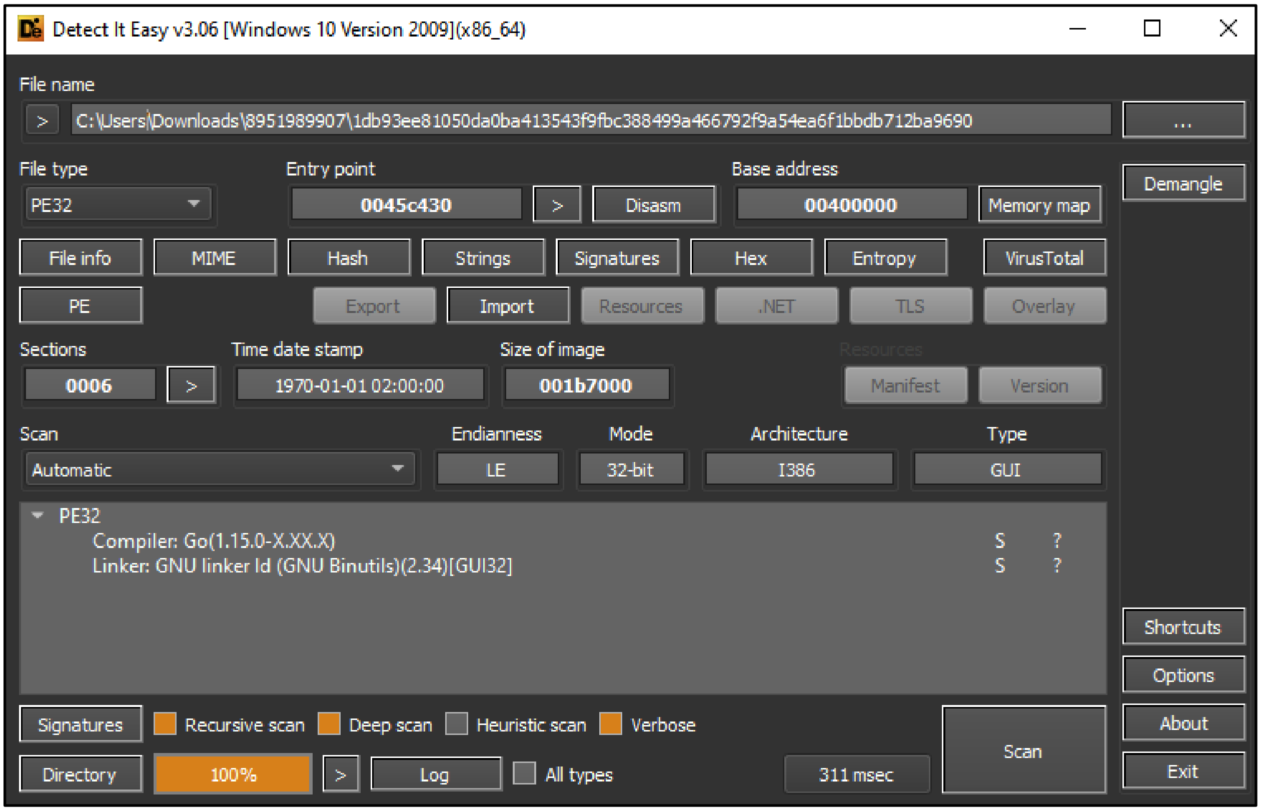
#BREAKING On January 25th #ESETResearch discovered a new cyberattack in 🇺🇦 Ukraine. Attackers deployed a new wiper we named #SwiftSlicer using Active Directory Group Policy. The #SwiftSlicer wiper is written in Go programing language. We attribute this attack to #Sandworm. 1/3

Function name	Segment	Start
type__eq_os_exec_Error	.text	004AF910
main_main	**.text**	**004AF9A0**
main_walkFunc	.text	004AFCD0
main_wipe	.text	004B0020
main_drives	.text	004B0420
main_paths	.text	004B05A0
main_GetNamedSecurityInfo	.text	004B0A60
main_SetNamedSecurityInfo	.text	004B0B40
main_SetEntriesInAcl	.text	004B0C20
main_Apply	.text	004B0CF0
main_Apply_func2	.text	004B0F20
main_Apply_func1	.text	004B0F60

Figure 1 - discover by ESET Research

Technical analysis

Utilizing Detect it Easy, we have determined that the SwiftSlicer malware has been written in the Go programming language and features a fabricated time stamp. This information can be observed in the accompanying illustration.



