The Spies Who Loved You: Infected USB Drives to Steal Secrets

In the first half of 2023, Mandiant Managed Defense has observed a threefold increase in the number of attacks using infected USB drives to steal secrets. Mandiant tracked all of the cases and found that the majority of the incidents could be attributed to several active USB-based operation campaigns affecting both the public and private sectors globally.

Previously, we covered one of the campaigns that leverages USB flash drives as an initial infection vector and concentrates on the Philippines. In this blog post, we are covering two additional USB-based cyber espionage campaigns that have been observed by Managed Defense:

- **SOGU Malware Infection via USB Flash Drives Across Industries and Geographies**

  This is the most prevalent USB-based cyber espionage attack using USB flash drives and one of the most aggressive cyber espionage campaigns targeting both public and private sector organizations globally across industry verticals. It uses USB flash drives to load the SOGU malware to steal sensitive information from a host.

  Mandiant attributes this campaign to TEMP.Hex, a China-linked cyber espionage actor. TEMP.Hex likely conducted these attacks to collect information in support of Chinese national security and economic interests. These operations pose a risk to a variety of industries, including construction and engineering, business services, government, health, transportation, and retail in Europe, Asia, and the United States.

- **SNOWYDRIVE Malware Infection via USB Flash Drives, Targets Oil and Gas Organizations in Asia**

  This campaign uses USB flash drives to deliver the SNOWYDRIVE malware. Once SNOWYDRIVE is loaded, it creates a backdoor on the host system, giving attackers the ability to remotely issue system commands. It also spreads to other USB flash drives and propagates throughout the network.

  Mandiant attributes this campaign to UNC4698, a threat actor that has targeted oil and gas organizations in Asia. Once the actor has gained access to the system, they execute arbitrary payloads using the Windows Command Prompt, use removable media devices, create local staging directories, and modify the Windows registry.

SOGU Malware Infection via USB Flash Drives Across Industries and Geographies

Managed Defense first observed this campaign while hunting for suspicious file write events in common directories that threat actors use for their malware, tools, or utilities.

