

# LDAP and Kerberos

## *Centralized User Administration*

David Hilley

davidhi@cc.gatech.edu

College of Computing  
Georgia Institute of Technology  
Atlanta, GA 30332

# What is NSS?

- Name Service Switch: `/etc/nsswitch.conf`
- Centralized administration for various things:
  - `/etc/passwd, shadow / master.passwd`
  - `/etc/groups`
  - `/etc/hosts`
  - protocols, services, networks, netgroup, aliases, etc.
- Hooks in the C library:
  - `getpwent, getgrent`
  - `gethostent, gethostbyname, etc.`
  - `getaliasent`
  - `getnetent, getnetbyaddr, etc.`
- Used in Linux, FreeBSD, NetBSD, Solaris, IRIX, HP-UX, AIX, etc.

# What is NSS? cont.

- Common backends supported:

- Files, BDB
- NIS, NIS+
- DNS
- LDAP
- Compat?
- Make your own!

- /etc/nsswitch.conf:

- `groups: nisplus [NOTFOUND=return] db files`
- `“[" ( “!”? <status> “=” <action> )+ “”`
- `<status>` → success | notfound | unavail | tryagain
- `<action>` → return | continue

# NSS Choices

- NIS (Network Information Service) – formerly “Yellow Pages”
  - + Widely used, widely supported.
  - Crusty and old, insecure.
- NIS+
  - + Newer, secure.
  - Proprietary, hard to administer, open-source efforts dead.
- Hesiod
  - + Uses DNS.
  - Uses DNS (stores all the data in DNS TXT records).
- Berkeley DB/Files?
- Lightweight Directory Access Protocol

# LDAP

- LDAP is a binary directory access protocol.
- LDIF is the LDAP Data Interchange Format.
- Follows the X.500 hierarchical object data model:
  - Tree-structured directory.
  - Each entry has a *distinguished name* (DN) and attributes.
  - Data follows a predefined schema (RFC 2307 for NSS).
- Backend can be a relational DB, BDB, DBM, etc.
- OpenLDAP – Many platforms, mature.
- Use to store passwords?
  - Use SSL for all lookups, secure permissions.
  - Or use Kerberos, which also provides other benefits.

# LDAP Examples

## ● Unix User/Group:

```
dn: uid=davidhi,ou=People,dc=abc,dc=org
uid: davidhi
cn: David Hilley
objectClass: account
objectClass: posixAccount
uidNumber: 1001
gidNumber: 100
loginShell: /bin/bash
homeDirectory: /home/davidhi
gecos: David Hilley
```

```
dn: cn=users,ou=Group,dc=abc,dc=org
objectClass: posixGroup
objectClass: top
cn: users
userPassword: {crypt}*
gidNumber: 100
memberUid: davidhi
...
```

```
● $ ldapsearch -x -h ldap.gatech.edu -b 'dc=gatech,dc=edu' 'cn=hilley'
...
# thilley3, gatech.edu
dn: uid=thilley3,dc=gatech,dc=edu
...
# gte556v, gatech.edu
dn: uid=gte556v,dc=gatech,dc=edu
...
```

# Kerberos

- Kerberos v5, RFC 1510 (MIT Kerberos or Heimdal).
- KDC (Key Distribution Center)
  - Stores keys derived from passwords.
  - Grants time-limited tickets.
- Provides single sign-on/authentication for Kerberized services.
- OpenSSH, Postfix, Dovecot, Samba, NFS, AFS, X11, etc.
- Requires synchronized clocks for systems (NTP).
- GSSAPI – Generic Security Services Application Programming Interface
  - Client-server authentication API.
  - Typically interfaces with Kerberos.
  - Often interfaced with via SASL.

