Open Directory & OpenLDAP

David M. O’Rourke
Engineering Manager

Overview

- Background on Apple's Open Directory Technology (8 minutes)
  - What is it
  - What is Directory Services
- How has Apple integrated OpenLDAP (20 minutes or less)
  - what has Apple added to OpenLDAP?
- Questions and Answers (remaining time)

Open Directory

- Open Directory is a technology name
  - Covers both client access technologies and server technologies
  - Integrates and promotes industry standard technologies
- Open Directory is built into Mac OS X & Mac OS X Server
  - Been there since 10.0
- Open Sourced as part of Darwin
  - http://developer.apple.com/darwin/projects/opendirectory/

What Is Directory Services

- Abstraction API for read/write access to system configuration and management data
  - Users, groups, mount records and others
  - Authentication abstraction

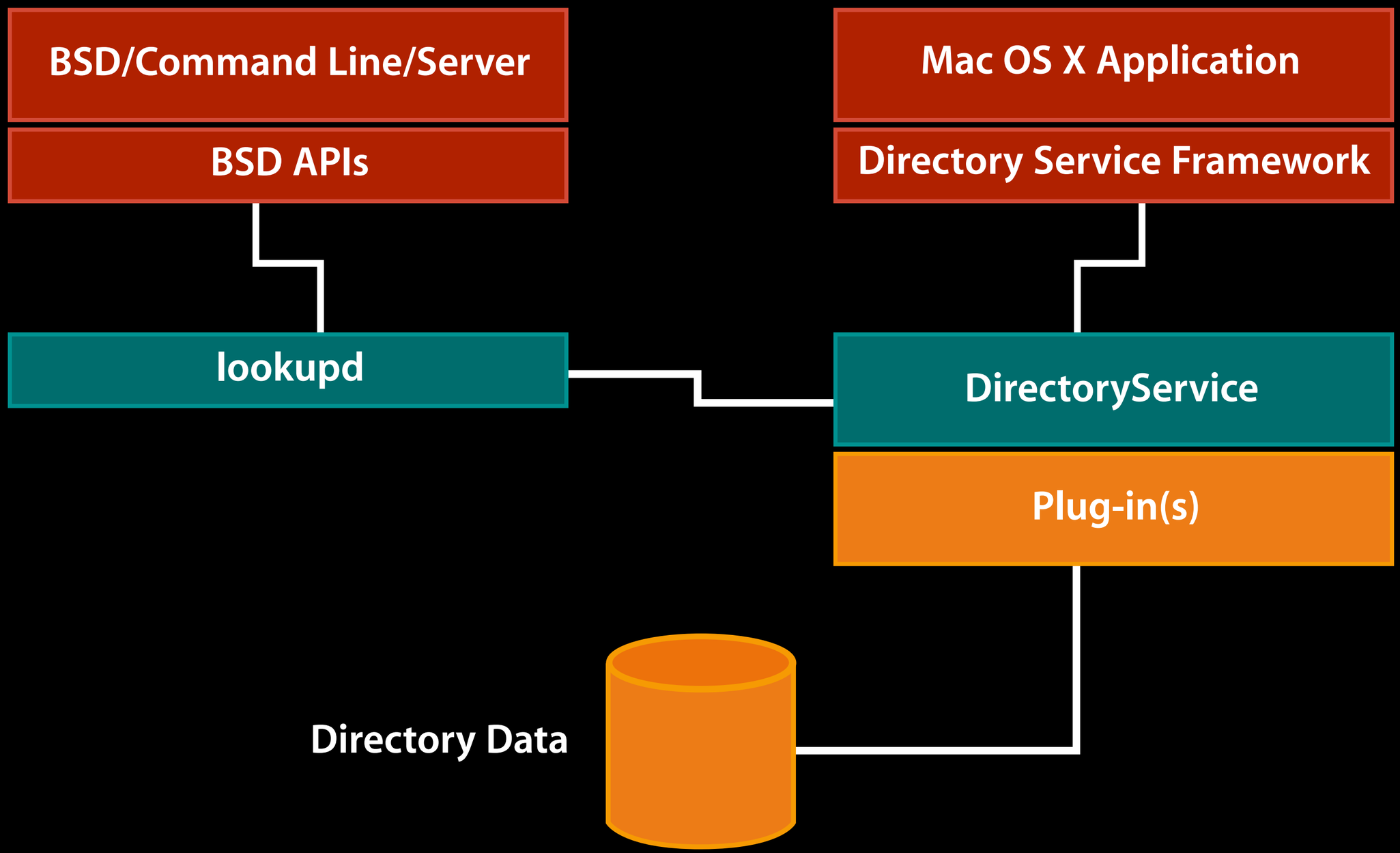
Directory Services in 10.3

- Includes
  - LDAPv3 (read/write), Native Active Directory, NetInfo, NIS, BSD/etc files
  - Service Discovery: Rendezvous, SMB, AppleTalk, and SLP
  - LDAPv3 client support replication fail over
- Documented Access API and plug-in API
  - SDK posted
    - Sample code, sample plug-in, notes
  - Directory Services Headers are installed in
    - /System/Library/Frameworks/DirectoryService.framework
  - Command line tool for directory editing ‘dscl’

10.3 Usage of Directory Services

- Login Window uses Directory Services for all user authentication
  - Managed Desktop
- All Security Framework authentication uses Directory Services
- Legacy Unix tools have been migrated to use PAM
  - Mac OS X default PAM module uses Directory Services
- Mac OS X Server processes and Administration Tools

How Does Mac OS X Use Directory Services?



Mac OS 10.3 Directory Server

- Mac OS 10.3 Server includes a complete Directory and Single Sign-on Authentication system
  - LDAPv3—based on OpenLDAP 2.1.22
  - Enhanced the Open Directory Password Server
    - SASL based network authentication
  - Added a fully integrated Kerberos Server
  - Support Windows Clients via integrated PDC
- Replication support for LDAP and Authentication data
- Apple is pursuing Kerberos as a unifying Single Sign-on Technology
  - Mac OS X Server all deploy Kerberos (customers don't even know it some cases)

Apple's Password Server

