An LDAP, SASL, GSSAPI and Kerberos primer

By Lawrence Kearney

Even amongst the technically savvy there is often confusion surrounding authenticating applications that use the Simple Authentication Services Layer (SASL), Generic Security Services Application Programing Interfaces (GSSAPI) and Kerberos. This writing serves as a high level overview of the components and how they interact with Kerberised LDAP servers.

SASL provides a layer of abstraction to remove proprietary dependencies for security protocols and application semantics for security frameworks (like OpenSSL for example).

GSSAPI mechanisms provide a similar abstraction layer for application developers. An application developer can write his application to utilize a GSSAPI mechanism without having to change the application if that mechanism is updated.

Kerberos uses a system that issues cryptographic “tickets” to users, and then validates those tickets based on identity to implement a secure authentication framework on potentially insecure networks. Different types of Kerberos tickets are used to authenticate to workstations, applications and file shares for example. One way of visualizing this system would be using one ticket that required your photo ID to allow access to a public movie theater, and another ticket (that still required the use of your photo ID, for the rowdy jazz hall you frequent following the movie.

GSSAPI in the now:
After some problems related to the secure negotiation of a GSSAPI mechanism beneath the SASL layer were identified with other protocols, the GSSAPI mechanism was retrofitted to only reference Kerberos v5. The GSSAPI Kerberos subsystem allows an application to pre-authenticate to Kerberos, and then use the initial security credentials to access other services securely, including directory services.

The SASL GSSAPI mechanism is only supported by LDAP v3 servers. LDAP servers that support the SASL GSSAPI mechanism include Windows 2000 (and better) Active Directory servers, OpenLDAP, Novell eDirectory 8, and the SunONE Directory Server v5.2.

The practice of adopting open protocols and then re-implementing them to marry with Microsoft's development needs encompasses GSSAPI and Kerberos as well. Microsoft’s implementation of GSSAPI is called “Security Support Provider Interface” (SSPI).

A common use of this framework implements Kerberos based authentication for applications to Active Directory domains over LDAP. Most applications that utilise LDAP are also SASL GSSAPI aware. As a result they can also use Kerberos to search for and authenticate users, and as a result be “Kerberised”. To use either standard Kerberos or Microsoft Kerberos (MsKerboros), your server and client must be joined to a domain (called a realm in Kerberos parlance). A Kerberos realm is a logical network which defines a group of systems under the same master Kerberos Key Distribution Center (KDC). The KDC is composed of trusted servers that issue Kerberos tickets to clients and servers allowing them to communicate securely. The first ticket issued to a principle (user, host and service account) is called a “ticket granting ticket” (pre-authentication) and is required for principles to request tickets for other Kerberised services, like file share access as mentioned previously.

Each realm must include a server that maintains the master copy of the principal database. This server is called the master KDC server. The KDC also incorporates an Admin Server facility which handles administrative commands such as adding, deleting, and modifying principals in the Kerberos database(s).

Statement made by Fulvio Ricardi in his Kerberos Protocol Tutorial:

Kerberos is "… an authentication protocol for trusted clients on untrusted networks."

So, if Kerberos is designed to trust on an untrusted network, it should be even more effective on a trusted corporate network.

When using the SASL GSSAPI mechanism, the following events occur.

1. The principle successfully authenticates to Kerberos realm.
2. The application assumes the identity of the authenticated principal.
3. GSSAPI negotiates the mechanism used to establish a “security context” for the authenticating application

In most cases the SASL aware application automatically uses the credentials of the authenticated user running the authenticating application.

Keytab files

As well as storing principles and their passwords, Kerberos servers (KDCs specifically) can store principles and their “keys” (similar to passwords). Principles and keys are used as part of the authentication process to verify which user, host or service is connecting to, or requesting access to a network service. Host and service accounts are generally both referred to as “service principals”.

Keytab files are static files on a host's file system that store principle credentials and keys used for encrypting and decrypting Kerberos tickets. These files are useful on systems that host services on platforms that don't have native access to the principle database but need to authenticate and consume services hosted in a Kerberos realm.

On Linux systems service principals are almost always stored in a keytab file. These service principles usually represent root-owned processes that interact with Kerberos realms and services on behalf of the local host. An example of this use case could involve providing authentication services for Active Directory users for an application running on the Linux host. Keytab files can be used by clients to authenticate directly to a Kerberised application, or they can be used by the application server to pre-authenticate to the KDC and then manage client authentication requests itself. More often keytab files are used to allow systems and/or applications to authenticate to Active Directory on startup. It's important to remember that keytab files are static and if the service principle credentials or keys change all local files referencing those identifiers must be updated manually.