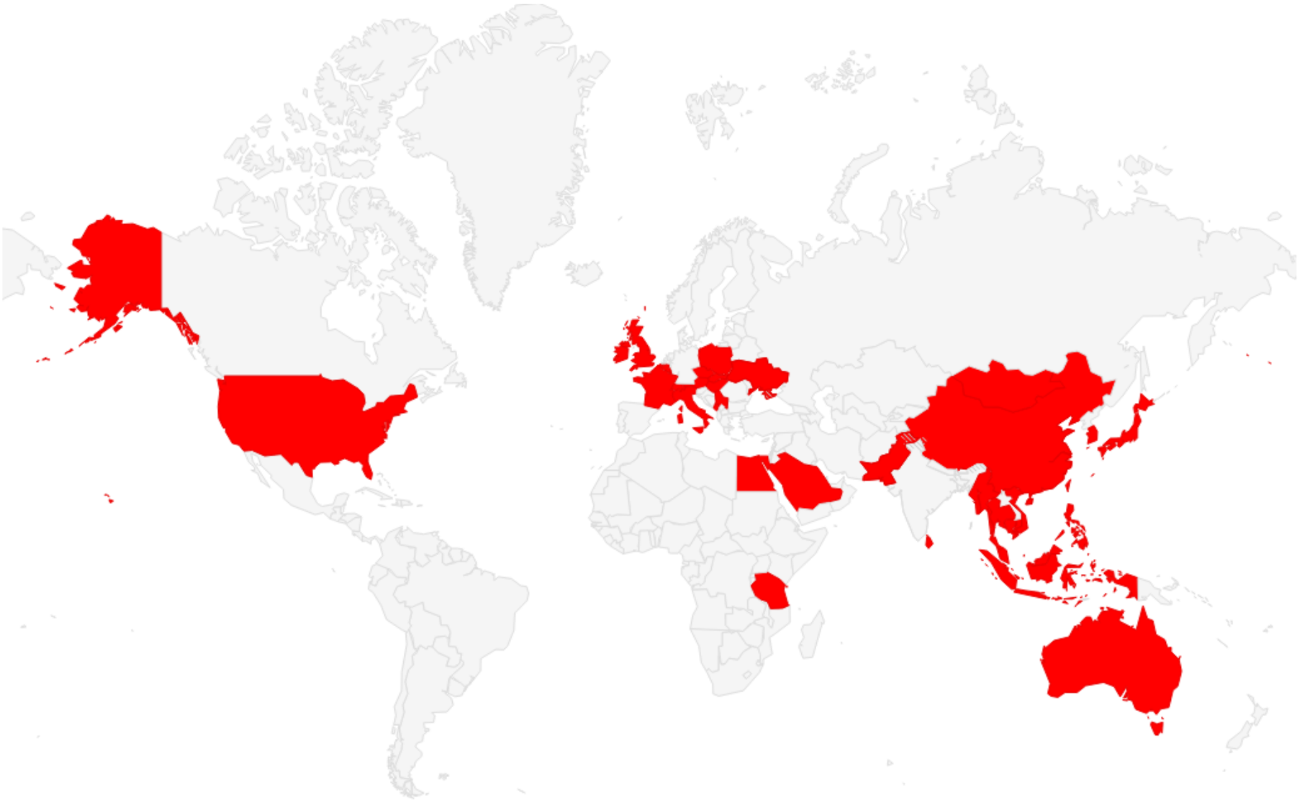


Figure 1: Geographic distribution of TEMP.HEX victims

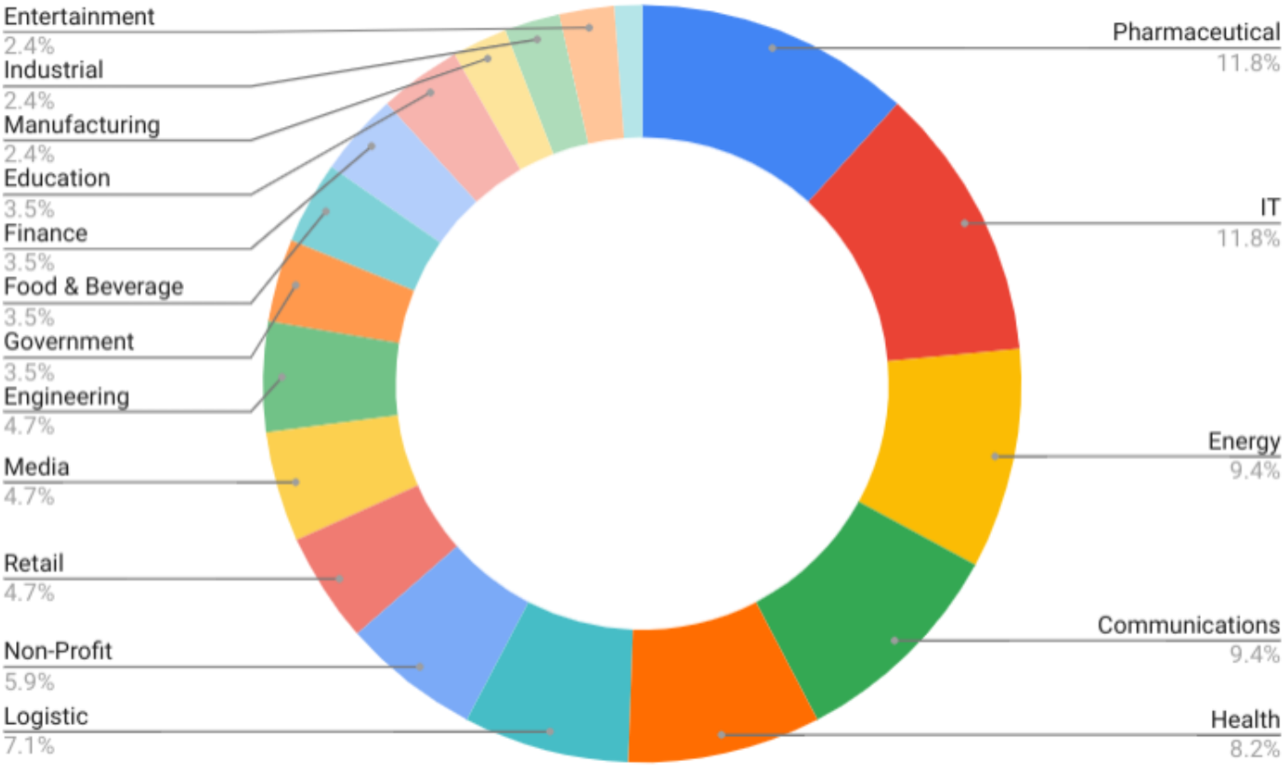


Figure 2: Managed Defense investigation breakdown by industry

The Initial Infection

An infected USB flash drive is the initial infection vector. The flash drive contains multiple malicious software that is designed to load a malicious payload in memory through DLL hijacking.

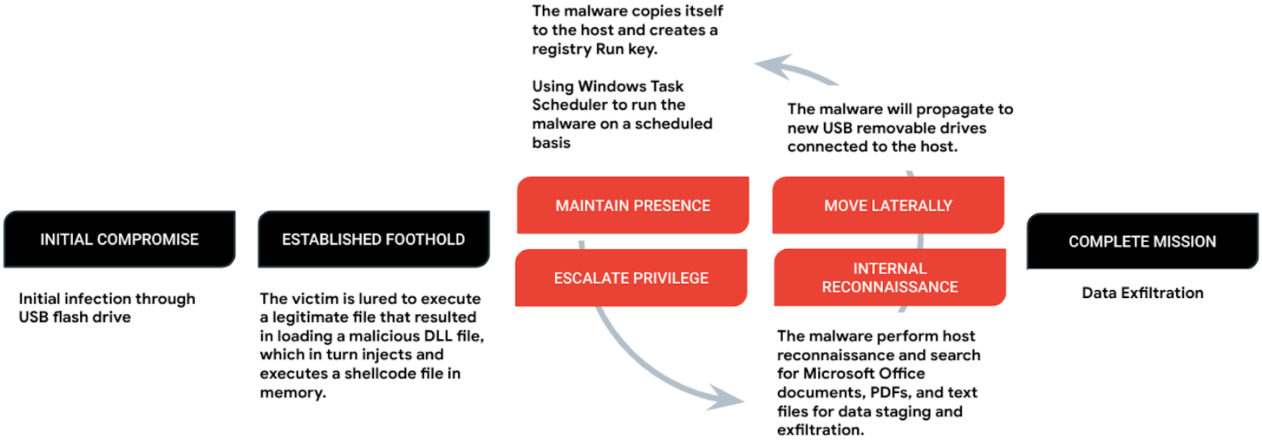

The malware copies itself to the host and creates a registry Run key.
Using Windows Task Scheduler to run the malware on a scheduled basis
The malware will propagate to new USB removable drives connected to the host.
INITIAL COMPROMISE
ESTABLISHED FOOTHOLD
MAINTAIN PRESENCE
ESCALATE PRIVILEGE
MOVE LATERALLY
INTERNAL RECONNAISSANCE
COMPLETE MISSION
Initial infection through USB flash drive
The victim is lured to execute a legitimate file that resulted in loading a malicious DLL file, which in turn injects and executes a shellcode file in memory.
The malware perform host reconnaissance and search for Microsoft Office documents, PDFs, and text files for data staging and exfiltration.
Data Exfiltration

