Kerberos Overview

Revision 20210324

Denodo Inc.: 525 University Avenue, Suite 31, Palo Alto, CA 94301. USA www.denodo.com

CONTENTS

1 OVERVIEW

In this document we will provide an introduction to the Kerberos protocol and how the Denodo Platform supports this protocol.
Kerberos is a computer-network authentication protocol that works on the basis of tickets to allow nodes communicating over a non-secure network to prove their identity to one another in a secure manner. The protocol was named after the character Kerberos from Greek mythology, the ferocious three-headed guard dog of Hades. Its designers aimed it primarily at a client–server model and it provides mutual authentication—both the user and the server verify each other's identity. Kerberos protocol messages are protected against eavesdropping and replay attacks.

Kerberos builds on symmetric key cryptography and requires a trusted third party, and optionally may use public-key cryptography during certain phases of authentication.

Each head of the three-head dog that guards the gates of the underworld is related to a crucial part of any network security scheme:

- Authentication, which refers to verifying the identity of a specific user.
- Authorization, which refers to granting or denying access to specific resources based on the requesting user's identity.
- Auditing, which refers to the capacity of auditing all actions taken by the authentication and authorization steps for future review by an administrator if necessary.

Before explaining how Kerberos works, it is necessary to define different concepts:

1.1 PRINCIPAL

A Principal is every entity contained within a Kerberos installation: individual users, computers and services running on servers. Principals are globally unique names, so each Kerberos principal is a unique identity to which Kerberos can assign tickets.

Every Principal is divided into 3 parts:

- The primary is the first part of the principal. In the case of a user, it's the same as your username. For a host, the primary is the word host.
- The instance is an optional string that qualifies the primary. The instance is separated from the primary by a slash. In the case of a user, the instance is usually null, but a user might also have an additional principal, with an instance called admin, which is used to administer a database. In the case of a host, the instance is the fully qualified hostname.
- The realm is your Kerberos realm. This is an administrative realm of control that is distinct from every other Kerberos installation.By convention, the Kerberos realm for a given DNS domain is the domain converted to uppercase.

The format of a typical Kerberos principal is `<primary>/<instance>@<REALM>`.

1.2 KEY DISTRIBUTION CENTER

The Key Distribution Center(KDC) consists on 3 logical components:

- A database of all principals and their associated encryption keys.
- The Authentication Server: It issues an encrypted Ticket Granting Ticket (TGT) to clients who want to login to the Kerberos realm.
- The Ticket Granting Server: It issues individual service tickets to clients when they request them. The TGS needs a ticket request that includes the principal name representing the service the client wants to connect and a TGT that has been issued previously by the Authentication Server. The TGS verifies the TGT is valid by checking that it is encrypted with the Kerberos server's TGT key and then issues the user the service ticket he requested.

1.3 KEYTAB

Service keys could be stored in the server providing the service in a special file called the keytab. There should be one keytab file for each service offered. Each Kerberized service should run as a different and unique username, and the keytab file for that service should be readable only by that username.

