Shellshock Attack Lab

1 Overview

On September 24, 2014, a severe vulnerability in Bash was identified. Nicknamed Shellshock, this vulnerability can exploit many systems and be launched either remotely or from a local machine. The learning objective of this lab is for you to get a first-hand experience on this interesting attack and understand how it works.

2 Lab Tasks

2.1 Task 1: Attack `Set-UID` programs

The Bash program in Ubuntu 16.04 has already been patched, so it is no longer vulnerable to the Shellshock attack. For the purpose of this lab, we have installed a vulnerable version of Bash inside the /bin folder; its name is bash shellshock. We need to use this Bash in our task. Please run this vulnerable version of Bash like the following and then design an experiment to verify whether this Bash is vulnerable to the Shellshock attack or not.

```
$ /bin/bash_shellshock
```

Experiment with Bash functions Reproduce the example in slide 8 of the class slides. Don't forget to use `bash_shellshock` instead of `bash`

2.2 Task 2: Attack CGI programs

In this task, we will launch the Shellshock attack on a remote web server. Many web servers enable CGI, which is a standard method used to generate dynamic content on Web pages and Web applications. Many CGI programs are written using shell script. Therefore, before a CGI program is executed, the shell program will be invoked first, and such an invocation is triggered by a user from a remote computer.

Step 1: Set up the CGI Program. You can write a very simple CGI program (called `myprog.cgi`) like the following. It simply prints out `"Hello World"` using shell script.

```
#!/bin/bash_shellshock

echo "Content-type: text/plain"
echo
echo
echo "Hello World"
```

Please place the above CGI program in the `/usr/lib/cgi-bin` directory and set its permission to 755 (so it is executable). You need to use the root privilege to do these (using `sudo`), as the folder is only writable by the root. This folder is the default CGI directory for the Apache web server. If you want to change this setting, you can modify `/etc/apache2/sites-available/default`, which is the Apache configuration file.

To access this CGI program from the Web, you can either use a browser by typing the following URL: `http://localhost/cgi-bin/myprog.cgi`, or use the following command line program `curl` to do the same thing:

```
$ curl http://localhost/cgi-bin/myprog.cgi
```

In our setup, we run the Web server and the attack from the same computer, and that is why we use `localhost`. In real attacks, the server is running on a remote machine, and instead of using `localhost`, we use the hostname or the IP address of the server.

Step 2: Launch the Attack. After the above CGI program is set up, you can launch the Shellshock attack. The attack does not depend on what is in the CGI program, as it targets the Bash program, which is invoked first, before the CGI script is executed. Your goal is to launch the attack through the URL `http://localhost/cgi-bin/myprog.cgi`, such that you can achieve something that you cannot do as a remote user. For example, you can delete some file on the server, or fetch some file (that is not accessible to the attacker) from the server.

Please describe how your attack works.

Question: Can you pinpoint from the Bash source code `variables.c` where the vulnerability is. You just need to identify the line in the `initialize_shell_variables()` function (between Lines 308 and 369).

2.3 Task 3: Remote Attack

In this task we attacking a remote machine. Start another VM (you should make a clone if you have not yet) and label it Server

Task 3A. Repeat the attack in Task2 but on the server machine instead of the localhost.

Task 3B. The web server in the VM hosts a web application (CMS) called Elgg. When a web application connects to its back-end databases, it needs to provide login passwords. These passwords are usually hard-coded in the program or stored in a configuration file. For Elgg, the config file is located */var/www/SeedElgg/engine/settings.php*.
Use the shellshock attack in 3A to acquire the username and password.

Task 3C. The goal of this task is to use the Shellshock attack to gain a remote shell on the server machine. First make sure you can successfully gain a remote shell using the commands used in the slides.

2.4 Task 4: Questions

Instead of putting an extra shell command after a function definition, we put it at the beginning (see the following example). We then run Bash, which is vulnerable to the Shellshock attack.
Will the shell command *echo world* be executed?

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

For the Shellshock vulnerability to be exploitable, two conditions need to be satisfied. What are these two conditions?

You notice that a server machine at your work is vulnerable to a Shellshock attack. What do you do to fix the problem.

2.5 Task 5: Remote Attack Server Side Code

Download the php file on Canvas and place it in /var/www/ Attack the code remotely using the techniques above.

3 Submission

You need to submit a detailed lab report to describe what you have done and what you have observed, including screenshots and code snippets. You also need to provide explanation to the observations that are interesting or surprising. Your can earn extra credit points for extra efforts.