Figure 3: Attacker lifecycle

Established Foothold

The entire infection chain usually consists of three files: a legitimate executable, a malicious DLL loader, and an encrypted payload. Table 1 shows the commonly observed malware file paths and file names observed throughout the campaign.

Table 1: The legitimate executables commonly observed were security software, such as Avast, Smadav, or Symantec. The working directory is usually either in RECYCLER.BIN or RECYCELRS.BIN

File Path	Benign Executable	Malicious DLL Loader	Encrypted Payload
<drive>:\RECYCLER.BIN\1\	CEFHelper.exe	wsc.dll	avastauth.dat
<drive>:\RECYCLERS.BIN\	Smadav.exe	smadhook32c.dll	smadavupdate.dat
<drive>:\RECYCLERS.BIN\	AdobeUpdate.exe	hex.dll	adobeupdate.dat

When the legitimate executable is run, it will side-load a malicious DLL file, which we tracked as **KORPLUG**. The **KORPLUG** malware will then load a decrypted shellcode, commonly observed in the form of a .dat file, and execute it in memory. The shellcode is commonly observed as a backdoor that Mandiant tracked as **SOGU**, a backdoor written in C.

Reconnaissance and Data Staging

The infection continues by dropping a batch file onto the RECYCLE.BIN file path. The batch file runs host reconnaissance commands and outputs the results to a file named **c3IzLmluZm8**. When decoded from Base64, the file name **c3IzLmluZm8** is “sys.info”. The following commands to gather specific system metadata are executed:

- tasklist /v
- arp -a
- netstat -ano
- ipconfig /all
- systeminfo

Subsequently, the malware searches the C drive for files with the following extensions: **.doc**, **.docx**, **.ppt**, **.pptx**, **.xls**, **.xlsx**, and **.pdf**. It encrypts a copy of each file, encodes the original filenames using Base64, and drops the encrypted files in the following directories:

- C:\Users\<user>\AppData\Roaming\Intel\<SOGU CLSID>\<filename in Base64>
- <drive>:\RECYCLER.BIN\<SOGU CLSID>\<filename in Base64>

Maintain Presence

To maintain its persistence on the system, the malware creates a directory that masquerades as a legitimate program and sets the directory's attribute to hidden. It then copies its main components to this directory, with the following commonly used file paths:

- C:\ProgramData\AvastSvcpCP
- C:\ProgramData\AAM UpdatesHtA
- C:\ProgramData\AcroRd32cWP
- C:\ProgramData\Smadav\SmadavNSK

Then, it creates a Run registry key with the same name as the directory created earlier. The Run registry keys are used to run programs automatically when a user logs on. The following are the commonly observed Run registry key entries.

- Value: AvastSvcpCP
- Text: C:\ProgramData\AvastSvcpCP\AvastSvc.exe
- Value: AAM UpdatesHtA
- Text: C:\ProgramData\AAM UpdatesHtA\AAM Updates.exe
- Value: AcroRd32cWP
- Text: C:\ProgramData\AcroRd32cWP\AcroRd32.exe
- Value: SmadavNSK
- Text: C:\ProgramData\Smadav\SmadavNSK\Smadav.exe

In some SOGU variants, an additional scheduled task may be created to run the malware every 10 minutes to maintain persistence.

- SCHTASKS.exe /create /sc minute /mo 10 /tn "Autodesk plugin" /tr """"C:\ProgramData\Smadav\SmadavNSK\Smadav.exe""" 644" /f

Complete Mission

At the last stage of the attack lifecycle, the malware will exfiltrate any data that has been staged. The malware may include HTTP, HTTPS, a custom binary protocol over TCP or UDP, and ICMP to communicate with its command and control server. The malware was also found to support a wide range of commands, including file transfer, file execution, remote desktop, screenshot capture, reverse shell, and keylogging.

The malware can also copy onto new removable drives plugged into an infected system. This allows the malicious payloads to spread to other systems and potentially collect data from air-gapped systems.

Mandiant tracks this event as Campaign 22-054.

SNOWYDRIVE Malware Infection via USB Flash Drives, Targets Oil and Gas Organizations in Asia

Managed Defense first observed this campaign while hunting Windows Explorer process execution with a suspicious folder path (e.g., “F:\”) specified on the command line. This behavior is commonly observed when a user is tricked into executing malware on USB drives. While this type of threat is not uncommon, Mandiant's relentless research and pursuit of every attack led to the discovery of yet another espionage campaign that uses USB drives to spread malware.

The Initial Infection

An infected USB flash drive is the initial infection vector. The victim is lured to click on a malicious file that is masquerading as a legitimate executable. Upon executing the malicious file, it triggers a chain of malicious executions, each designed to perform its specific task throughout the attacker's lifecycle.