- Allows customers to support all major LAN network protocols and required authentication methods
  - Apple has invested in a secure password database
  - Supports secure replication
  - can sync passwords with an MIT KDC
  - enforces password policies
  - Allows customers to support both Kerberos and non-Kerberos based network protocols and applications
  - users have a single password across all network services/ platforms
- Future plans call for us to work with MIT to extend Kerberos for non-password based authentication

Password Server Authentication Methods Supported by Mac OS X

Authentication Method	Jaguar	Panther	Tiger	Protocol Client
MD5 Digest	✓	✓	✓	Default, Login Window, Clear Text
CRAM-MD5	✓	✓	✓	IMAP & SMTP
NT-Lan Manager	✓	✓	✓	SMB File Sharing
APOP	✓	✓	✓	POP3 Mail Protocol
WebDAV Digest	✓	✓	✓	WebDAV File System
MS Chap2		✓	✓	PPP, PPTP, VPN
2 Way Random	✓			AFP
DHX (Diffe-Helman Exhange)	✓	✓	✓	AFP, and Mac OS X Setpassword
NTLMv2			✓	SMB File Sharing

LDAPv3 Client Features

- 10.3 LDAPv3 is a robust LDAPv3 client
  - Support for DHCP LDAP server discovery
  - Server-based or client side LDAP mappings
  - Integrated support for Open Directory Password Server
  - Client side awareness of LDAP replicas
  - API transparent failover when necessary
  - read/write replication support
- Auto-discovery and configuration for Kerberos usage

OpenLDAP, Mac OS X Server and OpenDirectory

Jason Townsend
Directory Services Engineer
mailto:jtownsend@apple.com

Apple and OpenLDAP

- LDAP is Apple's network directory system of choice
- Must be trivial for our customers to deploy
- OpenLDAP is used for both client and server side LDAP support since Mac OS X 10.2
- Apple is investing in OpenLDAP feature set—we are adding support for:
  - Directory based schema and access controls
    - Changes will be presented to OpenLDAP project
    - Changes will also be posted on Darwin
- Mac OS X Server will also expose support for LDAP organization units (ou's) in the GUI tools

No secrets!

- All Apple modifications have been and will continue to be presented to OpenLDAP project
- Regardless of acceptance our changes are posted to Apple's darwin OpenSource repository
  - http://developer.apple.com/darwin/
  - http://developer.apple.com/darwin/projects/opendirectory/
  - http://developer.apple.com/darwin/projects/kerberos/
- Darwin is the OpenSource project that Mac OS X is based on.

