Mac OS X Security

Leon Towns-von Stauber, Occam's Razor

LISA 2003

`http://www.occam.com/osx/`

Contents

X Opening Remarks

- I'm assuming basic familiarity with UNIX operating system design, user experience with Mac OS X, and some OS X sysadmin-level knowledge
  - Account management, filesystems, network services, etc.
- Where I'm coming from:
  - UNIX user and some-time admin since 1990
  - Full-time UNIX admin since 1995
  - NeXTstep/OS X user and admin since 1991
- This presentation primarily covers:
  - Mac OS X 10.2.6 (Darwin 6.6)
  - Mac OS X Server 10.2.6 (Darwin 6.6)
  - Includes some updates for Panther, Mac OS X 10.3 (Darwin 7.0)

Legal Notices

X Out-of-the-Box

- Apple's customers were used to a secure out-of-the-box experience
  - Legacy Mac OS offered very few services, and was rarely attacked
- As a result, Mac OS X has the most secure default configuration of any major UNIX platform
  - Finally, a UNIX vendor that takes security seriously!
- Almost no network services are enabled by default
  - Exceptions: NTP (UDP 123), Rendezvous multicast DNS (UDP 53, 5353)
    - Not 5353 in Panther
- The `root` account is disabled (i.e., the password is invalidated)
  - All superuser access requires authentication as a non-`root` user first

Out-of-the-Box

- NB: Neither of these points is true for Mac OS X Server
  - SSH, Server apps, SLP, QTSS can all be listening after a fresh install
  - `root` is sometimes necessary (e.g., authenticating in NetInfo Manager after enabling Password Server)
    - Panther: NetInfo Manager can authenticate Password Server users, so a valid `root` password may no longer be needed

Open Directory

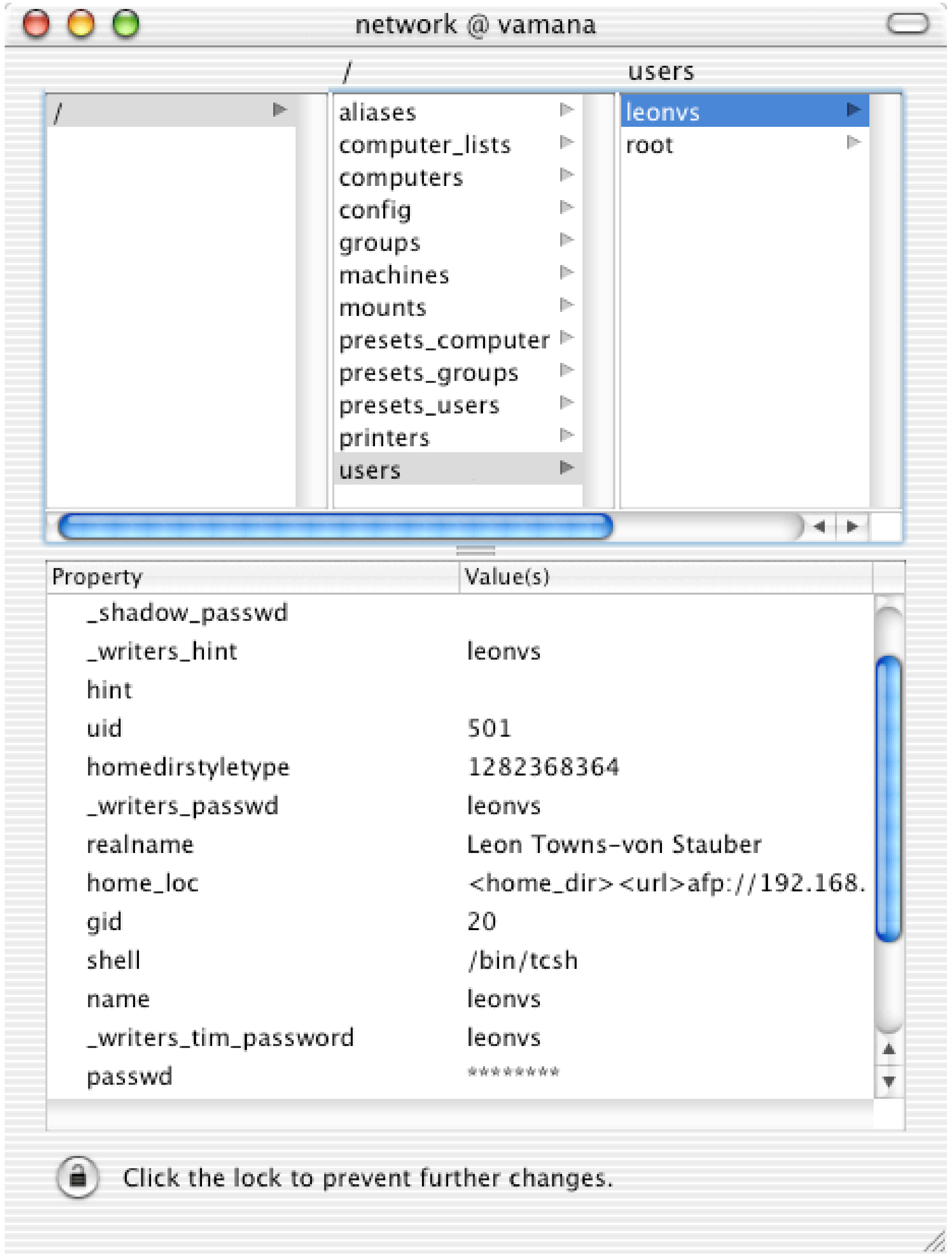
- Introduction
- NetInfo
- LDAP

Introduction

- "Open Directory" is a vague umbrella term referring to Apple's implementation of various directory services in Mac OS X
- I'm using the term to refer to the collection of software based on Open Directory domains, accessed by either NetInfo or LDAP, with on-disk data formatted as key/value pairs (e.g., NetInfo DB, Berkeley DB)
  - This is a subset of the full suite of directory software on OS X, which includes the Directory Services API, `lookupd`, DNS (via BIND), WINS (via Samba), NIS compatibility, service discovery protocols, etc.
- Directory services provide system config data, usually over a network
  - User accounts, file and printer sharing configuration, host info, etc.

Introduction

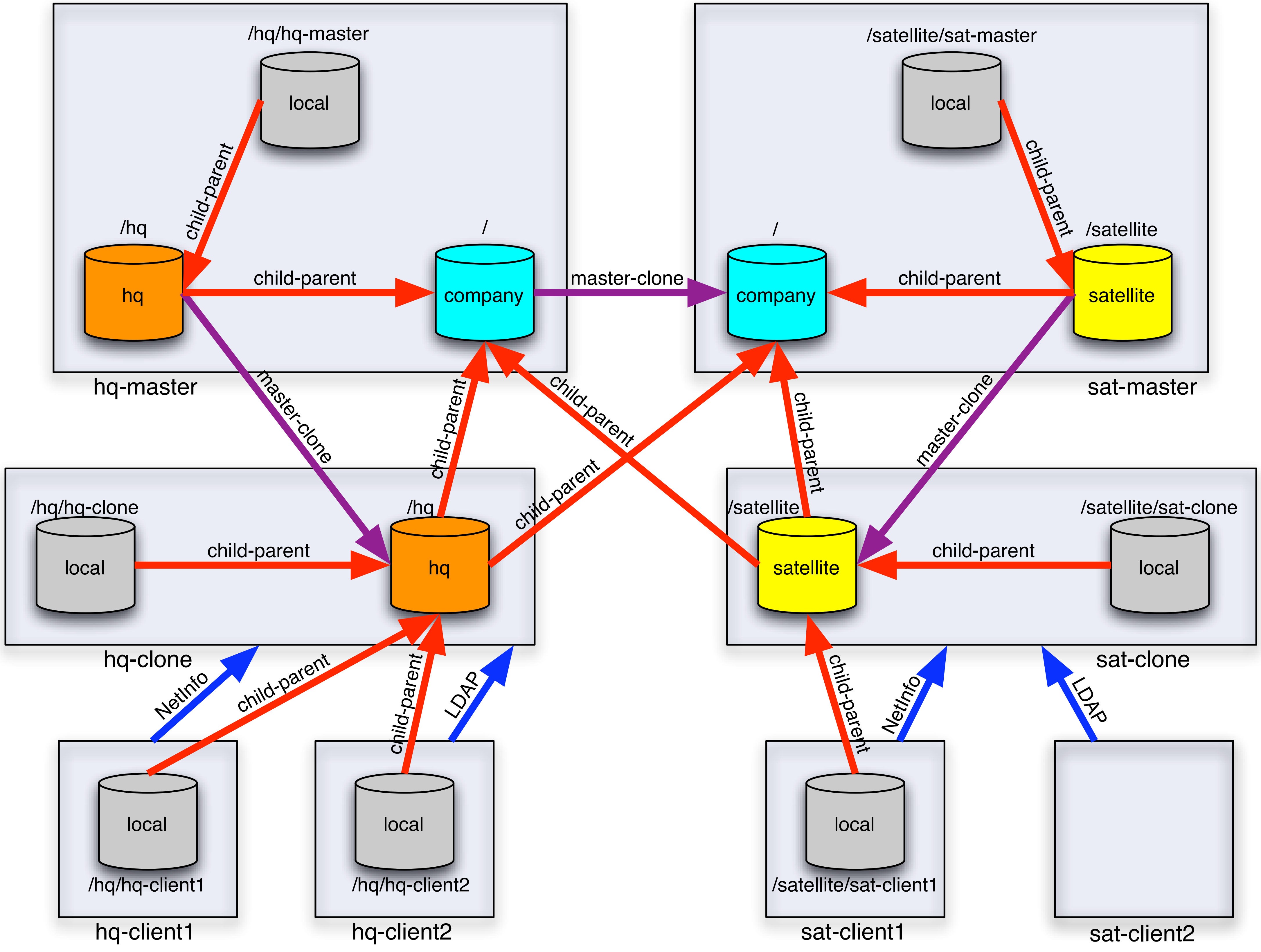
- The data in each domain is served from a single master database file
- Databases may be cloned to other servers
- Domains may have parent/child relationships, creating a hierarchy
- Every Mac OS X system hosts a `local` database, which may be a leaf node in a hierarchy
- Clients bind to servers using either NetInfo (NeXT legacy protocol) or LDAPv3
- Lookups go up through hierarchy
- Data within database consists of property/value pairs organized in filesystem-like directory hierarchy
    - E.g., my account data is in `/users/leonvs`, with properties like `name`, `passwd`, `shell`, etc.



