

## Challenge: [ZeroLogon Lab](#)

**Platform:** CyberDefenders

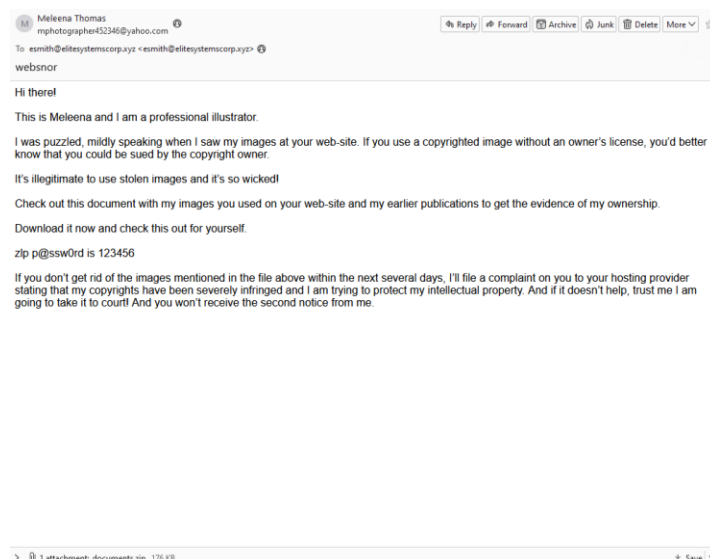
**Category:** Endpoint Forensics

**Difficulty:** Hard

**Tools Used:** MFTECmd, Timeline Explorer, LECmd, EvtxECmd, CyberChef, Notepad++, Event Log Explorer

**Summary:** This lab involves investigating a small-scale intrusion that began with a phishing email. The threat actor delivered a malicious ZIP file (documents.zip) containing a batch script (eyewear.bat) and an executable (easygoing.exe). Once executed, the malware established C2 communications with 42.63.200.142, conducted reconnaissance, and attempted privilege escalation via named pipe impersonation. Evidence in Sysmon logs revealed LSASS access for credential dumping and use of compromised credentials for lateral movement using WMIC. Persistence was achieved by creating a scheduled task running a malicious PowerShell script, while another script staged user data from C:\Users\ for exfiltration. Lateral movement and persistence continued as the threat actor installed AnyDesk and enabled RDP through registry modification. This lab was extremely enjoyable, honestly one of the best labs I have taken part in. For those who enjoy endpoint forensics, I recommend giving this a shot. Don't let the hard difficulty rating deter you, it's more of a medium level lab.

**Scenario:** Your role as a Tier 2 SOC Analyst at EliteSystems Corp is put to the test following an alert from the Tier 1 team about a confirmed phishing email leading to a potential network wide intrusion. With disk data already triaged and ready for analysis, you must uncover the extent of this intrusion and identify the compromised assets within the network.



**Analyzing the attack chain requires identifying the file that initiated the payload execution. Which shortcut file was generated after executing the payload-containing file extracted from the ZIP archive?**

**TLDR:** Use MFTECmd to parse the MFT file. Filter for the .zip file extension and focus on a suspicious zip file within the Downloads folder. If you filter for the parent path of this zip file, you can find the generated shortcut file.

In this lab, we are provided with triage images for 4 machines:

DC01	9/15/2024 12:33 PM	File folder
FileServer	1/2/2024 2:12 PM	File folder
FIN-PC	1/2/2024 2:12 PM	File folder
IT-PC	1/2/2024 2:12 PM	File folder

To find the ZIP archive sent in the phishing email, I am going to parse and analyse the MFT for each machine, starting with FIN-PC. The Master File Table (\$MFT) is a database that tracks all object (file and folder) changes on an NTFS filesystem. Each object has its own record in the \$MFT, containing metadata about that file or folder. To parse the MFT, I am going to use a tool called MFTECmd:

- `.\MFTECmd.exe -f "C:\Users\Administrator\Desktop\Start Here\Artifacts\FIN-PC\`$MFT" --csv . --csvf FIN-PC-MFT-OUT.csv`
  - -f specifies the path to the MFT file
  - --csv indicates where to save the output file
  - --csvf indicates the output filename

