Kerberos 101 – From Zero to Hero

blog.ahasayen.com/kerberos-101

Ammar Hasayen

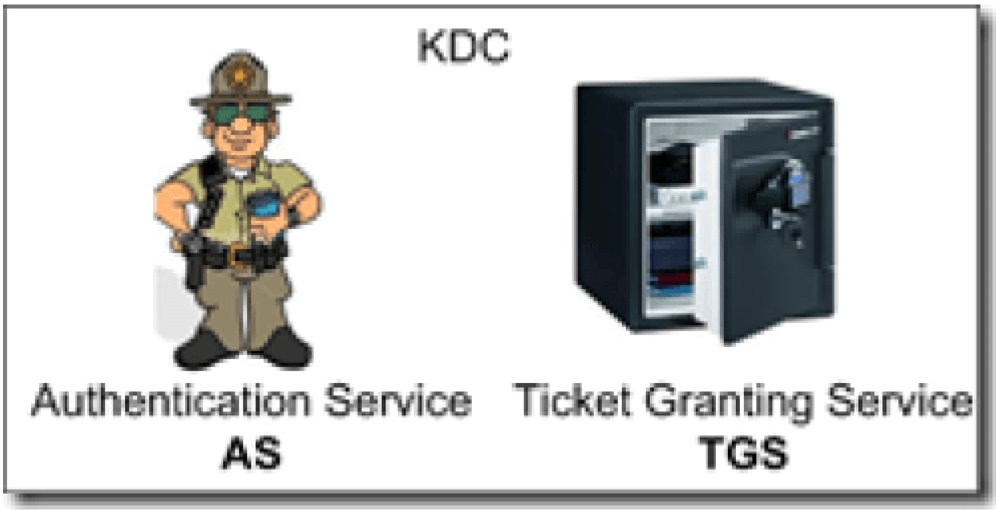
Kerberos 101 Introduction

In this Kerberos 101 post, we will talk about the basic concept of Kerbeors and how it works behind the scenes. Although Kerberos might seem like black magic to many system administrators, it is the main authentication protocol in Active Directory environment. It is used every time we log to domain joined machines, as well as when accessing resources such as share files and applications.

Kerberos is an authentication protocol and it doesn't perform any kind of authorization. Instead, Kerberos will provide the infrastructure (something called PAC) so that kerborized applications and services can decide by themselves whether the user is authorized and maybe creating a token for him.

I will start by naming an AD domain controller as KDC (Key Destruction Center) and defining two services running on each DC:

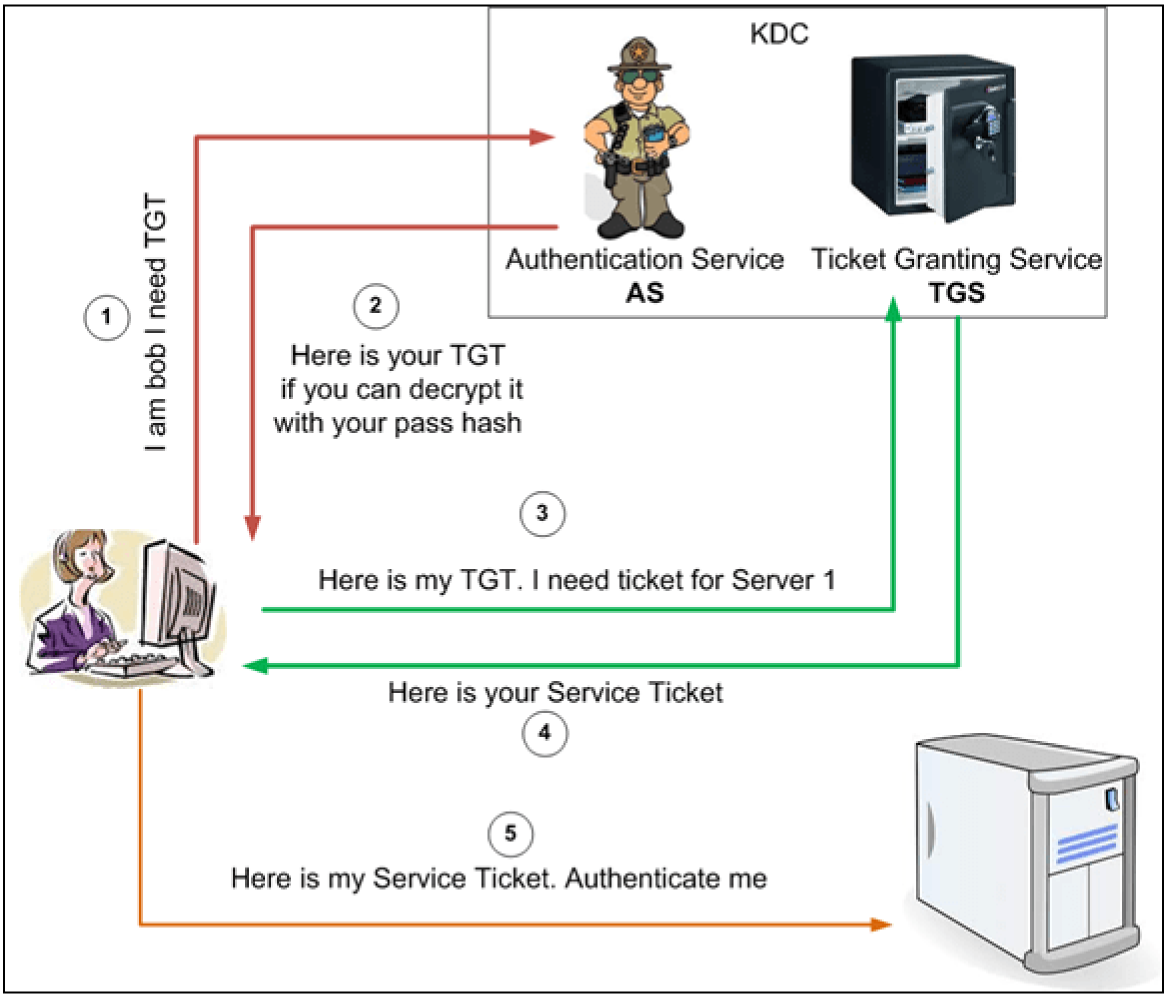
- **Authentication Service (AS)**: used to authenticate users.
- **Ticket Granting Service (TGS)**: used to enable users to gain access to resources.



Kerberos and the three parties

The Kerberos protocol name is based on the three- headed dog figure from Greek mythology known as Kerberos. The three heads of Kerberos comprise:

- Key Distribution Center (KDC).
- The client user.
- The server with the desired service to access.

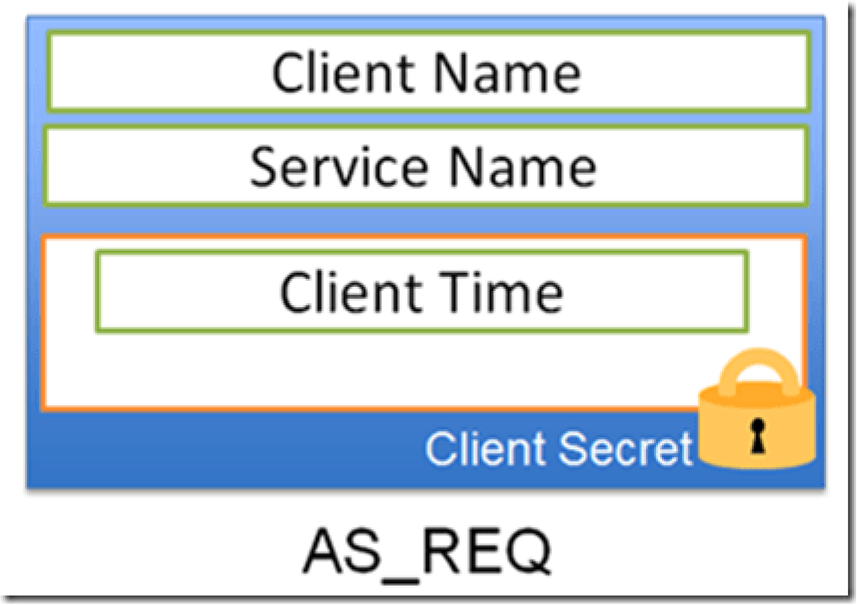


It is important to notice that AS and TGS are separated, which means that a user can obtain a TGT from one KDC and present it to the TGS on another KDC (maybe on another trusted domain).