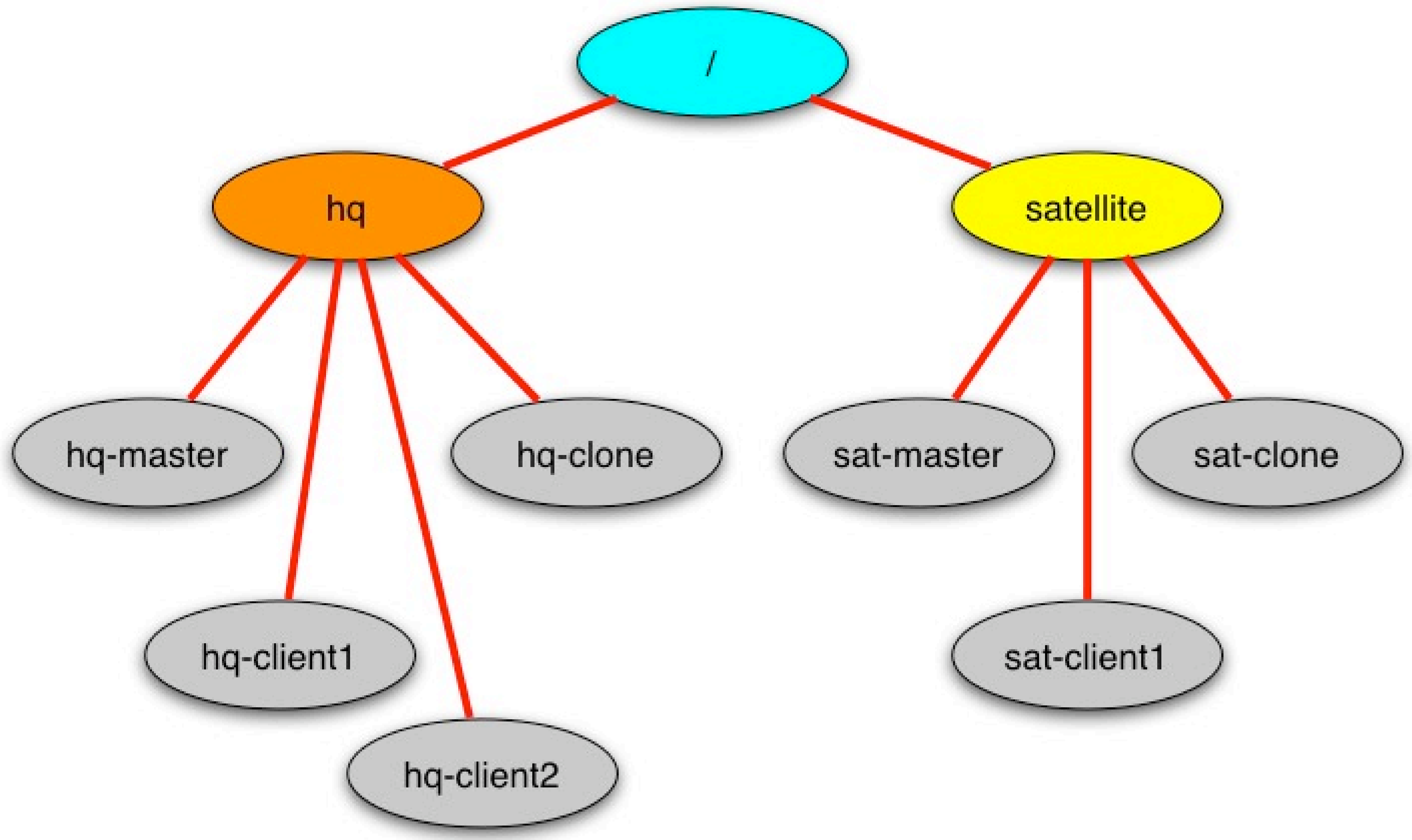
Property	Value(s)
_shadow_passwd	
_writers_hint	leonvs
hint	
uid	501
homedirstyletype	1282368364
_writers_passwd	leonvs
realname	Leon Towns-von Stauber
home_loc	<home_dir><url>afp://192.168.
gid	20
shell	/bin/tcsh
name	leonvs
_writers_tim_password	leonvs
passwd	********

Contents of Open Directory database, shown in NetInfo Manager

- The diagrams on the following two slides illustrate a typical three-tier OD hierarchy
  - `hq-master` hosts master databases (tagged `company` and `hq`) for the `/` and `/hq` domains
  - `sat-master` hosts master DB (tagged `satellite`) for `/satellite` domain, and cloned DB (tagged `company`) for `/` domain
  - `hq-clone` hosts cloned DB (tagged `hq`) for `/hq` domain
  - `sat-clone` hosts cloned DB (tagged `satellite`) for `/satellite` domain
  - Each host (except for non-OS X host lower right) hosts own `local` DB, bound into domain hierarchy under `/hq` or `/satellite`
  - `hq-client1` and `sat-client1` talk to local servers using NetInfo
  - `hq-client2` and `sat-client2` talk to local servers using LDAP



Sample three-tier Open Directory domain hierarchy

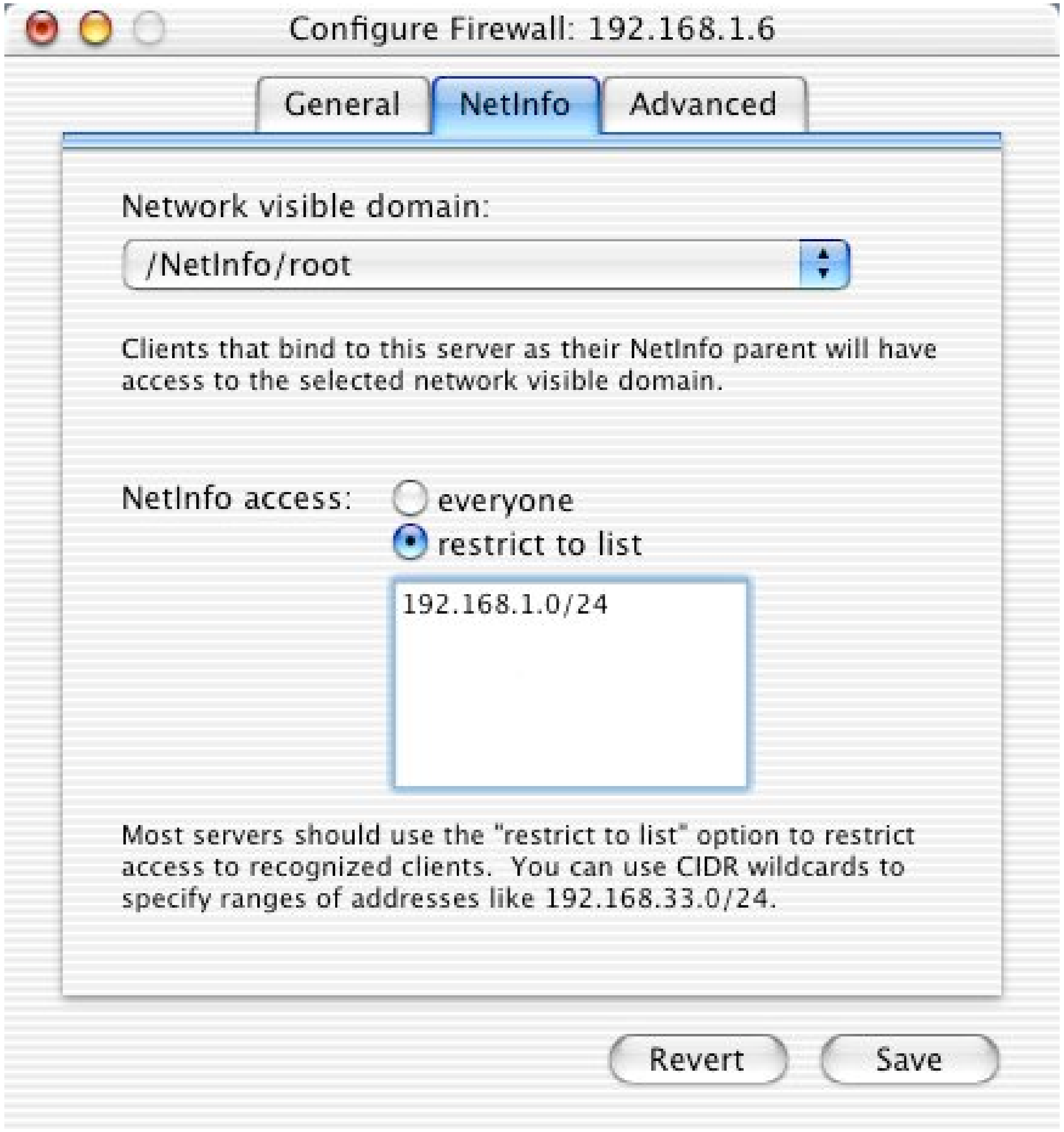


Sample three-tier Open Directory hierarchy: logical domain structure

NetInfo

- Restrict access by network address with `trusted_networks` property in root directory
  - Takes list of partial dotted-decimal addresses, or names defined in `/networks`
- Configure `netinfod` to run on specified ports with `port` property in root directory
  - Specify `tcp_port` and `udp_port` if you want them to be different
  - Normally `netinfod` binds to arbitary ports between 600 and 1023
    - Clients talk to `nibindd` to determine `netinfod` port numbers
  - Useful for firewalls
  - `nibindd` itself still listens on arbitrary ports, known to RPC portmapper
  - Mac OS X Server can limit access to NetInfo ports, even if dynamic
    - No more GUI for this in Panther Server

NetInfo



Network visible domain: /NetInfo/root

Clients that bind to this server as their NetInfo parent will have access to the selected network visible domain.

NetInfo access: everyone [✓ restrict to list]

192.168.1.0/24

Most servers should use the "restrict to list" option to restrict access to recognized clients. You can use CIDR wildcards to specify ranges of addresses like 192.168.33.0/24.

Limiting NetInfo access in Mac OS X Server Firewall config

NetInfo

- Access to domain data via NetInfo is read-only by default, unless you authenticate as the superuser, an administrative user, or a domain admin
  - Grant write access to properties with `_writers_`*`property`* property
    - E.g., `_writers_passwd:` *`username`*
  - Plain `_writers` property grants write access to all properties in the directory
  - `username` can be a list, or `*` (granting access to everyone)

NetInfo

- Shortcomings
  - No built-in support for secure network communications
    - Parent/child and master/clone interactions are unauthenticated (susceptible to rogue servers) and in the clear (susceptible to snoopers)
    - Use IPsec
  - All data is readable by any client, including password hashes
    - Use Password Server, or LDAP ACLs
    - Panther: By default, password hashes are now shadowed

Open Directory LDAP

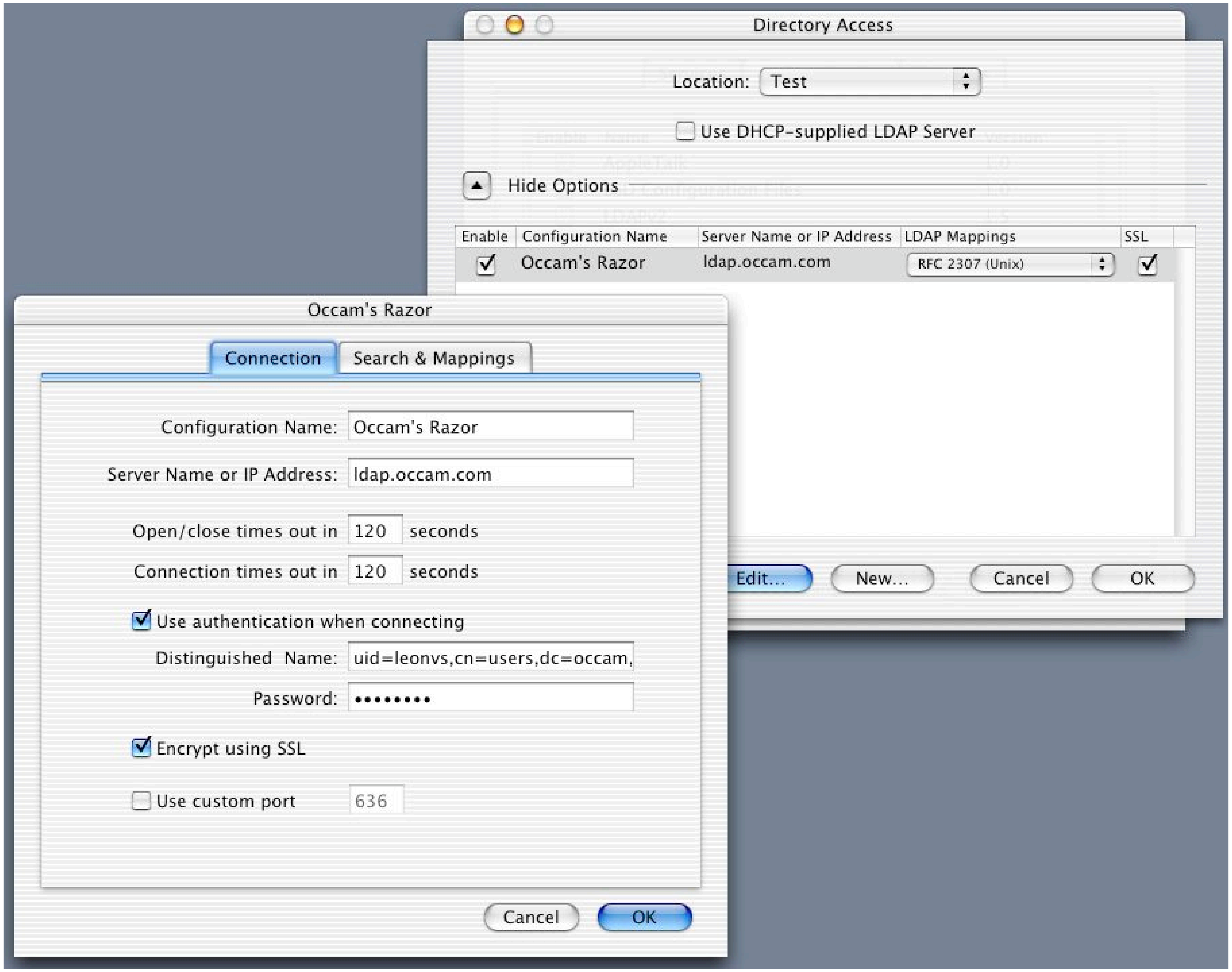
- OpenLDAP with an Open Directory database backend
- Set up access control lists in `/etc/openldap/slapd.conf`
  - Especially useful if all clients use LDAP instead of NetInfo
  - If the `DSENGINE_FLAGS_NATIVE_AUTHORIZATION` flag is set (which it is by default), read access can be restricted, but write access is overridden by the usual NetInfo behavior, as described earlier
- Increase strength of password hash by replacing `CRYPT` cipher in `userPassword attributemap` at end of `/etc/openldap/schema/netinfo.schema` with `SSHA`
  - Use `slappasswd -h '{SSHA}'` to generate initial password hashes
    - `nicl /`*`domain`*` -create /users/`*`username`*` passwd '`*`hash`*`'` to add
  - Users change passwords with `ldappasswd -x -W -S`
  - Useful if users authenticate <u>only</u> through LDAP

