A sysadmins guide to authentication and authorization

Christian Horn
chorn@fluxcoil.net

July 8, 2010

v1.45

introduction

Tom Servo, MST3K:

“Okay, what are we looking at and why are we looking at it?”

Authentication

What is Authentication?

- Making sure someone is who he claims to be.
- Authenticators are something the user
  - has (drivers license, passport, software token)
  - knows (pin, password, passphrase)
  - is or does (fingerprint, DNA sequence, voice recognition)

What is strong authentication?

- >=2 authenticators are used

Authorization/Policy

What is Authorization?

- Handling the users permissions

What are policies?

- Rules set up at companies, handling password-length etc.

What is a directory?

- A storage for information, database. I.e. phonebook

in the beginning...

How was all this solved earlier, in the computer stoneage, with computers like this Zuse Z3?

No different users, only security-barrier is physical access - same as with calculators nowadays.

file replication overview

Along came networking:

- multiple users, multiple computers
- data to be moved around while preserving ownerships
- gid/uid namespaces appear and span those computers

file replication overview

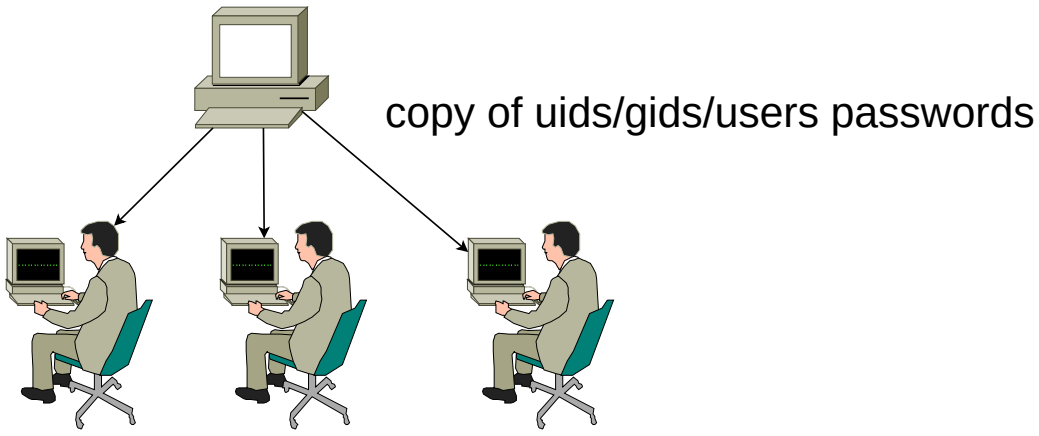
Along came networking:

- multiple users, multiple computers
- data to be moved around while preserving ownerships
- gid/uid namespaces appear and span those computers

Easiest solution:

- copy user/group/password information to all computers

file replication pros/cons



pros:

- simple

cons:

- security; all passwords around on all boxes
- you gotta trust root on all those boxes
- doesnt scale well with hundreds of boxes
- doesnt work for OSs hiding those informations

Overview

- First widely used directory appeared in 1983: DNS
- Initially just for lookups of ips/hosts - can now transport much more information, even functions for auth and dynamic modification of datasets
- extensions for better security exist (i.e. DNSSEC) but are not widely deployed yet

pros/cons

pros:

- essential standard, many network services rely on this
- IETF defined, good scalability
- will become even more important with rise of ipv6

cons:

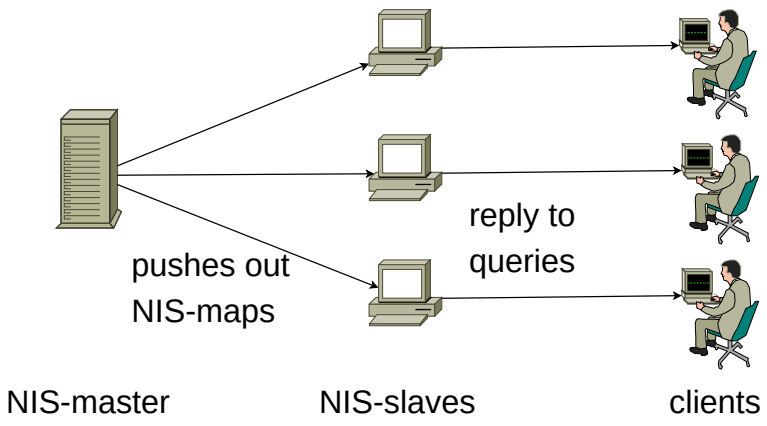
- widely used - essential to address flaws really fast