We can view the CSV output using another Eric Zimmerman tool called Timeline Explorer. Make sure to filter for the extension “.zip” to limit the results to Zip files:

Parent Path	File Name	Extension
.Program Files\WindowsApps\Microsoft.Microsoft3DViewer_6.1908.2042.0_x64__8wekyb3d8bbwe\Assets	Archive.zip	.zip
.Users\esmith\Downloads	documents.zip	.zip
.\Data	administrator.zip	.zip
.\Data	Administrator.FIN-PC.zip	.zip
.\Data	dlee.zip	.zip
.\Data	emilysmith.zip	.zip
.\Data	esmith.zip	.zip

A file that stands out is called documents.zip within the user esmith Downloads directory.

To find the shortcut file created after executing the payload-containing file extracted from the zip archive, we can use a tool called LECmd. For context, a Windows shortcut (LNK) file is simply a pointer to open a file or folder. LNK files can be created automatically by the OS when a user opens a file:

- `.\LECmd.exe -d "C:\Users\Administrator\Desktop\Start Here\Artifacts\FIN-PC\Users\esmith\AppData\Roaming\Microsoft\Windows\Recent" --csv . --csvf esmith-lnk-out.csv`
  - -d specifies to recursively parse a directory for LNK files
  - --csv indicates where to save the output file

- --csvf indicates the output filename

Here we can find the name of the LNK file:

```
\\FIN-PC\Users\esmith\AppData\Roaming\Microsoft\Windows\Recent\Documents.lnk
```

Alternatively, just filter for the parent path of the zip file and you can find the LNK file there as well.

Answer: documents.lnk

**It's essential to gather as much information as possible about the attack. Can you identify the malicious script inside the ZIP Archive?**

**TLDR:** Filter for the parent path of the zip file identified in the previous question to see what was extracted.

To identify what files were extracted from the zip archive, we can navigate back to the MFTECmd output and filter for the relevant parent path like as follows:

Parent Path	File Name	Extension	Is Directory	Has Ads	Is Ads	File Size	Created0x10
\\Users\esmith\Downloads			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	=	=
\\Users\esmith\Downloads	documents.zip	.zip	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	179970	2024-01-01 20:00:24
\\Users\esmith\Downloads	documents.zip:Zone.Identifier	.Identifier	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	50	2024-01-01 20:00:24
\\Users\esmith\Downloads	documents		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	2024-01-01 20:00:34
\\Users\esmith\Downloads\documents	max		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	2024-01-01 20:00:34
\\Users\esmith\Downloads\documents\max	easygoing.exe	.exe	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	315392	2024-01-01 20:00:37
\\Users\esmith\Downloads\documents\max	eyewear.bat	.bat	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	271	2024-01-01 20:00:37
\\Users\esmith\Downloads\documents	documents.lnk	.lnk	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1949	2024-01-01 20:00:37
\\Users\esmith\Downloads	desktop.ini	.ini	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	282	2023-12-30 23:24:35

We can see the original zip file, documents.zip, extracts to documents\max. Within the max folder, we can see three documents, a LNK file, easygoing.exe, and eyewear.bat. Note, we can determine that the zip archive contained these documents due to the parent path along with the creation timestamp (all three files within the max directory were created at the same time). A .bat file, or batch file, is a script file used in Windows, therefore, the malicious script is eyewear.bat.

Answer: eyewear.bat

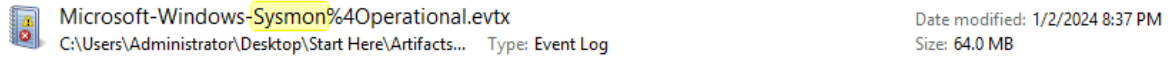
**By identifying the C2 IP address, we can gather clues about the attacker, such as their possible location, identity, or affiliation, and understand their motives. Can you identify the C2 IP address?**

**TLDR:** Filter for Event ID 3 in the Sysmon logs and search for a large number of network connections made to an external host. Focus on those originating from a binary extracted from the documents.zip file.