File name: C:\Users\Downloads\8951989907\1db93ee81050da0ba413543f9fbc388499a466792f9a54ea6f1bbdb712ba9690

File type: PE32
Entry point: 0045c430
Base address: 00400000

Sections: 0006
Time date stamp: 1970-01-01 02:00:00
Size of image: 001b7000

Scan: Automatic
Endianness: LE
Mode: 32-bit
Architecture: I386
Type: GUI

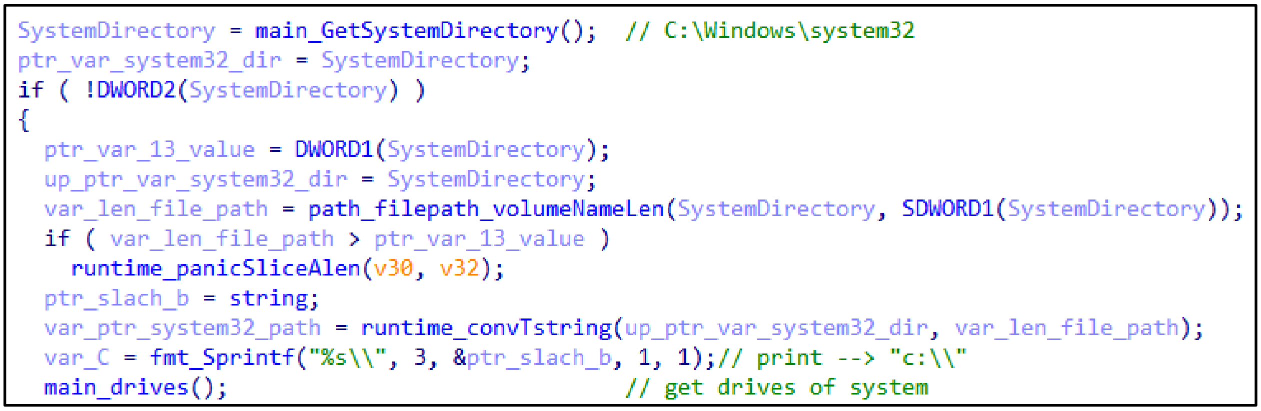
```
PE32
    Compiler: Go(1.15.0-X.XX.X)                              S    ?
    Linker: GNU linker ld (GNU Binutils)(2.34)[GUI32]        S    ?
```