LDAP

- Setting up SSL/TLS on server
  - 1) Self-signed CA cert: `openssl req -x509 -new -out ca_cert.pem -keyout ca_key.pem -days 3650`
  - 2) CSR: `openssl req -new -out req.pem -keyout tmp_key.pem`
  - 3) Strip passphrase: `openssl rsa -in tmp_key.pem -out key.pem`
  - 4) Sign: `openssl x509 -req -CA ca_cert.pem -CAkey ca_key.pem -CAcreateserial -in req.pem -out cert.pem -days 365`
  - 5) Configure `TLSCACertificateFile`, `TLSCertificateFile`, and `TLSCertificateKeyFile` in `/etc/openldap/slapd.conf`
  - 6) Start server as `slapd -h ldaps:///` in `/System/Library/StartupItems/LDAP/LDAP`
  - 7) `SystemStarter stop LDAP; SystemStarter start LDAP`
  - 8) Optional: Add `ldaps` entry (port 636) to `/etc/services`

LDAP

- Setting up SSL/TLS on client
  - 1) Copy `/etc/openldap/ca_cert.pem` from server to client
  - 2) Configure `TLS_CACERT` in `/etc/openldap/ldap.conf`
  - 3) Configure `URI` as `ldaps://server` in `/etc/openldap/ldap.conf`
  - 4) In Directory Access app, enable LDAPv3 and Configure...; enable SSL
  - 5) Also in Directory Access, under Authentication, add the LDAPv3 directory to a Custom search path
  - 6) Restart Directory Service server: `killall DirectoryService`



Location: Test

☐ Use DHCP-supplied LDAP Server

Enable	Configuration Name	Server Name or IP Address	LDAP Mappings	SSL
✓	Occam's Razor	ldap.occam.com	RFC 2307 (Unix)	✓

Configuration Name: Occam's Razor

Server Name or IP Address: ldap.occam.com

Open/close times out in 120 seconds

Connection times out in 120 seconds

☑ Use authentication when connecting

Distinguished Name: uid=leonvs,cn=users,dc=occam,

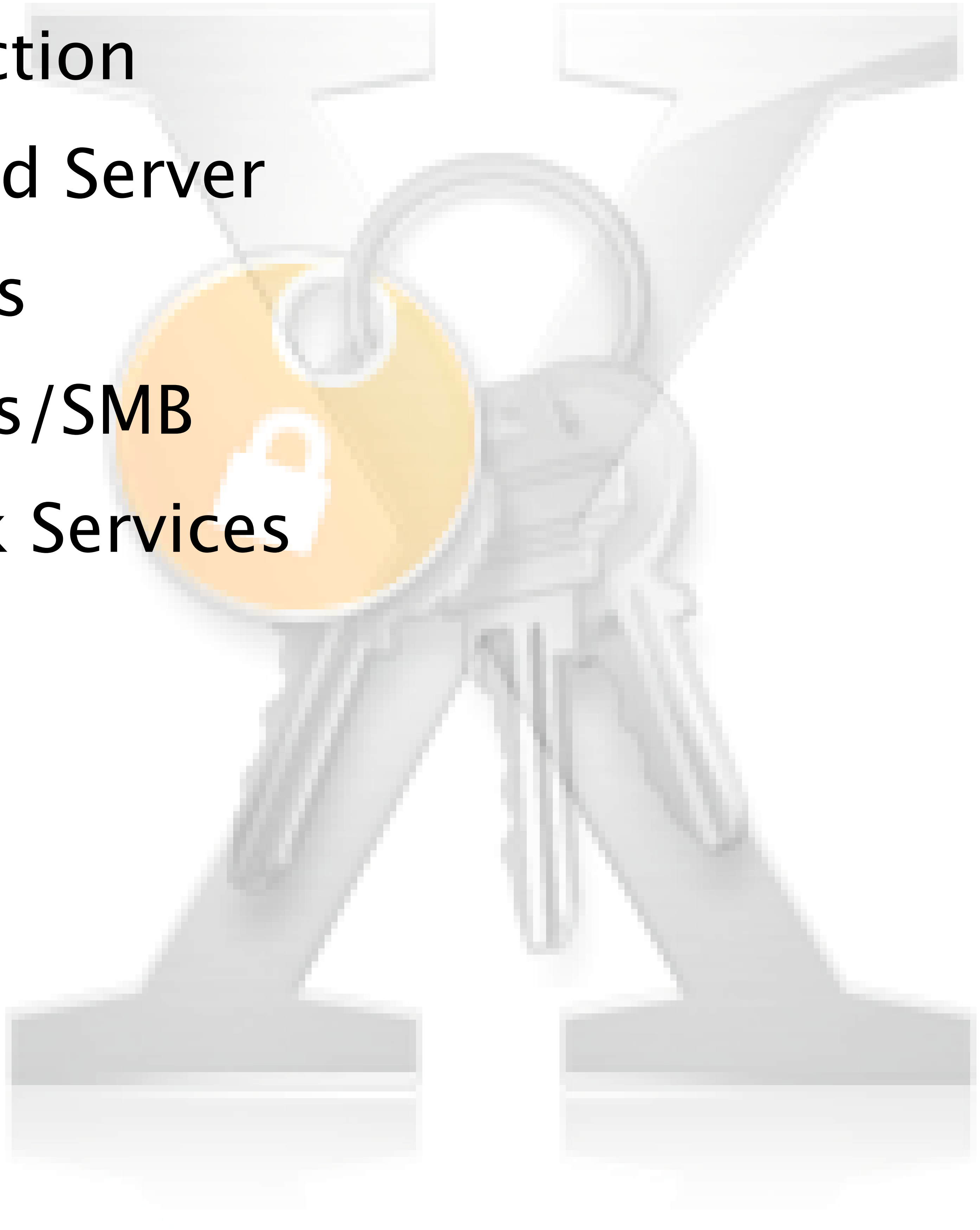
Password: ••••••••

☑ Encrypt using SSL

☐ Use custom port 636

Enabling LDAP over SSL in Directory Access

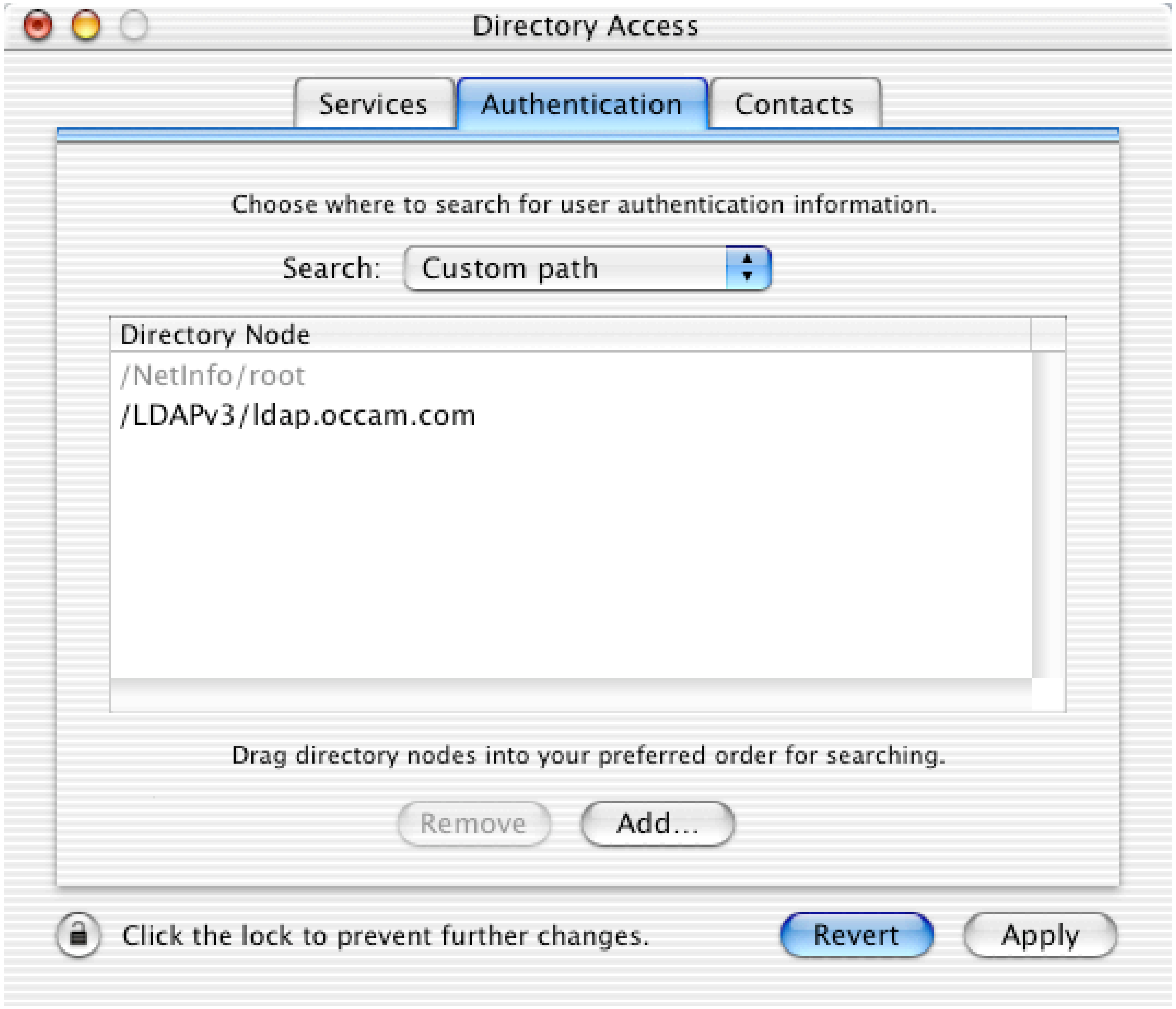
Authentication and Authorization



- Introduction
- Password Server
- Kerberos
- Windows/SMB
- Network Services

Introduction

- Mac OS X supports a variety of authentication methods
- Order of use is determined in the Directory Access application
- When NetInfo is the source, the key is the `authentication_authority` property in a user's profile
  - `Basic` (the default): `crypt` password hash
    - As already noted, no shadowing
  - `ApplePasswordServer`
  - `LocalWindowsHash`
  - `ShadowHash` (Panther)
- PAM supports authentication based on Directory Services for command-line and cross-platform UNIX programs



Choose where to search for user authentication information.

Search: Custom path

Directory Node
/NetInfo/root
/LDAPv3/ldap.occam.com

Drag directory nodes into your preferred order for searching.

Click the lock to prevent further changes.

