Module 2 – Exercise

Use Pyinstaller Library

1. Install Pyinstaller in your Windows 7 testing machine

2. Export your python script into EXE, your EXE should be :-

- One file only without any dependencies like DLL
- No console window should appear to your target machine when running the EXE file

3. Compare the file size with the one which we got from py2exe

4. When you run the EXE, does it ask for admin privilege? If yes, why?

5. Try to change the EXE file name, and then run it again, does it ask for admin privilege now? If no, why?

Module 2 – Exercise

Download Tools From Hacker Machine

Why we need to send tools to our target?

For post exploitation purposes, you will always need a way to transfer additional tools to the target machine like Windows Credentials Editor (wce.exe), Netcat and more.

Task

1. Update your client and server scripts where you can push a file from the attacker machine down to the target machine.

Hints

- ✓ Try to take an advantage of the 'transfer' function
- ✓ Add a new "if" statement like 'download'

Conditions

- No additional tunnels or sessions should be created – No FTP, SFTP, SCP or any other protocols are allowed.
- The transferred file should work as expected.

Module 2 – Exercise

Switching user-agent value

- ✓ Fire up Wireshark and monitor the HTTP GET/POST going back and forth
- ✓ Use the "follow TCP stream" feature on wireshark and write down the default value used in Requests user-agent
- ✓ Search on the web for user-agent values for a well known browsers like Chrome, Firefox
- ✓ Change the user agent on your HTTP client to make it looks like coming from Google chrome
- ✓ Run Wireshark again and verify the changes

Module 2

Persistence Outline

- Why Persistence?
- Make sure that your customer explicitly allows any modification on the target machine.

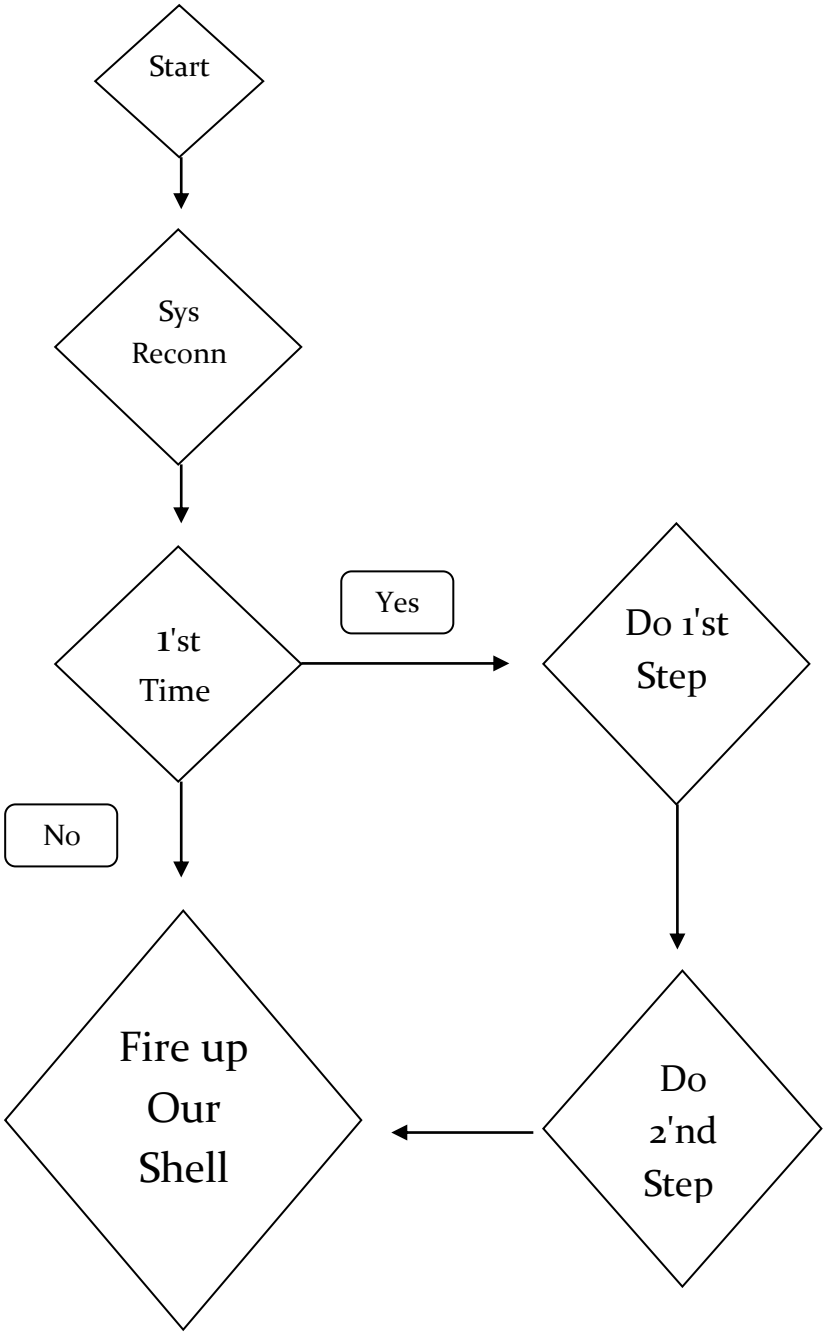
Action Plan

- First step: --- for first time we run on target machine
  - we need to make our shell copy itself into a different location - let's say in documents folder
  - Keep in mind in order to copy a file, you need to know the source path where file exists and the destination path (the documents folder in the user directory - C:\Users\<UserName>\Documents\ )

- Second step: -- for first time we run on target machine
  - we need to add a registry key pointing to our new location, so once the system boots up again our script will be executed

- Third step: -- each time we run on target machine
  - Once the computer gets rebooted we should NOT repeat the above two steps, instead, our reverse shell should get start

- System Reconnaissance
  - Since we don't know either the source or the destination, and each varies on each computer, we need to make our script a little intelligence to find output these parameters by itself.


