LOGPOINT

AgentTesla's Capabilities:

A Review and Detection Strategies

www.logpoint.com

FOREWORD

Over the past few months, at logpoint, we have been tracking the malware known as AgentTesla. Since 2014, AgentTesla has been utilized in various data theft campaigns by threat actors such as SWEED, Aggah, and SILVERTERRIER. Its capabilities to infiltrate systems, maintain persistence, collect and exfiltrate data while evading defense made it a popular choice among threat actors.

Anish Bogati

Logpoint Global Services and Security Research

Anish Bogati is a cybersecurity enthusiast and is working as a security researcher at Logpoint. He is passionate about creating effective detection rules that help organizations detect threats on their networks.

Find Anish on LinkedIn

TABLE OF CONTENTS

****All new detection rules are available as part of Logpoint's latest release**, as well as through the **Logpoint Help Center.** Customized investigation and response playbooks are available to all Logpoint Emerging Threats Protection customers.

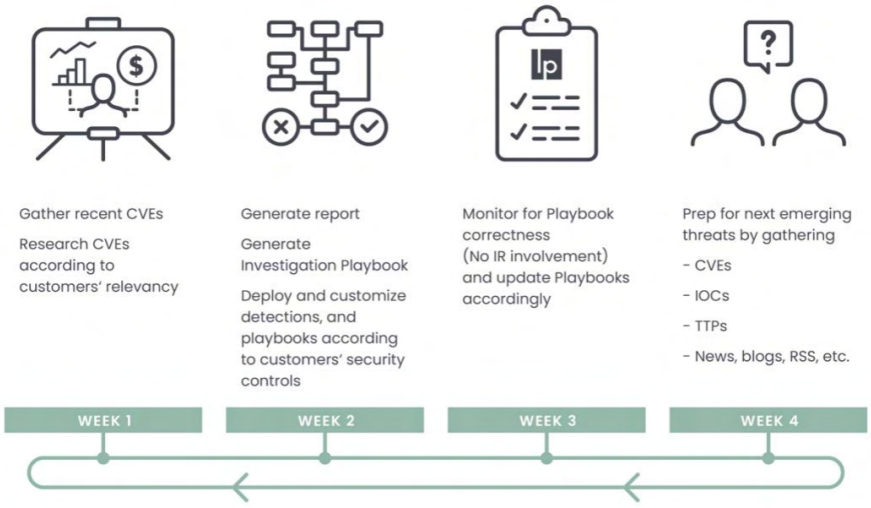
ABOUT LOGPOINT EMERGING THREATS PROTECTION

The cybersecurity threat landscape continuously changes while new risks and threats are discovered all the time. Not every organization has enough resources or the know-how to deal with evolving threats.

Emerging Threats Protection is a managed service provided by a Logpoint team of highly skilled security researchers that are experts in the field of threat intelligence and incident response. Our team keeps you informed on the latest threats and provides custom detection rules and tailor-made playbooks designed to help you investigate and mitigate emerging incidents.

****All new detection rules are available as part of Logpoint's latest release**, as well as through the **Logpoint Help Center.** Customized investigation and response playbooks are available to all Logpoint Emerging Threats Protection customers.

Below is a rundown of the incident, potential threats, and how to detect any potential attacks and proactively defend using Logpoint's SIEM+SOAR capabilities.



CASE STUDY

Menlo Security detected unknown threat actors targeting government sectors where the threat actors leveraged a loader malware known as PureCrypter to download AgentTesla in the victim system for credential harvesting and for the backdoor. The campaigns started via phishing, where victims were lured to open a link that redirected them to discord's CDN. By visiting the site a password-protected zip file was downloaded. When the password-protected file was unzipped a .Net Based malware PureCrypter was extracted. After PureCrypter was executed, it then downloaded AgentTesla into the system. The downloaded malware was obfuscated which helped in evading detection. When AgentTesla was executed it performed process hollowing **[T1055.012]** on `cvtres.exe` process. AgentTesla utilized the XOR operation to encode the strings in the file **[T1027]**. The AgentTesla sample utilized the FTP protocol for data exfiltration **[TA0010].**

We have seen similar cases in every instance where AgentTesla was used as a secondary malware.

From a similar incident response performed by the **Morphisec** team, the attack chain always began with a phishing attack. The phishing was masquerading as an order detail that the victim was receiving from a trusted third party but in this case, the third party was compromised. As user was convinced to execute the malicious attachment which was a Word file. The word file contained a payload to exploit a memory corruption vulnerability in the Equation editor. After the vulnerability was successfully exploited the second-stage payload was downloaded into the system. The dropped file was an image file. Steganography **[T1027.003]** was utilized to hide the malicious payload inside the image. After the payload was extracted it removed the file's ZoneIdentifier which prevents users to know the source of the file (**ZoneIdentifier** provides information

about the source of the file). After execution, it scheduled a task and sets the binary in the Run registry for persistence. Then the malware was detected performing process hollowing on the `regasm.exe` binary. All the list of browsers and their credential file were hardcoded inside the binary with XOR operation.

Decrypted XOR strings containing a list of browsers (Source: **Morphisec**)

METHODOLOGY

For the analysis of AgentTesla, we have used multiple samples to provide an all-encompassing detection and understanding. The samples used were retrieved from **MalwareBazaar** and for reference, samples from online sandbox **any.run** were utilized and are referenced in mentioned section. We performed a dynamic analysis of the samples, by detonating the malware in Microsoft Windows 10 Enterprise system.

We used Process Monitor (Procmon) to observe the processes as they ran. Besides that, to provide threat actor-specific information and the campaigns that the malware was used in we took reference from the above-mentioned case studies and other cyber defense blogs to make sure we didn't leave out any crucial information and be able to provide a comprehensive report as possible.

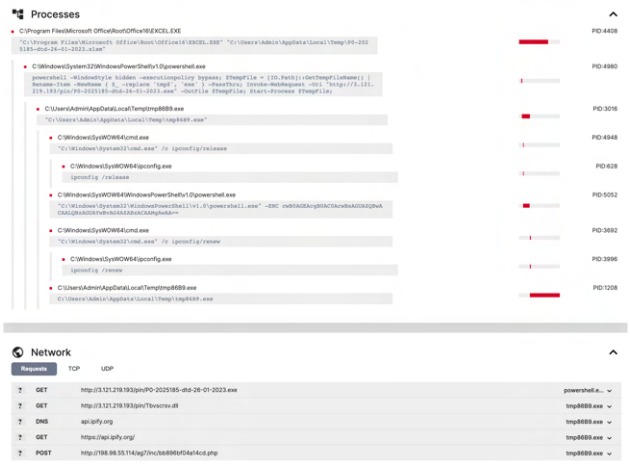
At a high level, below are some of **AgentTesla's** core capabilities:

- **Execution** - Social engineering user to execute a malicious file, scheduling tasks for timely execution, and using various PowerShell and windows commands for execution
- **Persistence** - Modifying AutoRun registry keys and scheduling tasks.
- **Defense Evasion** - Obfuscated payload and software packers for defense evasion.
- **Credential harvesting** - Retrieve credentials from password-containing files.
- **Collection** - Collect sensitive data from browsers, VPNs, and Mail Clients.
- **Exfiltration** - Utilizes various protocols and applications to exfiltrate data.

MALWARE ANALYSIS

Infection Chain

The initial payload delivery and techniques used to drop the main payload in the system are similar among various malware families such as AgentTesla and **Emotet**. Both are primarily delivered through phishing attachments, where unsuspecting victims are lured into executing malicious files disguised as Office documents, Shortcuts, RTFs, zip, and image files. Those initial payloads when executed connect to a remote Command and Control (C2) server to download later stages of the malware. In the case of a shortcut file, execution of the payload spawned a PowerShell process, which then triggered mshta **[T1218.005]** to run a remote HTA application and downloaded the second stage payload. In other cases payloads were downloaded by utilizing various commands such as `Invoke-WebRequest, curl, wget` through the use of PowerShell **[T1059.001]** or command prompt **[T1059.003]**. A similar tactic was used in the case of malicious office document execution **[T1204.002]**. In some cases, multiple payloads were attached in the same file which later dropped the payload as a new file in publicly writable directories such as TEMP. Adversaries were also heavily utilizing OneNote attachments to load AgentTesla in the victim system. For more detail on using OneNote as an initial read the blog mentioned **here**. Below is the image of the process tree of initial payloads such as LNK and excel file process tree.



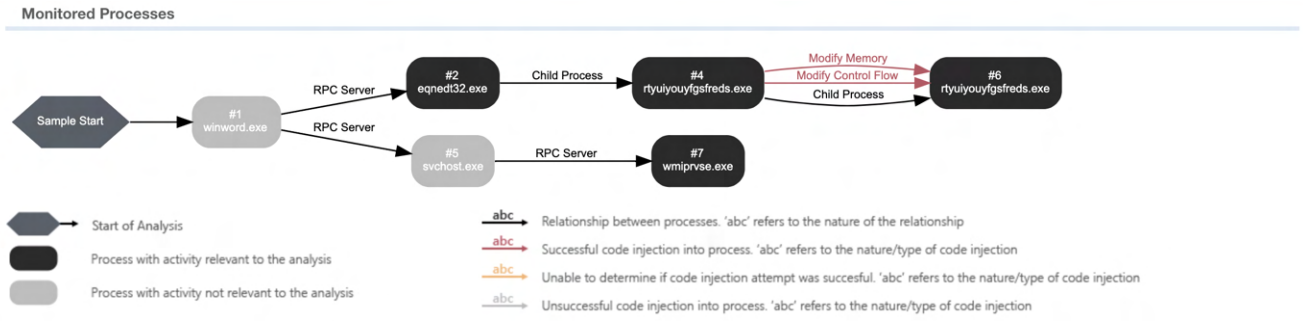
LNK process tree

Excel process tree & network connection

Some samples of AgentTesla were also discovered attempting to exploit the **CVE-2017-0199** and **CVE-2017-11882** vulnerabilities.

CVE-2017-0199 is a security vulnerability that affects Microsoft Office applications and WordPad. The exploitation of the vulnerability allows attackers to execute arbitrary code on a targeted system. The vulnerability is caused by the way that Microsoft Office and WordPad parse specially crafted files. The vulnerability allows attackers to execute a remote code if the user opens a specially crafted file containing a malicious OLE2link object, which can be hidden behind a hyperlink or an embedded image. When the user opens the file, the OLE2link object executes a command to download and run a malicious script from a remote location.

