Challenge: MeteorHit Lab

Platform: CyberDefenders

Category: Endpoint Forensics

Difficulty: Medium

Tools Used: Registry Explorer, Timeline Explorer, EVTXCmd, MFTECmd, VirusTotal

Summary: This lab involved using a variety of forensic tools to analyse a KAPE image. I absolutely loved the entire process, it's always fun to mess around with Eric Zimmerman tools and KAPE images. For those that love log analysis and NTFS forensics, I highly recommend giving this lab a shot.

Scenario: A critical network infrastructure has encountered significant operational disruptions, leading to system outages and compromised machines. Public message boards displayed politically charged messages, and several systems were wiped, causing widespread service failures. Initial investigations reveal that attackers compromised the Active Directory (AD) system and deployed wiper malware across multiple machines.

Fortunately, during the attack, an alert employee noticed suspicious activity and immediately powered down several key systems, preventing the malware from completing its wipe across the entire network. However, the damage has already been done, and your team has been tasked with investigating the extent of the compromise.

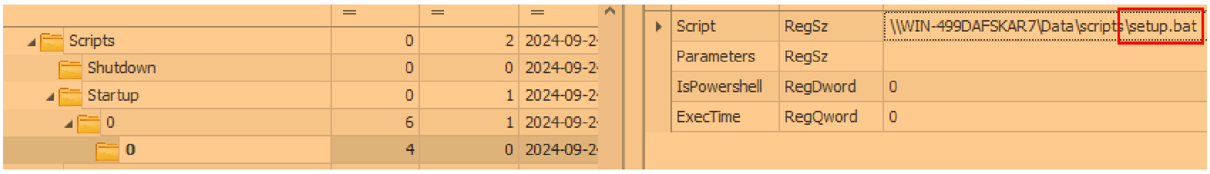
You have been provided with forensic artifacts collected via KAPE SANS Triage from one of the affected machines to determine how the attackers gained access, the scope of the malware's deployment, and what critical systems or data were impacted before the shutdown.

The attack began with using a Group Policy Object (GPO) to execute a malicious batch file. What is the name of the malicious GPO responsible for initiating the attack by running a script?

For context, a Group Policy Object (GPO) is a collection of settings that define how users and computers behave within an AD environment. GPOs are often abused by threat actors to deliver malware across multiple machines within the domain. You can find the malicious Group Policy Objects within the SOFTWARE registry hive, located at:

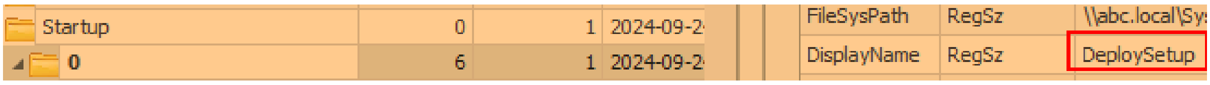
SOFTWARE\Microsoft\Windows\CurrentVersion\Group Policy\Scripts\Startup\0

You can open this up using Registry Explorer, which is an incredible tool made by Eric Zimmerman:



Scripts	0	2	2024-09-2
Shutdown	0	0	2024-09-2
Startup	0	1	2024-09-2
0	6	1	2024-09-2
0	4	0	2024-09-2

Script	RegSz	\\WIN-499DAFSKAR7\Data\scripts\setup.bat
Parameters	RegSz	
IsPowershell	RegDword	0
ExecTime	RegQword	0



Startup	0	1	2024-09-2
0	6	1	2024-09-2

FileSysPath	RegSz	\\abc.local\Sy
DisplayName	RegSz	DeploySetup

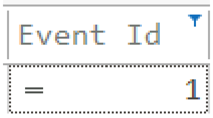
Answer: DeploySetup

During the investigation, a specific file containing critical components necessary for the later stages of the attack was found on the system. This file, expanded using a built-in tool, played a crucial role in staging the malware. What is the name of the file, and where was it located on the system? Please provide the full file path.