☑ Recursive scan ☑ Deep scan ☐ Heuristic scan ☑ Verbose

100% ☐ All types 311 msec

Figure 2 - Basic static analysis

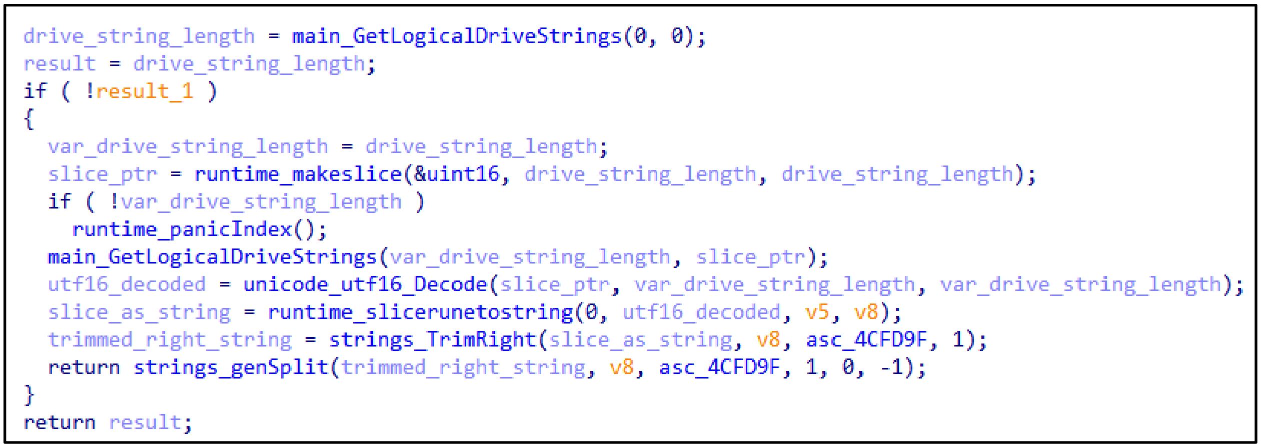
SwiftSlicer gets the system directory to determine the length of the volume and presents this information in the following illustration.



```
SystemDirectory = main_GetSystemDirectory();  // C:\Windows\system32
ptr_var_system32_dir = SystemDirectory;
if ( !DWORD2(SystemDirectory) )
{
  ptr_var_13_value = DWORD1(SystemDirectory);
  up_ptr_var_system32_dir = SystemDirectory;
  var_len_file_path = path_filepath_volumeNameLen(SystemDirectory, SDWORD1(SystemDirectory));
  if ( var_len_file_path > ptr_var_13_value )
    runtime_panicSliceAlen(v30, v32);
  ptr_slach_b = string;
  var_ptr_system32_path = runtime_convTstring(up_ptr_var_system32_dir, var_len_file_path);
  var_C = fmt_Sprintf("%s\\", 3, &ptr_slach_b, 1, 1);// print --> "c:\\"
  main_drives();                              // get drives of system
```

Figure 3 - Get system directories

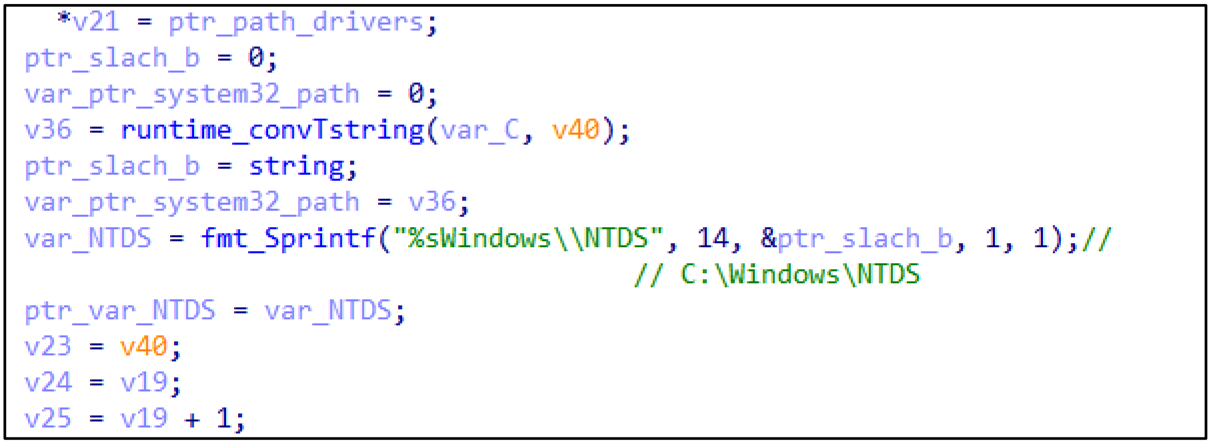
SwiftSlicer efficiently retrieves a comprehensive list of available drives in the system



```
drive_string_length = main_GetLogicalDriveStrings(0, 0);
result = drive_string_length;
if ( !result_1 )
{
  var_drive_string_length = drive_string_length;
  slice_ptr = runtime_makeslice(&uint16, drive_string_length, drive_string_length);
  if ( !var_drive_string_length )
    runtime_panicIndex();
  main_GetLogicalDriveStrings(var_drive_string_length, slice_ptr);
  utf16_decoded = unicode_utf16_Decode(slice_ptr, var_drive_string_length, var_drive_string_length);
  slice_as_string = runtime_slicerunetostring(0, utf16_decoded, v5, v8);
  trimmed_right_string = strings_TrimRight(slice_as_string, v8, asc_4CFD9F, 1);
  return strings_genSplit(trimmed_right_string, v8, asc_4CFD9F, 1, 0, -1);
}
return result;
```