Overview

Next step: SUN develops 'Yellow Pages' in 1985

- because BT owns that name it gets renamed to NIS
- first big standard supported by wide variety of OSs like SunOS, AIX and HP-UX in their defaultinstalls
- basically a directory: a NIS-master stores map-files; informations like users, groups, userpasswords, hostnames, nfs mounts
- on clients: implementations were first in libc, later moved to configurable nsslibs and pam (HP-UX's is still strange)

basics of operation



1. map-files are modified on NIS-master
2. after modificaton the whole files are pushed to NIS-slaves
3. NIS-clients query the informations from Slaves; resolve hostnames to ips, authenticate users etc.

pros/cons

pros:

- well understood, widely supported
- still more secure than unencrypted ldap ;)

cons:

- hashed passwords can be read & inspected on all slaves
- submission unencrypted (easily spoofable, not secret)
- requires rpc (remote procedure calls)
- doesnt scale well, maps not hashed for lookups, complete maps submitted every time

Network Information Service +

Some of NIS issues got adressed with NIS+ by SUN

- Secure RPC now used (clients and servers must authenticate)
- read-only replicas possible
- possible to transmit only diffs instead of full maps
- NIS+ table permissions so users can not access all informations

pros:

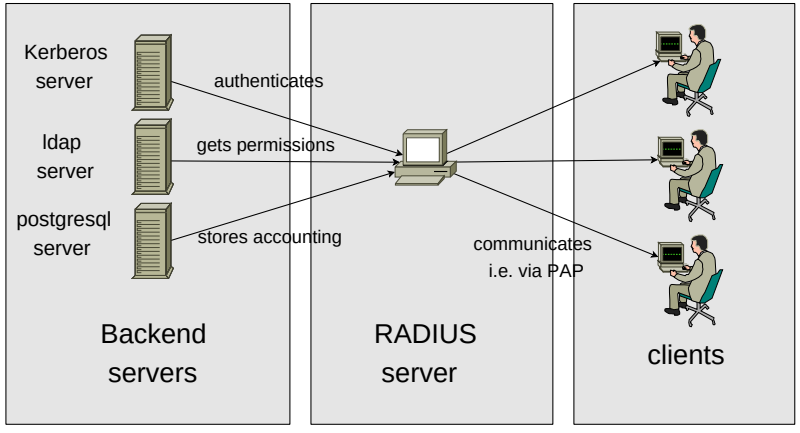
- NIS+ made multi-master configurations easier
- brought in security improvements

cons:

- no widely adoption outside the sun-world, no IETF standard
- bad reputation for manageability

Overview

- Radius (Remote Authentication Dial In User Service) gets defined in 1991, since 1997 its in the RFCs
- Basically a proxy, serving all three of AAA



- PAP, CHAP or EAP mostly used for communitation with user
- ldap/kerberos/sql/AD in backend

pros/cons

pros:

- wide acceptance in the areas ISPs/routers/wlan access points/Voip
- can be used to authenticate users directly for logins

cons:

- uids, gids etc. have to be taken out of directories like NIS/ldap
- passwords do not go over the net in the clear, but radius uses md5 for obfuscation which isnt regarded secure any more
- successor in sight: the Diameter protocol

Overview

- in 1993 Novell releases Netware Directory Services (NDS)
- later renamed to eDirectory
- NDS offers directory and authentication services; supports access by many OSs like linux, solaris, aix, hp-ux, netware and windows
- based on X.500, the heavy weight set of standards covering directory services
- access via ldap, odbc, soap etc. possible

pros/cons

pros:

- first mature, scalable directory
- supports most OSs via netware-protocol or ldap
- hierarchical data storage (think of ldap which is a descendant of X.500)
- fine grained access controls
- servers can carry subsets of data, replication to slaves possible