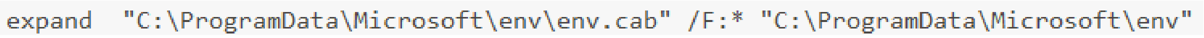
If you take a look at the winevt/logs folder within the KAPE image, you will see that the system where the image was taken from had Sysmon enabled. We can use this to search for Event ID 1 (Process Creation) to find any instances of a file being expanded. You can parse the Sysmon logs using EvtxECmd, or just use Windows Event Viewer:

.\EvtxECmd.exe -f "C:\Users\Administrator\Desktop\Start Here\Artifacts\C\Windows\System32\winevt\logs\Microsoft-Windows-Sysmon%4Operational.evtx" --csv . --csvf sysmon_out.csv

We can then open up this CSV output using Timeline Explorer, and filter for Event ID 1. If you navigate to the Executable Info field, we can see the command line associated with each process creation event:



Event Id
= 1



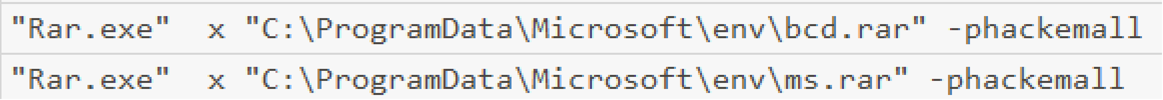
```
expand  "C:\ProgramData\Microsoft\env\env.cab" /F:* "C:\ProgramData\Microsoft\env"
```

The expand command is a built in Windows command that can be used to extract files from compressed cabinet (.cab) files.

Answer: C:\ProgramData\Microsoft\env\env.cab

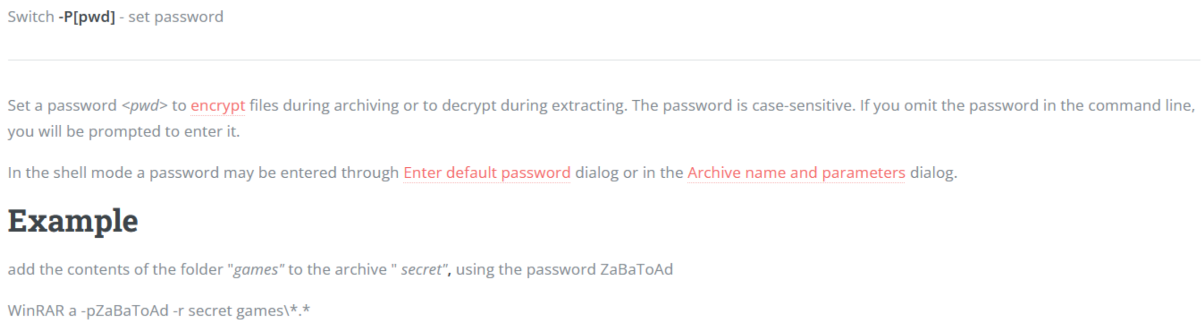
The attacker employed password-protected archives to conceal malicious files, making it important to uncover the password used for extraction. Identifying this password is key to accessing the contents and analyzing the attack further. What is the password used to extract the malicious files?

I knew that this would be found within the Sysmon logs as well. It took a while to realise, but with Rar.exe, there is no space following the -p parameter:



```
"Rar.exe"  x "C:\ProgramData\Microsoft\env\bcd.rar" -phackemall
"Rar.exe"  x "C:\ProgramData\Microsoft\env\ms.rar" -phackemall
```

Meaning the password is hackemail, and -p is used to signify that.



Switch **-P[pwd]** - set password

Set a password <*pwd*> to encrypt files during archiving or to decrypt during extracting. The password is case-sensitive. If you omit the password in the command line, you will be prompted to enter it.

In the shell mode a password may be entered through Enter default password dialog or in the Archive name and parameters dialog.

Example

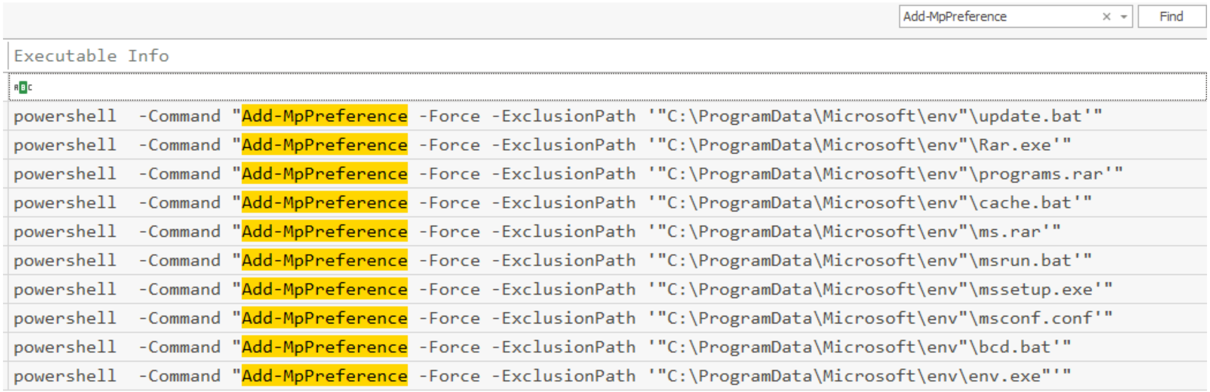
add the contents of the folder "*games*" to the archive " *secret*", using the password ZaBaToAd