If you navigate to:

- %SYSTEMROOT%\System32\winevt\logs

We can find all the event logs captured on FIN-PC. Fortunately, this host had Sysmon configured, which we can use to look for network connections:



We can use a tool called EvtxECmd to parse the Sysmon logs:

- `.\EvtxECmd.exe -f "C:\Users\Administrator\Desktop\Start Here\Artifacts\FIN-PC\Windows\System32\winevt\logs\Microsoft-Windows-Sysmon%4Operational.evtx" --csv . --csvf FIN-PC-SYSMON-OUT.csv`
  - -f specifies the path to the EVTX file
  - --csv indicates where to save the output file
  - --csvf indicates the output filename

Event ID 3 within Sysmon records network connections. If you group by the DestinationIp field, we can see that there were 73,787 network connections made to 42.63.200.142:

Payload Data6: DestinationIp: fe80:0:0:0:e9ba:304b:2270:361a (Count: 2)
Payload Data6: DestinationIp: fe80:0:0:0:7ed5:14fd:e25e:eeeb (Count: 10)
Payload Data6: DestinationIp: 42.63.200.142 (Count: 73,787)
Payload Data6: DestinationIp: 192.168.202.197 (Count: 3)
Payload Data6: DestinationIp: 192.168.202.154 (Count: 3)
Payload Data6: DestinationIp: 192.168.202.132 (Count: 3)
Payload Data6: DestinationIp: 192.168.202.126 (Count: 17)
Payload Data6: DestinationIp: 192.168.202.1 (Count: 10)

This is extremely suspicious relative to the other network connections, especially due to the volume which is consistent with C2 activity (beaconing). If you look at the event field, we can see that the source image behind these network connections is the easygoing.exe binary that was extracted from the documents.zip archive discovered previously:



**With escalated privileges, an attacker can typically do more damage. What command did the attacker use to attempt privilege escalation?**

After exploring process creation logs, I came across a suspicious cmd.exe command, with the parent process rundll32.exe:

```
C:\Windows\system32\rundll32.exe C:\Windows\system32\cmd.exe /c echo ddb867670d7 > \\.\pipe\308808
```

After doing research, this is consistent with named pipe impersonation for privilege escalation, commonly associated with Cobalt Strike. This seems likely as rundll32.exe is a default beacon spawned by Cobalt Strike.

If you filter for Pipe creation events (EID 17), we can see named pipes being created that are consistent with Cobalt Strike:

```
{ "EventData": { "Data": [ { "@Name": "RuleName", "#text": "-"}, { "@Name": "EventType", "#text": "CreatePipe"}, { "@Name": "UtcTime", "#text": "2024-01-02 20:53:52.786"}, { "@Name": "ProcessGuid", "#text": "9ba2f8d1-77e0-6594-a902-000000000d00"}, { "@Name": "ProcessId", "#text": "4276"}, { "@Name": "PipeName", "#text": "\\MSSE-3245-server"}, { "@Name": "Image", "#text": "\\127.0.0.1\\ADMIN$\\a962278.exe"}, { "@Name": "User", "#text": "NT AUTHORITY\\SYSTEM"} ] } }
```

Other named pipes, including those that begin with postex\_\* are also consistent with Cobalt Strike:

PipeName: \PSHost.13348699...
PipeName: \postex_41bd
PipeName: \postex_3db4
PipeName: \postex_00dd
PipeName: \MSSE-3245-server
PipeName: \postex_a40f
PipeName: \postex_9279
PipeName: \308808
PipeName: \postex_cb08
PipeName: \PSHost.13348706...
PipeName: \postex_3e52
PipeName: \postex_a161
PipeName: \PSHost.13348698...