CVE-2017-11882 is a memory corruption vulnerability in Microsoft Office's Equation Editor that could allow remote code execution on vulnerable devices. An attacker could exploit this vulnerability by tricking users into opening a specially crafted file, which could then allow the attacker to run arbitrary code in the context of the current user.



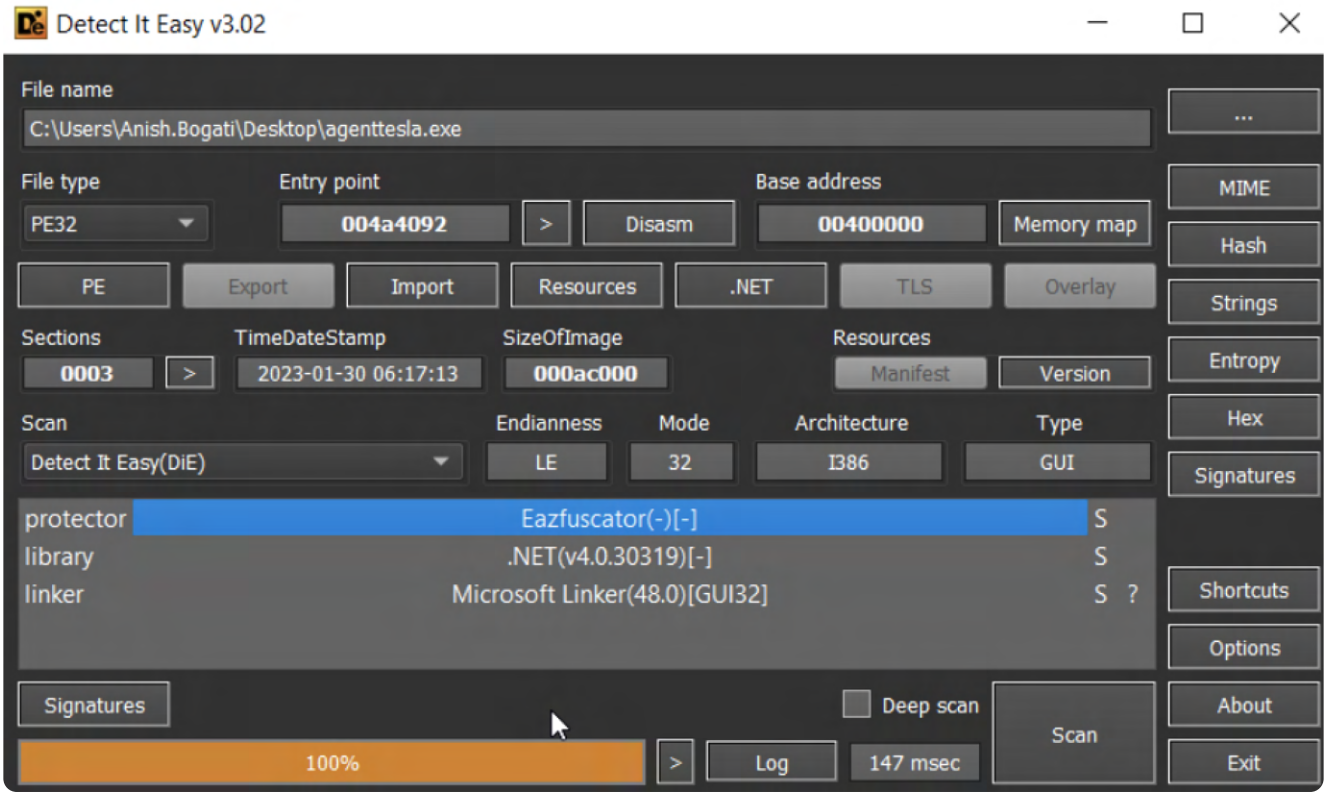
Process graph of CVE-2017-1182 **exploitation**

After the initial access and execution, AgentTesla is dropped into the system. The malware then performs activities to maintain persistence by utilizing various techniques such as scheduling tasks **[T1053]** and placing malware in startup folders or placing it under registry `Run` keys **[T1547.001]**. The malware also performs system information and network discovery. It then proceeds with data collection by retrieving data from browsers, mail, and VPN clients' files, if the services and applications are present in the system. After collecting data from the system, AgentTesla utilizes various protocols and applications such as SMTP, FTP, Telegram, and Discord for data exfiltration.

Behavioral Analysis

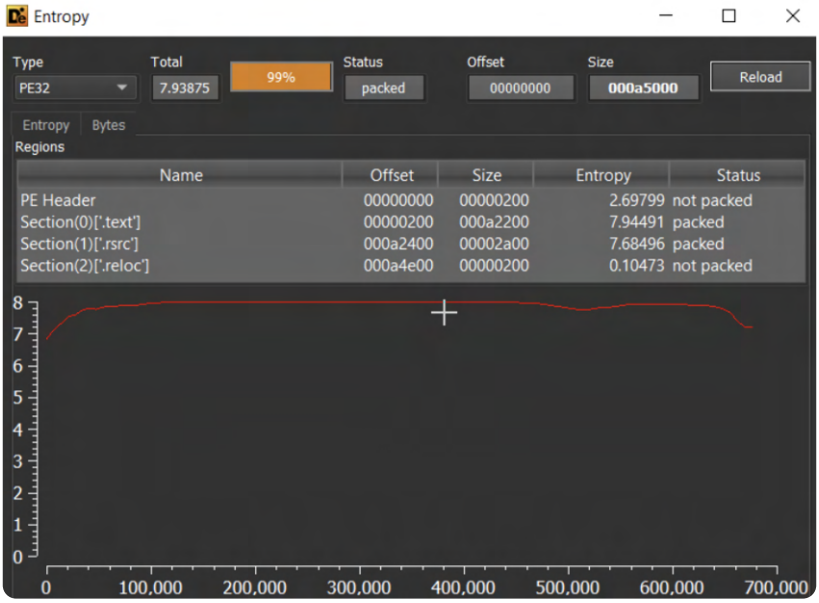
In the following contents, the malware run time behavior that we detected is mentioned, and techniques to detect and respond to such behavior are provided.

Before proceeding to analyze the sample behavior for creating detection rules, the **sample** was loaded into "Detect it Easy" to determine its properties and characteristics, including the presence of obfuscation techniques. The application's results indicated that the sample had been obfuscated using the .Net-based obfuscator Eazfuscator **[T1027.002]**.



Detect It Easy File Scan Result

While calculating the entropy of the malware in “Detect It Easy”, the result was 7+ on the entropy calculation which suggests that the binary is packed. Packing is a common technique used by malware authors to evade detection by security software and to make it more difficult for analysts to determine the functionality of the malware.

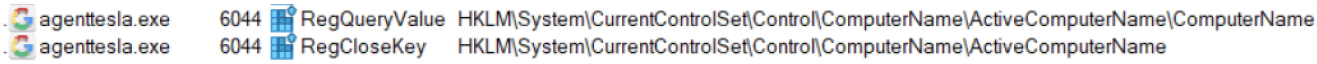


Entropy calculation of binary in Detect It Easy

Above data are provided to showcase the use of obfuscation techniques by the malware developers.

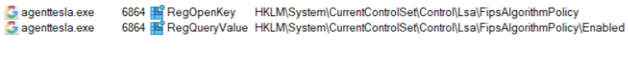
Note: In some SS outputs are filtered out to provide a better view

After executing the sample, we observed that the sample first queried the supported languages and system names by querying registry keys ComputerName and ActiveComputerName under `HKLM\System\CurrentControlSet\Control\ComputerName` **[T1082]**.

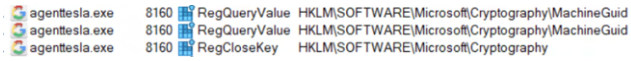


Querying System Name From the Registry

Then we observed that the malware was querying the registry key “HKLM\System\CurrentControlSet\Control\Lsa\FipsAlgorithmPolicy” which can be used to verify the level of encryption being used by the operating system and potentially find a way to bypass it. Then malware also queries the MachineGuid via the registry.

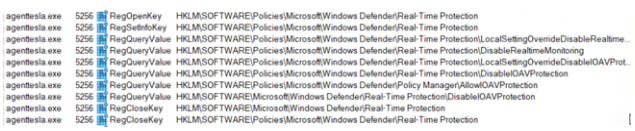


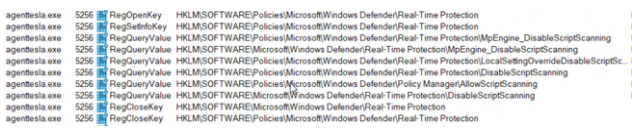
Querying LSA Policy



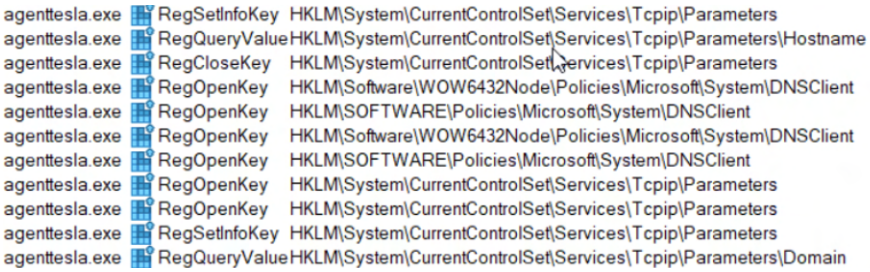
Querying MachineGuid

For the malware analysis, we disabled the antivirus on our sandbox to detonate the sample, so we only observed the malware querying information related to windows defenders such as paths, and policies which we can see in below two images.