WinRAR a -pZaBaToAd -r secret games*.*

Answer: hackemall

Several commands were executed to add exclusions to Windows Defender, preventing it from scanning specific files. This behavior is commonly used by attackers to ensure that malicious files are not detected by the system's built-in antivirus. Tracking these exclusion commands is crucial for identifying which files have been protected from antivirus scans. What is the name of the first file added to the Windows Defender exclusion list?

The Add-MpPreference PowerShell cmdlet is used to add exclusions for Windows Defender and is often abused by threat actors to evade Windows Defender. We can search for this within the Sysmon logs:



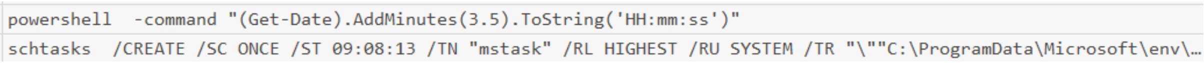
```
Executable Info
powershell  -Command "Add-MpPreference -Force -ExclusionPath '"C:\ProgramData\Microsoft\env"\update.bat'"
powershell  -Command "Add-MpPreference -Force -ExclusionPath '"C:\ProgramData\Microsoft\env"\Rar.exe'"
powershell  -Command "Add-MpPreference -Force -ExclusionPath '"C:\ProgramData\Microsoft\env"\programs.rar'"
powershell  -Command "Add-MpPreference -Force -ExclusionPath '"C:\ProgramData\Microsoft\env"\cache.bat'"
powershell  -Command "Add-MpPreference -Force -ExclusionPath '"C:\ProgramData\Microsoft\env"\ms.rar'"
powershell  -Command "Add-MpPreference -Force -ExclusionPath '"C:\ProgramData\Microsoft\env"\msrun.bat'"
powershell  -Command "Add-MpPreference -Force -ExclusionPath '"C:\ProgramData\Microsoft\env"\mssetup.exe'"
powershell  -Command "Add-MpPreference -Force -ExclusionPath '"C:\ProgramData\Microsoft\env"\msconf.conf'"
powershell  -Command "Add-MpPreference -Force -ExclusionPath '"C:\ProgramData\Microsoft\env"\bcd.bat'"
powershell  -Command "Add-MpPreference -Force -ExclusionPath '"C:\ProgramData\Microsoft\env\env.exe"'"
```

Answer: update.bat

A scheduled task has been configured to execute a file after a set delay. Understanding this delay is important for investigating the timing of potential malicious activity. How many seconds after the task creation time is it scheduled to run?

Note: Consider the system's time zone when answering questions related to time.

Schtasks.exe is used to schedule commands and programs to run periodically or at a specific time. We can search for this command to find scheduled tasks that were created by the threat actor.



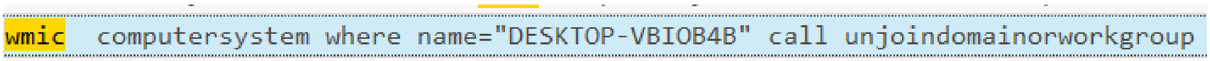
```
powershell  -command "(Get-Date).AddMinutes(3.5).ToString('HH:mm:ss')"
schtasks  /CREATE /SC ONCE /ST 09:08:13 /TN "mstask" /RL HIGHEST /RU SYSTEM /TR "\""C:\ProgramData\Microsoft\env\…
```

The schtasks command creates the scheduled task, and the PowerShell command above it appears to determine the delay. This PowerShell command adds 3.5 minutes (210 seconds) to the current system time and outputs it in HH:mm:ss format.

Answer: 210

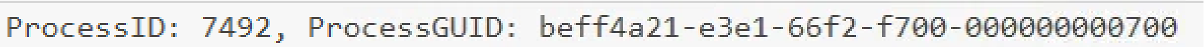
After the malware execution, the wmic utility was used to unjoin the computer system from a domain or workgroup. Tracking this operation is essential for identifying system reconfigurations or unauthorized changes. What is the Process ID (PID) of the utility responsible for performing this action?