Authentication search order in Directory Access

Password Server

- Bundled with Mac OS X Server
- Based on Cyrus SASL
- Supports many auth protocols: CRAM-MD5, APOP, NT, LM, Digest (for Apache), etc.
- Can set password policies (minimum length, expiration time, etc.)
- "Passwords go in, but they don't come out"
  - Some protocols, such as APOP, compromise this ideal by requiring access to decryptable passwords
  - `NeST -getprotocols` lists these as `Plain` (as opposed to `Hash`)
- In a user's profile, no password hash, just a 128-bit ID indexed to an entry in Password Server
  - Server IP identified in OD domain by `/config/passwordserver`
- Usually enabled with Open Directory Assistant; also with `NeST`

Password Server

- User credentials stored in `/var/db/authserver/authservermain`
- Planning issues
  - Does not support loginwindow on pre-10.2 clients
  - Enable lower-security protocols like APOP only if necessary
  - No built-in replication
    - Yikes!
    - Make your server as highly available as you can
    - Consider regular sync of `authservermain` to warm standby
    - Consider keeping one or more local admin accounts out of PS
    - Panther: Can replicate Password Server

Password Server

- Sometimes authentication as a PS admin stops working
  - Corrupted PS DB? May be related to sudden loss of network access.
  - Following procedure also applies if PS admin password is lost.
    - 1) Login to Password Server system.
    - 2) `su -`
    - 3) `nicl . -create /users/`*`username`* `authentication_authority ';Basic;'`
    - 4) `NeST -NOpasswordserver`
    - 5) `NeST -hostpasswordserver` *`username password`*

Kerberos

- Panther makes much more use of Kerberos, as part of its “Single Sign-On” features
  - Following information may only partially apply to Panther
- Using Kerberos (v5)
  - Set up user accounts with Basic authentication
    - Username must be same as Kerberos principal name
  - Copy the keytab file from the Kerberos server to `/etc/krb5.keytab`
  - Copy `edu.mit.Kerberos` to `/Library/Preferences/`, and to `~/Library/Preferences/` for each user
    - Sample file at `/System/Library/Frameworks/Kerberos.framework/Resources/`

Kerberos

- Setting up Kerberos clients
  - Login Window
    - Make Kerb auth mandatory by changing value for `system.login.console` in `/etc/authorization`: replace `authinternal` with `krb5auth:authenticate`
    - Make Kerb auth a side effect of logging in by changing value for `system.login.done`: add `krb5auth:login`
    - Syntax has changed on Panther; check comments in `/etc/authorization` for details
  - Mail application: in Accounts Preferences, select Kerberos authentication for SMTP, POP, or IMAP
  - Automatic for AFP, FTP, TELNET?

Kerberos

- Setting up Kerberos services on Mac OS X Server
  - Apple Mail Server: in Server Settings, under the Mail Service General preferences, select either Kerberos (to make it mandatory) or Any (to make it optional) authentication
    - Correspond to `kerberos_option_value` of `2` or `3`, respectively, under `/config/AppleMailServer` in the `local` OD domain
  - AFP: in Server Settings, under Apple Access preferences
    - `authentication_mode` of `kerberos` or `standard_and_kerberos` under `/config/AppleFileServer` in the `local` domain
  - FTP: in Server Settings, under FTP Advanced preferences
    - `auth_level` of `gssapi` or `both` in `/Library/FTPServer/Configuration/ftpaccess`

Authentication Windows/SMB

- Several options to provide authentication services to Windows clients
- Password Server: with NT or LanMan protocols
- Local Windows Hash
  - Method used when you check "Allow user to log in from Windows" under user account settings in System Preferences
  - Unlike Password Server, available in vanilla OS X
  - Stores password hash in `/var/db/samba/hash/`*`username`*
  - No longer needed on Panther
- Authentication Manager
  - Method to support Windows in 10.0 and 10.1, before Password Server
  - Stored encrypted password in user's `tim_passwd` property

Authentication Windows/SMB

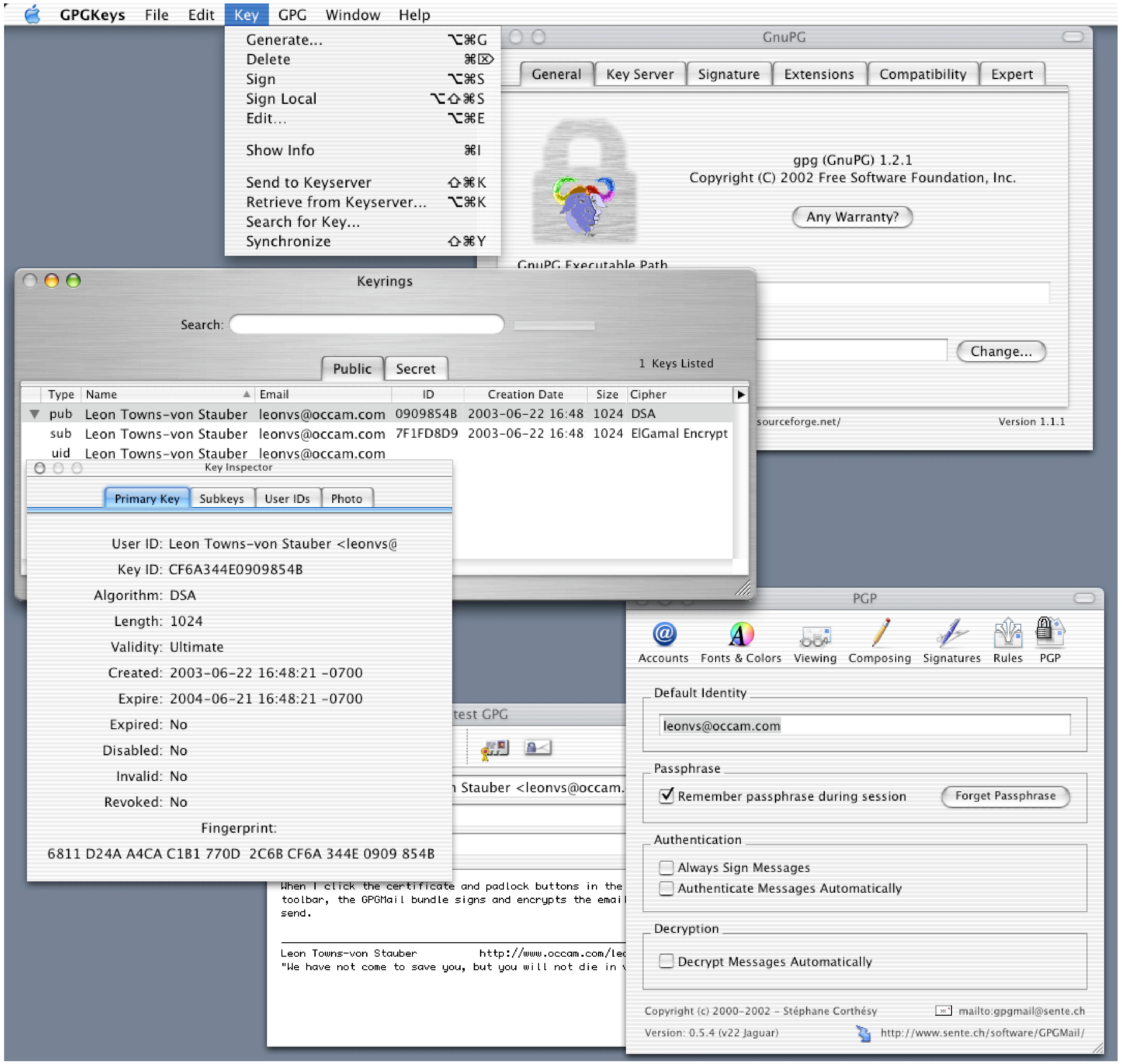
- Samba offers its own authentication options
  - Configured in `/etc/smb.conf`
    - Authentication schemes: `encrypt passwords`, `security`, `password server`, `lanman auth`, `min protocol`
    - Don't use these: `null passwords`, `hosts equiv`, `use rhosts`
    - Guest access: `public`, `map to guest`, `guest account`, `guest only`
    - When acting as PDC: `domain admin group`, `domain guest group`
    - User-based authorization: `username map`, `force user`, `force group`, `valid users`, `invalid users`, `admin users`, `printer admin`, `read list`, `write list`

X Authentication — Windows/SMB

- On Mac OS X, Samba is modified to use Directory Services for authentication
  - Panther: Samba modifications replaced by use of external modules
    - `auth methods` directive specifies `opendirectory` (which makes use of `/etc/auth/opendirectory.so`)
    - `passdb backend` directive specifices `opendirectorysam`

- Email
  - Mail application
    - Kerberos, MD5, APOP options for SMTP, POP, IMAP
    - Sign (and/or encrypt) messages with GPG (GNU Privacy Guard)
      - Open implementation of PGP (Pretty Good Privacy)
      - MacGPG (`http://macgpg.sourceforge.net/`), GPGMail (`http://www.sente.ch/software/GPGMail/`)
      - How-to:
        - 1) Download and unpack MacGPG quarterly release
        - 2) Install GnuPG and GPGPreferences packages, and GPGKeys application
        - 3) Open GnuPG pane in System Preferences to create default config files in `~/.gnupg/`

- Email (cont'd.)
  - Mail application (cont'd.)
    - GPG (cont'd.)
      - How-to (cont'd.):
        - 4) Generate key pair, either with GPGKeys or with `gpg -gen-key`
        - 5) Transform public key to ASCII with GPGKeys->File->Export->Key..., and distribute
        - 6) Install GPGMail from MacGPG `Contributed Software/` folder
        - 7) `defaults write com.apple.mail EnableBundles YES`
        - 8) Restart Mail to get new PGP preferences, menu items, and toolbar icons



gpg (GnuPG) 1.2.1
Copyright (C) 2002 Free Software Foundation, Inc.

sourceforge.net/ Version 1.1.1

1 Keys Listed

Type	Name	Email	ID	Creation Date	Size	Cipher
pub	Leon Towns-von Stauber	leonvs@occam.com	0909854B	2003-06-22 16:48	1024	DSA
sub	Leon Towns-von Stauber	leonvs@occam.com	7F1FD8D9	2003-06-22 16:48	1024	ElGamal Encrypt
uid	Leon Towns-von Stauber	leonvs@occam.com				

User ID:	Leon Towns-von Stauber <leonvs@
Key ID:	CF6A344E0909854B
Algorithm:	DSA
Length:	1024
Validity:	Ultimate
Created:	2003-06-22 16:48:21 -0700
Expire:	2004-06-21 16:48:21 -0700
Expired:	No
Disabled:	No
Invalid:	No
Revoked:	No

Fingerprint:
6811 D24A A4CA C1B1 770D 2C6B CF6A 344E 0909 854B

Stauber <leonvs@occam.

```
When I click the certificate and padlock buttons in the
toolbar, the GPGMail bundle signs and encrypts the emai
send.

Leon Towns-von Stauber          http://www.occam.com/le
"We have not come to save you, but you will not die in
```