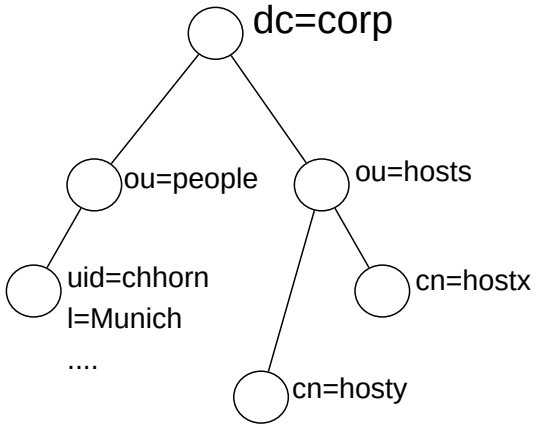
cons:

- only implemented by Novell
- code not open; vendor lock-in

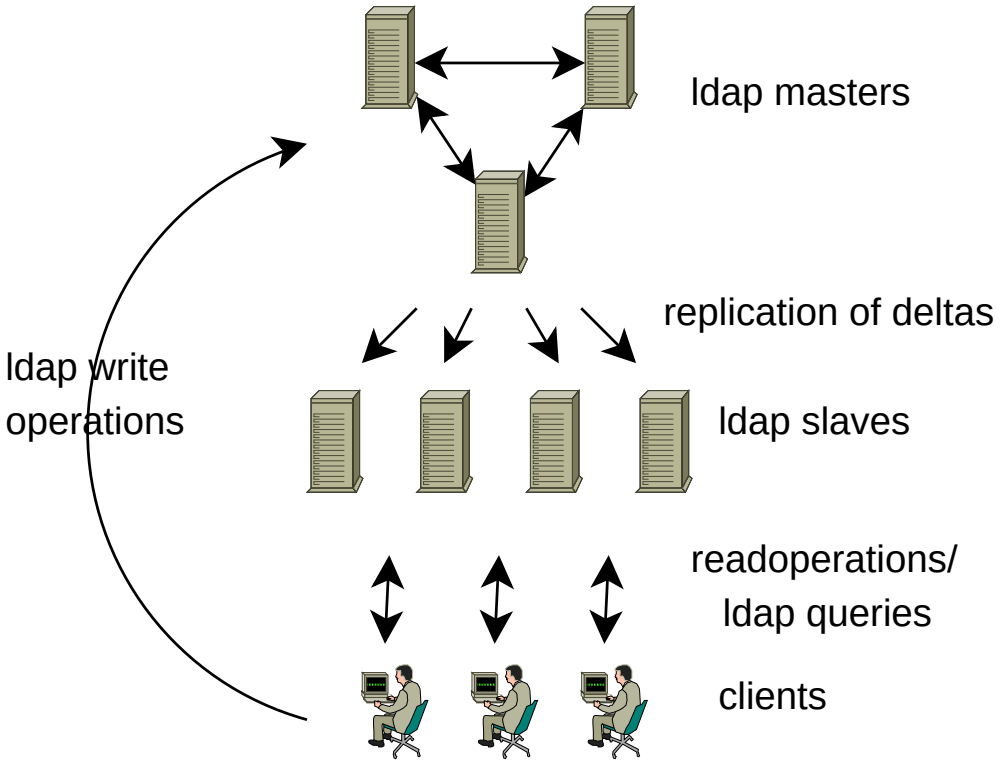
Overview

LDAP - developed in 1993
- to access a directory with data about people and other objects, extended with authentication/encryption/groups etc.

- Open standard protocol on how to access directory services, published in RFCs by the IETF
- data hierarchically structured, support for replication to slaves



ldap replication/slaves setup



pros/cons

pros:

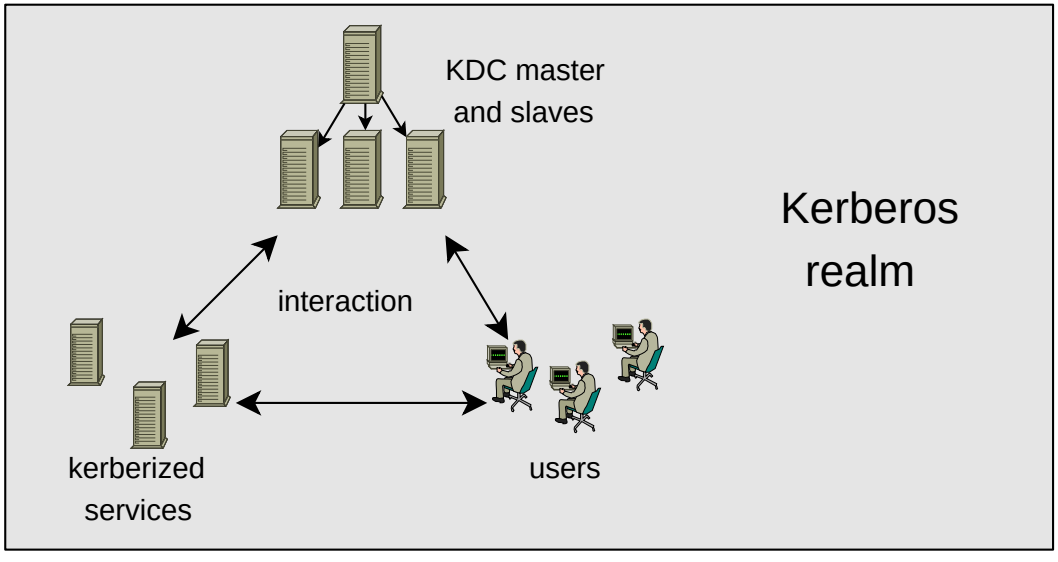
- widely supported, many server implementations: OpenLDAP, Fedora Directory Server, SunONE (form. iPlanet), NDS, AD
- authentication of OSusers supported natively on linux, solaris, aix and hp-ux, using additional opensource software (pgina) also on windows
- authentication of the server possible with x509-certificates or with kerberos
- communication can be secured with ssl/tls, enterprise ready (multimaster)

cons:

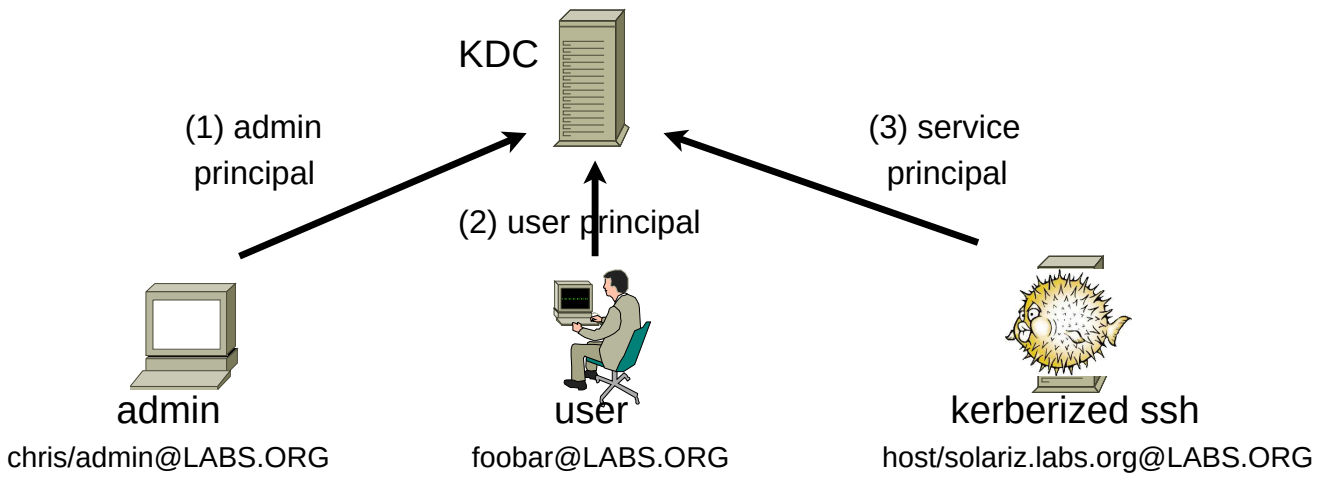
- without ssl/tls passwords go in the clear on the network

the history

... pure authentication protocol
... developed at the Project Athena (IBM, MIT, DEC)
... version 4 from late 1980s still in some use
... version 5 from 1993 is current