2 KERBEROS PROCESSES

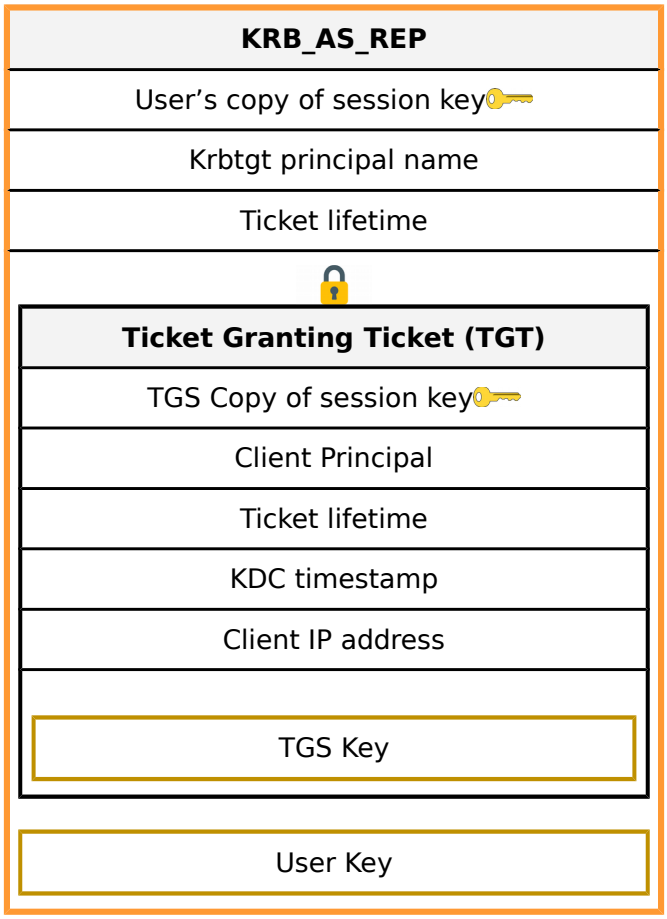
2.1 USER AUTHENTICATION PROCESS

1. Client sends to the Kerberos KDC the Authentication Server Request message (AS_REQ) in plain text.

KRB_AS_REQ
Client Identity
Client timestamp
Ticket Granting Server (krbtgt) Principal Name
Requested lifetime

2. The KDC verifies that the requested principal exists and that the client's timestamp is close to the KDC's local time.
3. The Authentication Server generates a random session key that will be shared between the client and the Ticket Granting Server. This key secures the ticket requests that the client makes to the Ticket Granting Server later on for specific Kerberized services. The KDC makes two copies of this session key: one for the client and another one for the Ticket Granting Server.

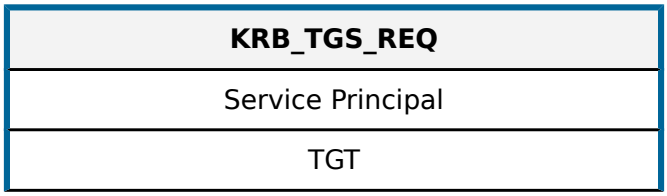
4. The KDC responds to the Client with an Authentication Server Reply message (AS_REP), which is encrypted with the user's key. This message also includes the Ticket Granting Ticket (TGT), which is encrypted with the Ticket Granting Server's key.

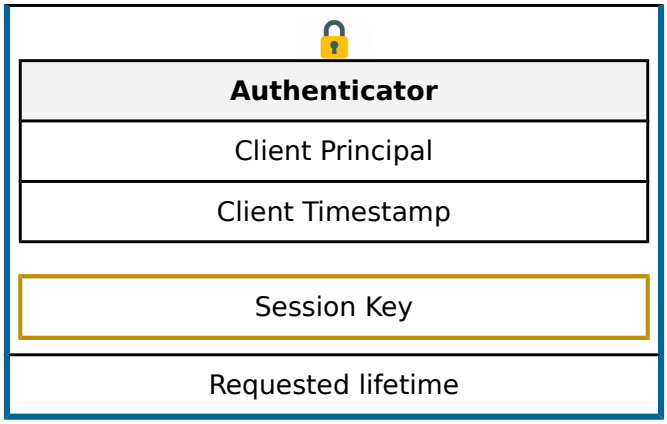


5. The client decrypts the AS_REP message with the user's key. If this decryption is successful then the client stores the Ticket Granting Ticket (TGT) and its copy of the session key in a credential cache. Notice that the client can not read the contents of the TGT because it is encrypted with the TGS's key.

