

Kerberos?

- Network authentication
- Written by MIT in 1980's, as part of Project Athena
- picky about network setup
- Common implementations: MIT krb5, Heimdal, Microsoft Active Directory

<http://web.mit.edu/Kerberos>



Kerberos is a network authentication system, allowing a user to prove that he is who he claims to be to a remote service without revealing sensitive information to that remote service.

Three parties: KDC (a trusted third party), a Client, a Service.

The client can be a user who authenticated and receives a ticket. It can also be a service who requests something from another service.

Kerberos does not do authorization, only authentication.

Project Athena is the project which also brought us the X Window System.

LDAP?

- Lightweight Directory Access Protocol
- "lightweight" – hah.
- Directory
- Common implementations: OpenLDAP, Microsoft Active Directory (almost)



LDAP is the "lightweight" directory access protocol, originally based on the Directory Access Protocol from the OSI stack. Called "lightweight" to distinguish from DAP and because it originally *was*, in fact, lightweight, but it absolutely isn't anymore, today. Quick word of advice: if you ever write some piece of software, never call it 'simple', 'small', or 'lightweight' – these things have the tendency of growing.

A "directory" is a system in which data is stored, but it is *not* a database. A directory is for data that is rarely updated, and for which a delay in the update is not a significant problem. For instance: a phone book, employee database, user account information, etc. LDAP is also commonly used for things like a shared addressbook—thunderbird, evolution, kmail, etc, have support for this.

See tutorial sunday morning in the ruby room – we'll touch it, but not in detail.

There's a RedHat implementation too, but they don't even ship it in their own distribution...

Authentication vs Authorization

- Authentication: allowing someone to prove they are who they say they are.
- Authorization: deciding whether users, based on their authentication credentials, are allowed to do certain things
- Kerberos does authentication only; LDAP can be used for both.



Kerberos terms

- Realm
- Principal
- Ticket
- KDC
- TGT



A kerberos realm is a group of principals that is administered by a common group of administrators. It is not possible – as with ldap – to limit certain administrator’s access to only part of the realm, but it is possible to "link" realms with cross-realm authentication. It is, however, possible to limit what *actions* a particular administrator can take (for instance, one can say that a particular administrator can change passwords, but not add or remove principals).

A kerberos principal is a name to which the system can assign tickets. A principal has a secret that is shared with the KDC.

A ticket is a proof of authentication. With a ticket, a principal can prove to another principal that the principal is who it claims to be, without them having to send passwords to eachother

A KDC, or Kerberos Distribution Center, is the central authority which holds the secrets to all principals and hands out tickets. Since it’s a trusted third party, a compromise of the KDC is a serious matter—that would require destroying and recreating the realm

The TGT, or Ticket Granting Ticket (some call it 'Ticket to Get Ticket') is the ticket one gets from the KDC after authenticating. With that ticket, it’s possible to get tickets for other services – other principals. It’s a ticket to the "ticket granting service".

Anatomy of a kerberos principal

host/samba.grep.be@GREP.BE

- host: primary
- samba.grep.be: instance
- GREP.BE: realm



The primary describes what the ticket serves. In this case, it allows 'host' access (e.g., ssh or telnet connections). For user principals, the primary is the user's username.

The instance is optional, and not used for user principals (though it is used for administrator principals).

Network pickyness

Kerberos has certain requirements...

- no clock skew (use ntp)
- reverse lookups must make sense for servers
- optionally, announce realm and kdc's through DNS (reduces setup on clients).



DNS example

- `_kerberos.grep.be. 86400 IN TXT "GREP.BE"`
- `_kerberos._udp.grep.be. 86400 IN SRV 0 0 88 samba.grep.be.`
- `_kerberos-adm._tcp.grep.be. 86400 IN SRV 0 0 749 samba.grep.be.`
- `_kpasswd.grep.be. 86400 IN SRV 0 0 464 samba.grep.be.`



The first record allows clients to figure out what the realm for a given domain is.

The second is used by clients who want to figure out where the KDC for a given realm is. There can be several KDC's for a given realm, so there can be several `_kerberos._udp` SRV records. Note that it is also possible for kerberos to use `tcp` (and, hence, to have `_kerberos._tcp` SRV records), but MIT `krb5` does not use that by default.

The `-adm` SRV record is used by `kadmin` (the administrator's interface) and can only be running on the master KDC. This is MIT-specific

The `kpasswd` service is used by `kpasswd` (or by `passwd` if PAM has been set up correctly) to allow users to update their own password. This can only be running on the master KDC, too.

Setting up a kerberos realm

- Debian packages: krb5-admin-server, krb5-kdc
- RedHat: krb5-server
- kdb5_util create (or: krb5_newrealm)
- kadmin.local: addprinc *user/admin*
- Start kadmin and kdc services
- use kadmin (with newly created admin user) to add more principals

Congratulations! You have a realm!



Using the kerberos realm

- kinit/kdestroy
- klist
- pam_krb5.so
- sshd: GSSAPIAuthentication yes



Wouter Verhelst

Single sign-on with Kerberos and LDAP

And...

- GSSAPI: "GSSAPI is just a funny name for Kerberos v5" – Peter Palfrader
- SASL: IMAP (dovecot, evolution, thunderbird, offlineimap, mutt, ...)
- SASL: LDAP (openldap – beware circular dependency)
- SASL: ...
- HTTP: Negotiate authentication (mod_auth_kerb, gecko-browsers, internet explorer, google chrome for windows (*not* chrome for Linux, yet), webkit, ...)
- NFS: sec=gss/krb5, sec=gss/krb5i, sec=gss/krb5p
- ...



Wouter Verhelst

Single sign-on with Kerberos and LDAP

Yes, NFS supports GSSAPI authentication. There are three options: krb5 (which authenticates users to the remote server by using kerberos), krb5i (which authenticates users on each and every RPC call), and krb5p (which encrypts all traffic between client and server).

Setting up LDAP

- Debian packages: slapd
- RedHat packages: openldap-servers
- Create slapd.conf (see tutorial, or use Debian)
- slapadd to add initial directory tree.
- objectClass required for unix accounts: posixUser, posixGroup, shadowAccount
- libnss_ldap.so
- pam_mkhome.so