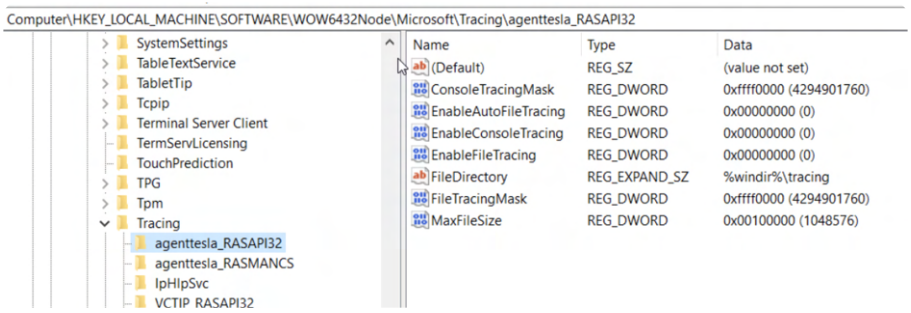


After querying the system-related information the malware proceeds to retrieve network-related information and in the process has been retrieving information related to `Hostname`, `DNSClient`, and `Domain` name by querying registry keys under `HKLM\System\CurrentControlSet\Services\Tcpip\Parameters`. Other network-related information such as previous network connections, proxy servers, network configuration, NameServer, and network adapter name was also collected by the malware.



Querying network-related information

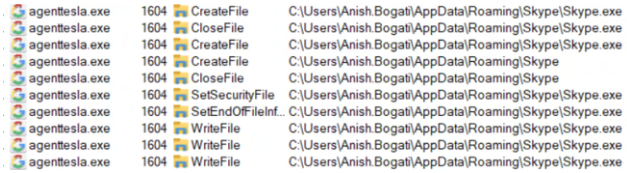
Before moving further the malware created various registry sub-keys under "HKLM\Software\WOW6432Node\Microsoft\Tracing". The malware updated the various values of the created registry keys. Then to evade defense and hide the malware's activities, AgentTesla attempted to disable event tracing for the malware's binary.



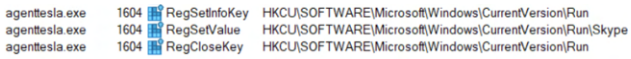
Creation of various Registry keys

Registry value set for above registry keys

After that, AgentTesla created a file with the legitimate binary name i.e. `skype.exe`, and write the payload in the file. After writing the payload in the disk, a new registry key under the `Run` registry was created from the payload name, and the payload's path was then included in the value. The file placed in the Run registry is executed whenever the system is started or a user logins to the system.

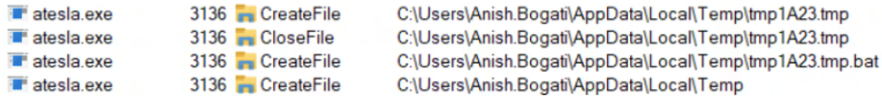


Creation of a new file



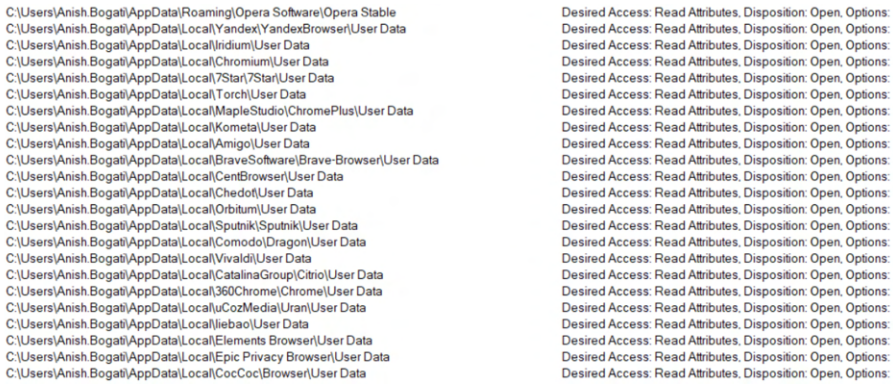
Adding a new file in the Run registry

In another **sample** that we analyzed, we found out that the malware dropped a payload in the temporary file. The binary then masqueraded as svchost.exe which is a legitimate binary name in windows. After creating a file, the file is scheduled by utilizing the schtasks.exe binary **[T1053.005]**.



Payload Dropped in Temp Folder

As it is an information stealer malware, it is used for retrieving various sensitive data from a victim's system. Then after performing the required actions and maintaining persistence the malware then proceeded to collect data and exfiltrate those data. We have found out that the malware has features to retrieve sensitive data such as cookies and credentials that are stored by various browsers **[T1555.003]**. In most browsers, sensitive data such as passwords, and web browsing activities are stored in `User Data` file under the browser's folder, so the malware attempts to read the file in an attempt to recover credentials from it. From our observation, the list of browsers that the malware can steal data from is long, and below is the list of browsers that we observed the malware trying to steal data from.

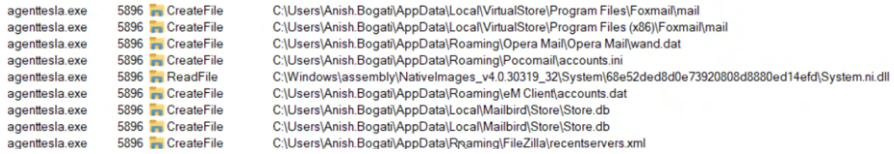


AgentTesla Acessing User Data file

Opera	**Amigo**	**Comodo**	**Flock**	**WaterFox**
Yandex	**Brave**	**Vivaldi**	**Epic Privacy**	**Comodo IceDragon**
Iridium	**CentBrowser**	**Catalina**	**Browser**	**PaleMoon**
Chromium	**QIP Surf**	**Chrome**	**CocCoc**	**IceCat**
7star	**Chedot**	**Uran**	**QIP Surf**	**WaterFox**
Torch	**Orbitum**	**Liebao**	**Coowon**	**QQBrowser**
Kometa	**Sputnik**	**Elements Browser**	**CyberFox**	

It not only holds the capability to steal data from multiple browsers but also has the capability to steal data from multiple email clients. After attempting to steal data from various browsers installed in the system, the malware then moved forward to steal data from email clients installed in the system. The data can be a username, user mail address, attachments, list of email recipients, etc. Below is the list of email clients from whom AgentTesla can steal data:

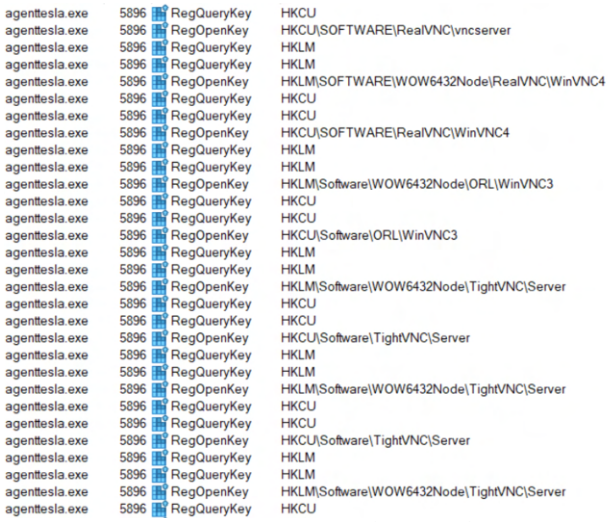
Mozilla ThunderBird	**BlackHawk**	**Postbox**	**Fox Mail**	**Poco Mail**
SeaMonkey	**K-Meleon**	**Flock**	**Opera Mail**	**Mail Bird**



Attempt to read files from mail clients

Note: The CreateFile operation in the above image occurred as a result of accessing the file.

After retrieving data from browsers and mail clients, the malware also verified if any VNC software was installed in the system. Besides VNC, it also tried to retrieve data from VPN applications. During analysis, we also observed the malware trying to retrieve data from NordVPN.



Attempt to retrieve VNC software informaation

Then we also observed that the malware attempted to read data from credential files of windows systems which are present under the following directory "C:\Users\[User Profile]\AppData\Roaming\Microsoft\Credentials" for windows vista and later versions.

Attempt to retrieve credentials from the system

Data in such files are not in plain text so, tools such as **CredentialsFileView** can be utilized to decrypt and retrieve plain text data.

Exfiltration Techniques

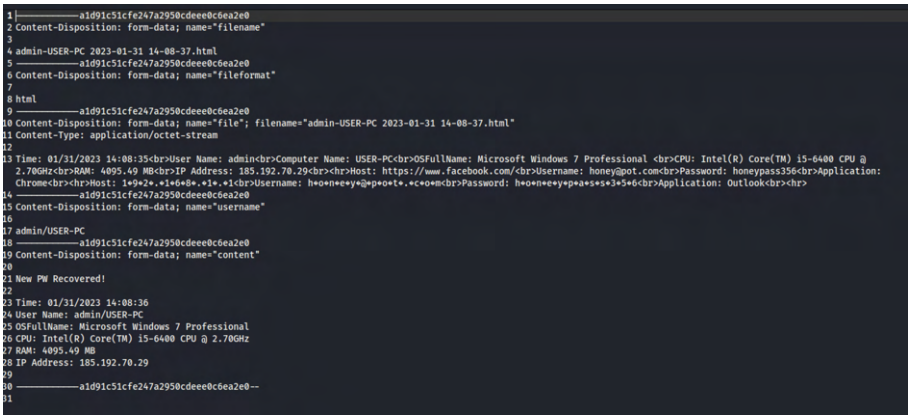
After harvesting credentials from the system, the malware can utilize various techniques to exfiltrate data from the system. From our observation, AgentTesla has utilized protocols such as SMTP, FTP, and HTTP and services such as Discord and Telegram for data exfiltration. The below sub-sections contain brief detail about the mentioned techniques.

Data Exfiltration via Discord WebHooks

While observing a sample in any.run, we found out that the malware tries to exfiltrate data via discord webhooks.

Data Exfiltration via Discord

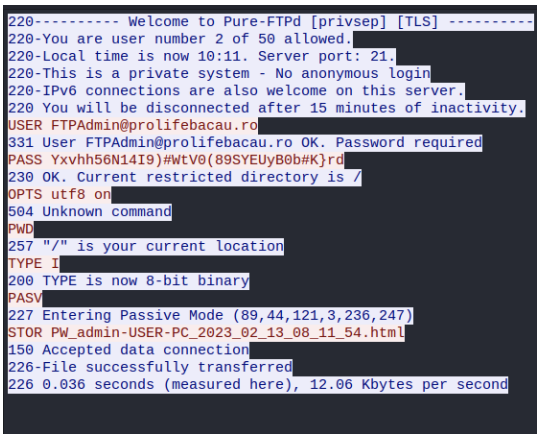
While Exfiltrating data via Discord, we observed that it was exfiltrating sensitive data such as user and system name, CPU and memory information, and collected credentials which we can see in the below images which can be retrieved from the above sample link.



