THREAT HUNT REPORT: HUNT FOR EXECUTION AND DISCOVERY

Lee Archinal March 27, 2023

EXECUTIVE SUMMARY

This report captures the findings from a hunt that consisted of two hunt packages. The first was searching for powershell executing a command that was encoded and the second hunted for activity that suggests a threat actor is actively enumerating our environment, which could be a follow on actions of a piece of malicious software executing in our environment.

The first hunt package that we used returned one event that captured encoded commands being executed. Once decoded we identified that it was a command to modify the CurrentVersion\Run registry key and added AnyDesk.exe to the registry value. This behavior suggested that the threat actor gained persistence through the AnyDesk executable which we confirmed was not the legitimate software. We then confirmed that AnyDesk.exe ran in our environment and produced a child process that was randomly named. We then found artifacts of enumeration using net.exe and whoami.exe from the randomly named executable.

The second hunt package we used returned multiple results that included different living-off-the-land binaries, aka LOLBINS, which are programs or software that come natively with the Windows Operating System). Our team focused on cmd.exe as the parent process because it was the parent of many of these LOLBINS. Once we expanded our search to see if me missed any cmd.exe as the parent process activity we found evidence of exfiltration via ftp.exe (File Transfer Protocol). Pivoting off ftp.exe we were able to determine a target internet protocol (IP) address, 10.10.30.98. We then expanded our search for any traffic to this IP address and we found that another AnyDesk.exe variant existed in the Windows Startup Folder and was actively reaching out to the same IP address of 10.10.30.98. This suggested that the threat actor had multiple levels of persistence and access to our environment.

Once we had enough evidence that suggested this was an incident, we gathered our findings and artifacts and escalated to the Digital Forensics and Incident Response Team (DFIR) who are currently working to eradicate the threat actors and return our environment to a non-compromised state.

ABSTRACT

Looks for valid variations of the -EncodedCommand parameter. This is commonly used to encode or obfuscate commands, and not all occurrences are malicious. For example, benign complex commands may require encoding to properly run on a target system. Analysis of the encoded command by base64 decoding the encoded data will be necessary.

HYPOTHESIS

Looks for valid variations of the -EncodedCommand parameter. This is commonly used to encode or obfuscate commands, and not all occurrences are malicious. For example, benign complex commands

may require encoding to properly run on a target system. Analysis of the encoded command by base64 decoding the encoded data will be necessary.

TECHNICAL SUMMARY

Overview

OUTCOME

Escalated

ESCALATION DETAILS

Encoded command modified registry keys specifically CurrentVersion\Run which lead to an executable launching from the following directory: C:\Users\jamesmurphy\Documents\AnyDesk.exe

The encoded command was decoded using the following recipe.
From_Base64('A-Za-z0-9+/=',true,false)

Regular_expression('User defined','',true,true,false,false,false,false,'Highlight matches')

The decoded command read as:
New-Item -Path 'HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run' -Name 'ClearCache' -Value C:\Users\jamesmurphy\Documents\AnyDesk.exe

Based on these findings we escalated to the IR team to begin remediation while we continued to gather indicators of attack.

Hunt Packages

POWERSHELL ENCODED COMMAND EXECUTION

Threat Description

Looks for valid variations of the -EncodedCommand parameter. This is commonly used to encode or obfuscate commands, and not all occurrences are malicious. For example, benign complex commands may require encoding to properly run on a target system. Analysis of the encoded command by base64 decoding the encoded data will be necessary.

Mitigation Recommendations

If the activity is confirmed to be malicious, host data for the affected system can be reviewed for the following data to gain a better understanding of what caused the malicious activity (root cause analysis): Process creation events prior to the observed activity while taking note of parent processes and privilege they re run at/with, and correlate available logs and network data based on the source host or hosts it was communicating with, such as SMB, HTTP(s), DNS, etc.. Once root cause is determined, utilize indicators (such as IPs, domains, hashes, urls, and process executions) to search across the enterprise in a 6 hour time window to identify additional hosts affected by the malicious activity. This can include searching for the same process execution (such as powershell), same confirmed download URL, or command and control IP/Domain. This will

determine the potential scope of the attack/activity for the Incident Response process.

MITRE Information

Mitre Attack Tactic

Defense Evasion, Execution

Mitre Attack Techniques

Obfuscated Files or Information, PowerShell

Mitre Attack Id

T1027, T1059.001

Analysis

1. CrowdStrike - edr

Query

```
ImageFileName="*\\powershell.exe"
| regex CommandLine="\-[Ee^]{1,2}[NnCcOoDdEeMmAaPpHh^‘]+\s+\"?[a-zA-Z0-
9+\/=]{6,}"
| stats values(_time) as Occurrences, values(CommandLine) as
commandLines, values(ParentBaseFileName) as parentProcessNames,
values(ImageFileName) as processPaths count by ComputerName
| convert ctime(Occurrences)
```

Notes

Initial: ImageFileName="*\powershell.exe" | regex CommandLine="-[Ee^]{1,2}[NnCcOoDdEeMmAaPpHh^‘]+\s+\"?[a-zA-Z0-9+\/=]{6,}" | stats values(_time) as Occurrences, values(CommandLine) as commandLines, values(ParentBaseFileName) as parentProcessNames, values(ImageFileName) as processPaths count by ComputerName | convert ctime(Occurrences)

Pivot Query 1: Pivot query logic that is based on the hostname that returned results.

Pivot 2: Pivot query logic that is based on a randomly named anomalous process that was discovered.