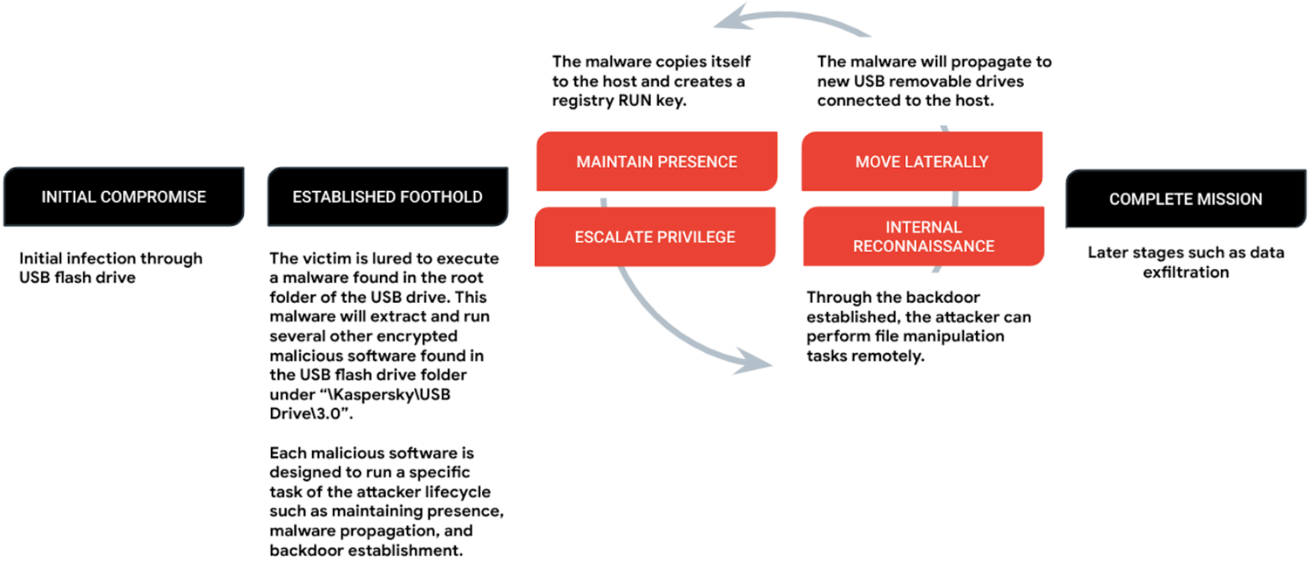

INITIAL COMPROMISE
Initial infection through USB flash drive
ESTABLISHED FOOTHOLD
The victim is lured to execute a malware found in the root folder of the USB drive. This malware will extract and run several other encrypted malicious software found in the USB flash drive folder under “\Kaspersky\USB Drive\3.0”.
Each malicious software is designed to run a specific task of the attacker lifecycle such as maintaining presence, malware propagation, and backdoor establishment.
The malware copies itself to the host and creates a registry RUN key.
MAINTAIN PRESENCE
ESCALATE PRIVILEGE
The malware will propagate to new USB removable drives connected to the host.
MOVE LATERALLY
INTERNAL RECONNAISSANCE
Through the backdoor established, the attacker can perform file manipulation tasks remotely.
COMPLETE MISSION
Later stages such as data exfiltration

Figure 4: Attacker lifecycle

Established Foothold

The infection chain typically starts with an executable that serves as a dropper. The dropper is responsible for writing malicious files to disk and launching them. In one instance, a dropper named **USB Drive.exe** wrote the following encrypted files to **C:\Users\Public\SymantecsThorvices\Data**:

- aweu23jj46jm7dc
- bjca3a0e2sfbs
- asdigasur3ase
- sf33kasliaeae
- sf24acvywsake

The encrypted files contain executables and DLLs that are extracted and written in the directory C:\Users\Public\SymantecsThorvices\Bin.

These files can be broken down into four components, each consisting of a legitimate executable and a malicious DLL that is loaded via DLL search order hijacking. As shown in Figure 5, each component is responsible for a task within the attack lifecycle.