Captured Data being Exfiltrated from Discord

Data Exfiltration via FTP

In another **sample**, we found out that the data collection mechanism was similar to other samples but data were exfiltrated using FTP protocol. Due to the use of unencrypted communication channels used by the malware, we were able to observe the following behavior.

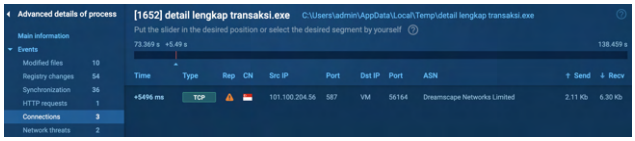


Captured Data being Exfiltrated from FTP

Data Exfiltration via SMTP

In another **sample**, we observed the malware after collecting data was exfiltrating data using SMTP, as the communication channel is encrypted we were not able to retrieve what data was exfiltrated.

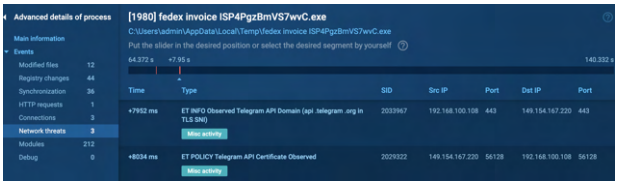
Data transfer to a remote host on an SMTP port



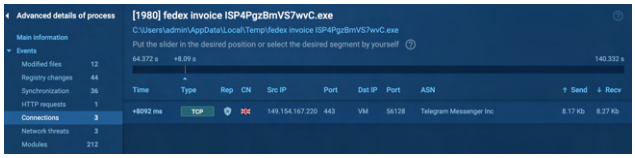
Suricata matched rule

Data Exfiltration via Telegram

In one **sample**, we observed that the malware was able to exfiltrate data through telegram. The malware utilizes telegram API to connect to **api.telegram.org** and exfiltrate data.



Matched Suricata rule



Communication with telegram

DETECTION USING LOGPOINT

With the right tools and proper visibility, it should be fairly simple to detect threats at any stage. Read below on how to use Logpoint SIEM to hunt and SOAR to remediate AgentTesla's artifact.

Log Source Needed

- Windows
- Windows Sysmon
- PowerShell Script Block Logging should be **enabled**.
- Firewall
- DNS
- Process Creation with Command Line Auditing should be **enabled**
- Registry Auditing should be **enabled**
- Object Access Auditing should be **enabled**

While explaining the process, we have mentioned suitable detection rules that we have tested in our lab environments. Below is the collection of alert rules applicable to the procedures carried out by AgentTesla malware. Note, as with many alert rules, some set of rules may need to be baselined for your unique environment and appropriate filters should be added for approved activities from certain users, systems, or applications.

Suspicious Execution of LNK File

Threat actors utilized LNK files to execute their initial payload so this alert is triggered whenever the execution of suspicious LNK files that either spawns PowerShell or command prompt and has high entropy in the command field is detected. For this alert to work "entropy" plugin is required. Entropy is our new plugin that helps to calculate randomness in a field's value.(Include a link to download Entropy Plugin)

```
label="Process" label=Create parent_process="*\explorer.exe"
"process" IN ["*\cmd.exe","*\powershell.exe"]
| process entropy(command) as command_entropy
| search command_entropy > 5
```

Depending upon the environment, analysts can set the entropy value to filter out the false positives. In our environment, legitimate use entropy was below 5 so we used an entropy value greater than 5 to filter out false positives. Analysts can use up to 90 days of data to establish a baseline to reduce false positives.

Microsoft Office product spawning Windows shell

In most cases, office documents such as Word and Excel are utilized to execute payload through macros. so this alert helps to detect suspicious child process creation from Microsoft Office Products. These events indicated malicious office file execution and as a result, suspicious child processes such as regsvr32, rundll32, PowerShell, and the command prompt are executed.

```
label="Process" label=Create parent_process IN
["*\WINWORD.EXE", "*\EXCEL.EXE", "*\POWERPNT.exe", "*\MSPUB.exe", "*\VISIO.exe",
"*\OUTLOOK.EXE","*\MSACCESS.EXE","*EQNEDT32.EXE"]
"process" IN ["*\cmd.exe", "*\powershell.exe", "*\pwsh.exe", "*\wscript.exe",
"*\cscript.exe", "*\sh.exe", "*\bash.exe", "*\scrcons.exe", "*\schtasks.exe",
"*\regsvr32.exe", "*\hh.exe", "*\wmic.exe", "*\mshta.exe", "*\rundll32.exe",
"*\msiexec.exe", "*\forfiles.exe", "*\scriptrunner.exe", "*\mftrace.exe",
"*\AppVLP.exe", "*\svchost.exe","*\msbuild.exe"]
```

Office product spawning suspicious child process

Suspicious OneNote Child Process

As macros are blocked by default, adversaries are utilizing **OneNote to deliver their payload**, so the below query can help to detect events where suspicious processes are spawned via OneNote. The spawned process indicates that the user executed the attached file from OneNote. We have filtered out the spawned process and command line to display only the execution of suspicious commands and processes.

```
label="Process" label=Create parent_process ="*\onenote.exe"
("process" IN ["*\RUNDLL32.exe","*\REGSVR32.exe","*\bitsadmin.exe","*\CertUtil.exe",
"*\InstallUtil.exe","*\schtasks.exe","*\wmic.exe","*\cscript.exe","*\wscript.exe",
"*\CMSTP.EXE","*\Microsoft.Workflow.Compiler.exe","*\RegAsm.exe","*\RegSvcs.exe",
"*\MSHTA.EXE","*\Msxsl.exe","*\IEExec.exe","*\Cmd.Exe","*\PowerShell.EXE","*\HH.exe",
"*\javaw.exe","*\pcalua.exe","*\curl.exe","*\ScriptRunner.exe","*\CertOC.exe",
```

```
"*\WorkFolders.exe","*\odbcconf.exe","*\msiexec.exe","*\msdt.exe"]
OR ("process"="*\explorer.exe" command IN
["*.hta*","*.vb*","*.wsh*","*.js*","*.ps*","*.scr*","*.pif*","*.bat","*.cmd*"])
OR "process" IN ["*\AppData\*","*\Users\Public\*","*\ProgramData\*",
"*\Windows\Tasks\*","*\Windows\Temp\*","*\Windows\System32\Tasks\*"])
```

Suspicious child process spawned from PowerShell

In many samples, we observed that AgentTesla used obfuscated PowerShell scripts and commands to execute its malicious payload. This alert helps to detect the aforementioned events.

```
label="Process" label=Create parent_process IN
["*\powershell.exe*","*\pwsh.exe*","*\powershell_ise.exe*"]
"process" IN ["*\sh.exe","*\bash.exe","*\schtasks.exe","*\certutil.exe",
"*\bitsadmin.exe","*\wscript.exe","*\cscript.exe","*\scrcons.exe","*\regsvr32.exe",
"*\hh.exe","*\wmic.exe","*\mshta.exe","*\rundll32.exe","*\forfiles.exe",
"*\scriptrunner.exe"]
```

Regsvr32 binary execution without DLL in the command line

According to **Microsoft**, "regsvr32.exe is a command-line utility to register and unregister OLE controls, such as DLLs and ActiveX controls in the Windows Registry." Regsvr32 is utilized to execute DLL files, so those events where regsvr32 execution is detected without any DLL files in the command line should be monitored.

```
label="Process" label=Create "process"="*\regsvr32.exe"
-command IN ["*.dll*","*.ocx*","*.cpl*","*.ax*","*.bav*","*.ppl*"]
```

The query helps to detect events where the execution of regsvr32.exe is not executing any expected file types.

Regsvr32 network activity

As adversaries have utilized regsvr32 to execute their malicious payload which then communicates with the C2 server and retrieves further information, so this alert detects network connections initiated by the regsvr32.exe binary.

```
norm_id=WindowsSysmon image="*\regsvr32.exe" event_id IN ["3", "22"]
```

Web request methods via PowerShell

We have observed the use of various PowerShell commands to perform web requests by the malware. This alert can help in detecting such events.

```
norm_id=WinServer event_id=4104 script_block IN ["*Invoke-WebRequest*","*iwr *",
"*wget *","*curl *","*Net.WebClient*","*Start-BitsTransfer*"]
```

Note: As the above query is only limited to the search log generated from the script block module.
The events can also be searched using process creation logs with command line auditing enabled and provided that the commands were not entered interactively.

```
label="Process" label=Create command IN ["*Invoke-WebRequest*","*iwr *",
"*wget *","*curl *","*Net.WebClient*","*Start-BitsTransfer*"]
```

Insecure Policy Set via Set-ExecutionPolicy

This alert is triggered whenever the Set-ExecutionPolicy command is utilized to set insecure policies such as Unrestricted, bypass, or RemoteSigned. Set-ExecutionPolicy is a PowerShell command that can change PowerShell execution policies for Windows systems. The "bypass" option allows the script to be executed without any warning or prompts. The "RemoteSigned" option allows the scripts downloaded from the internet to be executed. The "Unsigned" option will allow scripts that are not digitally signed to be executed.

```
norm_id=WinServer event_id=4104 script_block="*Set-ExecutionPolicy*"
script_block IN ["*Unrestricted*","*bypass*","*RemoteSigned*"]
-script_block IN ["*(New-Object System.Net.WebClient).DownloadString('https://
community.chocolatey.org/install.ps1')*",
"*(New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/
install.ps1)*","*\AppData\Roaming\Code\*"]
```

Note: The above query is only limited to the search log generated from the script block module. The events can be searched using Process creation logs with command line auditing enabled and below is the query:

```
label="Process" label=Create Command="*Set-ExecutionPolicy*"
command IN ["*Unrestricted*","*bypass*","*RemoteSigned*"]
-command IN ["*(New-Object System.Net.WebClient).DownloadString('https://
community.chocolatey.org/install.ps1')*","*(New-Object
System.Net.WebClient).DownloadString('https://chocolatey.org/
install.ps1)*","*\AppData\Roaming\Code\*"]
```