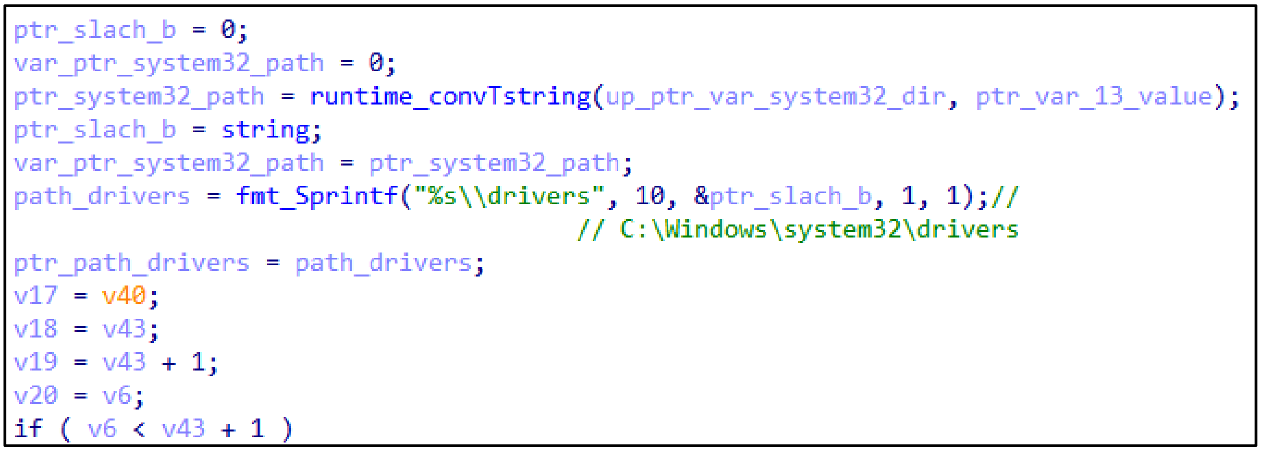
Figure 4 - Get local drives

Additionally, our analysis reveals that the malware has targeted the C:\Windows\system32\drivers and C:\Windows\NTDS directories. As shown in the accompanying illustration.



```
  *v21 = ptr_path_drivers;
ptr_slach_b = 0;
var_ptr_system32_path = 0;
v36 = runtime_convTstring(var_C, v40);
ptr_slach_b = string;
var_ptr_system32_path = v36;
var_NTDS = fmt_Sprintf("%sWindows\\NTDS", 14, &ptr_slach_b, 1, 1);//
                                        // C:\Windows\NTDS
ptr_var_NTDS = var_NTDS;
v23 = v40;
v24 = v19;
v25 = v19 + 1;
```

Figure 5 - NTDS directory

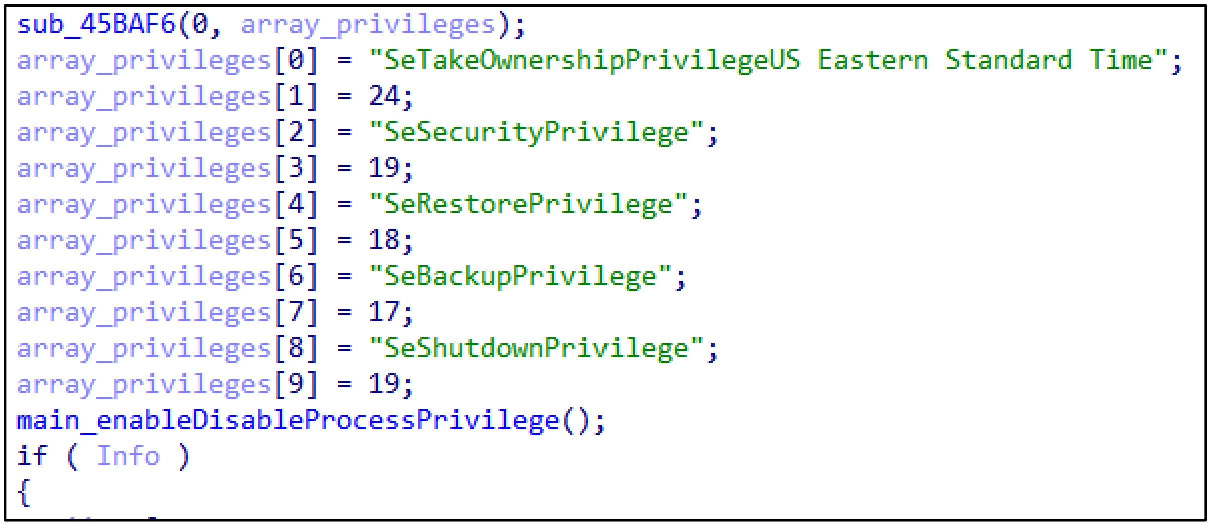


```
ptr_slach_b = 0;
var_ptr_system32_path = 0;
ptr_system32_path = runtime_convTstring(up_ptr_var_system32_dir, ptr_var_13_value);
ptr_slach_b = string;
var_ptr_system32_path = ptr_system32_path;
path_drivers = fmt_Sprintf("%s\\drivers", 10, &ptr_slach_b, 1, 1);//
                                         // C:\Windows\system32\drivers
ptr_path_drivers = path_drivers;
v17 = v40;
v18 = v43;
v19 = v43 + 1;
v20 = v6;
if ( v6 < v43 + 1 )
```

