

# Simple Obfuscation with PowerShell using Base64 Encoding

 Jun 15, 2017

· [POWERSHELL](#)

· Share on: 

I recently received a question from someone wanting to know how I encoded a string of text on my blog site. Back in January of 2013, I competed in Jeff Hicks PowerShell Challenge that was held by TrainSignal. One of the questions had an encoded command which you were to decode. I figured out that the `EncodedCommand` parameter of `PowerShell.exe` could not only be used to run commands that are encoded with Base64, that it could also be used to easily decode a string of text that was encoded with Base64.

```
powershell.exe /? BATCH
```

```
...
-EncodedCommand
    Accepts a base-64-encoded string version of a command. Use this parameter
    to submit commands to Windows PowerShell that require complex quotation
    marks or curly braces.
...
```

The help for `PowerShell.exe` also shows you how to encode a command with Base64:

```
...
# To use the -EncodedCommand parameter:
$command = 'dir "c:\program files" '
$bytes = [System.Text.Encoding]::Unicode.GetBytes($command)
$encodedCommand = [Convert]::ToBase64String($bytes)
powershell.exe -encodedCommand $encodedCommand
```

Encoding something like the domain name for this blog site is easy enough:

```
[Convert]::ToBase64String([System.Text.Encoding]::Unicode.GetBytes("mikefrobbins.com")) POWERSHELL
```

```
JwBtAGkAawB1AGYAcgBvAGIAYgBpAG4AcwAuAGMAbwBtACcA
```

While it could be decoded within PowerShell:

```
[System.Text.Encoding]::Unicode.GetString([System.Convert]::FromBase64String('JwBtAGkAawB1AGYAcgBvAGIAYgBpAG4AcwAuAGMAbwBtACcA')) POWERSHELL
```

```
'mikefrobbins.com'
```

Search...



Mike F. Robbins

Scripting | Automation | Efficiency

READ MORE

## Disclaimer

All information and code on this site is for informational purposes only and provided as-is. This site does not provide any warranty, either express or implied. All thoughts and opinions are my own.

## Recent Posts

- How to install PowerShell 7 and essential tools on Linux
- How to install PowerShell 7 and essential tools on Windows 11
- Find paired Azure region locations with Azure PowerShell
- Understanding the Clean block in PowerShell
- Detecting Windows Terminal with PowerShell
- How to resolve winget is unable to find or install packages
- Check out someone else's pull request using the GitHub CLI
- Generating PowerShell module documentation with platyPS

## Categories

- POWERSHELL

347
- ACTIVE DIRECTORY

34
- SCRIPTING GAMES

32
- SQL SERVER

20

Adding quotes around the domain name also allows it to be decoded with **PowerShell.exe** using the **EncodedCommand** parameter without having to encode it with a command such as **Write-Output:**

```
powershell.exe -encodedCommand JwBtAGkAawB1AGYAcgBvAGIAYgBpAG4AcwAuAGMAbwBtACcA
```

BATCH

```
mikefrobbins.com
```

The code shown in the previous example specifies the **NoProfile** parameter but it's not required.

μ

0 reactions



0 comments

DESIRED STATE CONFIGURATION18

AZURE13HYPER-V10

LINUX9VSCODE9

POWERSHELL SUMMIT8

GIT6IIS6

OFFICE 3656

PESTER6

AZURE POWERSHELL5

WINDOWS 105

ARCOLINUX4AST4

ALL CATEGORIES

## Tags

POWERSHELL340

ACTIVE DIRECTORY32

SCRIPTING GAMES30

SQL SERVER21DSC19

DESIRED STATE CONFIGURATION18

AZURE13HYPER-V11

VIDEO11PESTER10

LINUX9VSCODE9

AST8

POWERSHELL VERSION 58

GIT7IIS6

OFFICE 3656

POWERSHELL SCRIPT MODULE DESIGN6

ALL TAGS