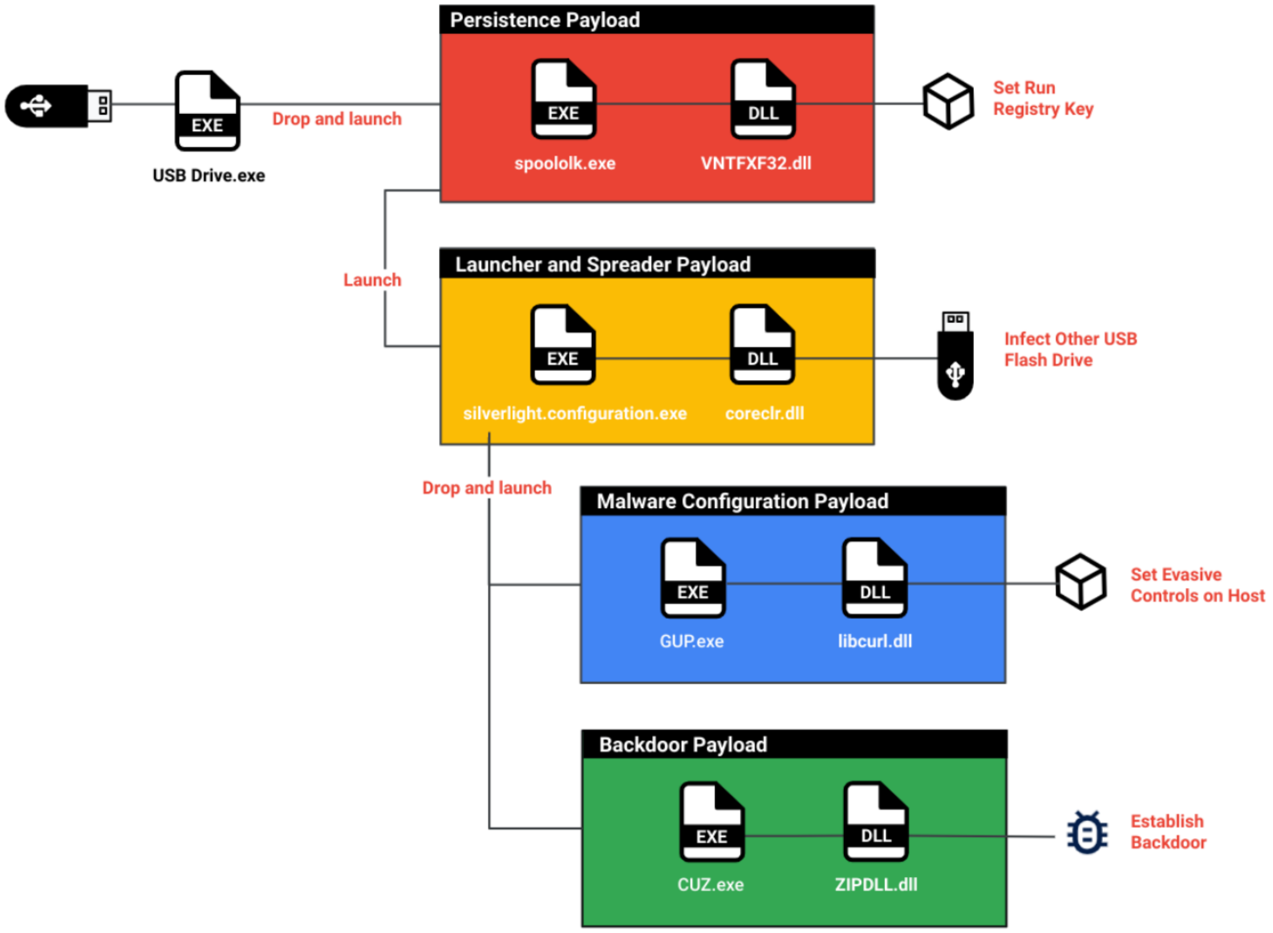


Figure 5: Components and the execution chain of this campaign

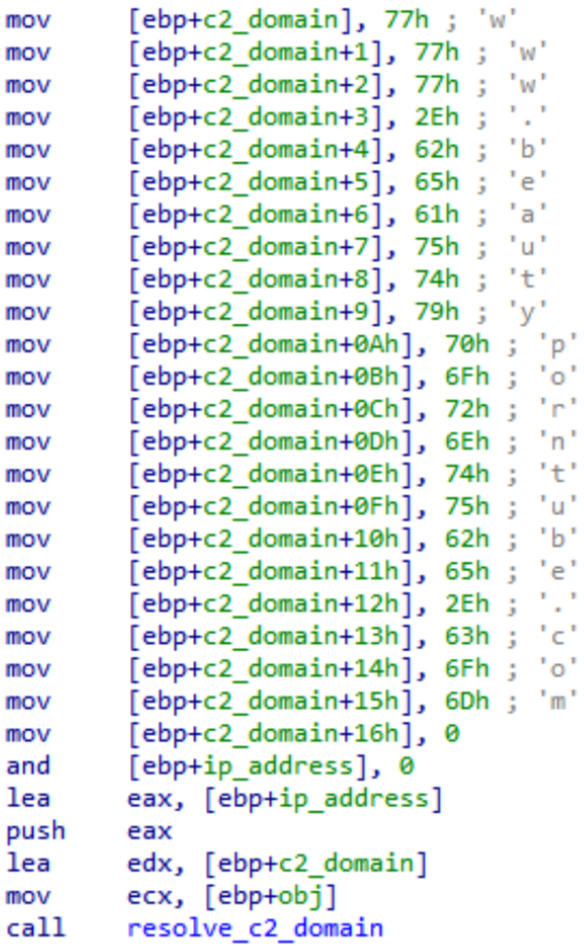
Table 2: Malware components

Filename	Purpose
GUP.exe	Legitimate WinGup for Notepad++
Silverlight.Configuration.exe	Legitimate Microsoft Silverlight
spoololk.exe	Legitimate VentaFax Software
CUZ.exe	Legitimate CAM UnZip Software
VNTFXF32.dll	A malicious DLL loaded by **spoololk.exe** to create registry persistence.
coreclr.dll	A malicious DLL loaded by **Silverlight.Configuration.exe**. This malware will • Drop and execute a shellcode-based backdoor. • Drop and execute a malicious utility that configures the host to evade detection. • Infect other attached USB flash drives.
libcurl.dll	A malicious DLL loaded by **GUP.exe**. It is an evasion utility that sets registry values to show hidden files, hide file extensions, and hide files that are marked "system" and "hidden".
ZIPDLL.dll	**ZIPDLL.dll** is a memory-only dropper that injects a shellcode-based

backdoor named **SNOWYDRIVE** into **CUZ.exe**.

