Kerberos (I): How does Kerberos work? – Theory

tarlogic.com/blog/how-kerberos-works

March 20, 2019

The objective of this series of posts is to clarify how Kerberos works, more than just introduce the attacks. This due to the fact that in many occasions it is not clear why some techniques works or not. Having this knowledge allows to know when to use any of those attacks in a pentest.

Therefore, after a long journey of diving into the documentation and several posts about the topic, we've tried to write in this post all the important details which an auditor should know in order to understand how take advantage of Kerberos protocol.

In this first post only basic functionality will be discussed. In later posts it will see how perform the attacks and how the more complex aspects works, as delegation.

If you have any doubt about the topic which it is not well explained, do not be afraid on leave a comment or question about it. Now, onto the topic.

What is Kerberos?

Firstly, Kerberos is an authentication protocol, not authorization. In other words, it allows to identify each user, who provides a secret password, however, it does not validates to which resources or services can this user access.

Kerberos is used in Active Directory. In this platform, Kerberos provides information about the privileges of each user, but it is responsibility of each service to determine if the user has access to its resources.

Kerberos items

In this section several components of Kerberos environment will be studied.

Transport layer

Kerberos uses either UDP or TCP as transport protocol, which sends data in cleartext. Due to this Kerberos is responsible for providing encryption.

Ports used by Kerberos are UDP/88 and TCP/88, which should be listen in KDC (explained in next section).

Agents

Several agents work together to provide authentication in Kerberos. These are the following:

- **Client or user** who wants to access to the service.
- **AP** (Application Server) which offers the service required by the user.
- **KDC** (Key Distribution Center), the main service of Kerberos, responsible of issuing the tickets, installed on the DC (Domain Controller). It is supported by the **AS** (Authentication Service), which issues the TGTs.

Encryption keys

There are several structures handled by Kerberos, as tickets. Many of those structures are encrypted or signed in order to prevent being tampered by third parties. These keys are the following:

- **KDC or krbtgt key** which is derivate from krbtgt account NTLM hash.
- **User key** which is derivate from user NTLM hash.
- **Service key** which is derivate from the NTLM hash of service owner, which can be an user or computer account.
- **Session key** which is negotiated between the user and KDC.
- **Service session key** to be use between user and service.

Tickets

The main structures handled by Kerberos are the tickets. These tickets are delivered to the users in order to be used by them to perform several actions in the Kerberos realm. There are 2 types:

- The **TGS** (Ticket Granting Service) is the ticket which user can use to authenticate against a service. It is encrypted with the service key.
- The **TGT** (Ticket Granting Ticket) is the ticket presented to the KDC to request for TGSs. It is encrypted with the KDC key.

PAC

The **PAC** (Privilege Attribute Certificate) is an structure included in almost every ticket. This structure contains the privileges of the user and it is signed with the KDC key.

It is possible to services to verify the PAC by comunicating with the KDC, although this does not happens often. Nevertheless, the PAC verification consists of checking only its signature, without inspecting if privileges inside of PAC are correct.

Furthermore, a client can avoid the inclusion of the PAC inside the ticket by specifying it in *KERB-PA-PAC-REQUEST* field of ticket request.