Answer: echo ddb867670d7 > \\.\pipe\308808

**We need to assess the severity of the breach. Was the attacker able to compromise any user account? Can you provide the password of the user account the attacker compromised?**

**TLDR:** Look for credentials within executed commands.

On Jan 2<sup>nd</sup>, at 21:28:43, shortly after the privilege escalation attempt discovered previously, the threat actor executed a WMIC command targeting user FileShareService at 192.168.202.126:

```
C:\Windows\system32\cmd.exe /C wmic /node:192.168.202.126 /user:FileShareService /password:MYpassword123# logicaldisk get caption,description,drivetype,providername,volumename
```

This command runs WMIC from cmd.exe to query the logical disks on the remote machine using the supplied credentials. Given this, we know that the user FileShareService has been compromised.

At 20:53:28 and 21:12:44, rundll32.exe was observed accessing lsass.exe, which is behaviour consistent with credential dumping. Both granted access values are associated with credential dumping through LSASS access:

GrantedAccess: Unknown code (0x1FFFFFF)
GrantedAccess: 0x1010 (PROCESS_QUERY_LIMITED_INFORMATION & PROCESS_VM_READ)

Given this, it's safe to assume that the threat actor was able to successfully dump and extract credentials from LSASS.

Answer: MYpassword123#

**To ensure complete eradication, identifying and removing persistence mechanisms is essential to ensure the attacker can no longer access the compromised system. What's the command used by the attacker to achieve persistence?**

**TLDR:** Search for common persistence mechanisms used by threat actors within the process creation logs.

Shortly after the threat actor queried 192.168.202.126, they were observed creating a scheduled task at 21:35:46:

```
C:\Windows\system32\cmd.exe /C schtasks /create /tn "ChromeUpdater" /tr "powershell -File 'C:\Users\esmith\AppData\Local\ChromeUpdater\ChromeUpdate.ps1'" /sc onlogon /ru System
```

The threat actor used schtasks to create a scheduled task called "ChromeUpdater" that executes ChromeUpdate.ps1 on logon as the System user.

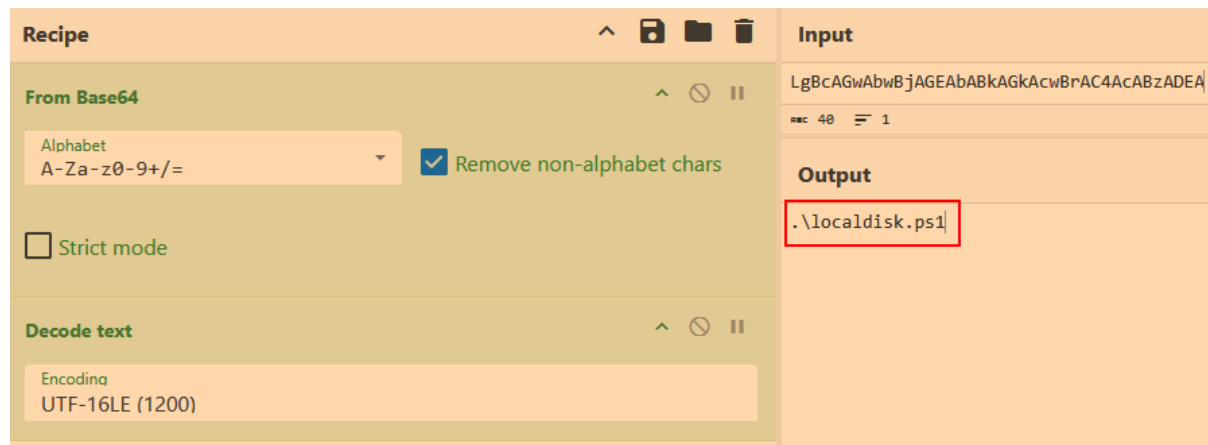
Answer: schtasks /create /tn "ChromeUpdater" /tr "powershell -File 'C:\Users\esmith\AppData\Local\ChromeUpdater\ChromeUpdate.ps1'" /sc onlogon /ru System

**Identifying the targeted data for exfiltration allows the organization to understand the potential impact of the breach and the data's confidentiality level. What's the full path of the folder whose data was targeted by the PowerShell script?**

**TLDR:** Examine encoded PowerShell commands found within the process creation logs. Once you discover the executed PowerShell script, navigate to its working directory and analyse the script.