Command and Control

The shellcode-based backdoor named SNOWYDRIVE generates a unique identifier based on the system name, username, and volume serial number. This identifier serves as a unique ID when communicating to its command and control (C2) server. The C2 domain is usually found hard-coded in the shellcode.



```
mov     [ebp+c2_domain], 77h ; 'w'
mov     [ebp+c2_domain+1], 77h ; 'w'
mov     [ebp+c2_domain+2], 77h ; 'w'
mov     [ebp+c2_domain+3], 2Eh ; '.'
mov     [ebp+c2_domain+4], 62h ; 'b'
mov     [ebp+c2_domain+5], 65h ; 'e'
mov     [ebp+c2_domain+6], 61h ; 'a'
mov     [ebp+c2_domain+7], 75h ; 'u'
mov     [ebp+c2_domain+8], 74h ; 't'
mov     [ebp+c2_domain+9], 79h ; 'y'
mov     [ebp+c2_domain+0Ah], 70h ; 'p'
mov     [ebp+c2_domain+0Bh], 6Fh ; 'o'
mov     [ebp+c2_domain+0Ch], 72h ; 'r'
mov     [ebp+c2_domain+0Dh], 6Eh ; 'n'
mov     [ebp+c2_domain+0Eh], 74h ; 't'
mov     [ebp+c2_domain+0Fh], 75h ; 'u'
mov     [ebp+c2_domain+10h], 62h ; 'b'
mov     [ebp+c2_domain+11h], 65h ; 'e'
mov     [ebp+c2_domain+12h], 2Eh ; '.'
mov     [ebp+c2_domain+13h], 63h ; 'c'
mov     [ebp+c2_domain+14h], 6Fh ; 'o'
mov     [ebp+c2_domain+15h], 6Dh ; 'm'
mov     [ebp+c2_domain+16h], 0
and     [ebp+ip_address], 0
lea     eax, [ebp+ip_address]
push    eax
lea     edx, [ebp+c2_domain]
mov     ecx, [ebp+obj]
call    resolve_c2_domain
```

Figure 6: Hard-coded domain observed in a SNOWYDRIVE variant

The backdoor supports the following commands:

Table 3: SNOWYDRIVE supported commands

Command ID	Description
0x2	Sleep
0x3, 0x4	Terminate reverse shell, exit
0x5	Create file
0x6	Write file or delete file
0x7	Initiate file upload
0x8	Continue file upload
0x9	Create cmd.exe reverse shell
0xA	Execute reverse shell command
0xB	Retrieve reverse shell command output
0xC	List logical drives

0xD	Start file/directory search
0xE	Continue file/directory search

Maintain Presence

The registry value HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\ushsguaei1hgba is used for persistence. It stores the path of Silverlight.Configuration.exe.

Lateral Movement

The malware copies itself to removable drives that are plugged into an infected system. It creates the folder “<drive_root>\Kaspersky\Usb Drive\3.0” on the removable drive and copies the encrypted files that contain the malicious components. An executable is extracted from the file “aweu23jj46jm7dc” and written to <drive_root>\<volume_name> .exe, which is responsible for extracting and executing the content of the encrypted files.

Outlook and Implications

Mandiant's investigation and research identified local print shops and hotels as potential hotspots for infection. While some threat actors targeted specific industries or regions, Campaign 22-054 appears to be more opportunistic in nature. This campaign may be part of a long-term collection objective or a later-stage follow-up for subjects of interest to state-sponsored threat actors.

Organizations should prioritize implementing restrictions on access to external devices such as USB drives. If this is not possible, they should at least scan these devices for malicious files or code before connecting them to their internal networks.

YARA Rules

SOGU

SOGU is a backdoor written in C. The network protocol varies between samples and may include HTTP, HTTPS, a custom binary protocol over TCP or UDP, and ICMP. Supported commands include file transfer, file execution, remote desktop, screenshot capture, reverse shell, and keylogging.

```
rule M_Code_SOGU

{

    meta:

        author = "Mandiant"

        description = "Hunting rule for SOGU"

        sha256 = "8088b1b1fabd07798934ed3349edc468062b166d5413e59e78216e69e7ba58ab"

    strings:

        $sb1 = { 8B [2] C7 ?? 01 03 19 20 8B [2] C7 ?? 04 01 10 00 00 8B [2] C7 ?? 08 00
00 00 00 8B [2] C7 ?? 0C 00 00 00 00 0F B7 }
```

```
        $sb2 = { 8B ?? 0C C7 ?? 01 03 19 20 8B ?? 0C C7 ?? 04 00 10 00 00 6A 40 E8 [4]
83 C4 04 8B ?? 0C 89 ?? 08 8B ?? 0C C7 ?? 0C 00 00 00 00 C7 [2] 00 00 00 00 EB 09 8B [2]
83 ?? 01 89 [2] 8B ?? 0C 8B [2] 3? ?? 08 7? ?? 68 FF 00 00 00 E8 [4] 83 C4 04 8B [2] 03
[2] 88 ?? 10 EB D4 }

    condition:

        (uint16(0) == 0x5A4D) and (uint32(uint32(0x3C)) == 0x00004550) and
(uint16(uint32(0x3C)+0x18) == 0x010B) and all of them

}
```

