Secure programing assignment 2

CSE 5382-001

Submitted by:

Name: Anuraag Venkatapuram Sreenivas

Student id: 1001716458

2.1 Task 1: Experimenting with Bash Function

Text, letter

Description automatically generated

Observation: we see that when the vulnerable shell is used to execute the shell command the code echo “attack” would be executed in the shell while in the patched shell the echo “attack” is not executed

Task 2: Passing Data to Bash via Environment Variable

Task2.a:

Graphical user interface, text

Description automatically generated

Task2.b:

Using curl -v

Text

Description automatically generated

Using curl -v -A

Text

Description automatically generated

Using curl -v -e

Graphical user interface, text, application

Description automatically generated

Using curl -v -H

Graphical user interface, text, application

Description automatically generated

Observation: curl -A, -e, -H all 3 options can be used to inject malicious code into the server

Task 3: Launching the Shellshock Attack

Task3.A: Get the server to send back the content of the /etc/passwd file.

Graphical user interface, text, application, email

Description automatically generated

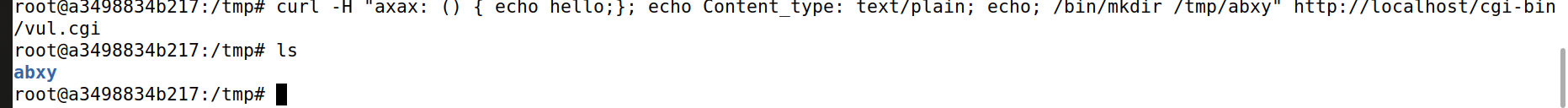
Observation: we use curl -A approach here and are able to steal the /etc/passwd file

Task 3.B: Get the server to tell you its process’ user ID. You can use the /bin/id command to print out the ID information.



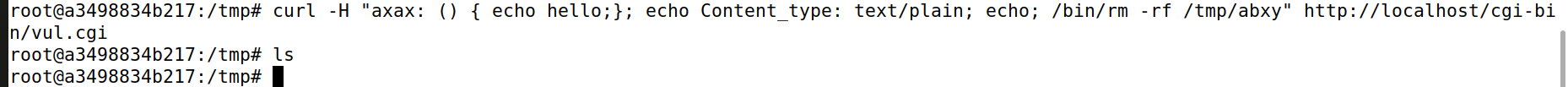
Observation: we use curl -e approach and are still able to inject code and get the output but we do not get the correct output as id command requires root privileges to display correct output and the current server we are using does not have root privileges as it uses user account other than root

Task 3.C: Get the server to create a file inside the /tmp folder. You need to get into the container to see whether the file is created or not, or use another Shellshock attack to list the /tmp folder.



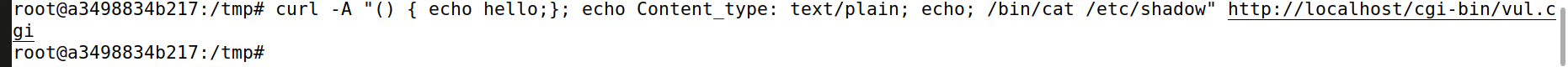
Observation: we are able to create the folder “abxy” in the tmp directory using curl -H

Task 3.D: Get the server to delete the file that you just created inside the /tmp folder.



Observation: we use the same approach as above to remover the created folder

Question 1: Will you be able to steal the content of the shadow file /etc/shadow from the server? Why or why not? The information obtained in Task 3.B should give you a clue.



content of /etc/shadow file can not be stolen because /etc/shadow file requires root privilege and the current server runs on user account other than root

Question2:

Text

Description automatically generated

This approach of trying to inject code payload using the GET method is not possible as the payload is attached at the end of the line and doesn’t get executed and the browser gives out error that it could not understand the request as the request is not a clean payload or address (you also don’t reach the website)

Task 4: Getting a Reverse Shell via Shellshock Attack:Graphical user interface, text, application

Description automatically generated

Graphical user interface, text

Description automatically generated

Table

Description automatically generated

Observation: we have got a reverse shell on the seed machine for the container server that has all the experimental data and cgi programs. Using the reverse shell, we are also able to use most commands except the commands that need root privileges

Task 5: Using the Patched Bash:

Text

Description automatically generated

Task5Sub : Redoing task3 using /bin/bash

A picture containing chart

Description automatically generated

Observation: using the patched bash we are unable to get the desired output as the shellshock vulnerability is already been patched in bash.