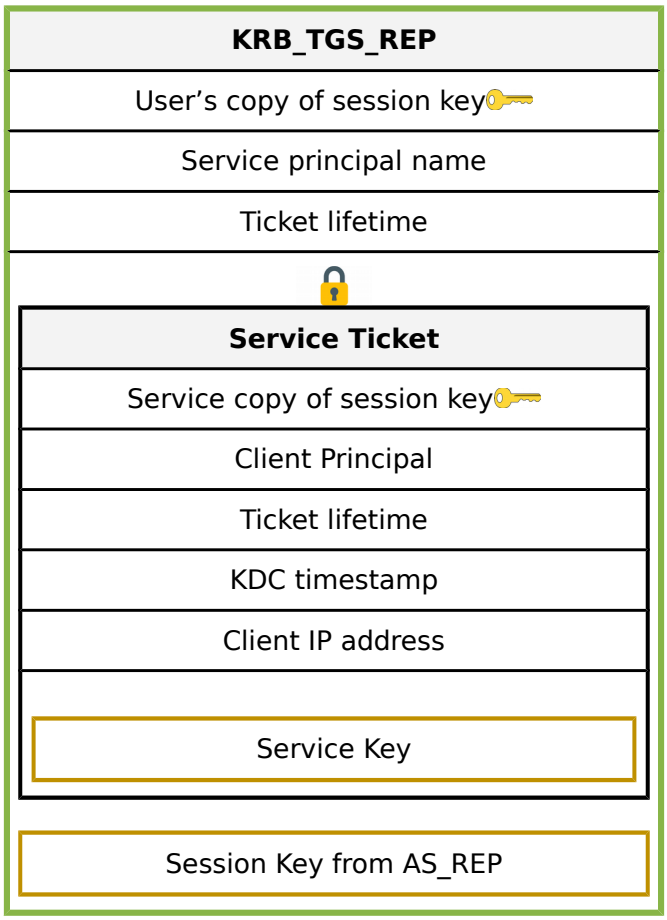
2.2 AUTHENTICATION TO A KERBERIZED SERVICE

1. The client sends a message to the TGS by including the TGS request, a copy of the TGT and an authenticator that serves to thwart replay.





2. The KDC replies to the Client a message that includes a new set of session keys shared between the client and the application server. The client's copy of the new session key is encrypted with the session key established during the Authentication Server exchange. The service's copy of the new session key is embedded inside of the Service Ticket and it is encrypted with the Service key. The client appends this service ticket and the new session key to its credential cache for later retrieval.



3. The client sends the server a message of type KRB_AP_REQ (Kerberos Application Request). This message contains an authenticator message that is encrypted with the key sent by the KDC for the session with the server and the ticket for sessions with the server.
4. The server receives KRB_AP_REQ, decrypts the ticket by using the session key, and extracts the user's authorization data and the session key.
5. The server uses the session key from the ticket to decrypt the user's authenticator message and evaluates the time stamp inside.
6. When the client receives KRB_AP_REP, it decrypts the server's authenticator message with the session key it shares with the server and compares the time sent back by the service with the time in its original authenticator message. If they match, the client is assured that the service is genuine, and the connection proceeds.