FROZENHILL

FROZENHILL is a launcher written in C++ that is configured to utilize existing files for execution and also infects newly attached storage volumes with additional malware.

```
rule M_Code_FROZENHILL

{

    meta:

        author = "Mandiant"

        description = "Hunting rule for FROZENHILL"

        sha256 = "89558b4190abcdc1a2353eda591901df3bb8856758f366291df85c5345837448"

    strings:

        $str1 = "path_symantec" ascii

        $str2 = "symantec_dir" ascii

        $str3 = "name_svchost" ascii

        $str4 = "run_cmd" ascii

        $str5 = "usb_dll_name" ascii

        $str6 = "name_mutex" ascii

        $str7 = "cmd /c \"%s\" %d" wide

        $str8 = { 8B 85 [4] 83 ?? 01 89 85 [4] 8B 85 [4] 3B 45 0C 74 ?? 8B 45 ?? 03 85
[4] 0F B6 08 33 8D [4] 81 E1 [4] 8B 95 [4] C1 EA ?? 33 94 8D [4] 89 95 [4] EB }

    condition:

        uint16(0) == 0x5A4D and uint32(uint32(0x3C)) == 0x00004550 and all of them

}
```

ZIPZAG

ZIPZAG is an in-memory dropper written in C++ that is configured to overwrite portions of the loading process with shellcode and transfer execution back to the process for execution.

```
rule M_Code_ZIPZAG
```

```
{

    meta:

        author = "Mandiant"

        description = "Hunting rule for ZIPZAG"

        sha256 = "8a968a91c78916a0bb32955cbedc71a79b06a21789cab8b05a037c8f2105e0aa"

    strings:

        $str1 = { C6 45 ?? 55 C6 45 ?? 8B C6 45 ?? EC C6 45 ?? 81 C6 45 ?? EC C6 45 ??
08 C6 45 ?? 01 C6 45 ?? 00 C6 45 ?? 00 C6 45 ?? C7 C6 45 ?? 45 C6 45 ?? FC C6 45 ?? 78
C6 45 ?? 56 C6 45 ?? 34 C6 45 ?? 12 C6 45 ?? 68 C6 45 ?? 04 C6 45 ?? 01 C6 45 ?? 00 C6
45 ?? 00 C6 45 ?? 8D C6 45 ?? 85 C6 45 ?? F8 C6 45 ?? FE C6 45 ?? FF C6 45 ?? FF C6 45
?? 50 C6 45 ?? FF C6 45 ?? 75 C6 45 ?? FC C6 45 ?? B8 C6 45 ?? 79 C6 45 ?? 56 C6 45 ??
34 C6 45 ?? 12 C6 45 ?? FF C6 45 ?? D0 C6 45 ?? FF C6 45 ?? 75 C6 45 ?? FC C6 45 ?? B8
C6 45 ?? 7A C6 45 ?? 56 C6 45 ?? 34 C6 45 ?? 12 C6 45 ?? FF C6 45 ?? D0 C6 45 ?? 8D C6
45 ?? 85 C6 45 ?? F8 C6 45 ?? FE C6 45 ?? FF C6 45 ?? FF C6 45 ?? 50 C6 45 ?? B8 C6 45
?? 7B C6 45 ?? 56 C6 45 ?? 34 C6 45 ?? 12 C6 45 ?? FF C6 45 ?? D0 C6 45 ?? C9 C6 45 ??
C3 }

        $str2 = "shellcode_size" ascii

    condition:

        uint16(0) == 0x5A4D and uint32(uint32(0x3C)) == 0x00004550 and all of them

}
```

SNOWYDRIVE

SNOWYDRIVE is a shellcode-based backdoor that communicates via a custom binary protocol over TCP. Supported commands include reverse shell creation, file transfer, file deletion, and disk enumeration.

```
rule M_Code_SNOWYDRIVE

{

    meta:

        author = "Mandiant"

        description = "Hunting rule for SNOWYDRIVE"

        sha256 = "964c380bc6ffe313e548336c9dfaabbd01a5519e8635adde42eedb7e1187c0b3"

    strings:

        $str1 = { C6 45 ?? 6B C6 45 ?? 65 C6 45 ?? 72 C6 45 ?? 6E C6 45 ?? 65 C6 45 ??
6C C6 45 ?? 33 C6 45 ?? 32 C6 45 ?? 2E C6 45 ?? 64 C6 45 ?? 6C C6 45 ?? 6C }

        $str2 = { C6 45 ?? 47 C6 45 ?? 65 C6 45 ?? 74 C6 45 ?? 50 C6 45 ?? 72 C6 45 ??
6F C6 45 ?? 63 C6 45 ?? 41 C6 45 ?? 64 C6 45 ?? 64 C6 45 ?? 72 C6 45 ?? 65 C6 45 ?? 73
C6 45 ?? 73 }

        $str3 = { C6 85 ?? FD FF FF 4C C6 85 ?? FD FF FF 6F C6 85 ?? FD FF FF 61 C6 85
?? FD FF FF 64 C6 85 ?? FD FF FF 4C C6 85 ?? FD FF FF 69 C6 85 ?? FD FF FF 62 C6 85 ??
FD FF FF 72 C6 85 ?? FD FF FF 61 C6 85 ?? FD FF FF 72 C6 85 ?? FD FF FF 79 C6 85 ?? FD
FF FF 41 }
```

```
        $str4 = { C6 85 ?? FC FF FF 57 C6 85 ?? FC FF FF 61 C6 85 ?? FC FF FF 69 C6 85
?? FC FF FF 74 C6 85 ?? FC FF FF 46 C6 85 ?? FC FF FF 6F C6 85 ?? FC FF FF 72 C6 85 ??
FC FF FF 53 C6 85 ?? FD FF FF 69 C6 85 ?? FD FF FF 6E C6 85 ?? FD FF FF 67 C6 85 ?? FD
FF FF 6C C6 85 ?? FD FF FF 65 C6 85 ?? FD FF FF 4F C6 85 ?? FD FF FF 62 C6 85 ?? FD FF
FF 6A C6 85 ?? FD FF FF 65 C6 85 ?? FD FF FF 63 }

    condition:

        uint16(0) != 0x5A4D and uint32(0) != 0x464c457f and uint32(0) != 0xBEBAFECA and
uint32(0) != 0xFEEDFACE and uint32(0) != 0xFEEDFACF and uint32(0) != 0xCEFAEDFE and all
of them

}
```