PowerShell Execution Policy Modification Detected

This alert is similar to the above alert as it also detects events where execution policies are set to insecure policies such as Unrestricted, bypass, and RemoteSigned by using registry events.

```
norm_id=WindowsSysmon event_id=13 event_type=setvalue target_object IN
2["*\ShellIds\Microsoft.PowerShell\ExecutionPolicy*",
3"*\Policies\Microsoft\Windows\PowerShell\ExecutionPolicy*"]
4detail IN ["*Bypass*","*RemoteSigned'*","*Unrestricted*"]
5-image IN ["C:\Windows\System32\*","C:\Windows\SysWOW64\*"]
```

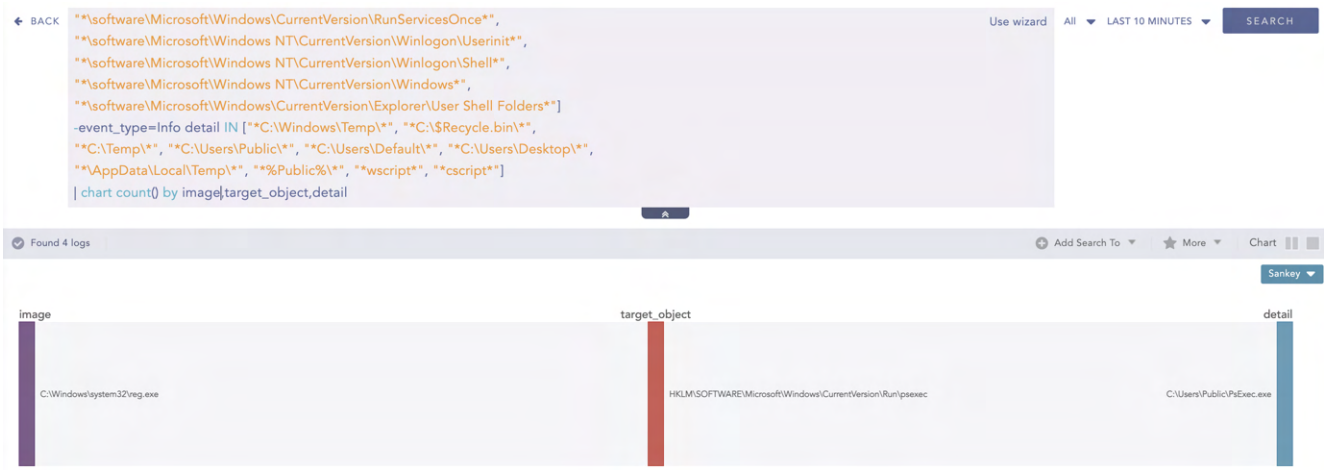
For this alert to trigger registry auditing for related registry keys needs to be **enabled**.

Autorun keys modification detected

To maintain persistence AgentTesla has been found modifying Run registry keys. This alert can help detect events where the binary is either referenced in Run registry keys or set up to be executed at startup.

```
label=Registry label=Set label=Value target_object IN [
"*\software\Microsoft\Windows\CurrentVersion\Run*",
"*\software\Microsoft\Windows\CurrentVersion\RunOnce*",
"*\software\Microsoft\Windows\CurrentVersion\RunOnceEx*",
"*\software\Microsoft\Windows\CurrentVersion\RunServices*",
"*\software\Microsoft\Windows\CurrentVersion\RunServicesOnce*",
"*\software\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit*",
"*\software\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell*",
"*\software\Microsoft\Windows NT\CurrentVersion\Windows*",
"*\software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders*"]
-event_type=Info detail IN ["*C:\Windows\Temp\*", "*C:\$Recycle.bin\*",
"*C:\Temp\*", "*C:\Users\Public\*", "*C:\Users\Default\*", "*C:\Users\Desktop\*",
"*\AppData\Local\Temp\*", "*%Public%\*", "*wscript*", "*cscript*"]
```

For this alert to trigger registry auditing for related registry keys needs to be **enabled**.



Browser Credential Files Accessed

As we have already discussed the credential harvesting techniques of AgentTesla. This alert helps to detect access to browser (Chrome, Edge, Brave & Firefox) files where sensitive data are stored by processes other than the browser itself.

```
label=File label=Access ((path IN ["*\AppData\Local\Google\Chrome\User
Data\Default\Network\Cookies*", "*\Appdata\Local\Chrome\User Data\Default\Login Data*",
"*\AppData\Local\Google\Chrome\User Data\Local State*"] object_name IN
["*\Appdata\Local\Microsoft\Windows\WebCache\WebCacheV01.dat", "*\cookies.sqlite"]) OR
object_name IN ["*\Microsoft\Edge\User Data\Default\Web Data",
"*Firefox*release\logins.json","*firefox*release\key3.db","*firefox*release\key4.db",
"*\BraveSoftware\Brave-Browser\User Data*"]) -"process" IN ["*\firefox.exe", "*\chrome.exe",
"C:\Program Files\*","C:\Program Files (x86)\*","C:\WINDOWS\system32\*","*\MsMpEng.exe",
"*\MpCopyAccelerator.exe","*\thor64.exe","*\thor.exe"] -parent_process IN ["C:
\Windows\System32\msiexec.exe"] -("process"=system parent_process=idle) "access"="ReadData*"
```

Note! In the alert we have only supported the most used browsers, so to monitor for access of credential files of other browsers include the credential file name and exclude the browser process name.

Detecting file tracing disabled events.

AgentTesla disables the file tracing through the registry, so the below query can detect events where file tracing is disabled.

```
norm_id=WindowsSysmon event_id=13 event_type=SetValue
target_object="*\Microsoft\Tracing\*Tracing" detail="DWORD (0x00000000)"
```

For this alert to trigger registry auditing for related registry keys needs to be **enabled**.

Detecting CVE-2017-11882

The query below detects events where the parent process is the equation editor.

```
label="Process" label=Create parent_process="*\EQNEDT32.EXE"
```

As discussed above **CVE-2017-11882** exists due to issues in Equation Editor, so this query attempts to detect the exploitation attempt by detecting the child process created by Equation Editor.

Detecting data transfer to discord

AgentTesla exfiltrates data from discord and in some campaigns have downloaded file through discord API. The below query detects events where a post request is made to the discord “api/webhooks” URL. Legitimate usage can trigger false positives but can be helpful in monitoring data transferred to the discord.

```
request_method=Post (url="*discord.com/api/webhooks*" OR (domain="*discord.com*" url="*api/
webhooks*"))
```

Detecting FTP connection

The below query detects network events where the destination or source port contains either TCP port 20 or 21. This query detects FTP connections which can be further filtered to detect an abnormal connection to a host.

```
(destination_port IN [20,21] OR source_port IN [20,21])
```

Detecting DNS query to telegram API sub-domain

The below query searches for events where DNS activities are performed and filter out DNS query to api.telegram.org only.

```
label=DNS (domain="*telegram.org" OR query="*telegram.org")
```

Suspicious outbound SMTP connection

AgentTesla has utilized SMTP protocols to exfiltrate data. The below query looks for network events where the destination port contains TCP ports 25,587,465,2525. To reduce false positives mail clients such as outlook and thunderbird were excluded. Also, mail binary provided by default on the windows system is also excluded.

```
norm_id=WindowsSysmon event_id=3 destination_port IN [25,587,465,2525] (-"process" IN ["*C:
\Program Files\Microsoft\Exchange Server*", "*\thunderbird.exe", "*\outlook.exe","C:\Program
Files\WindowsApps\microsoft.windowscommunicationsapps_*\HxTsr.exe"
```

Network Connection to Suspicious Server

Sites included in the below query such as pastebin.com, transfer.sh and mega.nz are legitimate websites that provide the ability for users to freely host, share, and store files on the server. As they are legitimate sites, threat actors can utilize those sites to freely host their payload and download it onto victim systems or exfiltrate data to those sites. This alert can help to detect connections to such sites.

```
((norm_id=WindowsSysmon event_id=3 "image" IN ["C:\Windows\*","C:\Users\Public\*"]
destination_host IN
["*dl.dropboxusercontent.com*","*.pastebin.com*","*.githubusercontent.com*",
"*cdn.discordapp.com/attachments*","*mediafire.com*","*mega.nz*","*ddns.net*",
"*.paste.ee*","*.hastebin.com/raw/*","*.ghostbin.co/*", "*ufile.io*","*anonfiles.com*",
"*send.exploit.in*","*transfer.sh*","*privatlab.net*","*privatlab.com*","*sendspace.com*",
"*pastetext.net*","*pastebin.pl*","*paste.ee*"]) OR (device_category IN ["Firewall",
"ProxyServer"] url IN
["*dl.dropboxusercontent.com*","*.pastebin.com*","*.githubusercontent.com*",
"*cdn.discordapp.com/attachments*","*mediafire.com*","*mega.nz*","*ddns.net*",
"*.paste.ee*","*.hastebin.com/raw/*","*.ghostbin.co/*", "*ufile.io*","*anonfiles.com*",
"*send.exploit.in*","*transfer.sh*","*privatlab.net*","*privatlab.com*","*sendspace.com*",
"*pastetext.net*","*pastebin.pl*","*paste.ee*"]))
```

Disable the Windows Task Manager application

In a sample, we have found that the malware tries to disable the task manager utilizing reg.exe binary. We can detect it either by process creation events or by registry events which are shown below respectively.

```
label="Process" label=Create "process"="*\reg.exe" command="* add *"
command="*DisableTaskMgr*"
command="*Software\Microsoft\Windows\CurrentVersion\Policies\System*" command="*/d 1*"
```

To detect events using the below query registry auditing for a particular key should be enabled.

```
norm_id=WindowsSysmon event_id=13 detail="*DWORD (0x00000001)*" event_type="SetValue"
target_object="*Software\Microsoft\Windows\CurrentVersion\Policies\System\DisableTaskMgr*"
```

Disable Command Prompt

In the same sample mentioned above, we have found out that it also utilizes PowerShell to disable the windows command prompt as a result users will not be able to run the Command Prompt application. We can detect such events through process creation logs and registry events, which are shown below respectively.

```
label="Process" label=Create "process"="*\reg.exe" command="* add *" command="*DisableCMD*"
command="*Software\Policies\Microsoft\Windows\System\*" command="*/d 1*"
```

Disable Windows Registry Tool

In another sample, we observed that the malware attempted to disable the windows registry tool as a result users would not be able to modify a registry entry. To detect such events we can utilize registry events.