WMIC is the WMI command-line utility that provides a command-line interface for Windows Management Instrumentation (WMI). Threat actors abuse WMI for many purposes, including executing processes or scripts, and often using it for lateral movement.



```
wmic  computersystem where name="DESKTOP-VBIOB4B" call unjoindomainorworkgroup
```

In the Payload Data1 field, you can see the ProcessID:

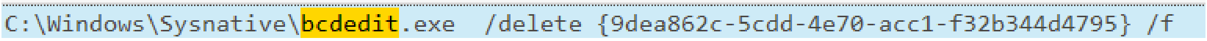


```
ProcessID: 7492, ProcessGUID: beff4a21-e3e1-66f2-f700-000000000700
```

Answer: 7492

The malware executed a command to delete the Windows Boot Manager, a critical component responsible for loading the operating system during startup. This action can render the system unbootable, leading to serious operational disruptions and making recovery more difficult. What command did the malware use to delete the Windows Boot Manager?

After a quick google search, I found out that you can deleted the Windows Boot Manager by using the bcdedit /delete {bootmgr} command.



```
C:\Windows\Sysnative\bcdedit.exe  /delete {9dea862c-5cdd-4e70-acc1-f32b344d4795} /f
```

Answer: C:\Windows\Sysnative\bcdedit.exe /delete {9dea862c-5cdd-4e70-acc1-f32b344d4795} /f

The malware created a scheduled task to ensure persistence and maintain control over the compromised system. This task is configured to run with elevated privileges every time the system starts, ensuring the malware continues to execute. What is the name of the scheduled task created by the malware to maintain persistence?

If you search for schtasks (the command used to create or modify scheduled tasks), we can find a scheduled task with the ONSTART parameter that runs as SYSTEM (denoted by /RU SYSTEM):

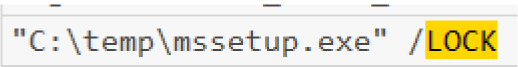
```
C:\Windows\System32\cmd.exe /c schtasks /CREATE /SC ONSTART /TN "Aa153!EGzN" /RL HIGHEST
/RU SYSTEM /TR "\"C:\ProgramData\Microsoft\env\env.exe\" \"C:\temp\msconf.conf\"" /F
```

The name of the scheduled task is the string after /TN, AKA "Aa153!EGzN"

Answer: Aa153!EGzN

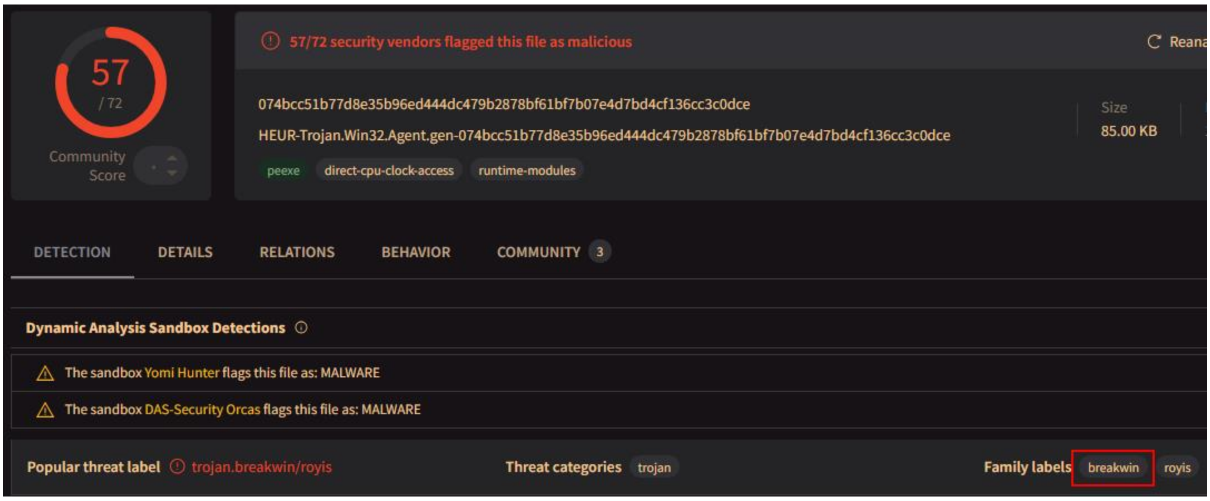
A malicious program was used to lock the screen, preventing users from accessing the system. Investigating this malware is important to identify its behavior and mitigate its impact. What is the name of this malware? (not the filename)

On a whim, I searched for "lock" within the Sysmon logs and came across this process command line:



```
"C:\temp\mssetup.exe" /LOCK
```

The presence of /LOCK along with mssetup.exe being within the temp directory raises many alarms. If you look at the Payload Data3 field, you can copy one of the hashes (MD5, SHA256, etc), and chuck it into VirusTotal:



57 / 72

57/72 security vendors flagged this file as malicious

074bcc51b77d8e35b96ed444dc479b2878bf61bf7b07e4d7bd4cf136cc3c0dce

HEUR-Trojan.Win32.Agent.gen-074bcc51b77d8e35b96ed444dc479b2878bf61bf7b07e4d7bd4cf136cc3c0dce

Size 85.00 KB

Dynamic Analysis Sandbox Detections

⚠ The sandbox Yomi Hunter flags this file as: MALWARE

⚠ The sandbox DAS-Security Orcas flags this file as: MALWARE

Popular threat label trojan.breakwin/royis

Threat categories trojan

Family labels breakwin royis

We can see that this executable is labelled as BreakWin, and has 57/72 detections which is extremely high.

Answer: breakwin

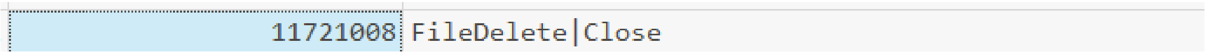
The disk shows a pattern where malware overwrites data (potentially with zero-bytes) and then deletes it, a behavior commonly linked to Wiper malware activity. The USN (Update Sequence Number) is vital for tracking filesystem changes on an NTFS volume, enabling investigators to trace when files are created, modified, or deleted, even if they are no

longer present. This is critical for building a timeline of file activity and detecting potential tampering. What is the USN associated with the deletion of the file msuser.reg?

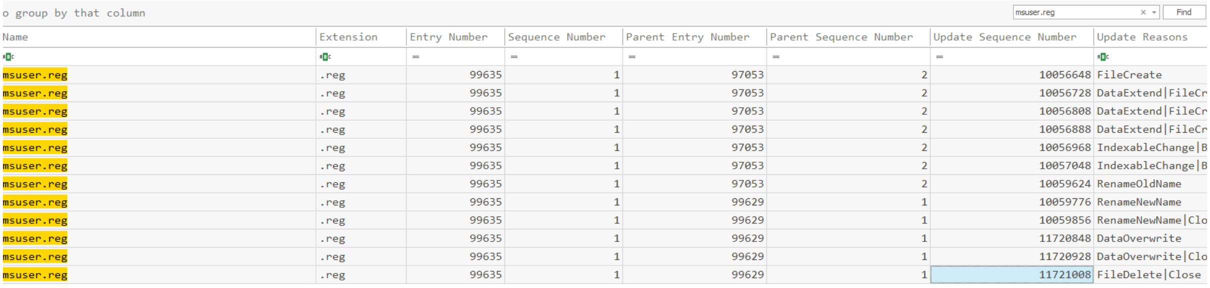
The USN Journal is a forensic artifact that maintains a record of changes made to the NTFS file system. The creation, deletion, or modification of files or directories are journalised/stored here. You can find the USN Journal in $Extend\$J. We can use MFTECmd.exe to parse the USN Journal file by executing the following command:

MFTECmd.exe -f "C:\Users\Administrator\Desktop\Start Here\Artifacts\C\`$Extend\`$J" --csv . --csvf usn_out.csv

Chuck the csv file into Timeline Explorer and search for msuser.reg, you can see a FileDelete|Close Update Reason, meaning that this file was deleted. Its USN (Update Sequence Number) is seen on the left:



11721008 FileDelete|Close



Name	Extension	Entry Number	Sequence Number	Parent Entry Number	Parent Sequence Number	Update Sequence Number	Update Reasons
msuser.reg	.reg	99635	1	97053	2	10056648	FileCreate
msuser.reg	.reg	99635	1	97053	2	10056728	DataExtend|FileCr
msuser.reg	.reg	99635	1	97053	2	10056808	DataExtend|FileCr
msuser.reg	.reg	99635	1	97053	2	10056888	DataExtend|FileCr
msuser.reg	.reg	99635	1	97053	2	10056968	IndexableChange|B
msuser.reg	.reg	99635	1	97053	2	10057048	IndexableChange|B
msuser.reg	.reg	99635	1	97053	2	10059624	RenameOldName
msuser.reg	.reg	99635	1	99629	1	10059776	RenameNewName
msuser.reg	.reg	99635	1	99629	1	10059856	RenameNewName|Clo
msuser.reg	.reg	99635	1	99629	1	11720848	DataOverwrite
msuser.reg	.reg	99635	1	99629	1	11720928	DataOverwrite|Clo
msuser.reg	.reg	99635	1	99629	1	11721008	FileDelete|Close

Answer: 11721008