Figure 6 - drivers directory

SwiftSlicer malware enables 5 privileges and we can see more information about them

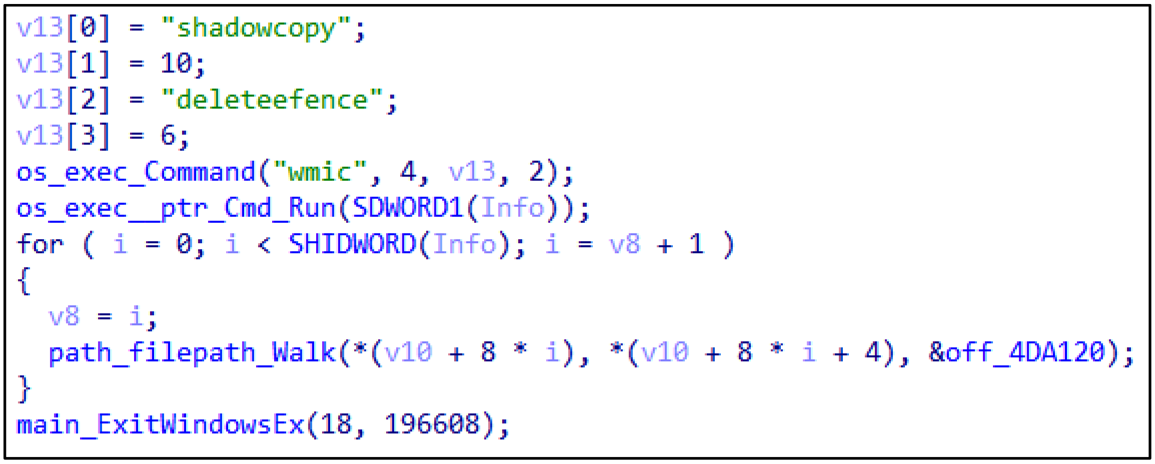
- SeTakeOwnershipPrivilege: allows a user to take ownership of any file or folder, even if the user doesn't have permission to do so.
- SeSecurityPrivilege: allows a user to modify security settings on a file or folder, such as permissions and auditing.
- SeRestorePrivilege: allows a user to restore files and directories that were backed up by the system.
- SeBackupPrivilege: allows a user to back up files and directories.
- SeShutdownPrivilege: allows a user to shut down the system.



```
sub_45BAF6(0, array_privileges);
array_privileges[0] = "SeTakeOwnershipPrivilegeUS Eastern Standard Time";
array_privileges[1] = 24;
array_privileges[2] = "SeSecurityPrivilege";
array_privileges[3] = 19;
array_privileges[4] = "SeRestorePrivilege";
array_privileges[5] = 18;
array_privileges[6] = "SeBackupPrivilege";
array_privileges[7] = 17;
array_privileges[8] = "SeShutdownPrivilege";
array_privileges[9] = 19;
main_enableDisableProcessPrivilege();
if ( Info )
{
```