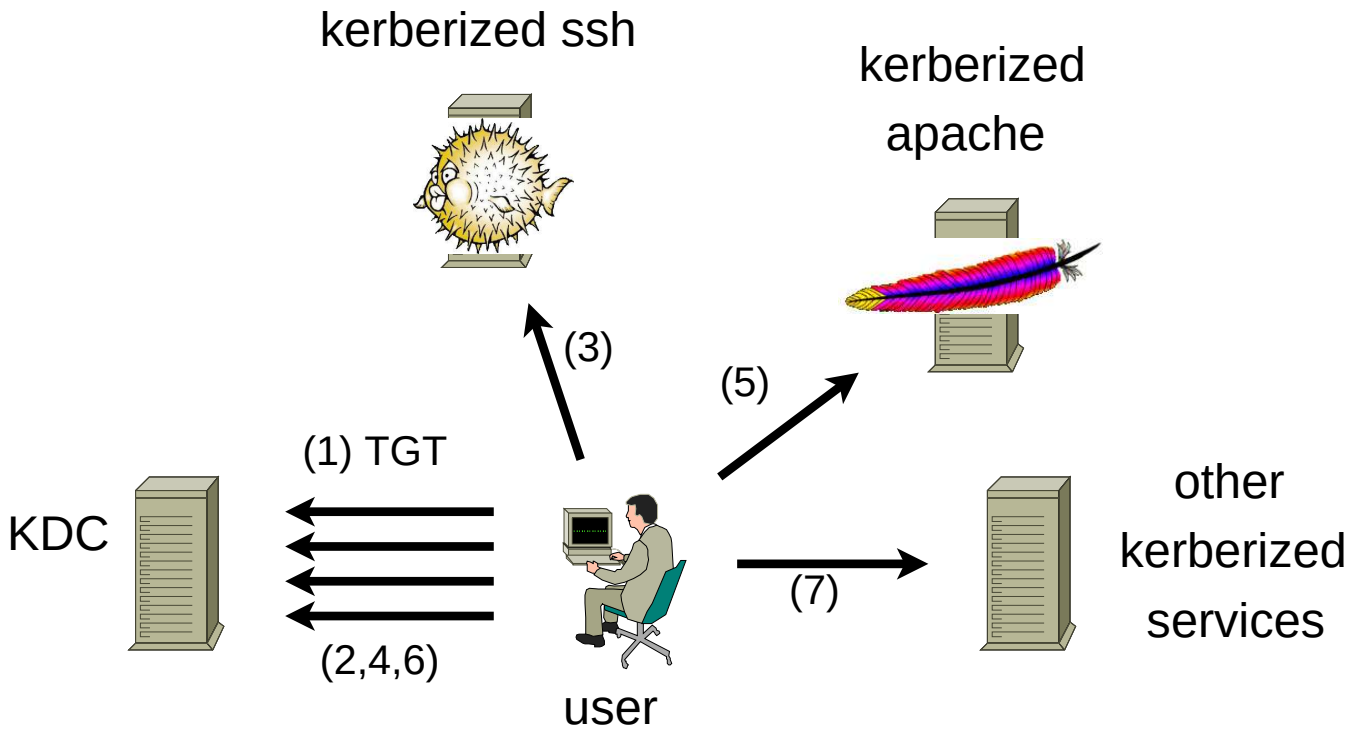


Setup of the principals in a realm



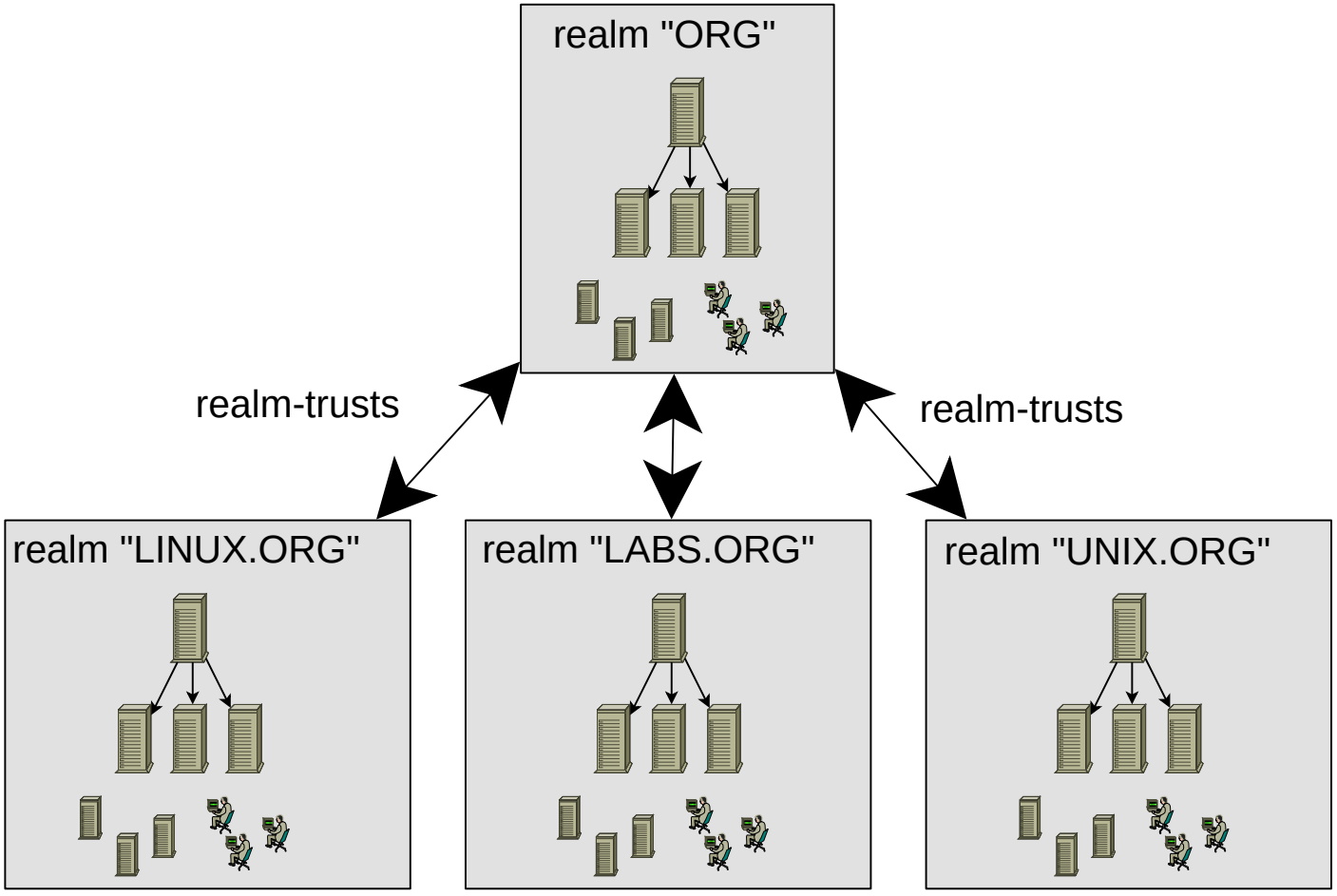
1. admin sets up his admin-principal using a password
2. user sets up a user-principal using a password
3. service-principal gets created w/ random input

SSO, details, simplified



(1) getting the TGT, i.e. kinit
(2,4,6) getting servicetickets
(3,5,7) usage of service

realm trusts: bigger realm environments



pros/cons

click here: http://SSO Crossrealm Kerberos Demo

pros:

- known for good security (mutual auth.), mature, scalable
- biggest protocol providing SSO
- wide support in industry from sun, google, apple, redhat, (microsoft) etc.
- many implementations: MIT, heimdal, commercial
- follows unix principle: does only auth – but that well
- apps: nfsv4, ssh, ftp, vpn, afs, jabber, apache, browsers..

cons:

- infrastructure requirements: timesynced hosts, dns
- if KDC gets compromised all keys have to be replaced
- compromised userkey allows everything the user is allowed to

AD overview

Chuck Yerkes @sage-member list

There's a new tool that MS has recently retroactively invented: Kerberos. It came out last year, and I've used it since 1993 or so.

- appeared in 1999 for win2k
- extended with releases of win2k3 and win2k8
- provides authentication, authorization, policies, delivery of critical updates
- employs ldap + kerberos + dns
- naming update: realm -> AD domain
- naming update: realms with trusts -> tree/forest

pros/cons

pros:

- easy & fast to deploy in pure windows environments
- tight integration with microsoft software like exchange

cons:

- focused on windows domainmembers
- modified kerberos is used, making interaction with protocol-compliant environments difficult
- AD software requires Windows, costs money
- sources not freely available -> no security auditing, no unrestricted improving/extending possible, vendor can enforce soft-releasechange in refusing delivery of updates
- debugging of problems awkward for unixers