Apple modifications summary

- Resolve any build issues on Mac OS X
- Full integration with the Apple Password Server
  - pass through of authentication data
- In directory schema and access controls
- slapconfig command line tool to automate standard setup process
  - GUI tools use slapconfig underneath
- configuration data stored in LDAP
- additional Apple schema
  - /Library/configuration/blah
- client side failover support in Directory Services LDAPv3 plug-in

Build support on Mac OS X

- symbol conflicts (sl_free)
- BIND 9 changes
- SASL header include
- Xcode project in OpenLDAP sources (for debugging)
  – autoconf/makefiles still used to generate shipping binaries

Integration with Password Server

- Password Server does not provide network access to either cleartext or hashes of the password
- The challenge is generated before the user is known
- To support secure authentication, challenge and response are relayed to Password Server for verification
  - accomplished using a custom SASL plug-in
- Cleartext authentication is proxied to a secure auth to Password Server

In-Directory Schema

- Storing schema directives in the directory
- Helpful for replicated environments
  - don't need to manually replicate .schema file changes
  - helps also in the case of remote-only administration
- schemaconfigdn directive indicates DN of schema record
  - attributetype directives become attributeTypesConfig values of this record
  - objectclass directives become objectClassesConfig values of this record
- modified files: servers/slapd/add.c, at.c, config.c, main.c, modify.c, oc.c, proto-slap.h, schema.c, schema_init.c, schema_prep.c, schemaparse.c, slap.h

In-Directory Schema Example

```
slapd.conf:
schemaconfigdn “cn=schema,cn=config,dc=example,dc=com”

LDIF:
dn: cn=schema,cn=config,dc=apple,dc=com
cn: schema
objectClass: top
objectClass: container
objectClass: extensibleObject
attributeTypesConfig: ( 2.16.840.1.113730.3.1.13 NAME 'mailLocalAddress'
  DESC 'RFC822 email address of this recipient' EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{256} )
attributeTypesConfig: ( 2.16.840.1.113730.3.1.18 NAME 'mailHost' DESC
  'FQDN of the SMTP/MTA of this recipient' EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{256} SINGLE-VALUE )
attributeTypesConfig: ( 2.16.840.1.113730.3.1.47 NAME 'mailRoutingAddress'
  DESC 'RFC822 routing address of this recipient' EQUALITY
  caseIgnoreIA5Match SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{256} SINGLE-VALUE)
objectClassesConfig: ( 2.16.840.1.113730.3.2.147 NAME
  'inetLocalMailRecipient' DESC 'Internet local mail recipient' SUP top
  AUXILIARY MAY ( mailLocalAddress $ mailHost $ mailRoutingAddress ) )
```

In-Directory Access Controls

- DN of access control record specified by access specified-in-directory directive
- attribute values include sequence number and access directive
  - 1000:access to * by * read
- Useful when using read access controls in a replicated environment
- modified files: servers/slapd/acl.c, aclparse.c, add.c, backend.c, config.c delete.c, main.c, modify.c, proto-slap.h, schema_prep.c, slap.h

In-Directory Access Controls Example

```
slapd.conf:
access specified-in-directory apple-acl
“cn=default,cn=accesscontrols,dc=example,dc=com”

LDIF:
dn: cn=default,cn=accesscontrols,dc=example,dc=com
cn: default
objectClass: apple-acl
objectClass: top
apple-acl-entry: 1000:access to attr=userPassword by self write by
  group/posixGroup/memberUid=”cn=admin,cn=groups,dc=example,dc=com” write
  by * read
apple-acl-entry: 1100:access to * by
  group/posixGroup/memberUid=”cn=admin,cn=groups,dc=example,dc=com” write
  by * read
```

slapconfig command line tool

- Manages the Open Directory server (including slapd and slurpd)
- Enables KDC when creating an Open Directory master
- Provides single commands for:
  - creating an Open Directory master
  - adding a replica
  - removing a replica
- Server Admin uses slapconfig as well so all GUI functionality is available thorugh the command line
- Log file is /Library/Logs/slapconfig.log

Example slapconfig commands

```
% slapconfig -createldapmaster <admin name>
Password:

% slapconfig -createreplica <master IP address>
<admin name>
Password:

% slapconfig -destroyldapserver
```

Other Configuration Data in LDAP

- OpenDirectory LDAP Server mappings
  - stored as an organizationalUnit record named macosxodconfig
  - description has XML plist for Directory Services LDAPv3 plug-in
  - needed for DHCP provided LDAP server (option 95)