EXCESSIVE WINDOWS DISCOVERY AND EXECUTION PROCESSES - POTENTIAL MALWARE INSTALLATION

Threat Description

This package utilizes a list of commonly abused LOLB which an attacker or malware would execute in quick succession. The presence of multiple executions of the programs within the list can be indicative of an infection or malicious activity occurring on a victim host. To reduce false positives, distinct counts per process name can be utilized to ensure over 5 unique processes from the list were executed versus just checking more than 6 events were generated on the host.

Mitigation Recommendations

If a confirmed event is identified, then analysts should proceed to review the process data related to alerted event(s). This includes reviewing the commands issued by the process, its parent process, what user or permission level the process ran as, and if there are any unusual discrepancies in the process chain. An example of a discrepancy that warrants further investigation could be an unusual or randomly named parent process (xyzxyz.exe), or if the process chain appears to not be user generated in nature. It should be determined if the LOLB's were executed in relation to legitimate administrative activity due to them being legitimate binaries native to Windows - this can be confirmed by speaking to your system administrators or the user of the impacted machine, although it should be considered uncommon for a significant amount to be executed in quick succession of one another. If a suspicious binary, script, or other artifact is identified in the investigation that is indicative of ransomware, wiper, or other destructive malware, then it is recommended to quarantine the host from the network and initiate typical incident response measures against such an infection. Hash values, strings, and other indicators derived from the analysis of the suspicious file can be searched across the environment for the identification of other potentially impacted hosts. Analysts can also review endpoint logs for further evidence of compromise, such as behavior indicative of file encryption, domain enumeration, privilege escalation, or lateral movement.

MITRE Information

Mitre Attack Tactic

Discovery

Mitre Attack Techniques

System Network Configuration Discovery

Mitre Attack Id

T1016

Analysis

1. CrowdStrike - edr

Query

```
ImageFileName IN ("*\\arp.exe" "*\\at.exe" "*\\attrib.exe"
"*\\cscript.exe" "*\\dsquery.exe" "*\\hostname.exe" "*\\ipconfig.exe"
"*\\mimikatz.exe" "*\\nbtstat.exe" "*\\net.exe" "*\\netsh.exe"
"*\\nslookup.exe" "*\\ping.exe" "*\\quser.exe" "*\\qwinsta.exe"
"*\\reg.exe" "*\\runas.exe" "*\\sc.exe" "*\\schtasks.exe" "*\\ssh.exe"
"*\\systeminfo.exe" "*\\taskkill.exe" "*\\telnet.exe" "*\\tracert.exe"
"*\\wscript.exe" "*\\xcopy.exe" "*\\pscp.exe" "*\\copy.exe"
"*\\robocopy.exe" "*\\certutil.exe" "*\\vssadmin.exe" "*\\powershell.exe"
"*\\wevtutil.exe" "*\\psexec.exe" "*\\bcedit.exe" "*\\wbadmin.exe"
"*\\icacls.exe" "*\\diskpart.exe" "*ver.exe" "*netstat.exe"
"*tasklist.exe" "*route.exe" "*driverquery.exe")
| bucket _time span=5m
| stats values(_time) as eventTimes, values(ImageFileName) as
processPaths, values(ParentBaseFileName) as parentProcessPaths,
values(CommandLine) as commandLines, dc(ImageFileName) as
```

```
uniqueProcessPathCount by ComputerName, _time
| where uniqueProcessPathCount >= 5
| convert ctime(eventTimes)
```

Notes
Initial: ImageFileName IN ("*arp.exe"* "\at.exe" "*attrib.exe"* "\cscript.exe" "*dsquery.exe"* "\hostname.exe" "*ipconfig.exe"* "\mimikatz.exe" "*nbtstat.exe"* "\net.exe" "*netsh.exe"* "\nslookup.exe" "*ping.exe"* "\quser.exe" "*qwinsta.exe"* "\reg.exe" "*runas.exe"* "\sc.exe" "*schtasks.exe"* "\ssh.exe" "*systeminfo.exe"* "\taskkill.exe" "*telnet.exe"* "\tracert.exe" "*wscript.exe"* "\xcopy.exe" "*pscp.exe"* "\copy.exe" "*robocopy.exe"* "\certutil.exe" "*vssadmin.exe"* "\powershell.exe" "*wevtutil.exe"* "\psexec.exe" "*bcedit.exe"* "\wbadmin.exe" "*icacls.exe"* "\diskpart.exe" "*ver.exe"* "\netstat.exe" "*tasklist.exe"* "\route.exe" "*driverquery.exe") | bucket _time span=5m | stats values(_time) as eventTimes, values(ImageFileName) as processPaths, values(ParentBaseFileName) as parentProcessPaths, values(CommandLine) as commandLines, dc(ImageFileName) as uniqueProcessPathCount by ComputerName, _time | where uniqueProcessPathCount >= 5 | convert ctime(eventTimes)

Hunt 2 Pivot 1: Pivot query logic that sets cmd.exe as the parent or grandparent process.

Hunt 2 Pivot 2: Pivot query logic that correlates executable activity with remote connection activity looking for ftp.exe traffic.

Hunt 2 Pivot 3: Pivot query logic that correlates executable activity with remote connection activity using the remote IP address found. | sort by firstConnection asc

CONCLUSION

We were able to determine that the threat actors (TA) had successfully planted a backdoor in the CurrentVersion\Run registry. Upon further investigation the program had executed and led to more activity from the threat actor. We found evidence of enumeration commands being executed by child processes associated with the TA. Upon examination of the logs we also found evidence of Remote Desktop Protocol (RDP) activity as well. This indicated that the TA had two means of access to our environment. Finally, we found evidence of exfiltration through the use of the file transfer protocol (FTP). Viewing the event logs we were able to confirm the IP of the destination and found that more processes were reaching out to it, including the AnyDesk.exe that was located in the CurrentVersion\Run registry key.