# Big Picture

- User and group data stored in LDAP database.
- Passwords handled by Kerberos.
- Concept of a *principal*
  - A distinguished entity.
  - Kerberos has them and OpenLDAP has them (DNs).
  - A Kerberos principle per user account.
- How does all this stuff work together?
  - NSS, GSSAPI
  - PAM – Pluggable Authentication Modules
  - SASL – Simple Authentiction and Security Layer

# Working Together

- NSS uses LDAP to grab the user/group info.
- PAM is used for authentication; PAM talks to Kerberos.
- LDAP talks to Kerberos via SASL, using SASL's GSSAPI method.
- OpenSSH interfaces with Kerberos via GSSAPI or PAM.
- sudo can use LDAP, but not through NSS.
- OpenSSH can use LDAP for public keys.
- Kerberos services use `/etc/krb5.conf`
- LDAP services use `/etc/ldap.conf`
- OpenLDAP tools use `/etc/openldap/ldap.conf`

# Setting up Kerberos

- First install and set up NTP.
- Binaries in `/usr/kerberos/bin` and `/usr/kerberos/sbin`
- Set up KDC, set your *realm*:
  - `/var/kerberos/krb5kdc`
  - `kdc.conf` (realm, key types)
  - Create the DB: `kdb5_util create -s`
  - Edit `kadm5.acl` ( `*/admin@ABC.ORG` `*` ).
  - Create `kadm5.keytab` if necessary with principles `kadmin/admin` and `kadmin/changepw`
  - Add an admin principle for yourself and user principles.
- Kerberos (88/UDP), `kadmin` (749/TCP).

# Kerberos Usage

- On clients edit `/etc/krb5.conf` (set realm, server, etc.).

- `kinit (-f)`, `klist`, `kpasswd` and `kadmin`

- Configure PAM to use Kerberos:

```
...
auth          sufficient    /lib/security/pam_krb5.so use_first_pass
...
password      sufficient    /lib/security/pam_krb5.so use_authtok
...
session       optional      /lib/security/pam_krb5.so
...
```

- Put `*K*` for password in `/etc/shadow` if not using LDAP.
- Set up replication to secondary KDCs (`kpropd`, 754/TCP).
- Set up directly Kerberized services.

# OpenSSH with GSSAPI

- Allows single sign-on and ticket forwarding.
- Make a Kerberos service principle for each host running OpenSSH.
  - `kadmin -q "addprinc -randkey host/<hostname>"`
  - `kadmin -q "ktadd -k /etc/krb5.keytab host/<hostname>"` (on host) **OR**
  - `kadmin -q "ktadd -k <tmpfile> host/<hostname>"` (and copy)
- **Edit `sshd_config` and `ssh_config`**

<code>KerberosAuthentication yes</code>	<code>GSSAPIAuthentication yes</code>
<code>KerberosOrLocalPasswd yes</code>	<code>GSSAPIDelegateCredentials yes</code>
<code>GSSAPIAuthentication yes</code>	<code>PreferredAuthentications gssapi-with-mic,...</code>
<code>GSSAPICleanupCredentials yes</code>	
- `ssh -o PreferredAuthentications=gssapi-with-mic` (testing)
- Depends on hostname; can get tricky.

# Setting up OpenLDAP

- Edit `/etc/openldap/slapd.conf`

- Include correct schemas: `include /etc/openldap/schema/nis.schema`

- Set suffix: `suffix "dc=abc,dc=org"`

- Set rootdn, rootpw for initial setup.

- Set SASL stuff:

```
sasl-secprops noanonymous,noplain,noactive
```

```
saslRegexp uid=([^\/*]*),cn=GSSAPI,cn=auth uid=$1,ou=people,dc=abc,dc=org
```

- Set ACLs:

```
access to attr=loginShell
```

```
    by dn.regex="uid=.* /admin,cn=GSSAPI,cn=auth" write
```

```
    by self write
```

```
    by * read
```

```
...
```

```
access to *
```

```
    by dn.regex="uid=.* /admin,cn=GSSAPI,cn=auth" write
```

```
    by * read
```