Authentication Service (AS) Deep Drive

Step 1 : Request to get TGT (AS_REQ)

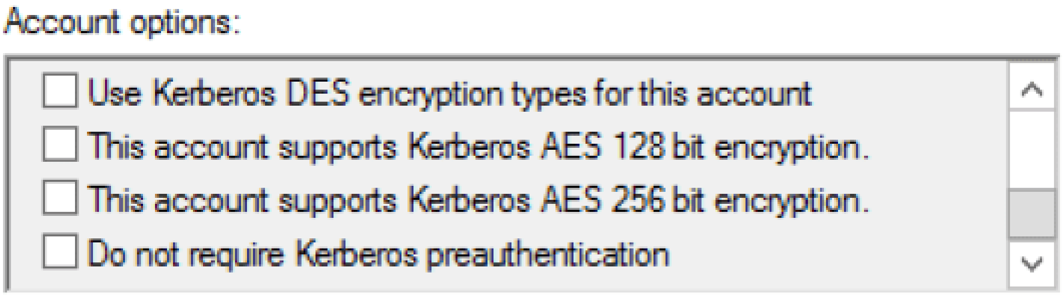
When you sit down at your workstation, and press Ctrl+Alt+Del to log on and enter your credentials, your machine begins the process of authentication.



The request will include plain text data that includes:

- Client Name: user UPN or legacy SAMAccountName.
- Service Name: which is the (krbtgt) service of the user's domain.

This is the Kerberos RFC standard for AS_REQ. It is plain text of data including the user name and the Kerberos service. It is a claim sent as plain text.
This request is subject to replay attach, so Microsoft added another field, which is called (Kerberos Pre-Authentication). This is simply the client time encrypted with the user password hash. If the encrypted time stamp isn't within five minutes of the current time, the request is rejected.
You can disable Kerberos Pre-Authentication for any user by un-checking the corresponding checkbox on his user property page in AD:



Account options:

☐ Use Kerberos DES encryption types for this account
☐ This account supports Kerberos AES 128 bit encryption.
☐ This account supports Kerberos AES 256 bit encryption.
☐ Do not require Kerberos preauthentication

Step 2: Getting the TGT (AS_REP)

When the KDC receives the AS_REQ, it will do the following:

- Tries to decrypt the timestamp using a copy of the user password hash.
- If this operation fails, then an error is returned to the client and the request doesn't proceed any further.
- If the decryption is successful and the timestamp is within acceptable limits, the KDC returns an AS_REP (Authentication Service Reply)

Note: if the user didn’t send the encrypted time stamp as part of the AS_REQ (if the Kerberos pre-authentication is disabled), the KDC will also return back the same AS_REP. The catch here is only the legitimate user can decrypt and understand the content of the AS_REP.
The AS_REP is encrypted with the user password hash, and contains:

- Session key: will be used to encrypt future communication with the KDC.
- Lifetime: to indicate when the user need to renew its TGT.
- The actual <TGT > Encrypted by the KDC Secret:
    - Session key: same one that is mentioned above.
    - Token information or PAC.
    - Lifetime: so the KDC can know when this TGT is expired.

At this point, the user’s machine will cache the TGT and session key for the lifetime of the TGT. By default, TGTs issued by AD KDCs expire after ten hours.

KDC Secret

When the first domain controller in a domain is created, a user named krbtgt is created with a random password. All writable DCs in that domain share the password of that user and they use it to encrypt the content of TGT. That means that a TGT issued by one domain controller can be decrypted and used on another domain controller to get a service ticket.
It is also important to notice that RODCs have their own krbtgt passwords and they don’t have knowledge of the domain krbtgt password. On the other hand, writable domain controllers have read access to the krbtgt of all RODCs.

PAC

Part of the TGT is the user token information or PAC. PAC is Privilege Attribute Certificate and it is also considered the (Authorization Data). It contains user groups membership and other user rights and info (like logon scripts path, home folder directory, etc).