Continuing with exploring process creation events, we can see another encoded PowerShell command being executed at 21:59:32:

ParentCommandLine: C:\Windows\System32\rundll32.exe powershell -nop -exec bypass -EncodedCommand LgBcAGwAbwBjAGEAbABkAGkAcwBrAC4AcABzADEA



This executes a PowerShell script called localdisk.ps1 located at:

- C:\Users\Administrator\AppData\Local\Temp\

If you navigate to this file, we can examine it using a tool like Notepad++:

```
localdisk.ps1
1 $destinationPath = "C:\Data"
2
3 if (-not (Test-Path -Path $destinationPath)) {
4     New-Item -ItemType Directory -Path $destinationPath
5 }
6
7 $directories = Get-ChildItem -Path "C:\Users\" -Directory
8
9 foreach ($dir in $directories) {
10     $zipFileName = "$($dir.Name).zip"
11     $zipFilePath = Join-Path $destinationPath $zipFileName
12     Compress-Archive -Path $dir.FullName -DestinationPath $zipFilePath
13     Write-Host "Compressed $($dir.FullName) to $zipFilePath"
14 }
15
16 Write-Host "All folders in C:\Users\ have been compressed."
```

This PowerShell script compresses every user folder in C:\Users\ and saves the resulting ZIP files to C:\Data. This behaviour is consistent with data staging prior to exfiltration. If you recall earlier when we were identifying the name of the zip file contained within the phishing email, we discovered multiple zip files within the MFT named after the users on the machine:



.\Data	administrator.zip
.\Data	Administrator.FIN-PC.zip
.\Data	dlee.zip
.\Data	emilysmith.zip
.\Data	esmith.zip

This tells us that the script successfully executed.

Answer: C:\Users

**To understand the spread of the intrusion and discover possible lateral movement attempts. What is the name of the malicious service installed remotely on FileServer?**

**TLDR:** Filter for Event ID 7045 in the System.evtx logs to identify service creation events on FileServer.

There are two primary log sources that detail service creation, that being EID 4697 in the Security.evtx file and EID 7045 in System.evtx. The former provides more comprehensive information but is not configured by default, and the latter is enabled by default. Unfortunately, EID 4697 is not logged in this environment, therefore, let's check out 7045 in the System event logs (note, I am going to use Event Log Explorer rather than EvtxECmd just for simplicity):

Type	Date	Time	Event	Source	Category	User	Computer
Information	1/2/2024	9:18:51 PM	7045	Service Control Ma	None	\SYSTEM	FileServer.elitesystems.loc
Information	1/2/2024	9:09:55 PM	7045	Service Control Ma	None	\S-1-5-21-145473079	FileServer.elitesystems.lo

There are two results, meaning two services were installed on the system, both within the same time range of the intrusion. The first is a service for AnyDesk, which is an RMM tool often abused by threat actors:

Description
A service was installed in the system.
Service Name: AnyDesk Service
Service File Name: "C:\Program Files (x86)\AnyDesk\AnyDesk.exe" --service
Service Type: user mode service
Service Start Type: auto start
Service Account: LocalSystem

The second service is for a suspicious binary within the ADMIN\$ share of the FileServer:

Description
A service was installed in the system.
Service Name: 075b12b
Service File Name: <a href="#">\\FILESERVER\ADMIN\$\075b12b.exe</a>
Service Type: user mode service
Service Start Type: demand start
Service Account: LocalSystem

This binary was observed executing rundll32.exe within the process creation logs, given that we know rundll32.exe to be malicious (likely a Cobalt Strike beacon), it's safe to say that this is a malicious service.