# Connect with Kerberos

- Create a Kerberos principal for the LDAP server: `ldap/<hostname>`
- Extract the principal's key to a keytab readable by the LDAP server.
- Define `KRB5_KTNAME` to the path of the keytab from within the OpenLDAP startup script (e.g. `KRB5_KTNAME=/etc/openldap/ldap.keytab`).
- If replicating: `slurpd` watches for changes and sends to replicas.
  - `slurpd` needs an active ticket.
  - Create a principle or use `host/<hostname>`.
  - Create a cronjob (for root) to refresh the ticket:

```
KRB5CCNAME=<file> /usr/kerberos/bin/kinit -k <principle>
```
  - `export KRB5CCNAME=<file>` (in `slurpd` startup)
  - Set `updatedn, updateref` in replica `slurpd.conf`
  - Configure `slurpd.conf` (`replica host=...`)

# Set up SSL/TLS

- LDAP w/ STARTTLS (port 389), LDAPS/SSL LDAP (port 636).
- Create a Certificate Authority and issue a cert:
  - Set up dir (serial, index.txt, openssl.cnf)
  - **Make CA:** `openssl req -new -x509 -keyout private/cakey.pem -out cacert.pem -days ... -config ../openssl.cnf`
  - **Make Cert:** `openssl req -nodes -new -x509 -keyout ldapkey.pem -out ldapreq.pem -days ... -config openssl.cnf`
  - **Generate Signing Request:** `openssl x509 -x509toreq -in ldapreq.pem -signkey ldapkey.pem -out tmp.pem`
  - **Sign Certificate:** `openssl ca -config openssl.cnf -policy policy_anything -out ldapcert.pem -infiles tmp.pem`
  - Delete request.

# LDAP SSL Certificate Madness cont.

- CN of the certificate must match the LDAP server hostname.
- `subjectAltName=DNS:ldap.abc.org,DNS:ldap-backup.abc.org`
- Put the CA certificate where clients and server can read.
- Put the server key and the certificate for OpenLDAP to read.
- Modify `/etc/openldap/slapd.conf`:  
`TLSCertificateFile /etc/openldap/certs/ldapcert.pem`  
`TLSCertificateKeyFile /etc/openldap/certs/ldapkey.pem`  
`TLSCACertificateFile /etc/openldap/cacerts/CA-ldap.pem`
- Configure startup script for `slapd -h "ldap:/// ldaps://"`

# Set up clients

- Make sure Cyrus SASL (w/ GSSAPI) is installed

- Configure `/etc/ldap.conf`:

```
host                ldap.abc.org ldap-backup.abc.org
base                dc=abc,dc=org
ssl                 start_tls
tls_checkpeer      yes
tls_cacertfile     /etc/openldap/cacerts/CA-ldap.pem
```

- Configure `/etc/openldap/ldap.conf`:

```
TLS_CACERTDIR     /etc/openldap/cacerts/
URI                ldaps://ldap.abc.org/
BASE               dc=abc,dc=org
```

- Run `c_rehash` in the CA certs directory

- Add LDAP to NSS – `passwd: ldap files`

# All together now

- Set up `nscd` – the name service cache daemon
- Test `getent passwd` or `getent group`
- `kinit -f <principle>, klist, kdestroy, kpasswd`
- `ldapsearch ... <expr>`
  - `-H ldap://<host>` or `-H ldaps://<host>`
  - `-ZZ: StartTLS`
  - `-x: Simple auth (no SASL)`
  - `-b: Bind DN (dc=abc ,dc=org)`
- Open the right ports:
  - Kerberos: 88 UDP, 749 TCP (remote admin)
  - LDAP: 389 TCP, 636 TCP (SSL)
  - NTP: 123 UDP

# Setting up sudo with LDAP

- Build sudo with LDAP enabled.
- Add the sudo.schema to OpenLDAP (include in slapd.conf).
- Make a sudoers subtree in your LDAP dir:

```
dn: ou=sudoers,dc=abc,dc=org
objectClass: organizationalUnit
ou: sudoers
```

- Add sudo entries under the sudoers OU:

```
dn: cn=entry1,ou=sudoers,dc=abc,dc=org
objectClass: sudoRole
cn: entry1
sudoUser: ...
sudoHost: ALL
sudoCommand: ALL
sudoCommand: !/bin/sh
sudoOption: !authenticate
sudoRunAs: root
```

# sudo/LDAP cont.

- Create a separate `ldap.conf` for sudo (compile-time option).