Default Identity
leonvs@occam.com

Passphrase
☑ Remember passphrase during session

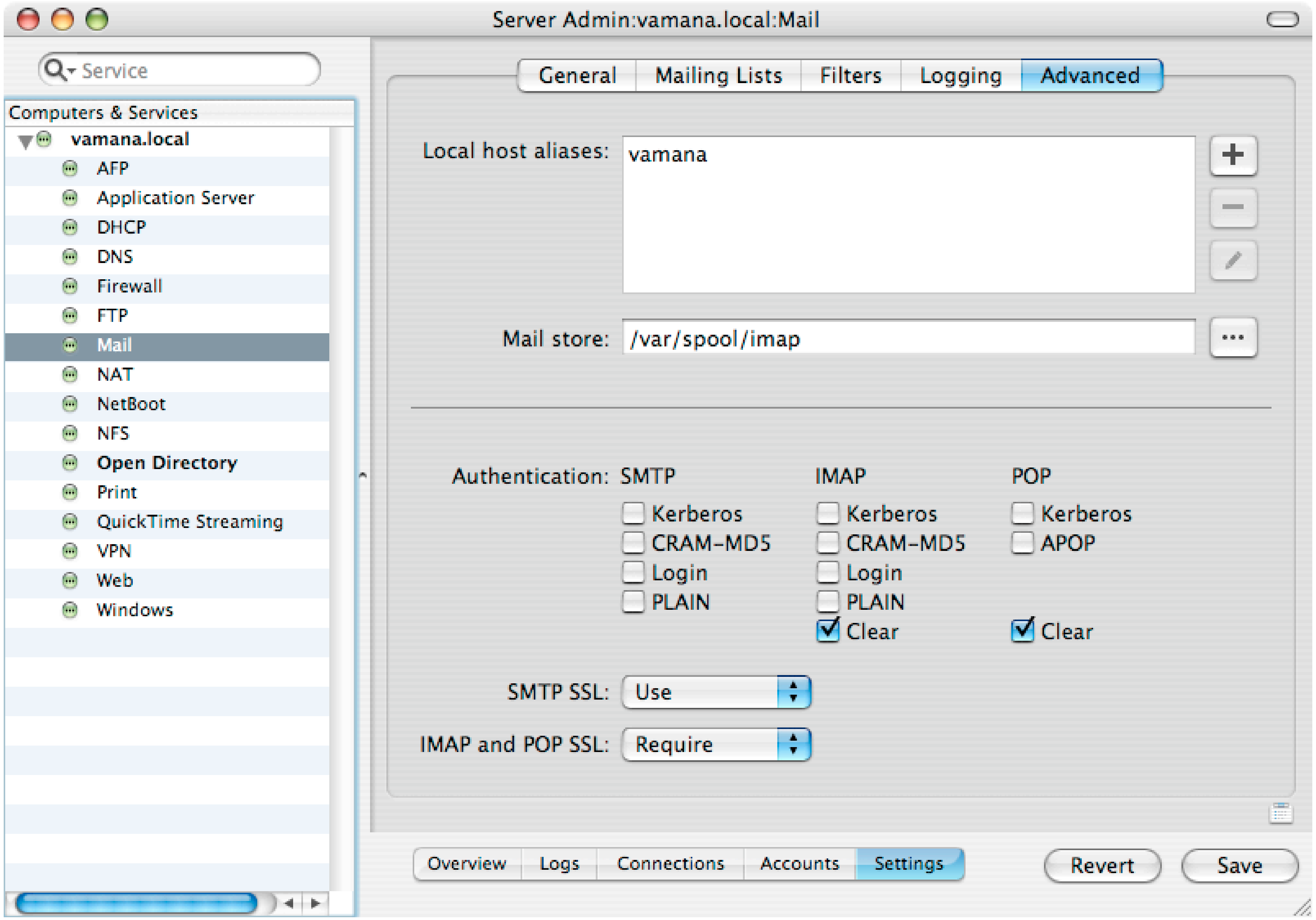
Authentication
☐ Always Sign Messages
☐ Authenticate Messages Automatically

Decryption
☐ Decrypt Messages Automatically

Copyright (c) 2000-2002 - Stéphane Corthésy mailto:gpgmail@sente.ch
Version: 0.5.4 (v22 Jaguar) http://www.sente.ch/software/GPGMail/

GPGKeys, GnuPG Preferences, GPGMail

- Email (cont'd.)
  - Apple Mail Server
    - Kerberos, MD5, APOP options for SMTP, POP, IMAP
  - Sendmail
- Web and Multimedia
  - Apache
    - Configured in `/etc/httpd/httpd.conf`
      - `AuthType`, `Auth*File`, `AuthName`, `Require`, `Satisfy`, `Anonymous_*`, `AllowOverride`, `AuthConfig`, `AccessFileName`
    - `mod_auth_apple` and `mod_digest_apple` modules (included with Mac OS X Server) allow authentication based on Directory Services



Local host aliases: vamana

Mail store: /var/spool/imap

Authentication:

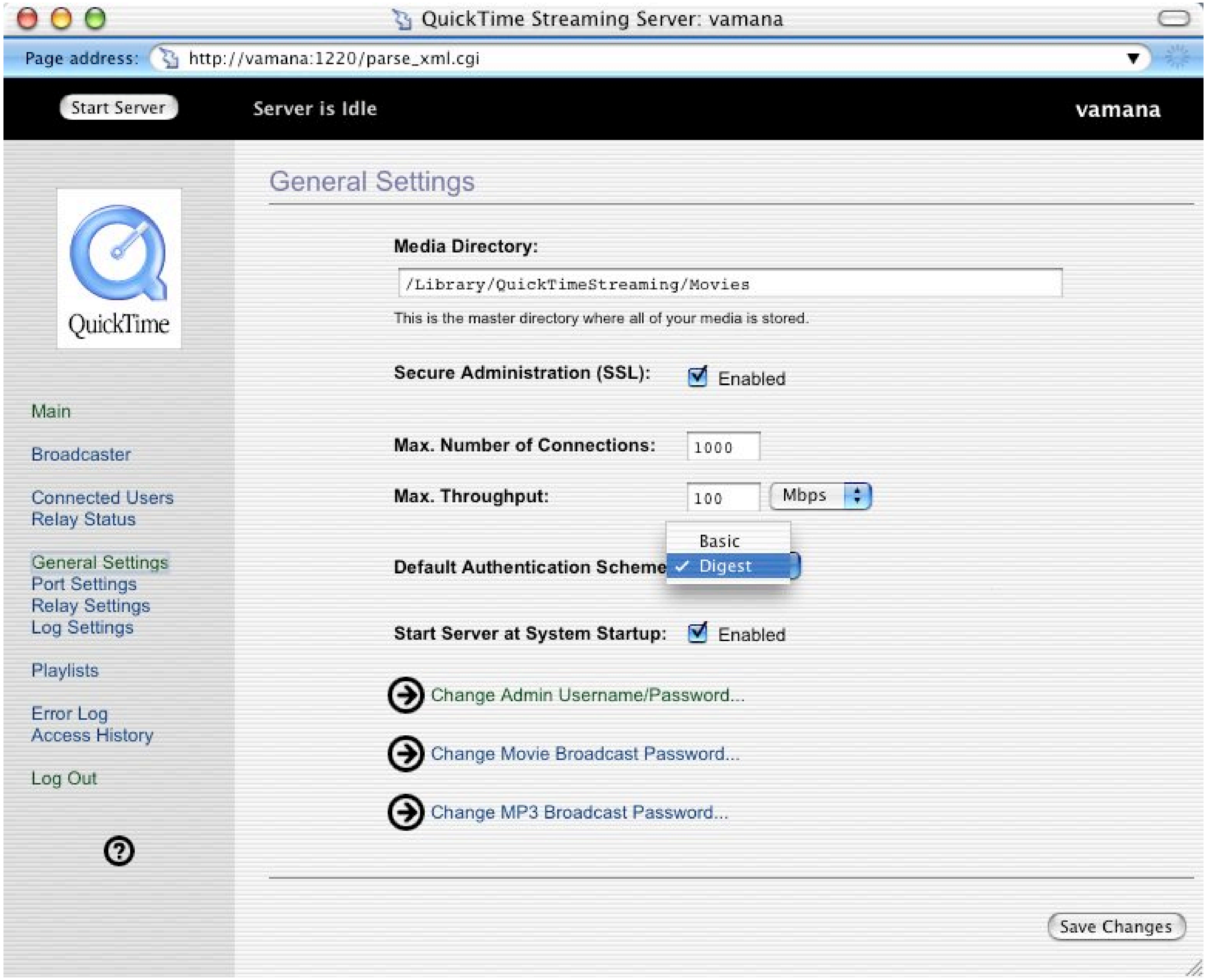
SMTP	IMAP	POP
☐ Kerberos	☐ Kerberos	☐ Kerberos
☐ CRAM-MD5	☐ CRAM-MD5	☐ APOP
☐ Login	☐ Login	
☐ PLAIN	☐ PLAIN	
	☑ Clear	☑ Clear

SMTP SSL: Use

IMAP and POP SSL: Require

Panther Server mail authentication options

- Web and Multimedia (cont'd.)
  - QuickTime Streaming Server (QTSS)
    - Bundled with Mac OS X Server
    - `/Library/QuickTimeStreaming/`
    - Users in `qtusers` (managed with `qtpasswd`), groups in `qtgroups`
    - `qtaccess` files in content directories
      - `AuthScheme` (`basic` or `digest`), `AuthName`, `AuthUserFile`, `AuthGroupFile`, `Require` (`user`, `group`, `valid-user`, `any-user`)
    - Streaming Server Admin (web UI) handles authorization internally
      - Must be in group `admin`
      - Sends browser a cookie with `qtsspassword`, containing Base64-encoded `username:password`; uses basic authentication



General Settings

Media Directory: /Library/QuickTimeStreaming/Movies

This is the master directory where all of your media is stored.

Secure Administration (SSL): ✓ Enabled

Max. Number of Connections: 1000

Max. Throughput: 100 Mbps

Default Authentication Scheme: Basic [✓ Digest]

Start Server at System Startup: ✓ Enabled

Change Admin Username/Password...

Change Movie Broadcast Password...

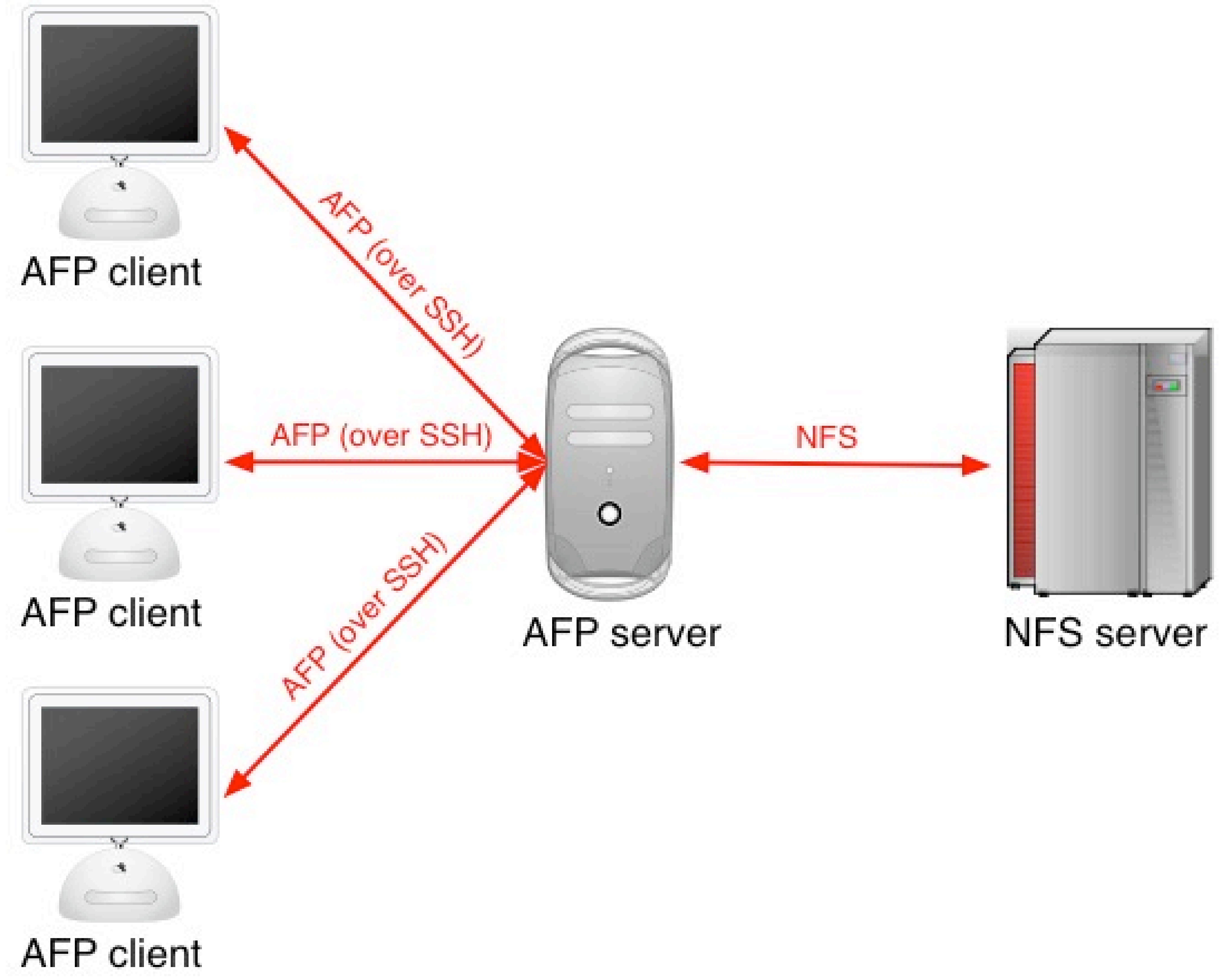
Change MP3 Broadcast Password...