CrossRealm setup Kerberos <-> AD-domain

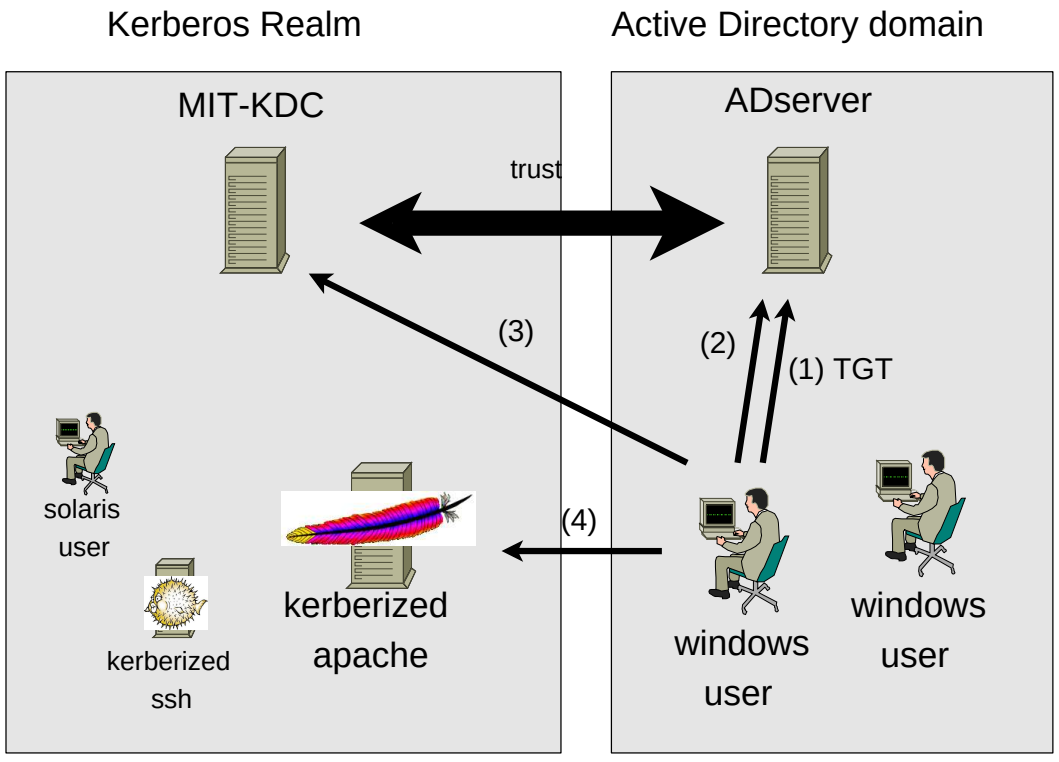
problems with solutions seen so far:

- nothing can provide SSO for everyone (unix, linux, mac, windows clients)
- solutions involve too much pain for the admin
- full support is needed (important in some places)

crossrealm setup contains:

- a Kerberos realm (I used MIT) with KDC and services
- an AD-domain with Server and clients
- exchanged principals + crossrealm trust

CrossRealm overview graphical



(1) getting TGT .. (2) referral ticket
(3) getting serv.ticket .. (4) using serv. ticket

CrossRealm pros/cons

pros:

- solution fits many needs: security, SSO
- works for all kinds of clients
- can be fully supported by vendors (think MIT)

cons:

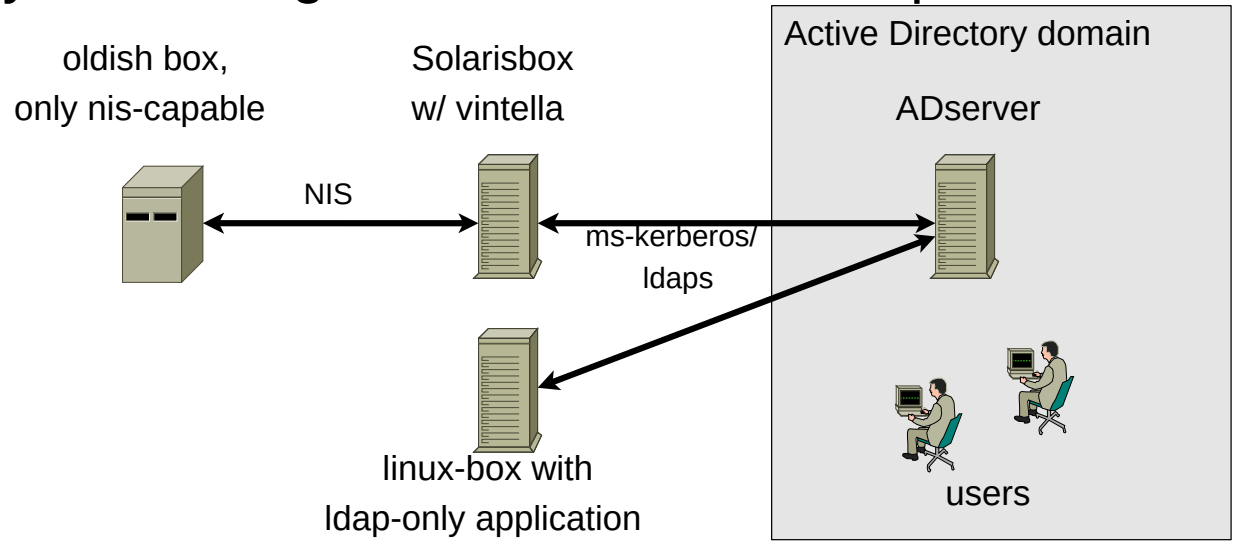
- see ldap/kerberos cons: infrastructure needs working DNS, ntp
- additionally solution to provide auth-data for the unix-servers needed (i.e. ldap)
- complex setup, gotta train your admins on that