Figure 7 - enable privileges

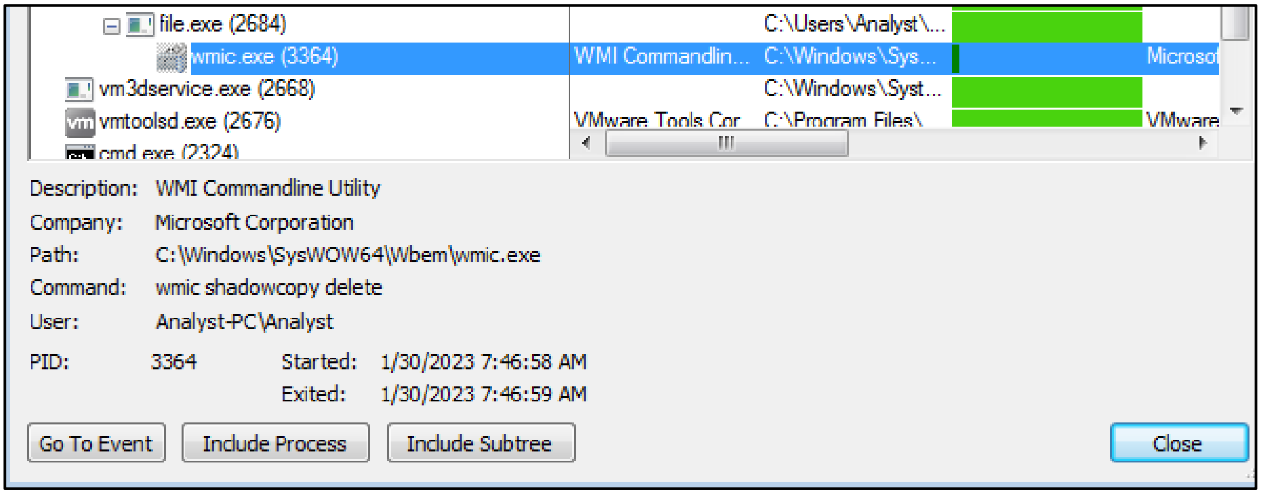
After that SwiftSlicer has successfully executed the 'wmic' command to delete the shadow copy which iis a feature in Windows operating system that allows users to take snapshots of the entire system or individual files at a certain point in time, as evident in the accompanying figure.



```
v13[0] = "shadowcopy";
v13[1] = 10;
v13[2] = "deleteefence";
v13[3] = 6;
os_exec_Command("wmic", 4, v13, 2);
os_exec__ptr_Cmd_Run(SDWORD1(Info));
for ( i = 0; i < SHIDWORD(Info); i = v8 + 1 )
{
  v8 = i;
  path_filepath_Walk(*(v10 + 8 * i), *(v10 + 8 * i + 4), &off_4DA120);
}
main_ExitWindowsEx(18, 196608);
```

Figure 8 - delete shadow copy

And we can see that malware creates a process to execute wmic and we can see that in the next figure.



```
⊟ file.exe (2684)                                   C:\Users\Analyst\...
    wmic.exe (3364)          WMI Commandlin...      C:\Windows\Sys...           Microsof
  vm3dservice.exe (2668)                            C:\Windows\Syst...
  vmtoolsd.exe (2676)        VMware Tools Cor       C:\Program Files\           VMware
  cmd.exe (2324)
```