QuickTime Streaming Server admin web interface

Authentication

- File Sharing
  - Apple Filesharing Protocol (AFP)
    - Kerberos
    - Guest access maps user to `nobody`
      - Must be enabled in both Workgroup Manager and Server Settings (or in Open Directory)
  - Network File System (NFS)
    - No user-based authentication
    - `root` on client usually "squashed" to `nobody` on server
      - `opts` value of `maproot=nobody` in `/exports/share_point` in OD

- File Sharing (cont'd.)
  - NFS (cont'd.)
    - NFS resharing (through AFP) provides user-based auth, as well as SSH encryption and legacy Mac OS support
      - AFP server's `root` must not be squashed on NFS server
      - 1) `sudo mkdir -m 0600 /nfs_reshares`
      - 2) Create mount points within `/nfs_reshares/`
      - 3) Mount NFS shares on mount points
      - 4) In Workgroup Manager, create share points from the NFS mounts, and set up AFP sharing

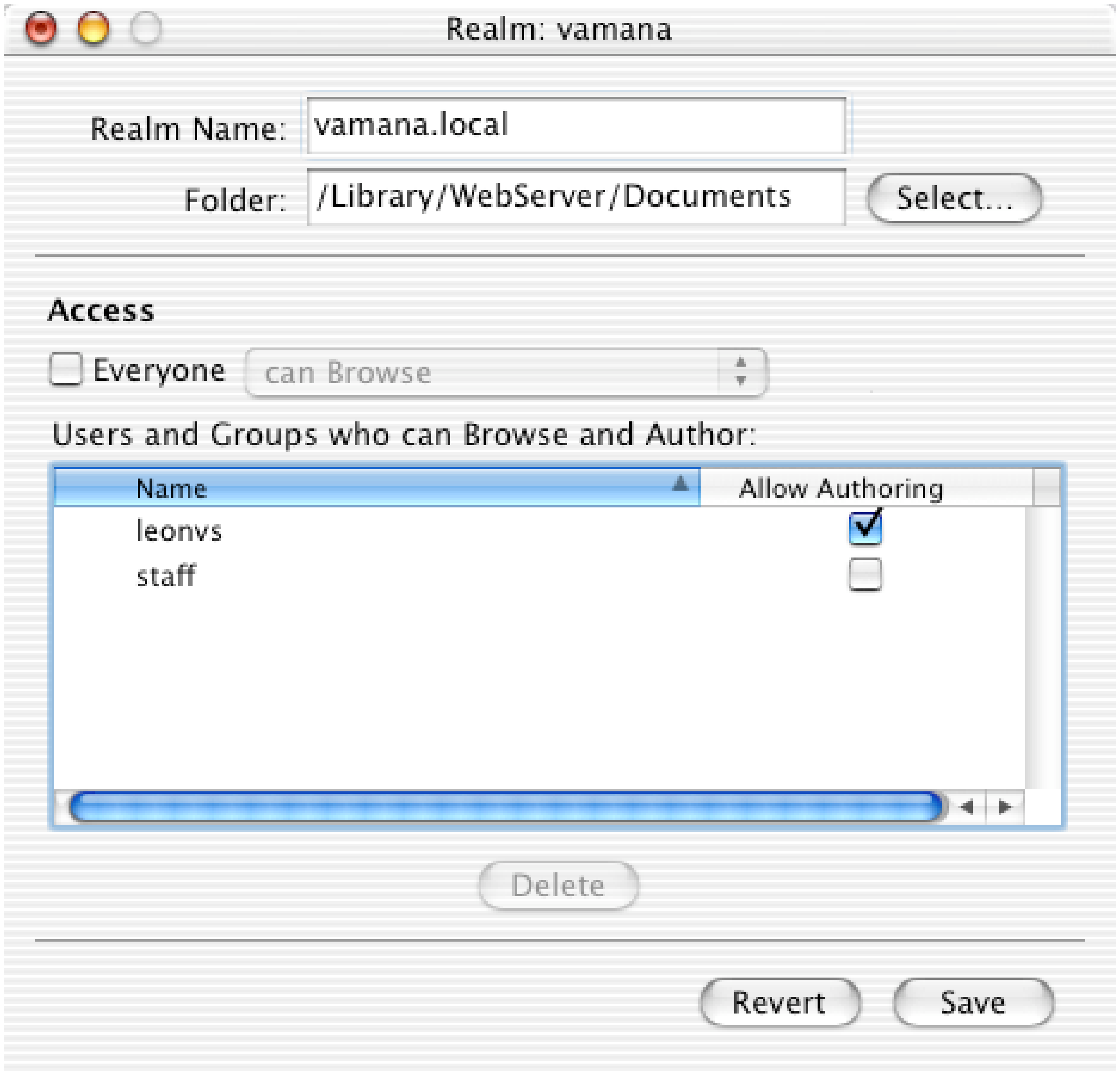


NFS resharing

- File Sharing (cont'd.)
  - File Transfer Protocol (FTP)
    - `lukemftpd` (vanilla OS X)
      - `/etc/ftpusers` (see Network Access Restrictions section later)
    - `xftpd` (OS X Server)
      - Kerberos
      - Anonymous ("guest") access
      - Configured in `/Library/FTPServer/Configuration/ftpaccess`
        - `anonFTP`, `auth_level`, `autogroup`, `allow-gid`, `deny-gid`, `allow-uid`, `deny-uid`, `loginfails`, `passwd-check`, `private`, `upload`
      - `ftpusers` (see Network Access Restrictions section later)

X Authentication

- File Sharing (cont'd.)
  - Web-based Distributed Authoring & Versioning (WebDAV)
    - Same as Apache
    - Set up browse (read) and author (write) privileges
    - Finder on 10.2.4 and later displays warning if WebDAV server uses basic authentication (instead of digest)
    - Basic authentication is default on Mac OS X Server
      - Can enable digest if users are set up in Password Server
        - `WEBDAV-DIGEST` must be among enabled protocols listed with `NeST -getprotocols`


Realm: vamana

Realm Name: vamana.local

Folder: /Library/WebServer/Documents

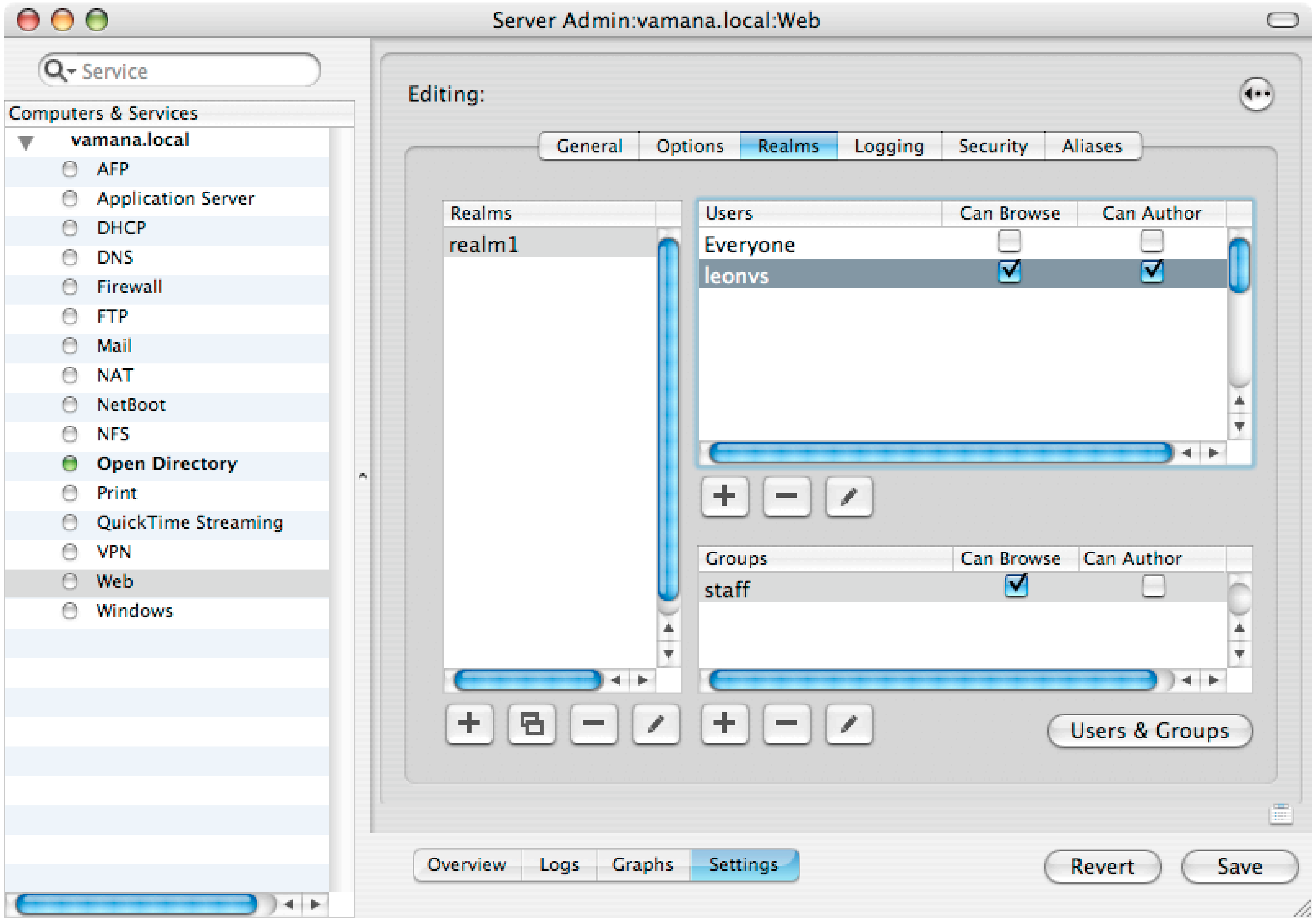
Access

☐ Everyone — can Browse

Users and Groups who can Browse and Author:

Name	Allow Authoring
leonvs	✓
staff	☐

WebDAV realm access in Server Settings



Realms
realm1

Users	Can Browse	Can Author
Everyone	☐	☐
leonvs	✓	✓

Groups	Can Browse	Can Author
staff	✓	☐

WebDAV realm access in Panther’s Server Admin

- File Sharing (cont'd.)
  - WebDAV (cont'd.)
    - To enable digest auth:
      - 1) Add to `/etc/httpd/httpd.conf`:
        - `LoadModule apple_digest_module /usr/libexec/httpd/mod_digest_apple.so`
        - `AddModule mod_digest_apple.c`
      - 2) In `/etc/httpd/httpd_macosxserver.conf`, replace `AuthType Basic` with `AuthType Digest`
      - 3) Restart web server
    - Panther: Digest auth enabled by default
      - And `httpd_macosxserver.conf` has been rolled up into `httpd.conf`

- Other open source software
  - Common UNIX Printing System (CUPS)
    - Configured in `/etc/cups/cupsd.conf`
      - `AuthClass`, `SystemGroup`, `Require`, `Satisfy`, `AuthType`, `RemoteRoot`
  - Berkeley Internet Name Daemon (BIND)
    - Configured in `/etc/named.conf`
    - TSIG to sign transactions between servers
    - DNSSEC not well supported in BIND 8
      - Panther includes BIND 9
        - `man -k dnssec` for information

- Other open source software (cont'd.)
  - OpenSSH
    - `AllowUsers`, `AllowGroups` in `/etc/sshd_config`
    - `from` in `authorized_keys`

- Other open source software (cont'd.)
  - Network Time Protocol (NTP)
    - Configured in `/etc/ntp.conf`
    - How-to:
      - 1) `sudo ntp-genkeys -h /etc`
      - 2) To `/etc/ntp.keys.`*`timestamp`*, add a human-memorable key
      - 3) Transfer key file to other hosts (using secure transfer method)
      - 4) On each host, `ln -s ntp.keys.`*`timestamp`*` /etc/ntp.keys`
      - 5) Add `keys /etc/ntp.keys` to `ntp.conf`
      - 6) Add `trustedkey`, `controlkey`, and `requestkey` directives
      - 7) Add `key` argument to `server`, `peer`, etc. directives
      - 8) `sudo SystemStarter restart "Network Time"`

