OBFUSCATION TECHNIQUES

by Howard Figueroa*

*Pentester of Open-Sec. He has security certifications like eJPT, and eWPT. He has experience performing infrastructure and web applications tests. He also performs penetration tests for ATMs. (https://www.linkedin.com/in/hwru/)

Obfuscation is a technique utilized in programming to hide the purpose or functionality of a code or a system. This is done in order to make it difficult for unauthorized parties to understand, modify or even analyze. Obfuscation is commonly used to protect the software and the intellectual property, as well as to prevent hijacking and inverse engineering

However, obfuscation does not guarantee an absolute security, as analysts with sufficient resources and abilities can, eventually, crack that code. In addition, obfuscation can also be counterproductive in terms of software compatibility and maintenance. Therefore, it is important to consider the pros and cons of obfuscation before deciding to use it.

Some of the most common obfuscation techniques are described below:

- **Variable Renaming:** This technique consists on changing the name of variables and functions to arbitrary names in order to not reveal their purpose. This process can make it more difficult to understand the code and identify the critical points.
- **Codification:** This process consists in writing the code in a different language or format to the original one, so that it can confuse the analysts. Codification can include the creation of a personalized programming language or the use of an automated codification tool.
- **Flow Control Obfuscation:** This technique consists of modifying the flow control structure of a program to make the comprehension of it harder. This can be achieved by removing the id and reordering the instructions, among other techniques.
- **Encryption:** This technique consists of encrypting critical parts of a code or data to hide them from the analyzers. This can be done by utilizing strong encryption algorithms and secret passwords.
- **Information Concealment:** This process consists in hiding critical information, like passwords, in the code or in the system to make it difficult to analyze.

WINDOWS COMMAND LINE OBFUSCATION TECHNIQUES TO AVOID THREAT DETECTIONS

The most experienced hackers are used to utilize the obfuscation techniques to avoid signature-based or blacklist-based detections and ensure the success of their attacks.

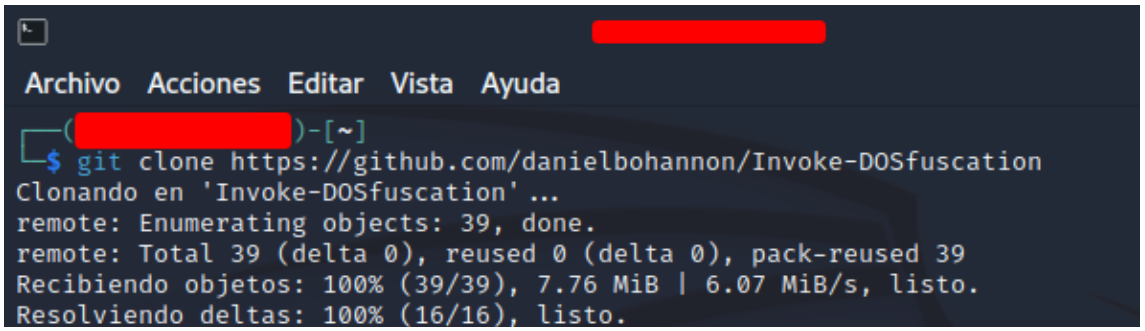
They will look for the way to pass without being notice, that 's why they can use the flexibility of the command line to succeed in evading detection. Simple techniques such as upper & lower case variability or full command line obfuscation can be used to break word-based detection and to hide original commands, respectively.

DOSFUSCATION

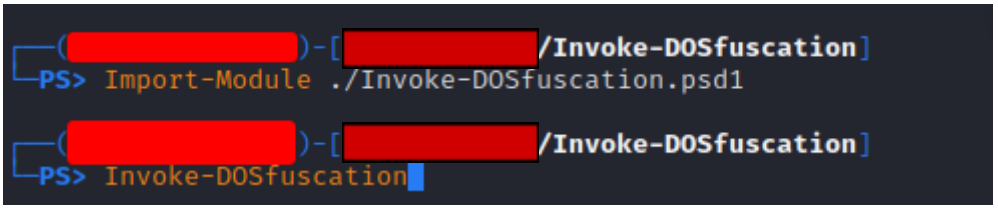
Invoke-DOSfuscation is a security tool from PowerShell that helps obfuscate scripts (from PowerShell) in order to avoid malware or trespassing detection. Here you have the follow-up-steps to install and use Invoke-DOSfuscation:

Download Link: https://github.com/danielbohannon/Invoke-DOSfuscation

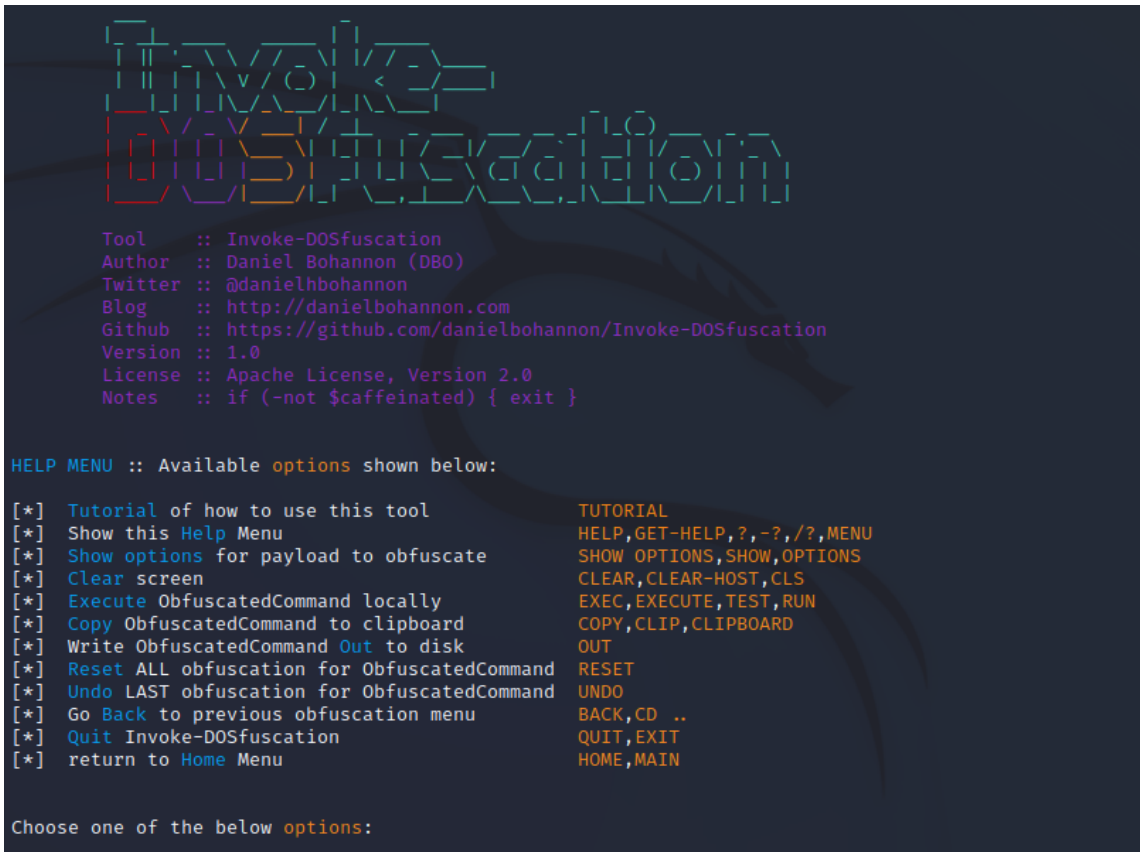
Installation: To install Invoke-DOSfuscation, you must have PowerShell 5.0 or a later version of it installed on your system. Install the tool from GitHub.



The next step is to charge the module to get access to a PowerShell module. After that, run the tool.



Invoke-DOSfuscation offers a variety of obfuscation options that can be used to personalize the obfuscation process. For example, you can specify different levels of obfuscation. disable some command execution and more.



To establish the command to be obfuscated, we gotta use the command **SET COMMAND** and then write the command CMD.