Description: WMI Commandline Utility

Company: Microsoft Corporation

Path: C:\Windows\SysWOW64\Wbem\wmic.exe

Command: wmic shadowcopy delete

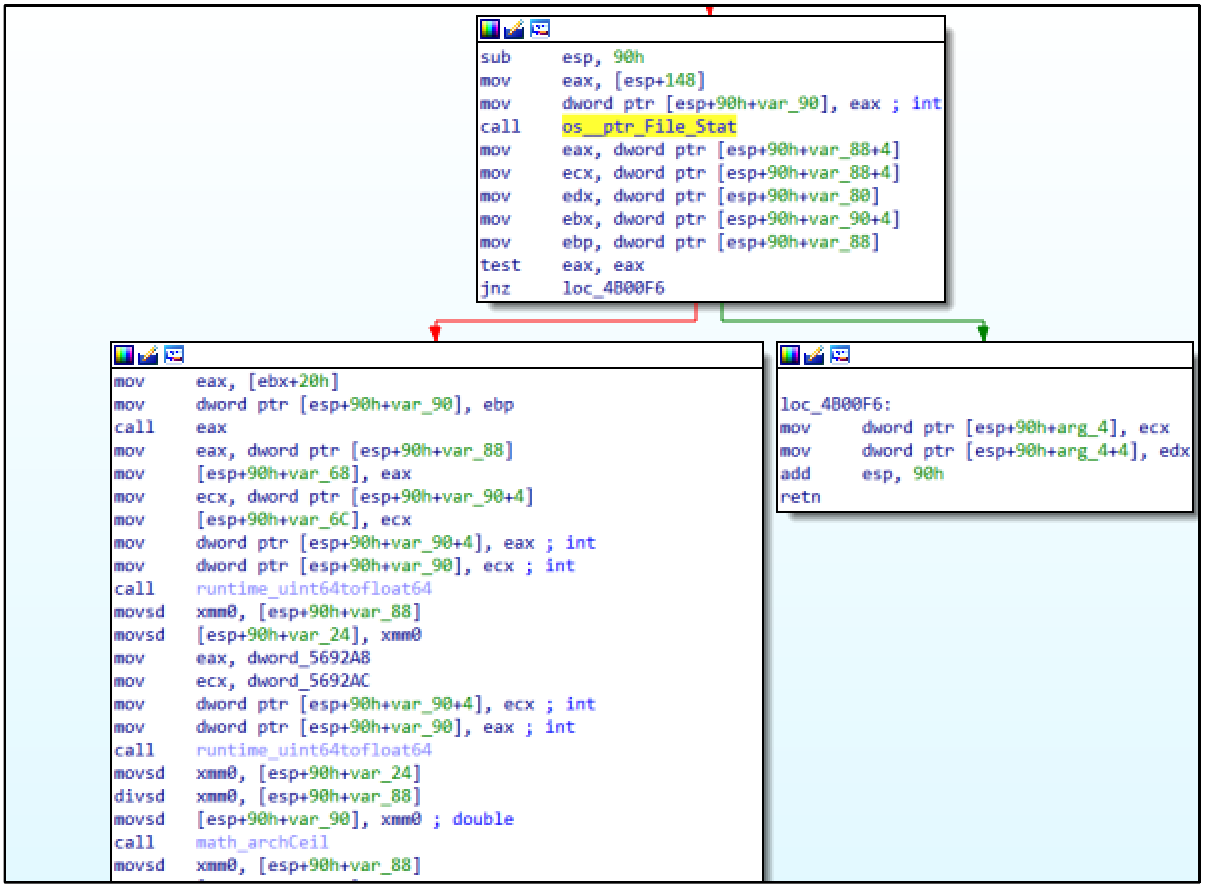
User: Analyst-PC\Analyst

PID: 3364 Started: 1/30/2023 7:46:58 AM

Exited: 1/30/2023 7:46:59 AM

Figure 9 - wmic.exe process

The SwiftSlicer malware operates in a highly efficient way, utilizing 4096 byte blocks to overwrite targeted data. The blocks are filled with randomly generated bytes, ensuring complete and thorough destruction of the targeted information. Once the data destruction process is complete, the malware reboots the system, leaving no residual evidence of the previous data.



```
sub     esp, 90h
mov     eax, [esp+148]
mov     dword ptr [esp+90h+var_90], eax ; int
call    os__ptr_File_Stat
mov     eax, dword ptr [esp+90h+var_88+4]
mov     ecx, dword ptr [esp+90h+var_88+4]
mov     edx, dword ptr [esp+90h+var_80]
mov     ebx, dword ptr [esp+90h+var_90+4]
mov     ebp, dword ptr [esp+90h+var_88]
test    eax, eax
jnz     loc_4800F6
```

```
mov     eax, [ebx+20h]
mov     dword ptr [esp+90h+var_90], ebp
call    eax
mov     eax, dword ptr [esp+90h+var_88]
mov     [esp+90h+var_68], eax
mov     ecx, dword ptr [esp+90h+var_90+4]
mov     [esp+90h+var_6C], ecx
mov     dword ptr [esp+90h+var_90+4], eax ; int
mov     dword ptr [esp+90h+var_90], ecx ; int
call    runtime_uint64tofloat64
movsd   xmm0, [esp+90h+var_88]
movsd   [esp+90h+var_24], xmm0
mov     eax, dword_5692A8
mov     ecx, dword_5692AC
mov     dword ptr [esp+90h+var_90+4], ecx ; int
mov     dword ptr [esp+90h+var_90], eax ; int
call    runtime_uint64tofloat64
movsd   xmm0, [esp+90h+var_24]
divsd   xmm0, [esp+90h+var_88]
movsd   [esp+90h+var_90], xmm0 ; double
call    math_archCeil
movsd   xmm0, [esp+90h+var_88]
```

```
loc_4800F6:
mov     dword ptr [esp+90h+arg_4], ecx
mov     dword ptr [esp+90h+arg_4+4], edx
add     esp, 90h
retn
```