- Other open source software (cont'd.)
  - UC-Davis Simple Network Management Protocol (SNMP)
    - Panther: Updated to Net-SNMP 5 (`http://www.net-snmp.org/`)
    - Configured primarily in `/usr/share/snmp/snmpd.conf`
    - Community strings
    - Engine ID
    - User-based Security Model (USM)
      - Setting up USM account:
        - 1) `sudo killall snmpd`
        - 2) Add `createUser` *`username`* `SHA` *`password`* to `/var/db/ucd-snmp/snmpd.conf`
        - 3) Restart `snmpd`

- Other open source software (cont'd.)
  - SNMP (cont'd.)
    - View-based Access Control Model (VACM)
      - `group` statements assign USM users to groups
      - `view` statements define OID subtrees
      - `access` statements define access to views by groups
      - `com2sec` statements map community strings to USM security names
      - `rocommunity`, `rwcommunity`, `rouser`, `rwuser` statements simplify configs

Privileged Access

- The `root` Account
- Administrative Accounts
- Domain Administrators

X Privileged Access `root`

- `root` logins disabled on Mac OS X, even locally (by an invalid password)
  - On Mac OS X Server, `root` password same as initial admin user
    - Consider changing one or the other, so they're not the same
- Enable/disable `root` logins under NetInfo Manager->Security
  - Or set a password with `sudo passwd root`
  - Panther includes `dsenableroot`
- Allow local `root` login while denying remote
  - Add to service file `/etc/pam.d/`:
    - `auth required pam_securetty.so debug`
  - Create `/etc/securetty` with `console` as only entry
- SSH: `PermitRootLogin` in `/etc/sshd_config`
- AFP: `allow_root_login` OD property in `/config/AppleFileServer`

Admin Users

- In lieu of logging in as `root`, Mac OS X encourages the use of administrative user accounts
  - Defined by membership in group `admin`
  - The account created during OS install is automatically in `admin` group
- Admin users have several sources of privileged access
  - Some are configurable, and can be removed from `admin` group members or reassigned to others
  - Others are hard-coded advantages

Admin Users

- Directory permissions
    - `/Applications/`, `/Library/`, and `/Developer/` are owned and writable by `admin` group, permitting software installations
    - Permissions can be reset
- `sudo`
    - Members of `admin` group given full access in `/etc/sudoers`, allowing execution of CLI utils as superuser (or any other user) by preceding command line with `sudo`
    - Configuration can be changed
- `su`
    - Can only `su` to `root` if in group `admin` or `wheel`
    - Configurable in `/etc/pam.d/su`

Privileged Access Admin Users

- Authorization Services
  - Component of the Security framework
  - Admin users can perform authorized actions with superuser privileges
  - Used by System Preferences, Installer, administrative apps, etc.
  - Can open files as superuser with `/usr/libexec/authopen`
  - Configurable in `/etc/authorization`
- Open Directory
  - Admin users have full write access to OD domain contents, via either NetInfo, or LDAP when NI authorization is enabled

Admin Users

- AFP (Apple Filesharing Protocol)
  - Admin users have special abilities determined by OD properties in `/config/AppleFileServer`
    - `admin_gets_sp` (Boolean, default `0`): Lets admins mount share points instead of volumes
    - `attempt_admin_auth` (Boolean, default `1`): Lets admins use their own passwords to authenticate as other users
    - `permissions_model`: When set to `unix_with_classic_admin_permissions`, admins can change ownership of files
    - `special_admin_privs` (Boolean, default `0`): Grants admins read access to folders regardless of permissions
  - Can be turned off, but `admin` group is hard-coded

Admin Users

- Apple Mail Server
  - If IMAP Administrator Access in enabled on an alternate port (from Server Settings), an admin user can connect to that port with an IMAP client and view stored email for all users, as well as queued messages
  - OD properties in `/config/AppleMailServer`: `admin_port_value`, `enable_admin_port_flag`
  - Can be turned off, but `admin` group is hard-coded
  - Not enabled by default
- CUPS (Common UNIX Printing System)
  - Specifying `AuthClass System` in `/etc/cups/cupsd.conf` requires auth as member of group `admin`
  - Configurable with `SystemGroup` directive
  - Not enabled by default

Admin Users

- Samba
  - Admin users have no extra privileges by default
  - You could extend admin users' privileges to SMB shares with something like this in `/etc/smb.conf`:
    - `admin users = +admin`
- QuickTime Streaming Server
  - Members of group `admin` can make configuration changes through QTSS web interface, but this group is completely defined within `/Library/QuickTimeStreaming/Config/qtgroups`, and is unrelated to the system group `admin`

X Privileged Access — Domain Admins

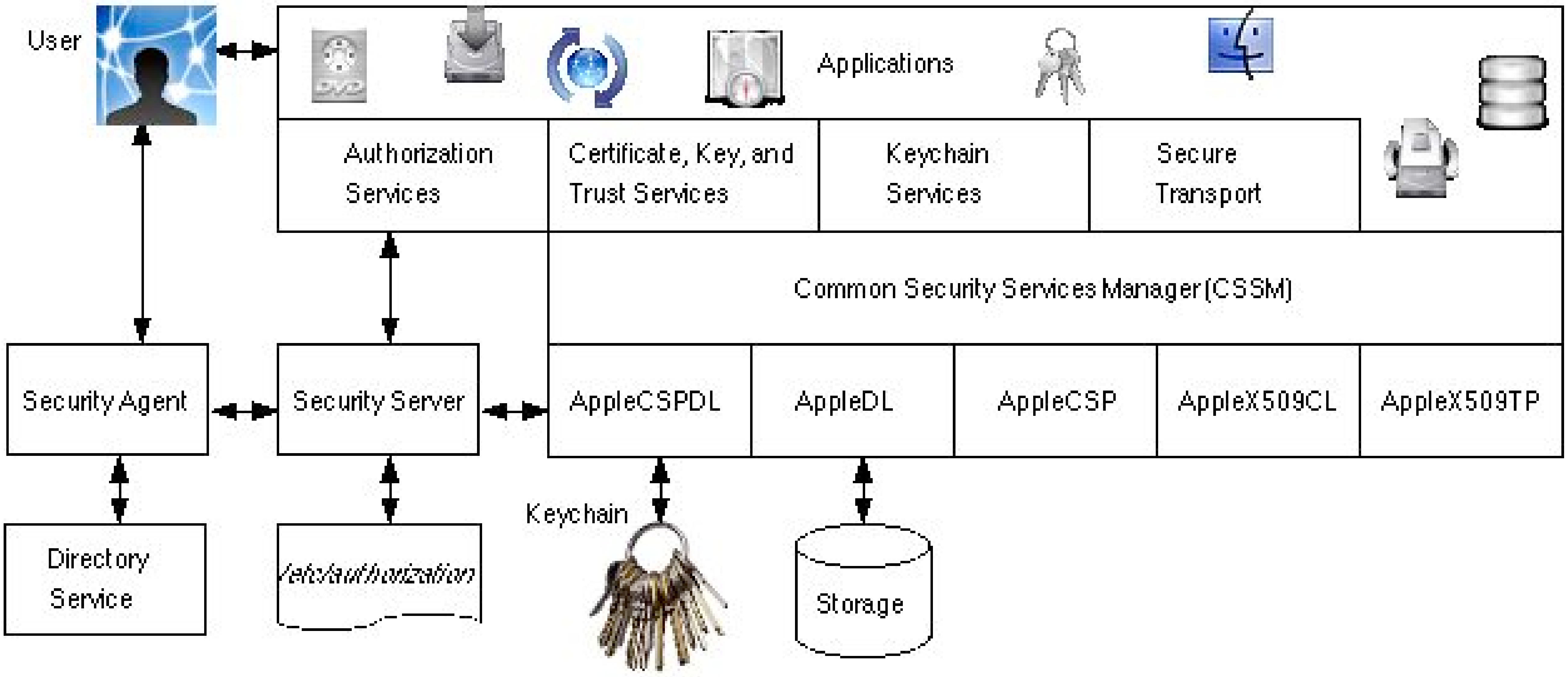
- Users can be designated administrators of particular Open Directory domains, without being made admin users (i.e., added to `admin` group)
  - Panther: Domain admins must be in `admin` group
- Specify which aspects of users, groups, and machines can be managed
  - Accounts and/or preferences (i.e., things controlled by Workgroup Manager)
  - Can limit to specific users, groups, and machines
    - Can't specify elements NOT to be managed, which might be nice
- Configured in Workgroup Manager
  - Stored in `admin_limits` property in user's OD entry

The Security Framework

- Introduction
- Daemons
- Authorization Services
- Common Data Security Architecture
- Keychains

X Security Framework Introduction

- Provides infrastructure for security-related functionality to Mac OS X apps
    - Privileged access, encryption, certificate handling, password storage, etc.
- Examples
    - Login Window uses the framework to authenticate graphical logins
    - Installer and Software Update verify permission to install software
    - Disk Copy creates encrypted disk images
    - Apple Mail Server supports SSL connections
    - Keychain Access is entirely dependent on the Security framework



Security framework and associated components

Daemons

- Every app linked to the Security framework maintains its own instance of the framework in its address space
    - Think of the large box in the preceding diagram as the address space of a single process
- To put distance between apps and sensitive data, external daemons handle most passwords and private keys
- The Security Server processes authorization requests, stores keys in memory, performs cryptographic computations on keys, evaluates ACLs in keychains, and manages keychain master keys
- The Security Agent handles user interaction, acquiring user secrets (passwords, biometric data, smart card keys, etc.), thus further separating such data from the app requesting access
- Both daemon executables located in `/System/Library/CoreServices/`

Authorization Services

- Authorization Services enables programs (typically GUI apps) to determine whether a user should be permitted to take certain actions
- An action is associated with an authorization right, configured in a policy database
  - The policy database is an XML file, `/etc/authorization`
    - The format of `/etc/authorization` has changed somewhat on Panther

Authorization Services

- Example entry in `/etc/authorization`, used by System Preferences:

```
<key>system.preferences</key>
<dict>
     <key>group</key>
     <string>admin</string>
     <key>shared</key>
     <true/>
     <key>allow-root</key>
     <true/>
</dict>
```

  - `system.preferences`: The name of the right
  - `group`: Users who authenticate as members of group `admin` satisfy the requirements for this right
  - `shared`: Other apps need not reauthenticate within the default timeout
  - `allow-root`: No authentication required if logged in as `root`

- Default entry (no name) specifies `group admin`, `shared`, and a `timeout` of 300 seconds (5 minutes)
- Another example, used by loginwindow:

```
<key>system.login.console</key>
<dict>
     <key>eval</key>
     <string>loginwindow_builtin:login,authinternal, \
loginwindow_builtin:success</string>
</dict>
```

  - This right is requested by loginwindow after bootup, causing the Security Server to start up a Security Agent that displays the login dialog
  - The `authinternal` value creates a shared credential used by the Security Server to authorize a user upon login, negating the need for reauthentication by applications after logging in

Authorization Services

- Another example, used by PAM:

```
<key>system.login.tty</key>
<dict>
    <key>eval</key>
    <string>push_hints_to_context,authinternal</string>
</dict>
```

  - This right is requested by the `pam_securityserver.so` module, referenced in several service config files in `/etc/pam.d/`
  - Thanks to this, successful password-based authentication through a PAM-enabled service results in the creation of a shared credential within the context of that service (SSH session, etc.)
  - Possible for remote logins due to `push_hints_to_context`, which forwards auth info to the Security Server, bypassing the Security Agent (which normally needs to put up a graphical dialog)

Authorization Services

- When an application requests authorization for an action (such as starting a network service from System Preferences), here's what happens:
  - 1) The application uses the Authorization Services API to contact the Security Server, requesting a named right (like `system.preferences`)
  - 2) The Security Server looks for the right in the policy database (`/etc/authorization`), and determines the requirements
  - 3) If the right requires authorization as a member of some group, and the Security Server doesn't already possess cached credentials that can be shared with the application, it triggers the Security Agent
  - 4) The Security Agent prompts the user logged into the console, then attempts to auth the user through the Directory Services framework
  - 5) Directory Service reports success or failure, which is passed back through the Security Agent to the Security Server, then back to the application