Start
Sys
Reconn
1'st
Time
Yes
Do 1'st
Step
No
Fire up
Our
Shell
Do
2'nd
Step

Module 2 – Exercise

Make a Persistant Netcat Backdoor

- ✓ On your kali device, make a netcat **listener** on port 4444
- ✓ Repeat the same process which I did with putty.exe but this time use netcat.exe
- ✓ On your windows 7, when the system boots up, **initiate** a **reverse** netcat connection back to kali machine.

Hint

Check out the registry key value which used in the below article and do any necessary changes to netcat arguments to make the above scenario works.

https://www.offensive-security.com/metasploit-unleashed/persistent-netcat-backdoor/

Module 2 – Exercise

Make a Persistant TCP Reverse Shell

Add Persistence feature to our TCP reverse shell in a similar way with what we just did in HTTP reverse shell, **but**:-

- ✓ Use a different directory rather than C:\Users\<UserName>\Documents\
- ✓ When you copy the backdoor, make it hidden*
- ✓ Use a random name for the copied file.

Make sure to reflect those changes when creating the "path" and "destination" variables otherwise the registry value will point to wrong directory.

*making a file hidden is not a security feature but it may protect our backdoor from getting deleted by mistake.

Module 2 – Exercise

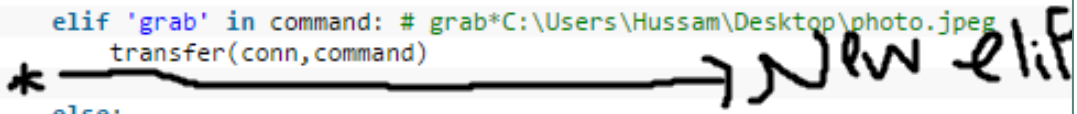
Overcome Empty String

In a previous video we have seen that when we hit enter key multiple times our shell breaks due to improper handling of empty string. In this exercise you have to fix this problem.

Hint

The simplest way to solve this issue is by adding a new line in our server code saying if the user input was empty string ' ' then we do nothing or we may send a trivial command like "whoami".

```
01. while True:
02.     command = raw_input("Shell> ")
03.     if 'terminate' in command:
04.         conn.send('terminate')
05.         conn.close() # close the connection with the host
06.         break
07.
08.     elif 'grab' in command: # grab*C:\Users\Hussam\Desktop\photo.jpeg
09.         transfer(conn,command)
10.
11.     else:
12.         conn.send(command) # send command
13.         print conn.recv(1024) # print the result that we got back
```



Module 2

Final Notes

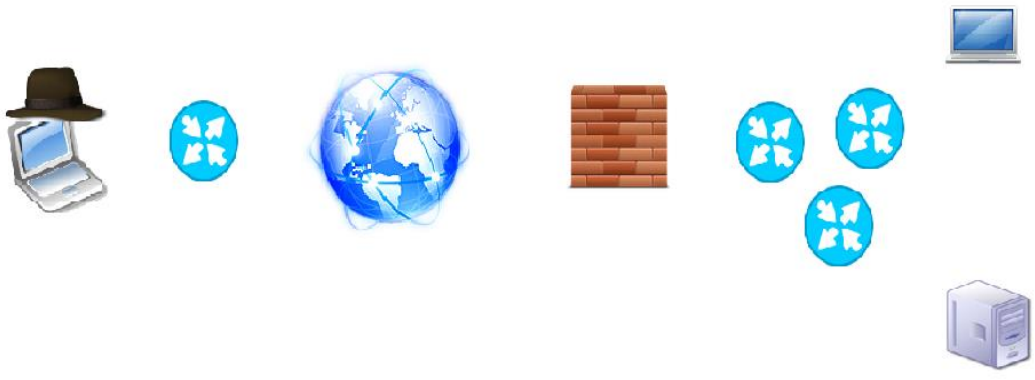
Don't Break Your Shell

This is a Shell NOT Terminal! We explained earlier that creating a shell is done via creating a subprocess then we pass the received argument into cmd.exe process. Some commands are designed to work on a terminal but not for shell since the shell will fail to handle the output properly.

For instance, "cls" and "clear" are commands which will clear a terminal screen; the shell will fail to handle those types of commands also, any interactive commands, will not work in shell such as telnet.

Python Expect

https://pexpect.readthedocs.org/en/latest/examples.html



Module 2

Countermeasures

- ✓ Securing the human from social engineering attacks
  - Not only boring slides and sessions, show them how the attacks happen on a high level (without explaining the coding part)
  - People forget, do these sessions on regular basis with enforcement from C-Level managers like CTO or CISO.

- ✓ Never blindly rely on
- Anti-Virus
- Sandbox
- Vmware

- ✓ Never use crack files, most likely it came embedded with backdoor!
- ✓ Never trust any exe/script unless you are 100% about the source - use md5 checksum to double check.
- ✓ Use DLP (Data Leaking Prevention)
- ✓ Install Host-Based Intrusion Detection System (HIDS) on all your endpoints, SOC analysts should actively raise a flag on any suspicious activities.
- ✓ If possible, limit what processes are allowed to run based on your business needs.
- ✓ Report any suspicious emails, files or behaviors immediately.