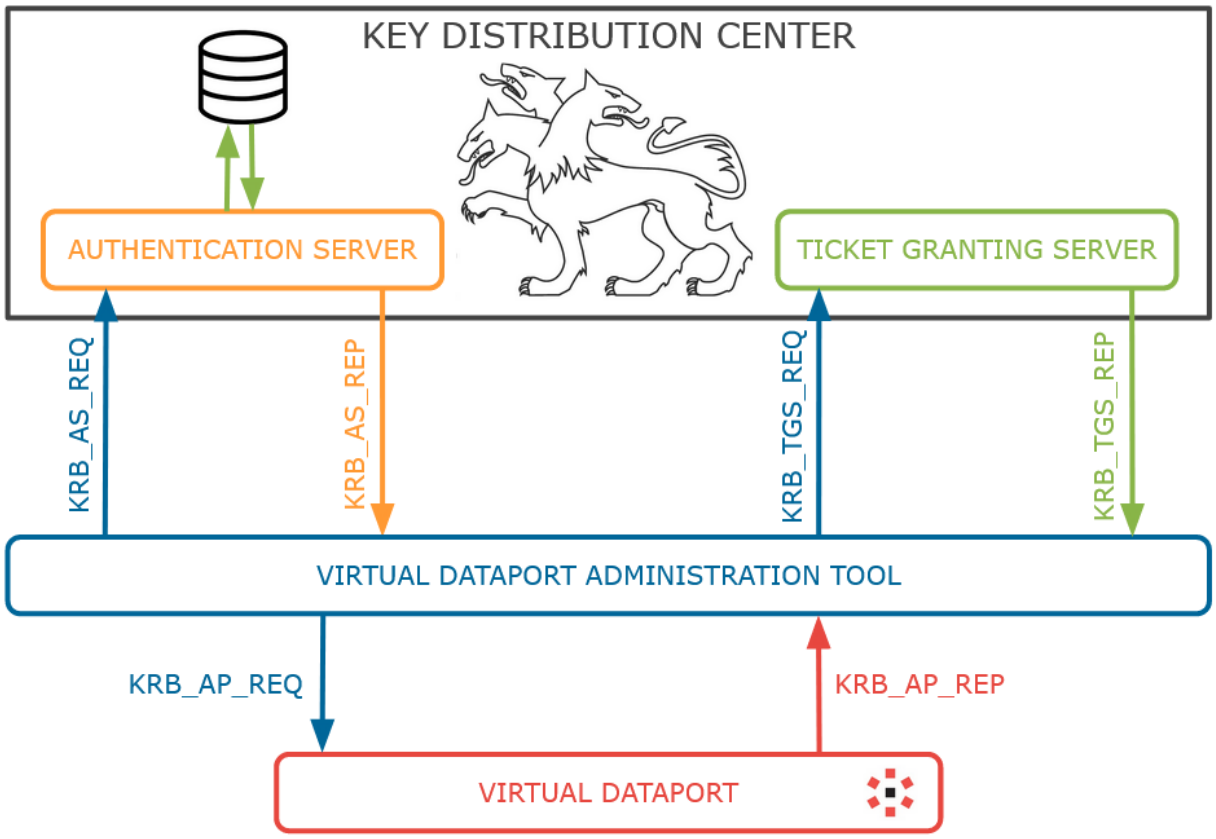
Notice that KDC does not specify policy on what principals are authorized to access a given service. The KDC will ensure the validity of the TGT sent by the client and then will issue a ticket for any service that it knows about, but authorization decisions are made by the individual application servers. The possession of a valid service ticket for a given service **does not** imply that the user should be granted access to that service.

3 KERBEROS AUTHENTICATION IN THE DENODO PLATFORM

Kerberos authentication can be configured for different servers and tools in the Denodo Platform.

3.1 VIRTUAL DATAPORT SERVER

In this case, the Virtual DataPort Server and the Virtual DataPort Administration Tool have to be configured to use Kerberos. The Virtual DataPort Server is the Kerberized Service and the Virtual DataPort Administration Tool is the Client.



3.2 WEB ADMINISTRATION TOOLS

There would be a similar situation when configuring the Data Catalog or the Design Studio for using Kerberos Single Sign On. In this case, the client would be the web browser, and each one of the Web Administration Tools would be a Kerberized Service. Also, the Virtual DataPort Server will be the Kerberized service for the Data Catalog or the Design Studio.

3.3 SCHEDULER SERVER

In the case of Scheduler, the web browser would be the client for the Scheduler Web Administration Tool, and the Scheduler Web Administration Tool would be the client for the Scheduler Server.

4 REFERENCES

Kerberizing Denodo for SSO - Step by step guide
Kerberos Authentication with the Active Directory
Kerberos configuration and troubleshooting