Figure 10 - wipe function

Yara Rule

```
rule Detect_SwiftSlicer_wiper: SwiftSlicer wiper
{
  meta:
     description = "Detect_SwiftSlicer_wiper"
     author = "@MalGamy12"
     date = "2023-02-1"
       hash =
"1db93ee81050da0ba413543f9fbc388499a466792f9a54ea6f1bbdb712ba9690"

    strings:

       $wipe_fun = {89 5C 24 ?? 89 74 24 ?? 89 6C 24 ?? 89 54 24 ?? C6 84
24 [4] ?? C7 84 24 [4] [4] C7 84 24 [4] [4] C7 84 24 [4] [4] 8D 84 24 [4]
89 04 24 C7 44 24 [4] ?? C7 44 24 [4] ?? E8 [4] 8B 05 [4] 8B 0D [4] 8B 54
24 ?? 89 04 24 89 4C 24 ?? 89 54 24 ?? E8 [4] 8B 44 24 ?? 89 44 24 ?? 8B
0D [4] 8B 15 [4] 89 54 24 ?? 89 0C 24 E8 [4] 8B 05 [4] 89 C1 8B 44 24 ??
89 C3 F7 E1 8B 6C 24 ?? 89 EE 29 C5 8B 7C 24 ?? 0F AF CF 01 D1 8B 15 [4]
0F AF D3 01 D1 29 C6 8B 44 24 ?? 19 C8 F2 0F 10 44 24 ?? F2 0F 11 44 24 ??
89 44 24 ?? 89 2C 24 E8 [4] F2 0F 10 44 24 ?? F2 0F 11 04 24 E8 [4] F2 0F
10 44 24 ?? F2 0F 11 04 24 E8 [4] 8B 44 24 ?? 89 44 24 ?? 8B 4C 24 ?? 89
4C 24 ?? 8B 54 24 ?? 89 54 24 ?? 8B 5C 24 ?? 89 1C 24 E8 [4] 8B 44 24 ??
8B 4C 24 ?? 8B 54 24 ?? 85 D2}

       $s0 = "main.wipe" ascii
       $s1 = "main.walkFunc" ascii
       $s2 = "main.GetLogicalDriveStrings" ascii
       $s3 = "main.lookupPrivilegeName" ascii
       $s4 = "main.LoadDLL" ascii
       $s5 = "main.GetSystemDirectory" ascii



    condition:
        uint16(0) == 0x5A4D and all of them
```

Mitre ATT&CK Tactics and Techniques

Tactic	**Technique ID**	**Technique Name**
Execution	T1047	Windows Management Instrumentation
Defense Evasion	T1070.004 T1070.006	File Deletion Timestomp
Discovery	T10832 T1083 T1518.001	System Information Discovery File and Directory Discovery Security Software Discovery
Impact	T1485	Data Destruction

IOCs

- 1db93ee81050da0ba413543f9fbc388499a466792f9a54ea6f1bbdb712ba9690

References

- https://www.welivesecurity.com/2023/01/27/swiftslicer-new-destructive-wiper-malware-ukraine/

ThreatMon

45305 Catalina cs St 150, Sterling VA 20166