Samba 4/7/franky)

- samba-teams approach to provide free AD-implementation
- that code is now included in usual samba 3.x tarball
- uses customized kerberos, ldap, ntp code
- already there with samba 3.5.x, i.e. in RHEL6:
  - windows/samba-boxes can join the samba-hosted domain
  - windows can be used as additional domaincontrollers in a samba-hosted domain
  - password-changes of domainmembers, smb encryption
- todos: readonly DCs, multi domain forests, ..
- samba4 making progress, full AD replication
- samba 4.0.0 release planned for 2011

Commercial solutions: vintella authentication services

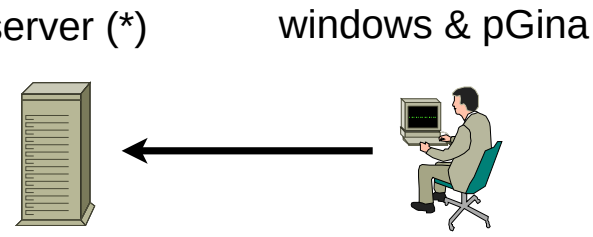
Basically interfacing AD-domains to other protocols



Likewise Enterprise

- specialized to connect linux/unix-boxen onto AD-domains
- credential cache allows userlogins also while KDC unreachable, i.e. laptops
- policy handling also on linux-boxen possible: i.e. enforce after what time an idle desktop activates the screensaver
- commercial product, free trialversions available

pGina



- OpenSource login-module for windows (kind of PAM)
- enables authentication via NIS, POP3, IMAP, LDAP, MySQL, OpenAFS, PAM, PostgreSQL, RADIUS etc.
- unclear if Kerberos-ticket can be grabbed after userlogin

389DS / RedHat Directory Server

- UMICH slapd -> Netscape Directory Server NDS -> forked to (SunONE, FDS) -> 389DS
- features: enterprise focus, multi-master replication, scalable, codebase around for long time, ssl, passwd policy, virtual views, plugin interface, resource limits, mostly online operations (reconfigs etc.)
- syncing of user/group/password with AD or NT-domain controller
- GUI to manage users/groups etc.
- backend berkley db (others via plugin), non openssl ssl-implementation used: Mozilla NSS

FreeIPA

- free Identity Policy Audit, freeipa.org
- approach from RedHat to create open security information management solution
- opensource
- backend-storage for everything is fedora directory server
- status: 1.2.x is stable, 2.x available as alpha
- features in stable: kerberos, ldap (users, groups, hosts, hostgroups), ntp, webfrontend/commandline for administration

FreeIPA cont'd

- IPA 2.0 will be shipped with RHEL 6.1
- features of version2: DNS, CA (request/approve/revoke certs), NIS, policy engine (via PolicyKit), audit log collection
- upcoming: SSSD, System Security Services Daemon, pam+nss on steroids. providing opensource credential cache (KDC can be unreachable), support of multiple different identity sources and policies (enforce GNOME-settings etc.)
- can be used in crossrealm setups
- freeIPA for everyone, as part of RHEL later supported by RedHat

Questions?

ありがとうございました [1]

`http://fluxcoil.net` - my site

chorn@fluxcoil.net

[1] Thank you.