Answer: 075b12b

**Credential dumping can significantly expand the breach's impact, giving attackers access to numerous systems and data. Can you identify the process name that dumped credentials?**

**TLDR:** Filter for Event ID 10 (process access) and target image "lsass.exe" in the Sysmon logs.

If you filter for Event ID 10 in the Sysmon logs, which records when a process accesses another process, we can see at 20:53:28 and 21:12:44, rundll32.exe was observed accessing lsass.exe, which is behaviour consistent with credential dumping. Both granted access values are associated with credential dumping through LSASS access:

GrantedAccess: Unknown code (0x1FFFFFF)
GrantedAccess: 0x1010 (PROCESS_QUERY_LIMITED_INFORMATION & PROCESS_VM_READ)

Answer: rundll32.exe

**Attackers usually install software on a target system to maintain long-term access, move laterally, access other systems, and expand their reach. What remote access software did the attacker install on one of the machines?**

**TLDR:** Look at commands executed on FileServer indicating installation of a known RMM tool.

As discovered by the created services on FileServer, we know that an AnyDesk service was present on the machine. Using the following command:

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

We can parse the Sysmon logs on FileServer and examine them using Timeline Explorer. If you focus on process creation logs (EID 1) and filter for the string "anydesk" we can see cmd.exe being used to install anydesk.exe (note the parent process of rundll32.exe):

Executable Info	
	anydesk
C:\Windows\System32\rundll32.exe	C:\Windows\system32\cmd.exe /C start anydesk.exe --install "C:\Program Files (x86)\AnyDesk" --start-with-win --create-desktop-icon
C:\Windows\system32\cmd.exe /C s...	anydesk.exe --install "C:\Program Files (x86)\AnyDesk" --start-with-win --create-desktop-icon
C:\Windows\Explorer.EXE	"C:\Program Files (x86)\AnyDesk\AnyDesk.exe" --control
C:\Windows\Explorer.EXE	"C:\Program Files (x86)\AnyDesk\AnyDesk.exe" --new-install

Answer: AnyDesk

## What password did the attacker set for the installed software?

Right after installing AnyDesk, we can see the threat actor creating a password via cmd.exe:

```
C:\Windows\System32\rundll32.exe C:\Windows\system32\cmd.exe /C echo Qwerty123!@#_! | AnyDesk.exe --set-password
```

Answer: Qwerty123!@#\_!

## Attackers often enable RDP for more control. What command was used by the attacker to enable RDP?

**TLDR:** Look for reg add commands that enable RDP in registry.

Using the following command:

- `.\EvtxECmd.exe -f "C:\Users\Administrator\Desktop\Start Here\Artifacts\DC01\Windows\System32\winevt\logs\Microsoft-Windows-Sysmon%40operational.evtx" --csv . --csvf DC-01-SYSMON-OUT.csv`

We can parse the Sysmon logs on DC01. To enable RDP, the threat actor likely needs to modify something in registry, so let's filter for commands that contain "reg":

```
Executable Info
reg
C:\Windows\system32\svchost.exe -k localService -p -s RemoteRegistry
C:\Windows\system32\cmd.exe /c C:\Windows\system32\reg.exe query hklm\software\microsoft\windows\softwareinventorylogging /v collectionstate /reg:64
C:\Windows\system32\reg.exe query hklm\software\microsoft\windows\softwareinventorylogging /v collectionstate /reg:64
C:\Windows\system32\sc.exe start pushtoinstall registration
cmd.exe /Q /c reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Microsoft Defender" /v DisableAntiSpyware /t REG_DWORD /d 1 /f 1> \\127.0.0.1\ADMIN$\_1704190945.703671 2>&1
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Microsoft Defender" /v DisableAntiSpyware /t REG_DWORD /d 1 /f
cmd.exe /Q /c reg add "hklm\system\currentcontrolset\control\terminal server" /f /v fDenyTSConnections /t REG_DWORD /d 0 1> \\127.0.0.1\ADMIN$\_1704190945.703671 2>&1
reg add "hklm\system\currentcontrolset\control\terminal server" /f /v fDenyTSConnections /t REG_DWORD /d 0
```

We can see the RemoteRegistry service being started, which enables remote users to connect to the Windows Registry over the network. We can then observe reg.exe being used to query the registry, followed by reg add being used to disable Microsoft Defender and enable RDP. Both commands redirect the output to a local ADMIN\$ share file.

Answer: `reg add "hklm\system\currentcontrolset\control\terminal server" /f /v fDenyTSConnections /t REG_DWORD /d 0`