- Create an entry for access control:

```
dn: cn=sudoers,dc=abc,dc=org
objectClass: organizationalRole
objectClass: simpleSecurityObject
cn: sudoers
userPassword: {SSHA}QR3os.....
```

- Add ACLs to OpenLDAP config (in `slapd.conf`):

```
access to dn.subtree="ou=sudoers,dc=abc,dc=org"
    by dn.base="cn=sudoers,dc=abc,dc=org" read
    by dn.regex="uid=.* /admin,cn=GSSAPI,cn=auth" write
```

- Make `ldap.conf.sudo` mostly like `ldap.conf`, but 0440 and add:

```
...
binddn cn=sudoers,dc=abc,dc=org
bindpw ...
sudoers_base ou=sudoers,dc=abc,dc=org
```

# OpenSSH-LPK

- Patch against OpenSSH to read `authorized_keys` from LDAP.
- Modify `sshd_config`:  
`UseLPK yes`  
`LpkLdapConf /etc/ldap.conf`  
`LpkUserDN ou=People,dc=abc,dc=org`  
`LpkServerGroup ...`
- Add `sshPublicKey` user attributes and `ldapPublicKey` object class:  
`dn: uid=davidhi,ou=People,dc=abc,dc=org`  
`uid: davidhi`  
`objectClass: posixAccount`  
`objectclass: ldapPublicKey`  
`...`  
`sshPublicKey: ssh-rsa AAAA...`
- Augment OpenLDAP ACLs to allow users to modify their public keys:  
`access to attr=sshPublicKey`  
`by dn.regex="uid=.* /admin,cn=GSSAPI,cn=auth" write`  
`by self write`  
`by * read`

# Other uses to consider

- Kerberos:

- telnet, rsh, rlogin, ftp, X11
- Samba, NFSv4, AFS, cvs, Apache
- Anything using SASL (Dovecot, Postfix, Sendmail, etc.)

- LDAP:

- Samba
- PowerDNS with LDAP backend or ldapdns
- Qmail-LDAP, Postfix, Sendmail, Cyrus IMAP, etc.
- Apache, Squid, Zope, etc.
- MUAs for address book/white pages searches

# Resources/Bibliography

- Replacing NIS with Kerberos and LDAP HOWTO  
<http://www.ofb.net/~jheiss/krbldap/howto.html>
- OpenLDAP, OpenSSL, SASL and KerberosV HOWTO  
<http://www.bayour.com/LDAPv3-HOWTO.html>
- Building Powerful Central Authentication  
<http://nermus.its.ac.id/show/main.php?track0=3&track1=0&&howto=central-auth>
- Building a modern LDAP based security framework  
[http://dev.gentoo.org/~lcars/ldap/pacsec\\_2005.pdf](http://dev.gentoo.org/~lcars/ldap/pacsec_2005.pdf)
- Very brief introduction to create a CA and a CERT  
<http://www.sendmail.org/~ca/email/other/cagreg.html>
- Linux's nsswitch.conf manual page

# Resources/Bibliography cont.

- Sudo's README.LDAP file  
[http://www.gratisoft.us/sudo/readme\\_ldap.html](http://www.gratisoft.us/sudo/readme_ldap.html)
- OpenLDAP Software 2.3 Administrator's Guide  
<http://www.openldap.org/doc/admin23/>
- Kerberos 5 Release 1.5.1 Documentation  
<http://web.mit.edu/Kerberos/krb5-1.5/#documentation>
- OpenSSH LDAP Public Key Patch  
<http://www.opendarwin.org/en/projects/openssh-lpk/>
- Debian's Wiki: LDAP topic  
<http://wiki.debian.org/LDAP>
- Wikipedia's articles on NSS, Kerberos, LDAP, etc.