To detect events using the below query registry auditing for a particular key should be enabled.

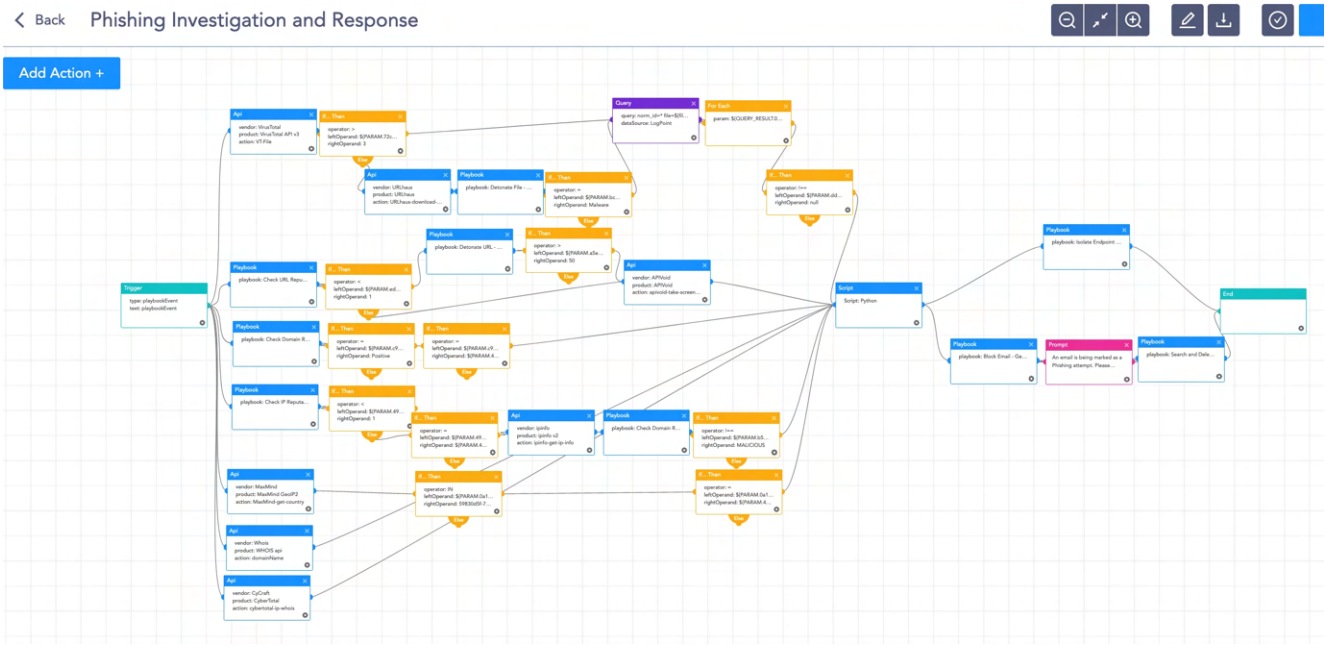
```
norm_id=WindowsSysmon event_id=13 detail="*DWORD (0x00000001)*" event_type="SetValue"
target_object="*\Software\Microsoft\Windows\CurrentVersion\Policies\System*"
```

INVESTIGATION AND RESPONSE USING LOGPOINT

Logpoint SOAR can greatly assist in automating the task of investigation and responding to intrusion and other various attacks. Leveraging SOAR can help to investigate malicious behavior and protect the network by blocking the indicator or in the case of end devices isolating them from the network. To accelerate the TDIR process, Logpoint counts with a native endpoint solution, **called AgentX**, which collects logs and telemetry and uses them to enrich SOAR events for faster malware detection and remediation. There are a tremendous number of useful playbooks that are already available, so only a few playbooks are showcased in this blog.

Phishing Investigation

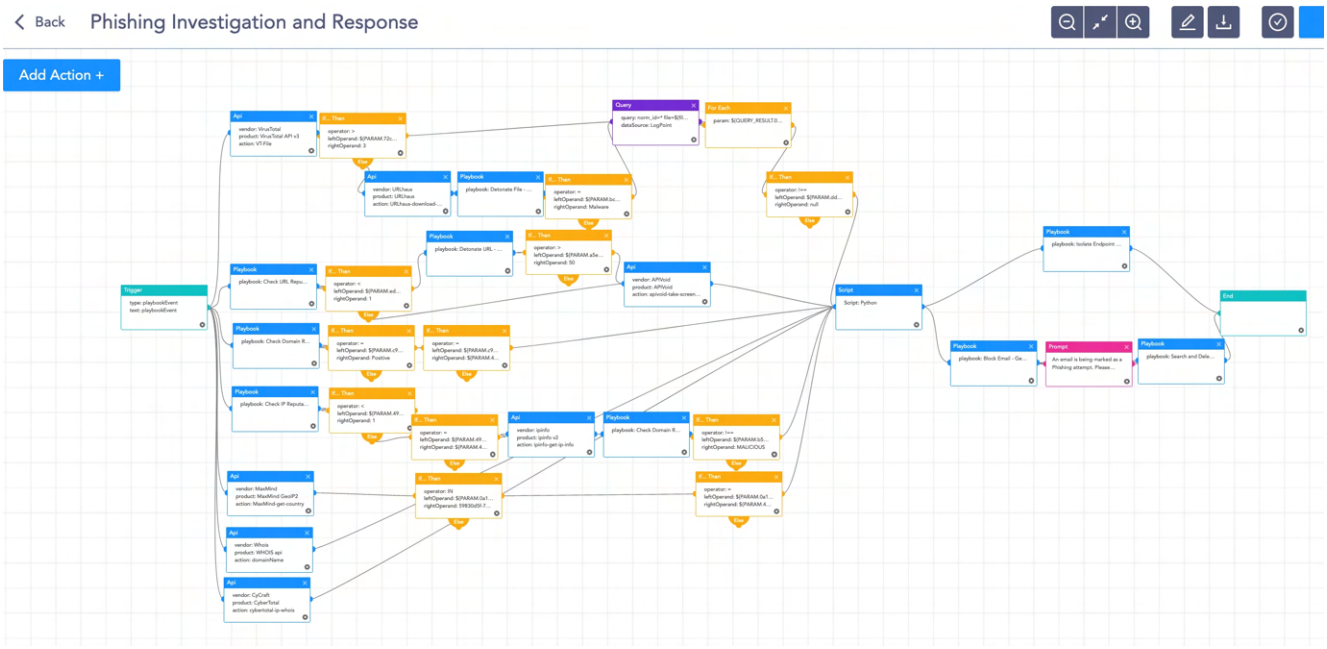
As it all starts with phishing attachments, we do have a phishing investigation and response playbook. This playbook investigates potential phishing attacks and provides automated responses which help to reduce the incident response time.



To detect events using the below query registry auditing for a particular key should be enabled.

OneNote Attachment Investigation and Response

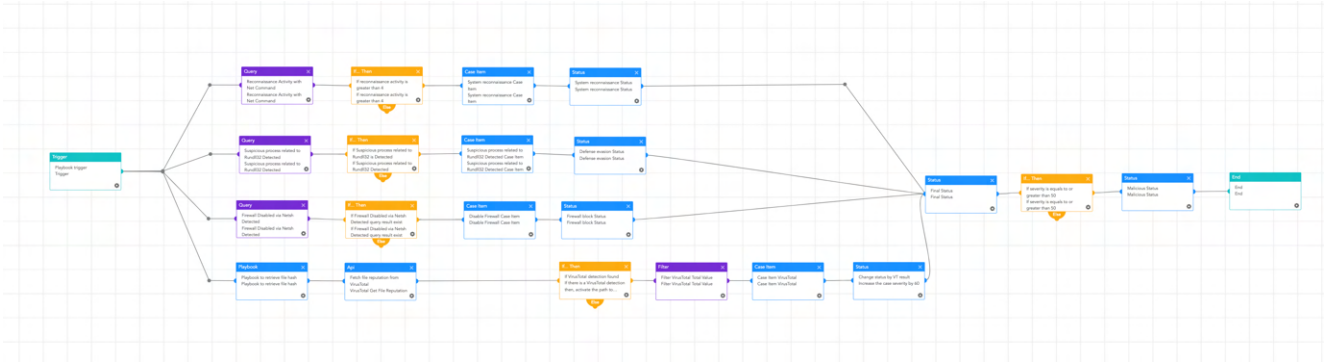
AgentTesla and other malware families have been found dropped into the system via malicious OneNote attachments, so we have already created playbooks that can investigate malicious OneNote files and provide a response to mitigate further incidents from the dropped payloads.



Malicious OneNote Remediation Playbook

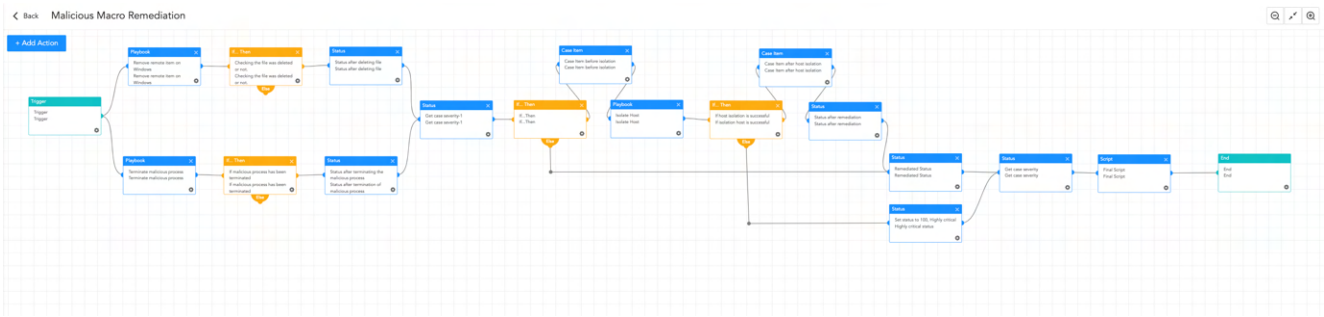
Malicious Macros Detection and Automated Response

As office documents containing malicious macros are the most used payload for dropping AgentTesla and other malware families alike. By utilizing Logpoint SOAR with AgentX we can perform an investigation on the execution of such macros.