YARA-L Hunting Rules

The YARA-L syntax is derived from the YARA language developed by VirusTotal. The language works in conjunction with the Chronicle Detection Engine and enables you to hunt for threats and other events across large volumes of data.

Find out more about Google Chronicles.

```
rule hunting_T1091_User Execution: Malicious File

{

  meta:

    rule_name = "Replication Through Removable Media"

    description = "This rule detects a file write event from a RECYCLER/S named path to
another directory"

author = "Mandiant Managed Defense"

    mitre_technique_name = "User Execution: Malicious File"

    mitre_technique = "T1204"

    mitre_tactic_name = "Execution"

    platform = "Windows"

  events:

       $e.target.process.path = ":\RECYCLER.BIN\" nocase or

       $e.target.process.path = ":\RECYCLERS.BIN\" nocase



    }

    condition:

        $e

}
rule hunting_T1091_Replication_Through_Removable_Media

{
  meta:

    rule_name = "Replication Through Removable Media"

    description = "This rule detects windows explorer process execution with a
suspicious folder path specified on the command line"

author = "Mandiant Managed Defense"

    mitre_technique_name = "Replication Through Removable Media"

    mitre_technique = "T1091"

    mitre_tactic_name = "Lateral Movement,Initial Access"

    platform = "Windows"

  events:

    $e.target.process = "explorer.exe" and

    (

      re.regex($e.principal.process.command_line, = `/explorer.exe?(\")?\s+(\")?[A-BD-
Za-bd-z]:\\/`) nocase and

      re.regex($e.principal.process.full_path, `:\\[^\\]+\.exe
```

```
) nocase

    )

    condition:

        $e

}
```

Indicators of Compromise

Malware Family	File Name	MD5
SOGU	AvastAuth.dat	ebb7749069a9b5bcda98d89f04d889db
SOGU	hex.dll	b061d981d224454ffd8d692cf7ee92b7
SOGU	adobeupdate.dat	38baabddffb1d732a05ffa2c70331e21
SOGU	SmadHook32c.dll	fc55344597d540453326d94eb673e750
SOGU	smadavupdate.dat	028201d92b2b41cb6164430232192062
SOGU	wsc.dll	722b15bbc15845e4e265a1519c800c34
SOGU	SmadavMain.exe	ab5d85079e299ac49fcc9f12516243de

FROZENHILL	coreclr.dll	848feec343111bc11cceb828b5004aad
ZIPZAG	ZIPDLL.dll	e1cea747a64c0d74e24419ab1afe1970

Malware Family	Network IOCs
SNOWYDRIVE	www.beautyporntube[.]com
SOGU	45.142.166[.]112
SOGU	103.56.53[.]46
SOGU	45.251.240[.]55
SOGU	43.254.217[.]165

About Managed Defense Hunting

Cyber security hunting missions are a way to look for security breaches that bypass an organization's security controls. Managed Defense hunting missions based on Mandiant's real-time intelligence mapped to the MITRE ATT&CK framework.

Find out more about Managed Defense.

About Threat Campaigns

Greater visibility into attacker operations: Threat Campaigns provides you with detailed information about active campaigns, including the tactics, techniques, and infrastructure used by attackers. This information can help you identify new threats and vulnerabilities, and prioritize your defensive actions.

Find out more about Threat Campaigns.

Mandiant Security Validation Actions

Mandiant Advantage Security Validation can automate the following process to give you real data on how your security controls are performing against these threats.

The following table is a subset of MSV actions for one of the malware variants. Find out more about Mandiant Security Validation.

VID	Name
A106-036	Protected Theater - TEMP.Hex, SOGU, Execution, Variant #1
A106-037	Protected Theater - TEMP.Hex, SOGU, Execution via Malicious LNK, Variant #1
A106-046	Command and Control - TEMP.Hex, SOGU, Beacon, Variant #1
A106-045	Protected Theater - TEMP.Hex, SOGU, Create Install Directory, Variant #1
A106-049	Host CLI - TEMP.Hex, SOGU, Establish Persistence via Registry Run Key, Variant #1
A106-051	Protected Theater - TEMP.Hex, SOGU, Establish Persistence via Registry Run Key, Variant #1
A106-052	Protected Theater - TEMP.Hex, SOGU, Network Registry Key Change, Variant #1

A106-060	Host CLI - TEMP.Hex, SOGU, Enumeration, Variant #1
S100-257	Malicious Activity Scenario - TEMP.Hex Campaign Spreading SOGU via Infected USB Drives, Variant #1

Acknowledgements

This blog post is dedicated to the analysts in the Managed Defense team for their tireless work to develop new ways in defending our clients around the clock.

Special thanks to Matt Williams for his assistance in analyzing the malware samples and Matthew Hoerger and Lexie Aytes for creating the Mandiant Security Validation (MSV) actions. Martin Co for his inputs and review of this blog post.