Ticket Granting Service (TGS) Deep Drive

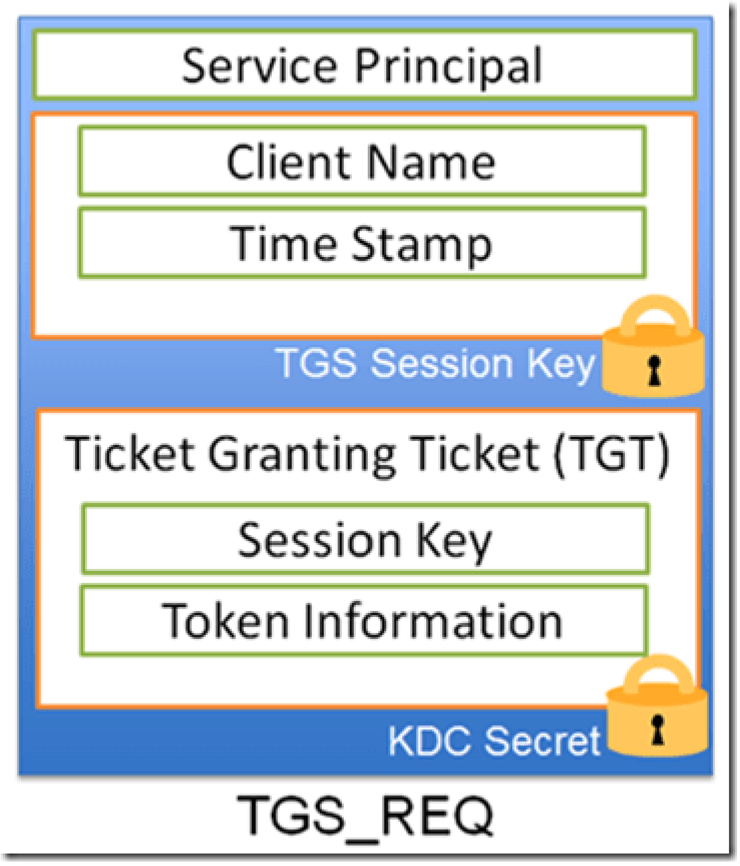
Step 1 : Request Service Ticket (TGS_REQ)

To access a service, you need to present a service ticket to the service. The first step is to identify the service principal name (SPN) of the service you want to access. Your machine or the application involved (e.g., Internet Explorer) is responsible for forming the SPN.

TGS_REQ contains the following info :

- Service Principal Name .
- The same TGT encrypted with the KDC Secret.

- Client name (username) and time stamp encrypted with the session key the client received as part of the AS_REP earlier.This information is again used to prevent replay attacks whereby an attacker reuses a request message.



Step 2 : Get Ticket (TGS_REP)

When the KDC receives a TGS_REQ message, if a single entry for the SPN is specified, the time stamp is within range, and the TGT is valid (and unexpired), the client will receive a service ticket as part of a TGS_REP message.

TGS Reply contains the following information:

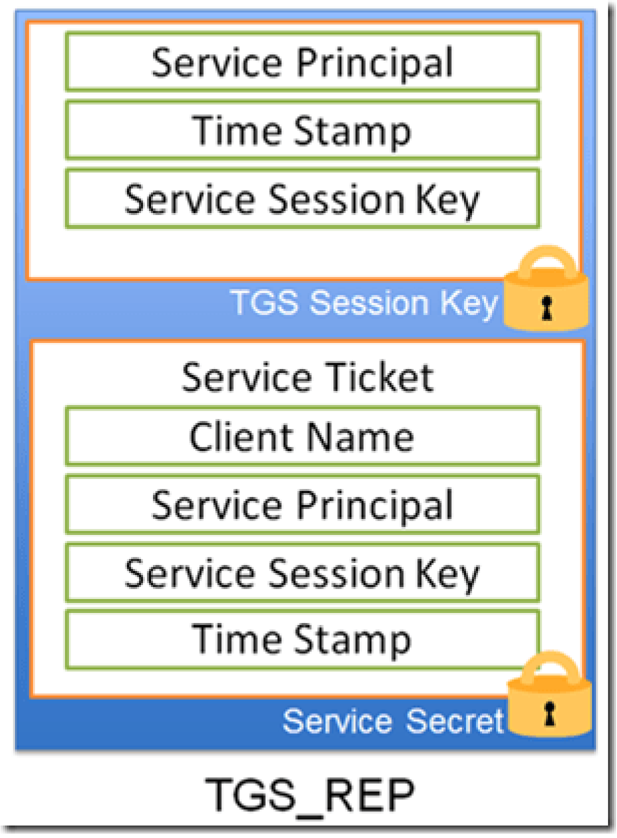
Encrypted with the same session key exchanged on the AS_REP:

- Service SPN.
- Time stamp.
- Service Session Key (symmetric key to be used between the user and the service).