Malicious Macro Investigation Playbook

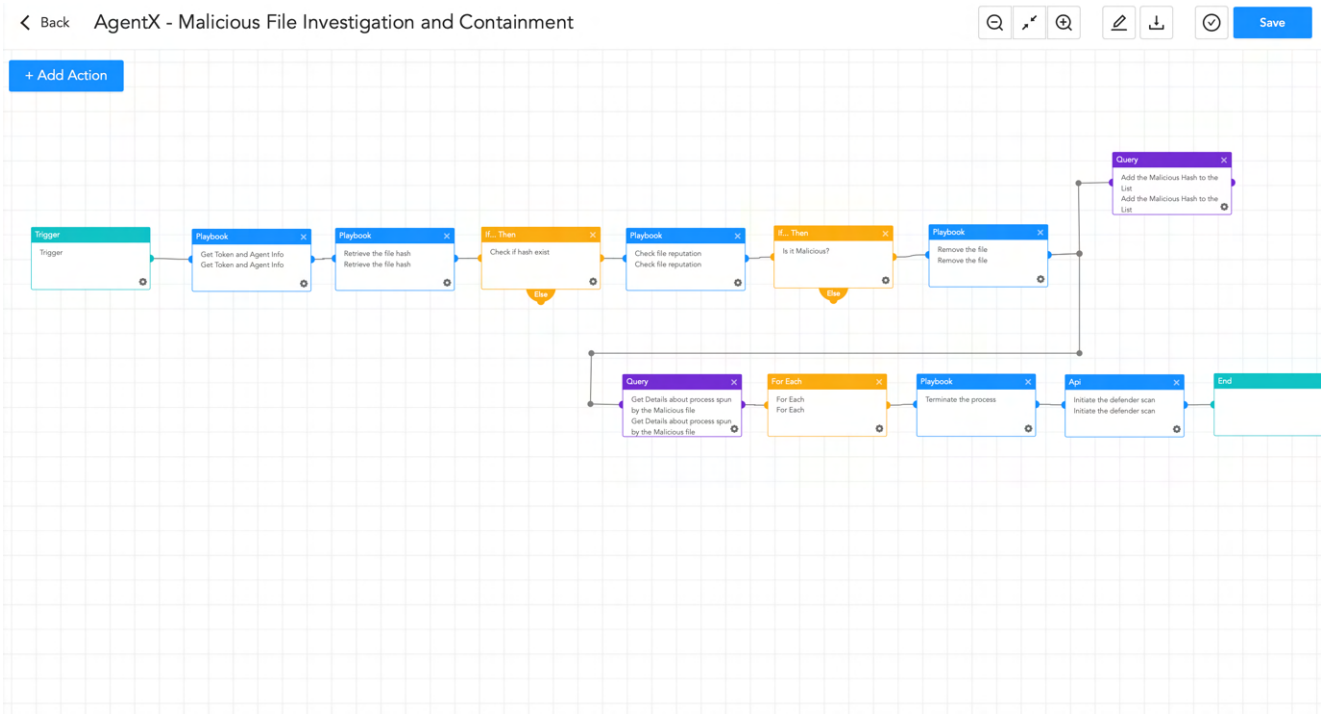
After detecting incidents, the Malicious Macro Remediation playbook can be leveraged to remediate threats.



Malicious Macro Remediation Playbook

Malicious File Investigation and Containment

Besides utilizing a playbook to provide automated investigation and response to malicious macros and OneNote attachments, we have a playbook that can query files in threat intelligence sites to check the legitimacy of the file and provide an automatic response.



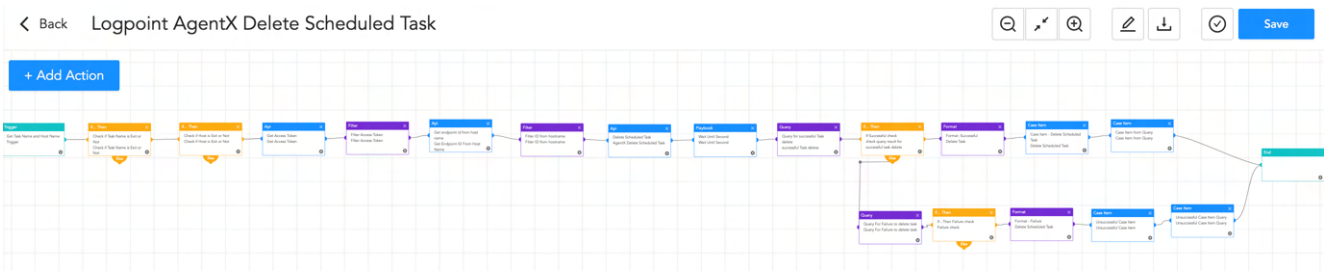
Malicious File Investigation and Containment Playbook

Scheduled task

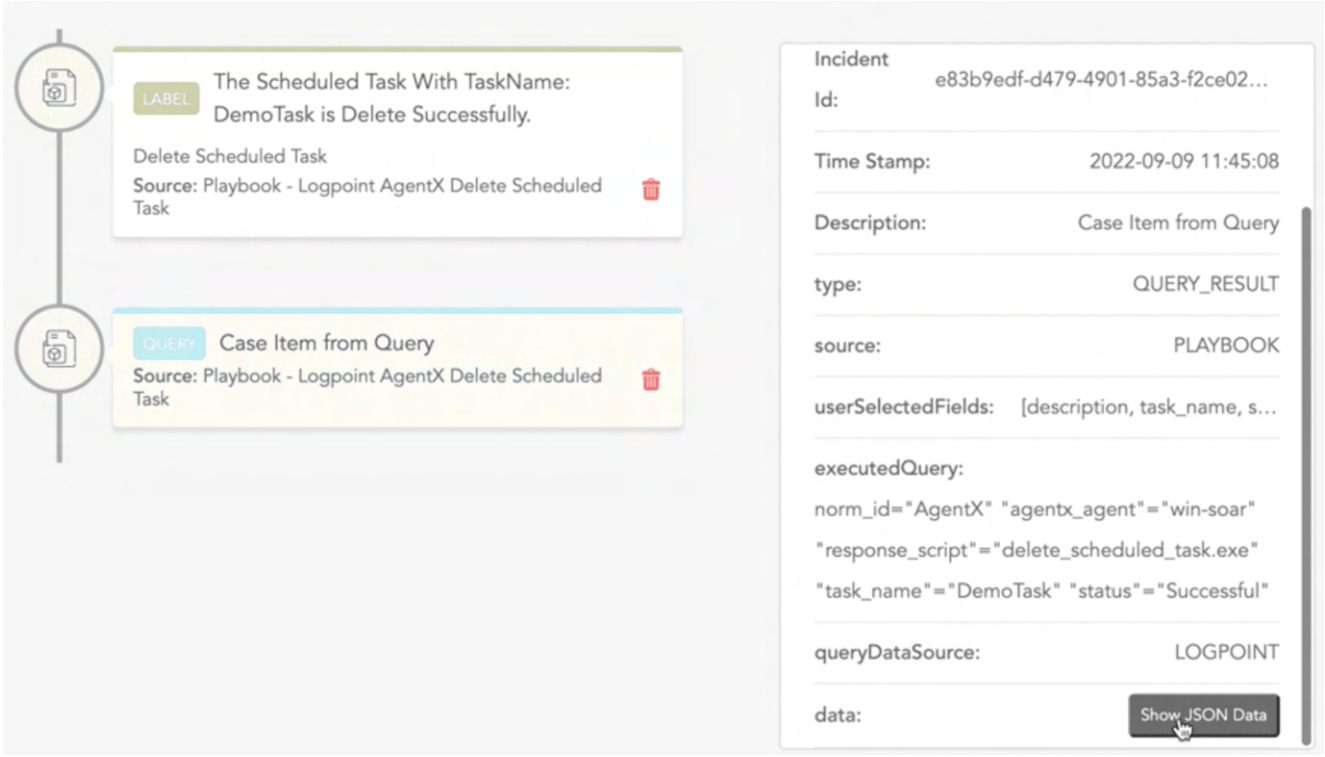
We have a playbook to retrieve a list of scheduled tasks from a host via OSquery. If an analyst is not sure about a scheduled task after reviewing the list of scheduled tasks, then the Logpoint AgentX Disable Scheduled Task playbook can be leveraged to disable it and perform an investigation on the service.

Playbook Name	Tags	Category	Run	Actions
Osquery Get enabled scheduled tasks - windows Subplaybook		Investigate		
Osquery Get enabled scheduled tasks - windows		Investigate		
Logpoint AgentX Disable Scheduled Task		Respond		

After the investigation, if a scheduled task is found malicious then the Logpoint AgentX Delete Scheduled Task response playbook can be utilized to delete a suspicious scheduled task on a windows host.