Messages

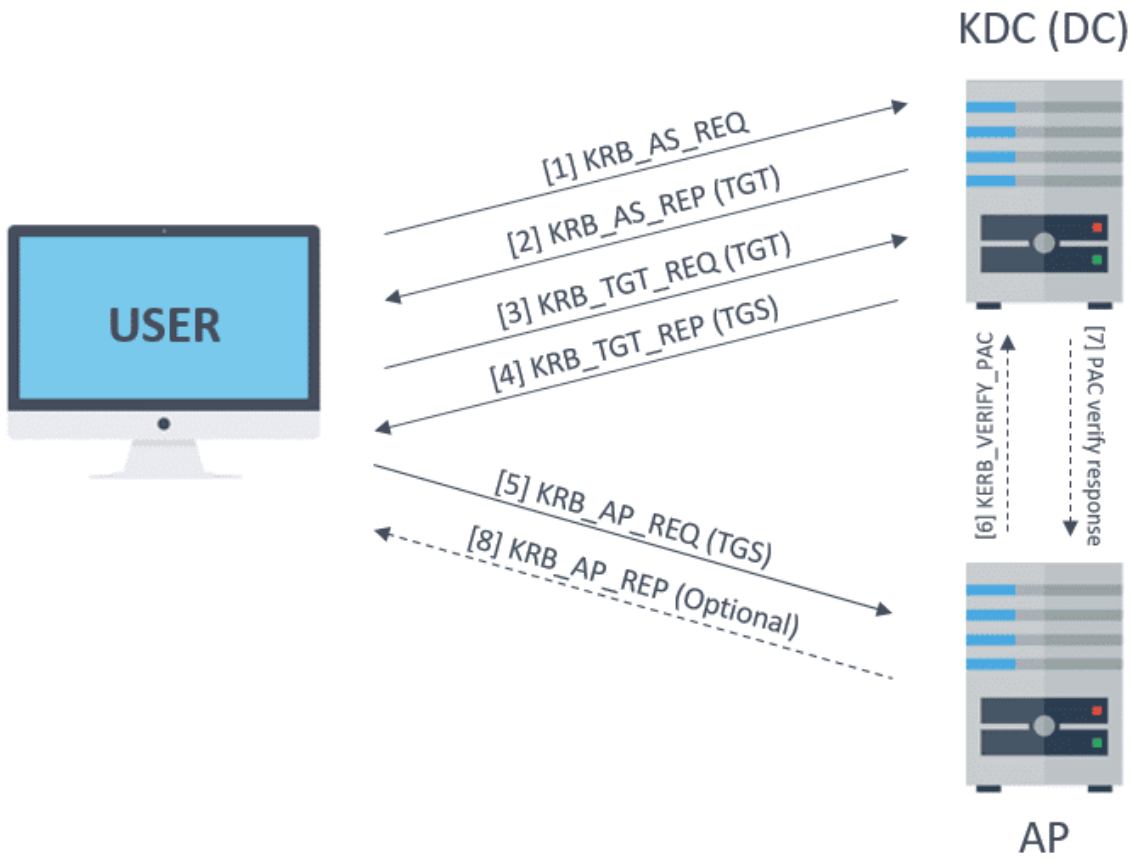
Kerberos uses differents kinds of messages. The most interesting are the following:

- **KRB_AS_REQ**: Used to request the TGT to KDC.
- **KRB_AS_REP**: Used to deliver the TGT by KDC.

- **KRB_TGS_REQ**: Used to request the TGS to KDC, using the TGT.
- **KRB_TGS_REP**: Used to deliver the TGS by KDC.
- **KRB_AP_REQ**: Used to authenticate a user against a service, using the TGS.
- **KRB_AP_REP**: (Optional) Used by service to identify itself against the user.
- **KRB_ERROR**: Message to comunicate error conditions.

Additionally, even if it is not part of Kerberos, but NRPC, the AP optionally could use the **KERB_VERIFY_PAC_REQUEST** message to send to KDC the signature of PAC, and verify if it is correct.

Below is shown a summary of message sequency to perform authentication:



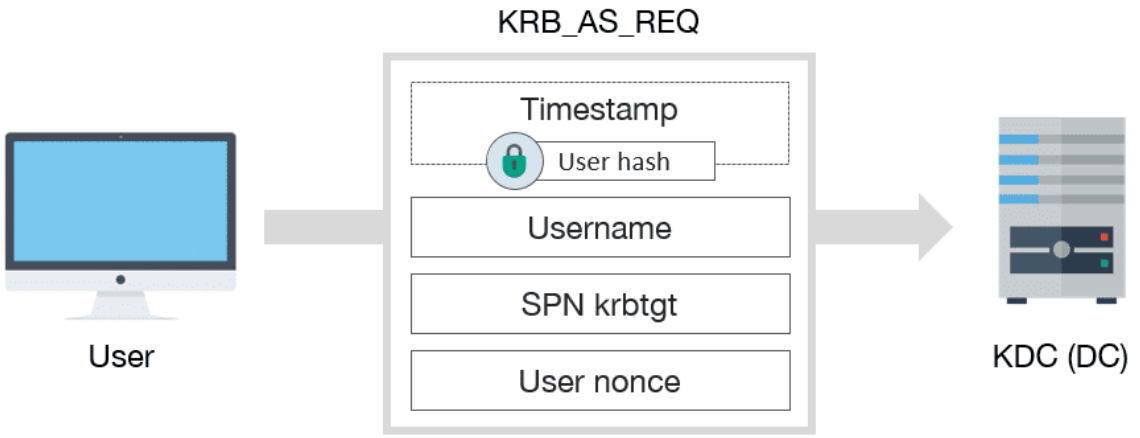
Kerberos messages summary

Authentication process

In this section, the sequency of messages to perform authentication will be studied, starting from a user without tickets, up to being authenticated against the desired service.

KRB_AS_REQ

Firstly, user must get a TGT from KDC. To achieve this, a KRB_AS_REQ must be sent:



KRB_AS_REQ schema message

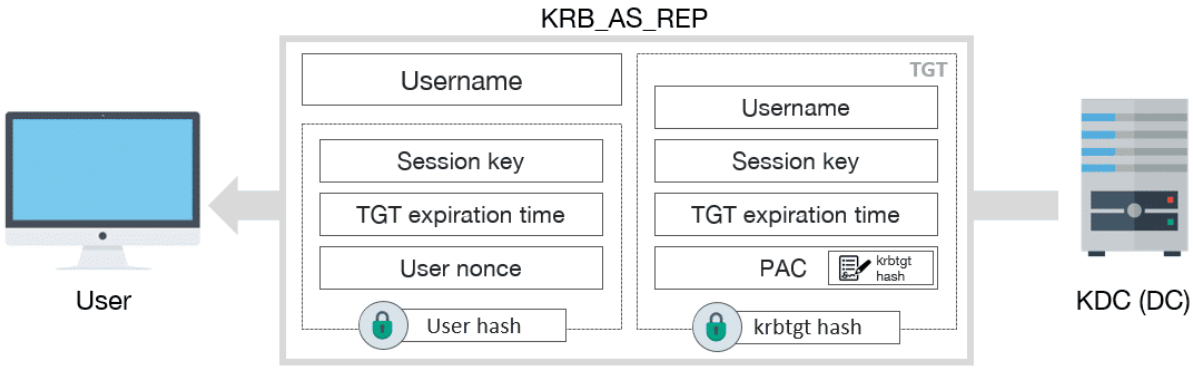
KRB_AS_REQ has, among others, the following fields:

- A encrypted **timestamp** with client key, to authenticate user and prevent replay attacks
- **Username** of authenticated user
- The service **SPN** associated with **krbtgt** account
- A **Nonce** generated by the user

Note: the encrypted timestamp is only necessary if user requires preauthentication, which is common, except if *DONT_REQ_PREAUTH* flag is set in user account.

KRB_AS_REP

After receiving the request, the KDC verifies the user identity by decrypting the timestamp. If the message is correct, then it must respond with a *KRB_AS_REP*:



KRB_AS_REP schema message

KRB_AS_REP includes the next information:

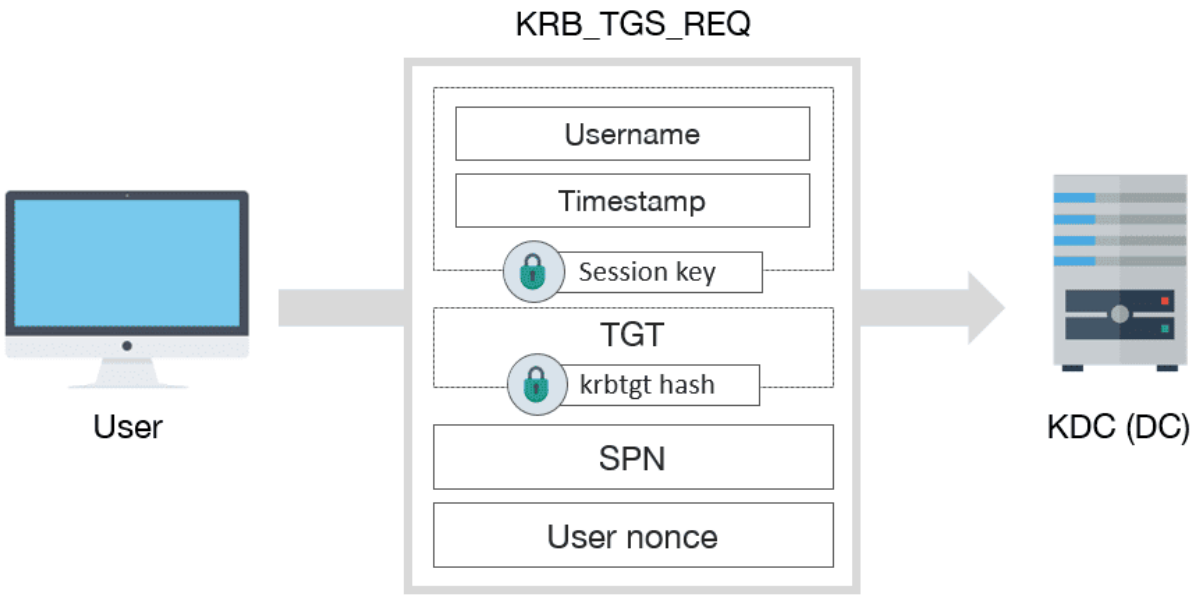
- **Username**

- **TGT**, which includes:
  - **Username**
  - **Session key**
  - **Expiration date** of TGT
  - **PAC** with user privileges, signed by KDC
- Some **encrypted data** with user key, which includes:
  - **Session key**
  - **Expiration date** of TGT
  - User **nonce**, to prevent replay attacks

Once finished, user already has the TGT, which can be used to request TGSs, and afterwards access to the services.

KRB_TGS_REQ

In order to request a TGS, a *KRB_TGS_REQ* message must be sent to KDC:



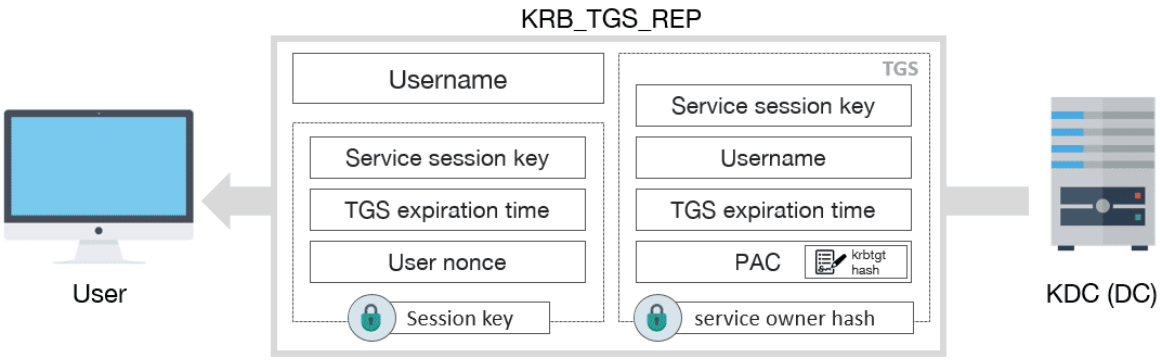
KRB_TGS_REQ schema message

KRB_TGS_REQ includes:

- **Encrypted data** with session key:
  - **Username**
  - **Timestamp**
- **TGT**
- **SPN** of requested service
- **Nonce** generated by user

KRB_TGS_REP

After receiving the *KRB_TGS_REQ* message, the KDC returns a TGS inside of *KRB_TGS_REP*:



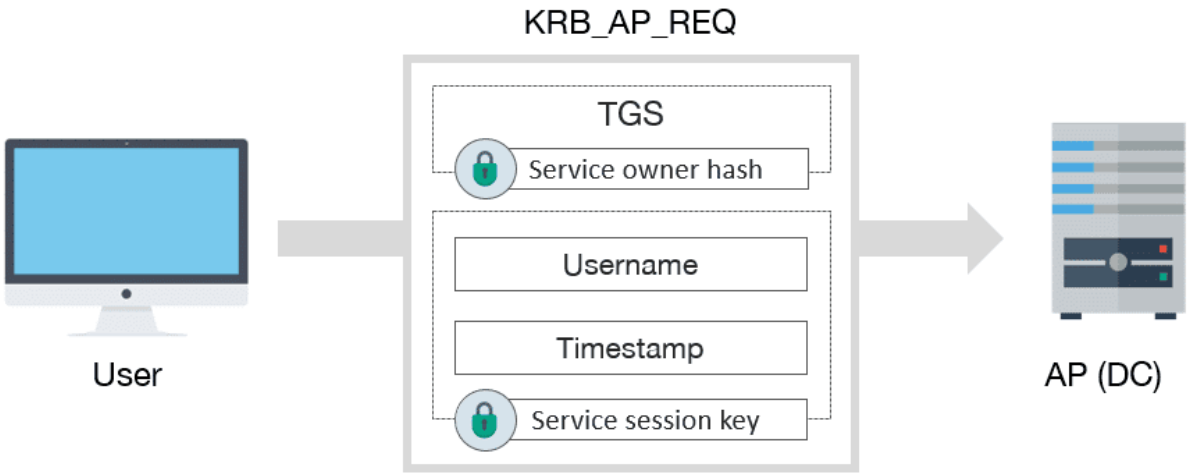
KRB_TGS_REP schema message

KRB_TGS_REP includes:

- **Username**
- **TGS**, which contains:
  - **Service session key**
  - **Username**
  - **Expiration date** of TGS
  - **PAC** with user privileges, signed by KDC
- **Encrypted data** with session key:
  - **Service session key**
  - **Expiration date** of TGS
  - User **nonce**, to prevent replay attacks

KRB_AP_REQ

To finish, if everything went well, the user already has a valid TGS to interact with service. In order to use it, user must send to the AP a *KRB_AP_REQ* message:



KRB_AP_REQ schema message

KRB_AP_REQ includes:

- **TGS**

- **Encrypted data** with service session key:
  - **Username**
  - **Timestamp**, to avoid replay attacks

After that, if user privileges are rigth, this can access to service. If is the case, which not usually happens, the AP will verify the PAC against the KDC. And also, if mutual authentication is needed it will respond to user with a *KRB_AP_REP* message.

Attacks

Based on previous explained authentication process the attacks oriented to compromise Active Directory will be explained in this section.

Overpass The Hash/Pass The Key (PTK)

The popular Pass The Hash (PTH) attack consist in using the user hash to impersonate the specific user. In the context of Kerberos this is known as Overpass The Hash o Pass The Key.

If an attacker gets the hash of any user, he could impersonate him against the KDC and then gain access to several services.

User hashes can be extracted from SAM files in workstations or NTDS.DIT file of DCs, as well as from the lsass process memory (by using Mimikatz) where it is also possible to find cleartext passwords.

Pass The Ticket (PTT)

Pass The Ticket technique is about getting an user ticket and use it to impersonate that user. However, besides the ticket, it is necessary obtain the session key too in order to use the ticket.

It is possible obtain the ticket performing a Man-In-The-Middle attack, due to the fact that Kerberos is sent over TCP or UDP. However, this techniques does not allow get access to session key.

An alternative is getting the ticket from lsass process memory, where also reside the session key. This procediment could be performed with Mimikatz.

It is better to obtain a TGT, due to TGS only can be used against one service. Also, it should be taken into account that the lifetime of tickets is 10 hours, after that they are unusable.

Golden Ticket and Silver Ticket

The objective of Golden Ticket is to build a TGT. In this regard, it is necessary to obtain the NTLM hash of krbtgt account. Once that is obtained, a TGT with custom user and privileges can be built.

Moreover, even if user changes his password, the ticket still will be valid. The TGT only can be invalidate if this expires or krbtgt account changes its password.

Silver Ticket is similar, however, the built ticket is a TGS this time. In this case the service key is required, which is derived from service owner account. Nevertheless, it is not possible to sign correctly the PAC without krbtgt key. Therefore, if the service verifies the PAC, then this technique will not work.

Kerberoasting

Kerberoasting is a technique which takes advantage of TGS to crack the user accounts passwords offline.

As seen above, TGS comes encrypted with service key, which is derived from service owner account NTLM hash. Usually the owners of services are the computers in which the services are being executed. However, the computer passwords are very complex, thus, it is not useful to try to crack those. This also happens in case of krbtgt account, therefore, TGT is not crackable neither.