Encrypted with the Service Password Hash :

- Client Name (username).
- SPN.
- Service Session Key.
- Time Stamp.

Service tickets have a maximum lifetime (which is ten hours by default in AD’s implementation of Kerberos) for which they can be reused.



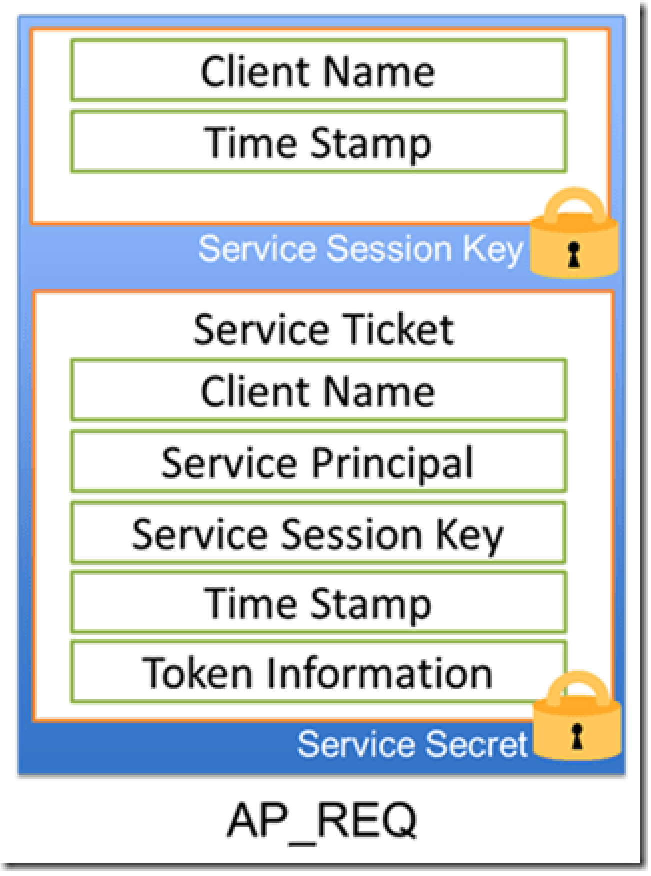
Accessing Services

After the client has a service ticket, the application accessing the service can present that ticket to the service and request access. The mechanics of presenting the service ticket aren't nearly as standardized as for obtaining the ticket because every application is different. In the case of an HTTP service, the service ticket is embedded in the headers of the HTTP request.

The service ticket is presented to the application in the form of a Kerberos AP_REQ message. The service decrypts the service ticket and obtains the session key, which it can use to decrypt the time stamp and client name fields, which are in turn used to validate the authenticity of the service ticket itself. It's important to note that even if the service accepts the service ticket, at this point the client has merely authenticated to the service. The task of authorization is still up to the service, based on the information it has about the client.

The service ticket typically also includes data known as the Privilege Attribute Certificate, or PAC. the PAC is called *Token Information. This i*s the same token information the KDC included in the user's TGT. The PAC is composed of information such as the user's SID, group membership information, and user security rights/privileges. When a user presents a TGT to the KDC to request a service ticket, the KDC copies the token information from the TGT and includes it in the service ticket's PAC field. This is the information that the service uses to construct an access token for the user and to verify the user's authorization, typically based on group membership.

An additional Kerberos message known as an AP_REP or Application Reply is permissible after the user presents a service ticket in the AP_REQ . The Application Reply message is optional; in general, the application won't send such a message unless an error occurs. One example of when an AP_REP message would be generated is in the case of a client that requests (in the AP_REQ message) that a service prove its identity through a process known as mutual authentication.



Final Thoughts

I hope this Kerberos 101 article helped you understand the main components of Kerberos and how TGT and service principal names play a big role in such wonderful protocol.

Blog Post Notification

Be the first to get notification when key blog post articles are released. No marketing material.