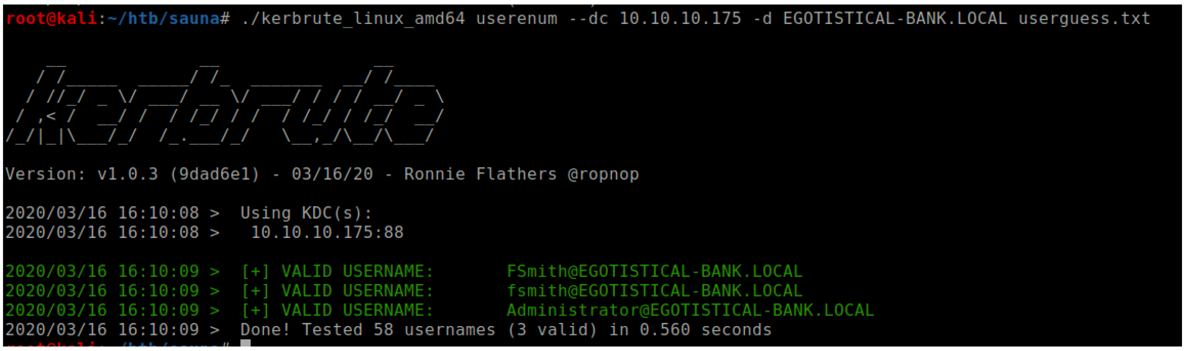
Kerberos

Kerbrute : user enumeration and bruteforcing

user enumeration
./kerbrute_linux_amd64 userenum --dc 10.10.10.175 -d EGOTISTICAL-BANK.LOCAL userlist.txt



```
root@kali:~/htb/sauna# ./kerbrute_linux_amd64 userenum --dc 10.10.10.175 -d EGOTISTICAL-BANK.LOCAL userguess.txt

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

Version: v1.0.3 (9dad6e1) - 03/16/20 - Ronnie Flathers @ropnop

2020/03/16 16:10:08 >  Using KDC(s):
2020/03/16 16:10:08 >   10.10.10.175:88

2020/03/16 16:10:09 >  [+] VALID USERNAME:       FSmith@EGOTISTICAL-BANK.LOCAL
2020/03/16 16:10:09 >  [+] VALID USERNAME:       fsmith@EGOTISTICAL-BANK.LOCAL
2020/03/16 16:10:09 >  [+] VALID USERNAME:       Administrator@EGOTISTICAL-BANK.LOCAL
2020/03/16 16:10:09 >  Done! Tested 58 usernames (3 valid) in 0.560 seconds
```

password bruteforcing
./kerbrute_linux_amd64 bruteuser --dc 10.10.10.175 -d <domain> rockyou.txt <username>

Notes

https://cyberx.tech/kerberos-authentication/

How does Kerberos authentication work?

Kerberos is a network authentication protocol that works by using secret key cryptography. Clients athenticate with a Key Distribution Center and get temporary keys to access locations on the network. This allows for strong and secure authentication without transmitting passwords.

The Kerberos ticket exchange is built around two main principles:
First, strong mutual authentication and second, restricted access.
Let's look at the authentication steps:

1: Client Authentication Request

The client sends an authenticator that includes the date and time to the Kerberos Key Distribution Center.
Part of the message is plain text and the other part is encrypted.
The encrypted part of the authenticator is encrypted with the client's password.
Note how the password is not transmitted.
This is because the developers of the network authentication system wanted it to be strong enough to operate in a hostile environment. That is One where attackers are present and sniffing communications.

2: KDC checks the client's credentials

The Key Distribution Center – the kerberos server – also the domain controller in Active Directory validates that the user is who they claim to be.
It does this by decrypting the authenticator message that the client sent.
Since the KDC has all users' passwords stored securely in its database, it attempts to decrypt the authentication message from the client.
If it can, it assumes that the user is who they claim to be as only they should know their password.
Once the client is authenticated, the authenticator is discarded. It's no longer needed.

3: The KDC creates a ticket

At this point in the process, the key distribution center creates a ticket that it can give to the client.
This Ticket Granting Ticket (TGT) is what the client uses henceforth to make access requests.
The KDC encrypts the TGT with a password that only the server knows.
Why?
Because no one else needs to be able to see the contents.
It's for the server to keep track of the client.
So, the KDC sends the TGT to the client.
The TGT will be stored in the Kerberos tray in RAM so that it is volatile.
Should the system crash or go down, the TGT is not stored anywhere.
TGTs expire after a set period – usually eight hours.

4: Client uses TGT to request access

When the client needs to access the file server in our example, it checks the Kerberos tray and since it doesn't have one, it makes a request.

The client sends a copy of the TGT to the key distribution centers and requests access to the file server.
When the KDC receives the new request, it knows the user is already authenticated so it doesn't need to do that again.
It just checks if it can decrypt the TGT using the password that it remembers encrypting the TGT with.
If it can, it must be the same one that the KDC sent earlier.

5: The KDC creates a ticket for the file server

Now, the KDC will create a ticket that the client can use to access the file server.
And since the KDC knows the file server's password, it encrypts the ticket with that credential.
Access instructions for the client are included in the ticket.
That way the file server knows what the client can access.

Note that the client never sees the contents of these tickets.
It just stores and uses them as necessary.
That's one of the reasons that Kerberos authentication works so well.
The new ticket for the file server is then given back to the client to store in the Kerberos tray.
For the next eight hours, or while the file server ticket is valid, whenever the client needs to access a file, it sends the file server its ticket.

6: The client uses the file ticket to authenticate

From this point forward, when the client needs to access a file on the file server, it sends a copy of the ticket.
The file server can decrypt it, verify the user, and grant them the appropriate access.

Remember, inside that ticket are all of the instructions for the client's groups and access.

Kerberos attacks

https://www.tarlogic.com/en/blog/how-to-attack-kerberos/
https://www.cyberark.com/blog/kerberos-attacks-what-you-need-to-know/
https://blog.stealthbits.com/complete-domain-compromise-with-golden-tickets/

The most powerful service account in any Active Directory environment: the KRBTGT account. By obtaining the password hash for this account, an attacker is able to compromise every account within Active Directory, giving them unlimited and virtually undetectable access to any system connected to AD.