All the same, on some occasions the owner of service is a normal user account. In these cases it is more feasible to crack their passwords. Moreover, this sort of accounts normally have very juicy privileges. Additionally, to get a TGS for any service only a normal domain account is needed, due to Kerberos not perform authorization checks.

ASREPRoast

ASREPRoast is similar to Kerberoasting, that also pursues the accounts passwords cracking.

If the attribute *DONT_REQ_PREAUTH* is set in a user account, then it is possible to built a KRB_AS_REQ message without specifying its password.

After that, the KDC will respond with a *KRB_AS_REP* message, which will contain some information encrypted with the user key. Thus, this message can be used to crack the user password.

Conclusion

In this first post the Kerberos authentication process has been studied and the attacks has been also introduced. The following posts will show how to perform these attacks in a practical way and also how delegation works. I really hope that this post it helps to understand some of the more abstract concepts of Kerberos.

References

- Kerberos v5 RFC: https://tools.ietf.org/html/rfc4120
- [MS-KILE] – Kerberos extension: https://msdn.microsoft.com/en-us/library/cc233855.aspx

- [MS-APDS] – Authentication Protocol Domain Support: https://msdn.microsoft.com/en-us/library/cc223948.aspx
- Mimikatz and Active Directory Kerberos Attacks: https://adsecurity.org/?p=556
- Explain like I'm 5: Kerberos: https://www.roguelynn.com/words/explain-like-im-5-kerberos/
- Kerberos & KRBTGT: https://adsecurity.org/?p=483
- Mastering Windows Network Forensics and Investigation, 2 Edition . Autores: S. Anson , S. Bunting, R. Johnson y S. Pearson. Editorial Sibex.
- Active Directory , 5 Edition. Autores: B. Desmond, J. Richards, R. Allen y A.G. Lowe-Norris
- Service Principal Names: https://msdn.microsoft.com/en-us/library/ms677949(v=vs.85).aspx
- Niveles funcionales de Active Directory: https://technet.microsoft.com/en-us/library/dbf0cdec-d72f-4ba3-bc7a-46410e02abb0
- OverPass The Hash – Gentilkiwi Blog: https://blog.gentilkiwi.com/securite/mimikatz/overpass-the-hash
- Pass The Ticket – Gentilkiwi Blog: https://blog.gentilkiwi.com/securite/mimikatz/pass-the-ticket-kerberos
- Golden Ticket – Gentilkiwi Blog: https://blog.gentilkiwi.com/securite/mimikatz/golden-ticket-kerberos
- Mimikatz Golden Ticket Walkthrough: https://www.beneaththewaves.net/Projects/Mimikatz_20_-_Golden_Ticket_Walkthrough.html
- Attacking Kerberos: Kicking the Guard Dog of Hades: https://files.sans.org/summit/hackfest2014/PDFs/Kicking%20the%20Guard%20Dog%20of%20Hades%20-%20Attacking%20Microsoft%20Kerberos%20%20-%20Tim%20Medin(1).pdf
- Kerberoasting – Part 1: https://room362.com/post/2016/kerberoast-pt1/
- Kerberoasting – Part 2: https://room362.com/post/2016/kerberoast-pt2/
- Roasting AS-REPs: https://www.harmj0y.net/blog/activedirectory/roasting-as-reps/
- PAC Validation: https://passing-the-hash.blogspot.com.es/2014/09/pac-validation-20-minute-rule-and.html
- Understanding PAC Validation: https://blogs.msdn.microsoft.com/openspecification/2009/04/24/understanding-microsoft-kerberos-pac-validation/
- Reset the krbtgt acoount password/keys: https://gallery.technet.microsoft.com/Reset-the-krbtgt-account-581a9e51
- Mitigating Pass-the-Hash (PtH) Attacks and Other Credential Theft: https://www.microsoft.com/en-us/download/details.aspx?id=36036
- Fun with LDAP, Kerberos (and MSRPC) in AD Environments: https://speakerdeck.com/ropnop/fun-with-ldap-kerberos-and-msrpc-in-ad-environments?slide=58

Discover our work and cybersecurity services at www.tarlogic.com