Example Server Mappings Record

```
dn: ou=macosxodconfig,cn=config,dc=apple,dc=com
ou: macosxodconfig
objectClass: top
objectClass: organizationalUnit
description:: PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPCFET0NUWVBF
 IHBsaXN0IFBVQkxJQyAiLS8vQXBwbGUgQ29tcHV0ZXIvL0RURCBQTElTVCAxLjAvL0VOIiAiaHR0c
...
 N0cmluZz50b3duc2VuZDk8L3N0cmluZz4KPC9kaWN0Pgo8L3BsaXN0Pgo=

description is an XML plist:
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN" "http://
www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        <key>Attribute Type Map</key>
        <array>
...
```

Other Configuration Data in LDAP

- Replica lists to support client side failover
  - LDAP is in ldapreplicas config record
  - passwordserver config record
  - KerberosClient record

Example ldapreplicas record

```
dn: cn=ldapreplicas,cn=config,dc=example,dc=com
cn: ldapreplicas
apple-ldap-replica: ldap://17.221.43.149
apple-ldap-replica: ldap://17.221.43.141
apple-ldap-writable-replica: ldap://17.221.43.149
objectClass: apple-configuration
objectClass: top
```

Other Configuration Data in LDAP

- KerberosClient record also used to automatically populate client's Kerberos config file
- KerberosKDC record contains contents of kdc.conf
  - used when adding a new replica
- Also have a method for storing key-tab information securely in LDAP (requires password server)
  - allows for delegated Server administration - File server administrator doesn't have to be a Directory administrator to add a new Kerberos service

KerberosClient Record

```
<plist version="1.0">
<dict>
        <key>edu.mit.kerberos</key>
        <dict>
                <key>domain_realm</key>
                <dict>
                        <key>.apple.com</key>
                        <string>ODDJOB.APPLE.COM</string>
                        <key>apple.com</key>
                        <string>ODDJOB.APPLE.COM</string>
                </dict>
                <key>libdefaults</key>
                <dict>
                        <key>default_realm</key>
                        <string>ODDJOB.APPLE.COM</string>
                </dict>
                <key>realms</key>
                <dict>
                        <key>ODDJOB.APPLE.COM</key>
                        <dict>
                                <key>KADM_List</key>
                                <array>
                                        <string>oddjob.apple.com</string>
                                </array>
                                <key>KDC_List</key>
                                <array>
                                        <string>oddjob.apple.com</string>
                                </array>
                        </dict>
                </dict>
        </dict>
        <key>generationID</key>
        <integer>1143721936</integer>
</dict>
</plist>
```

KerberosKDC Record

```
[kdcdefaults]
        kdc_ports = 88

[realms]
        ODDJOB.APPLE.COM = {
                kadmind_port = 749
                max_life = 10h 0m 0s
                max_renewable_life = 7d 0h 0m 0s
                master_key_type = des3-hmac-sha1
                supported_enctypes = des3-hmac-sha1:normal
arcfour-hmac-md5:normal des-cbc-crc:normal des-cbc-crc:v4
                kdc_supported_enctypes = des3-hmac-sha1:normal
arcfour-hmac-md5:normal des-cbc-crc:normal des-cbc-crc:v4
                acl_file = /var/db/krb5kdc/kadm5.acl
                admin_keytab = /var/db/krb5kdc/kadm5.keytab
                database_name = /var/db/krb5kdc/principal
        }
[logging]
        kdc = FILE:/var/log/krb5kdc/kdc.log
        admin_server = FILE:/var/log/krb5kdc/kadmin.log
```

Additional Apple schema

- Stored in /etc/openldap/schema/apple.schema
- Used to support Mac OS X/Mac OS X Server features
  - Managed desktop
  - AFP home directories
- OID prefix: 1.3.6.1.4.1.63

Future Ideas for OpenLDAP

What Apple thinks we need

- improved handling of upgrade scenarios
  - dump to LDIF and reimport can be expensive on large databases
- incremental indexing - better performance for adding to large databases
- unification with authentication services (Password Server/ SASL and Kerberos)
  - three databases currently which must be kept in sync

Q&A

Optional material follows

Kerberos Auth Authority

- Used by loginwindow to determine if the user is Kerberized
- ;Kerberosv5;[guid];[principal];realm;[realm key]
- Minimum
  - ;Kerberosv5;;;FOO.APPLE.COM;
  - Principal name will be user_shortname@FOO.APPLE.COM
- Recommended
  - Kerberosv5;;user@FOO.APPLE.COM;FOO.APPLE.COM;
  - May want to add the principal name as an additional shortname for the user