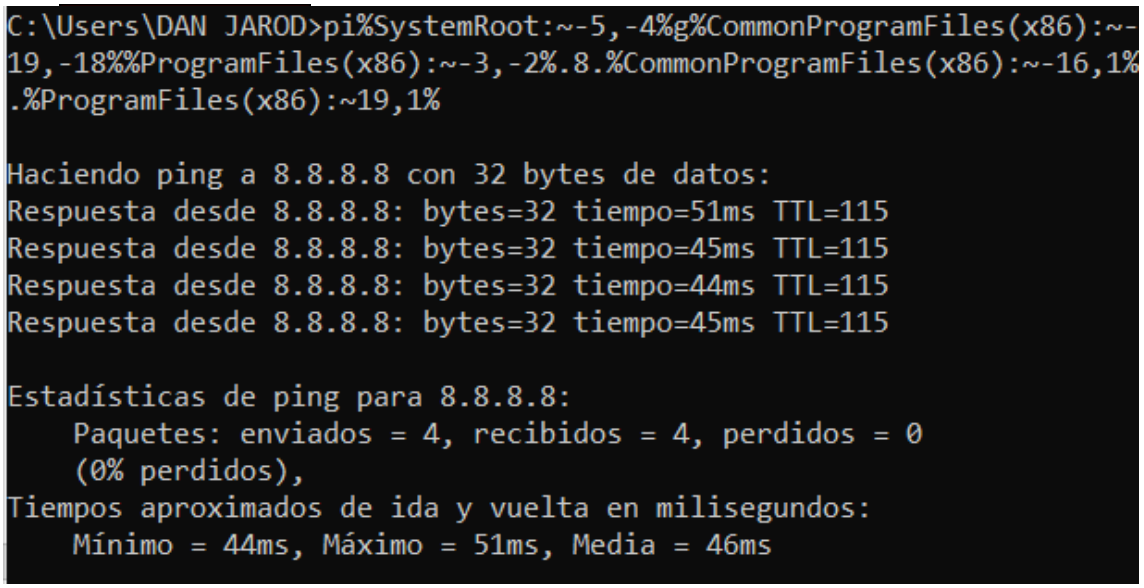
```
Invoke-DOSfuscation> SET COMMAND ping 8.8.8.8
```

Frem here, we can choose different obfuscation methods from basic to a high level. After selecting the options we want, we will get as a result a completely obfuscated command.

```
Successfully set Command:
ping 8.8.8.8


Choose one of the below options:

[*] BINARY      Obfuscated binary syntax for cmd.exe & powershell.exe
[*] ENCODING    Environment variable encoding
[*] PAYLOAD     Obfuscated payload via DOSfuscation


Invoke-DOSfuscation> ENCODING


Choose one of the below Encoding options to APPLY to current payload:

[*] ENCODING\1          Basic env var encoding
[*] ENCODING\2          Medium env var encoding
[*] ENCODING\3          Intense env var encoding


Invoke-DOSfuscation\Encoding> 2

Executed:
  CLI:  Encoding\2
  FULL: Out-EnvVarEncodedCommand -StringToEncode $Command -ObfuscationLevel 2 -MaintainCase

Result:
pi%SystemRoot:~-5,-4%g%CommonProgramFiles(x86):~-19,-18%%ProgramFiles(x86):~-3,-2%.8.%CommonProgramF
iles(x86):~-16,1%.%ProgramFiles(x86):~-19,1%
```

We can check that the functionality of the original command and the obfuscated one is the same.

```
C:\[redacted]>ping 8.8.8.8

Haciendo ping a 8.8.8.8 con 32 bytes de datos:
Respuesta desde 8.8.8.8: bytes=32 tiempo=45ms TTL=115
Respuesta desde 8.8.8.8: bytes=32 tiempo=45ms TTL=115
Respuesta desde 8.8.8.8: bytes=32 tiempo=45ms TTL=115
Respuesta desde 8.8.8.8: bytes=32 tiempo=44ms TTL=115

Estadísticas de ping para 8.8.8.8:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 44ms, Máximo = 45ms, Media = 44ms
```



It is important to keep in mind that obfuscation is not a magic solution for security and that the obfuscated scripts can still be detected by the security systems. It's better to be sure and combine the obfuscation with other security measures like the user's authentication, data encryption and more.