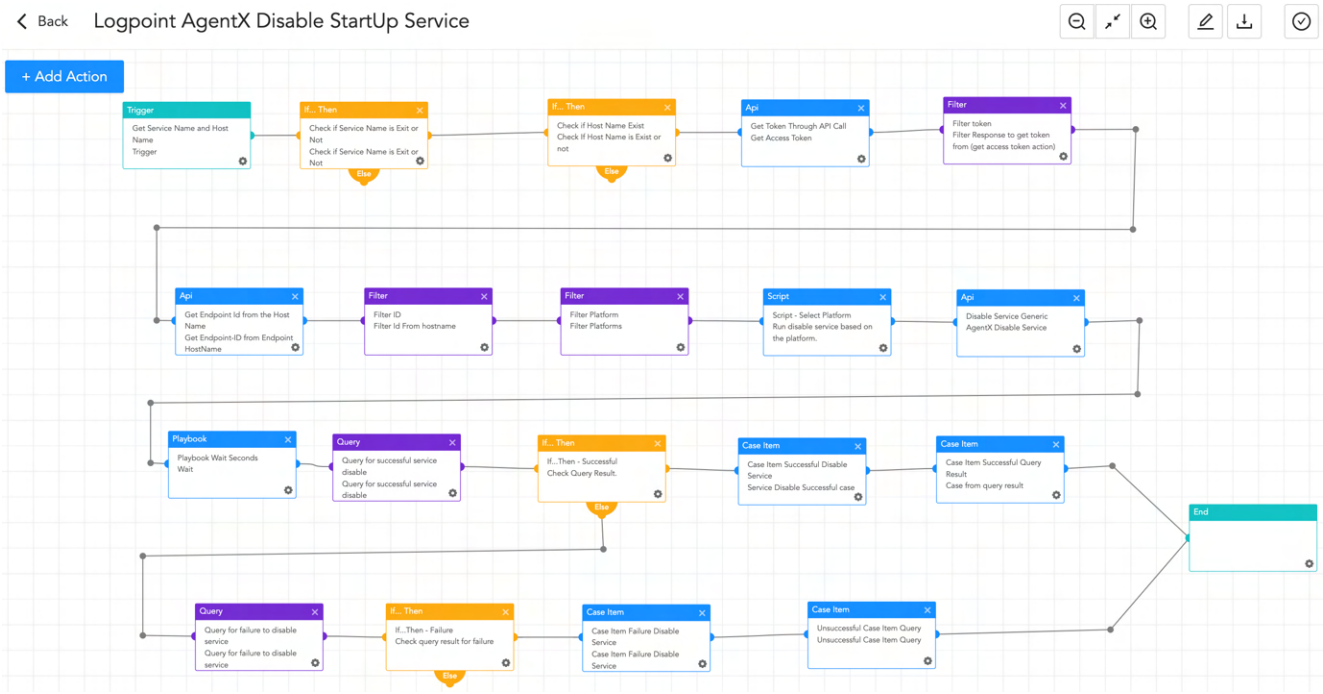
Delete Schedule Task Playbook



Schedule Task Deleted Case

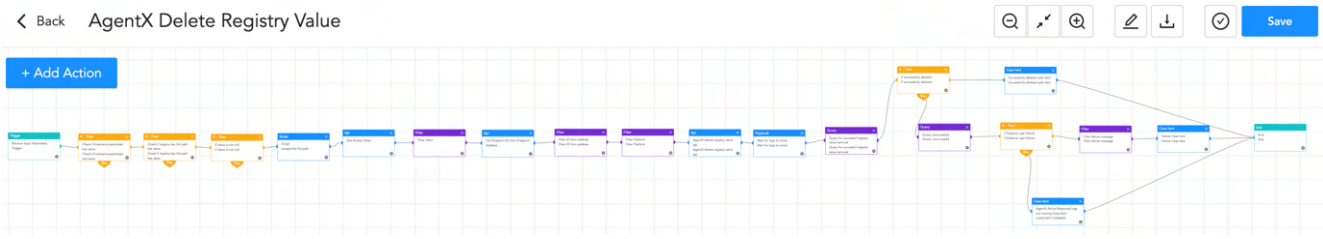
Disable Startup Service

The **Logpoint AgentX Disable Startup Service** response playbook reduces the burden of manually disabling a suspicious startup service. This playbook requires the analyst to provide the hostname of the machine, the manager IP address, and the startup service name.

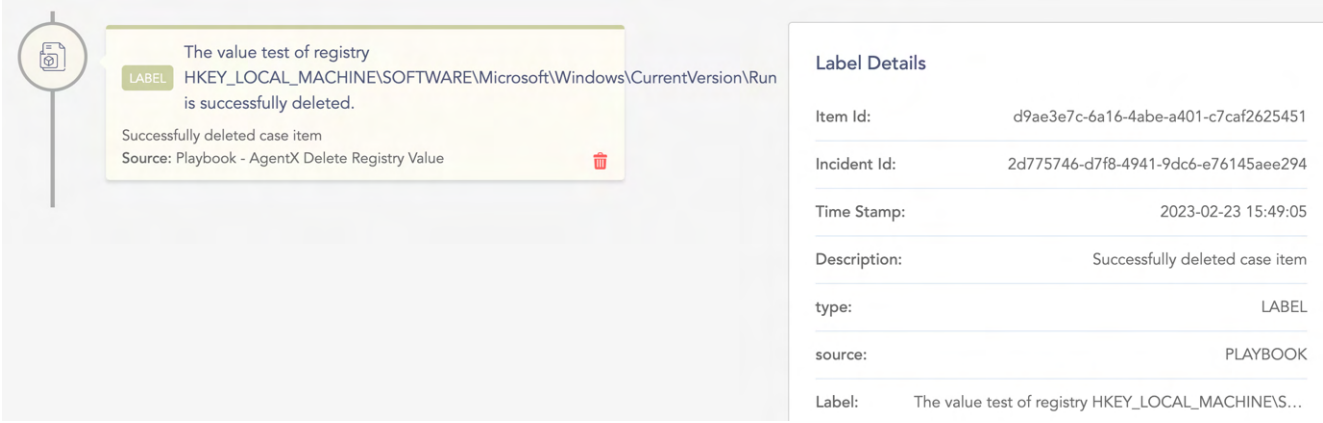


Delete Registry Value

As we have already provided alerts to investigate suspicious binaries placed in the Run registry or startup folders. An analyst can utilize the **AgentX Delete Registry Value** playbook to delete the registry value created under AutoRun Registry.

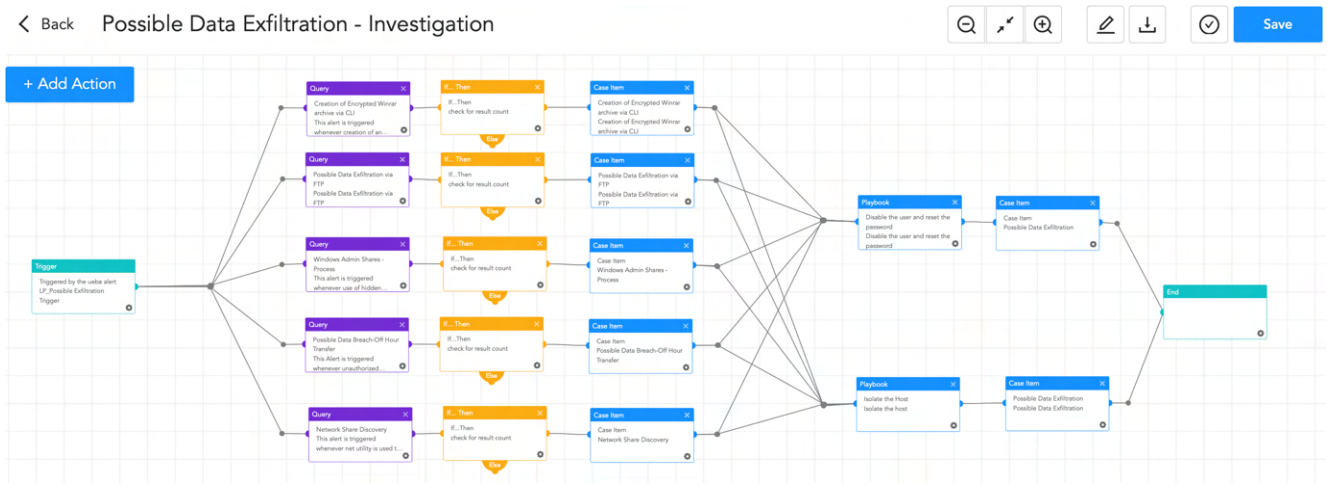


Below is the case result of the successful deletion of the registry value from the above playbook.



Possible Data Exfiltration

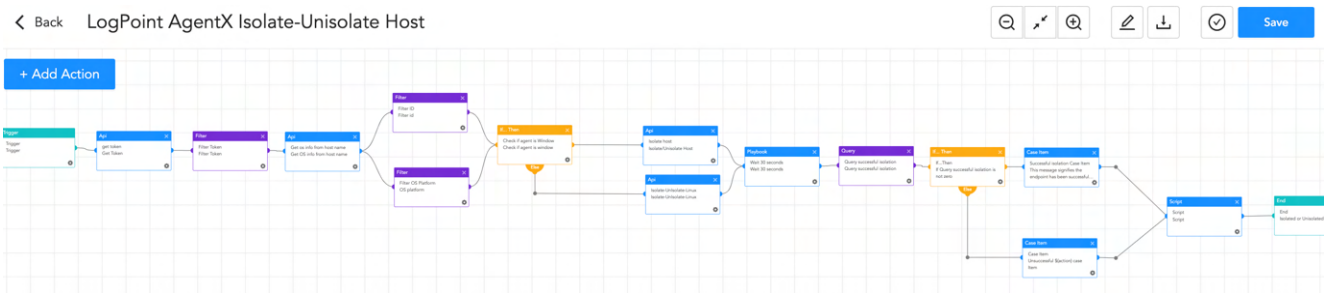
As we already have discussed various techniques used for data exfiltration, Possible Data Exfiltration - Investigation playbook can be utilized to investigate possible data exfiltration events.



Possible Data Exfiltration Investigation Playbook

Isolate Host

Once the alert rule to detect credential harvesting events is triggered, it is crucial to prevent those data from being exfiltrated, so an analyst can isolate the host by running the Logpoint AgentX Isolate-Unisolate Host playbook.



CONCLUSION

In conclusion, **AgentTesla** is a highly dangerous malware that can steal sensitive information and exfiltrate it. After conducting an analysis of the malware capabilities, it leads us to create effective detection rules that can help to detect the threat in the system and network. If not all, most of the AgentTesla behavior or traces can be detected by having proper auditing of systems and visibility of an organization's assets.

Enabling auditing of systems and leveraging Logpoint's SIEM can help to detect AgentTesla in various stages of infection changes. Whereas by deploying Logpoint's SOAR, organizations can proactively defend against AgentTesla and other malware threats by automating security operations and incident response workflows.
Besides removing suspicious registry run keys using AgentX, SOAR can be utilized to perform investigation actions and provide automatic responses to threats. AgentX is a lightweight application that enriches SIEM+SOAR events to provide increased endpoint protection.

If you would like to know more about **AgentX,** contact your Logpoint representative for further information.

ABOUT LOGPOINT

Logpoint is the creator of a reliable, innovative cybersecurity operations platform — empowering organizations worldwide to thrive in a world of evolving threats.
By combining sophisticated technology and a profound understanding of customer challenges, Logpoint bolsters security teams' capabilities while helping them combat current and future threats.
Logpoint offers SIEM, UEBA, and SOAR technologies in a complete platform that efficiently detects threats, minimizes false positives, autonomously prioritizes risks, responds to incidents, and much more.
Headquartered in Copenhagen, Denmark, with offices around the world, Logpoint is a multinational, multicultural, and inclusive company.
For more information visit **www.logpoint.com**