X Security Framework CDSA

- The Common Data Security Architecture (CDSA) is a standard originally developed by Intel, and now promoted by The Open Group
- Apple has implemented it as part of the Mac OS X Security framework
- At the base are plug-in modules of various types; Apple's are:
  - CSP (Cryptographic Service Provider): random number generation, encryption/decryption, key generation, hashes, digital signatures
  - DL (Data Library): file-based storage of certificates, keys, etc.
  - CSP/DL: manages keychains
  - X509CL (Certificate Library): manages X.509 certificates in memory
  - X509TP (Trust Policy): determines validity of X.509 certs
- Access to module functionality is through the Common Security Services Manager (CSSM) API, the centerpiece of the CDSA

Security Framework CDSA

- Atop the CSSM, Apple provides higher-level APIs as part of the Security framework
  - Secure Transport: implements SSL/TLS
    - Offers cleaner integration and abstraction for developers, but OpenSSL is more familiar and cross-platform
  - Certificate, Key, and Trust Services: manages certs and public keys
  - Keychain Services: programmatic interface to keychains

X Security Framework Keychains

- A keychain is a file containing keys, certificates, passwords, and other secured data
- Contents are encrypted, protected by an access password
- Convenience: a single password unlocks access to a multitude of passwords used for web sites, mail servers, file shares, etc.
  - Makes it practical to use unique, well-chosen passwords for each
- A default keychain is created in `~/Library/Keychains/` upon first login
  - Keychain password same as login password, synced when changing login password through GUI
  - If keychain and login passwords are the same, the keychain is automatically unlocked upon login
  - Panther makes a distinction between **default** and **login** keychains

Keychains

- Each key in the keychain has an ACL, managed by the Security Server
- Keychain items you see in Keychain Access aren't actually keys
    - Each item (password, secure note, etc.) is stored in the keychain, and encrypted with its own key, which is also stored in the keychain
    - ACLs are applied to per-item encryption keys, but they're made to look as if they're properties of the items the keys protect
- A master signing key is used to sign the per-item keys and their ACLs
- Per-item keys and the master signing key are themselves encrypted with a master key
- The master key is encrypted with the keychain password
- The master key and master signing key are also stored in the keychain, making the keychain file completely portable, requiring only the password to unlock its contents

Keychains

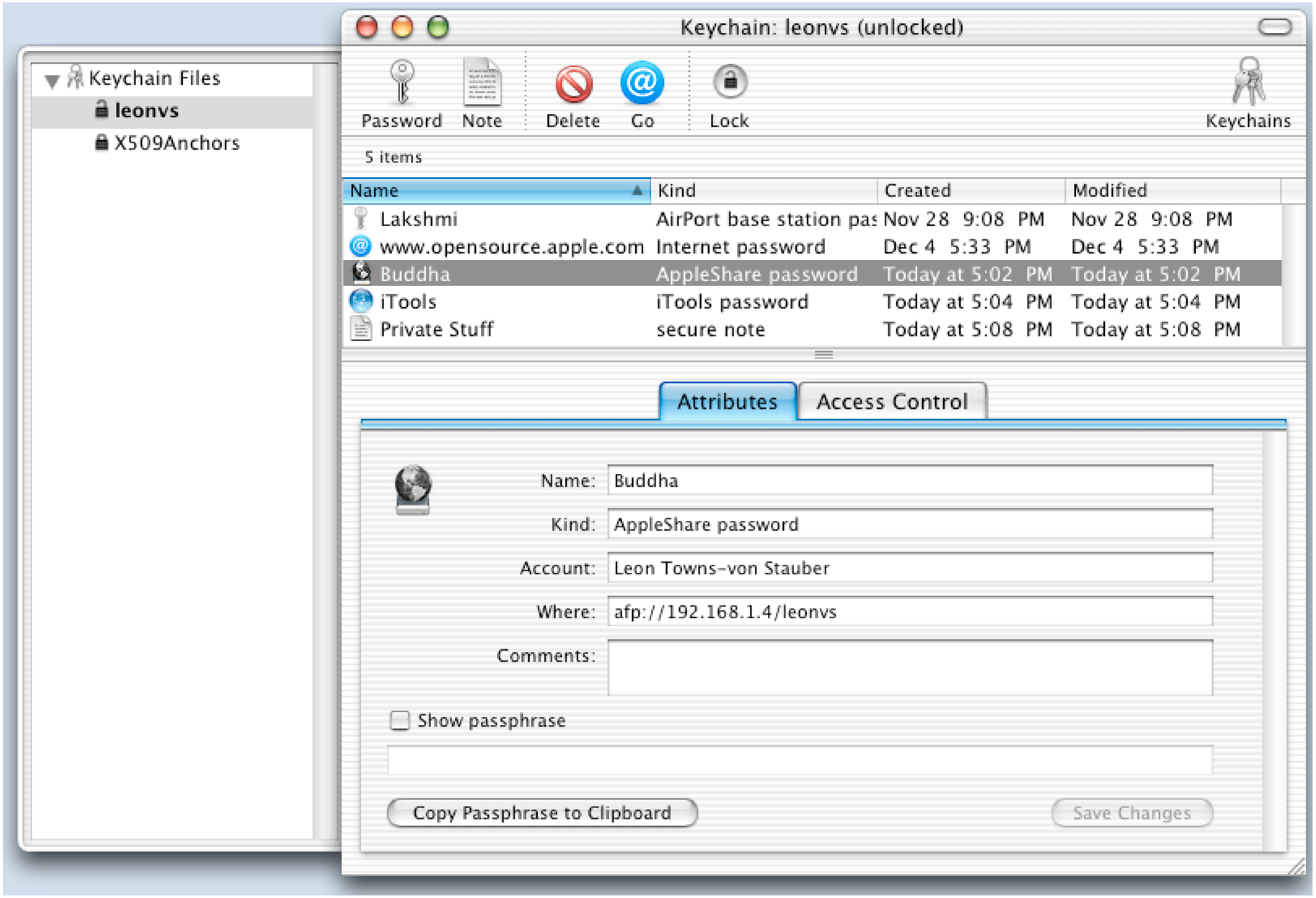
- Here's what happens when an app desires access to a keychain item:
  - 1) App makes a request with Keychain Services, through the CSSM, which calls on the AppleCSPDL
  - 2) If default keychain is locked, AppleCSPDL retrieves encrypted master key and hands it to the Security Server
  - 3) Security Server has Security Agent prompt user for keychain password, which is passed back to Security Server
  - 4) Security Server uses password to decrypt master key, then caches it in memory (A keychain is unlocked when the Security Server has its decrypted master key cached in memory.)
  - 5) Once keychain is unlocked, AppleCSPDL retrieves desired item, item's encryption key, and key's ACL, handing them to Security Server

Keychains

- Accessing a keychain item (cont'd):
  - 6) Security Server verifies signature on key and ACL with master signing key
  - 7) If signature checks out, Security Server processes ACL, resulting in denial, permission, or another prompt through the Security Agent
  - 8) If access is permitted, Security Server decrypts keychain item and hands it to AppleCSPDL, which passes it back up the software stack to the application
- Note that the application process never sees the user's keychain password, nor any of the decrypted keys in the keychain

Keychains

- Keychain Access is primary UI for keychain management
- In the Settings dialog (under the Edit menu), you can set the keychain to lock after some idle timeout, and/or when the machine sleeps
  - "Locking" means to have the Security Server throw away its cached copy of the keychain's decrypted master key
  - You should consider lowering the idle timeout
- Panther includes the `security` command, which can help manage keychains from the CLI
  - `security list-keychains`
  - `security lock-keychain` *`keychain`*
  - `security show-keychain-info` *`keychain`*
  - `security dump-keychain` *`keychain`*
  - See man page for more



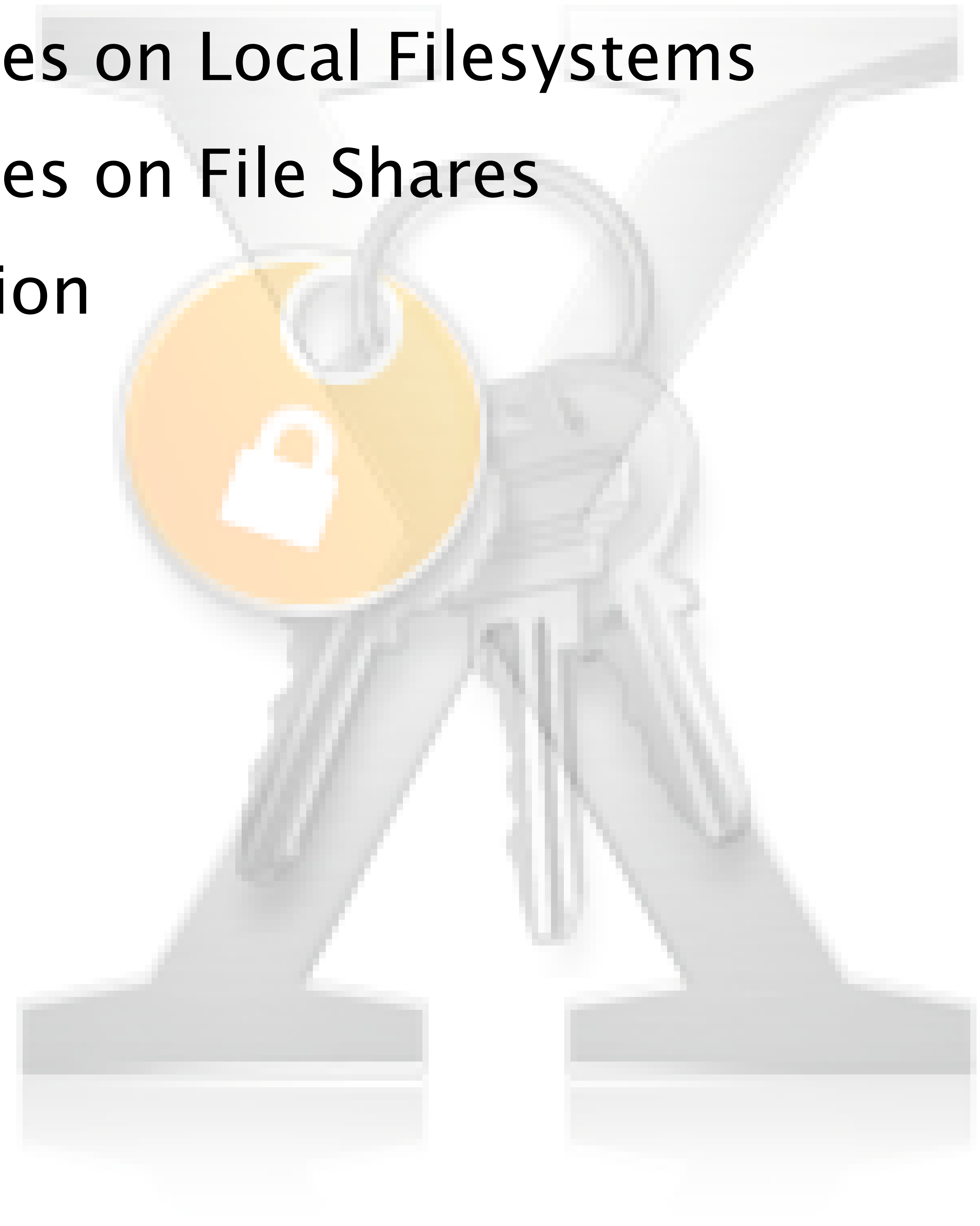
Name	Kind	Created	Modified
Lakshmi	AirPort base station pas	Nov 28 9:08 PM	Nov 28 9:08 PM
www.opensource.apple.com	Internet password	Dec 4 5:33 PM	Dec 4 5:33 PM
Buddha	AppleShare password	Today at 5:02 PM	Today at 5:02 PM
iTools	iTools password	Today at 5:04 PM	Today at 5:04 PM
Private Stuff	secure note	Today at 5:08 PM	Today at 5:08 PM

Name: Buddha
Kind: AppleShare password
Account: Leon Towns-von Stauber
Where: afp://192.168.1.4/leonvs
Comments:

Keychain Access

File Security

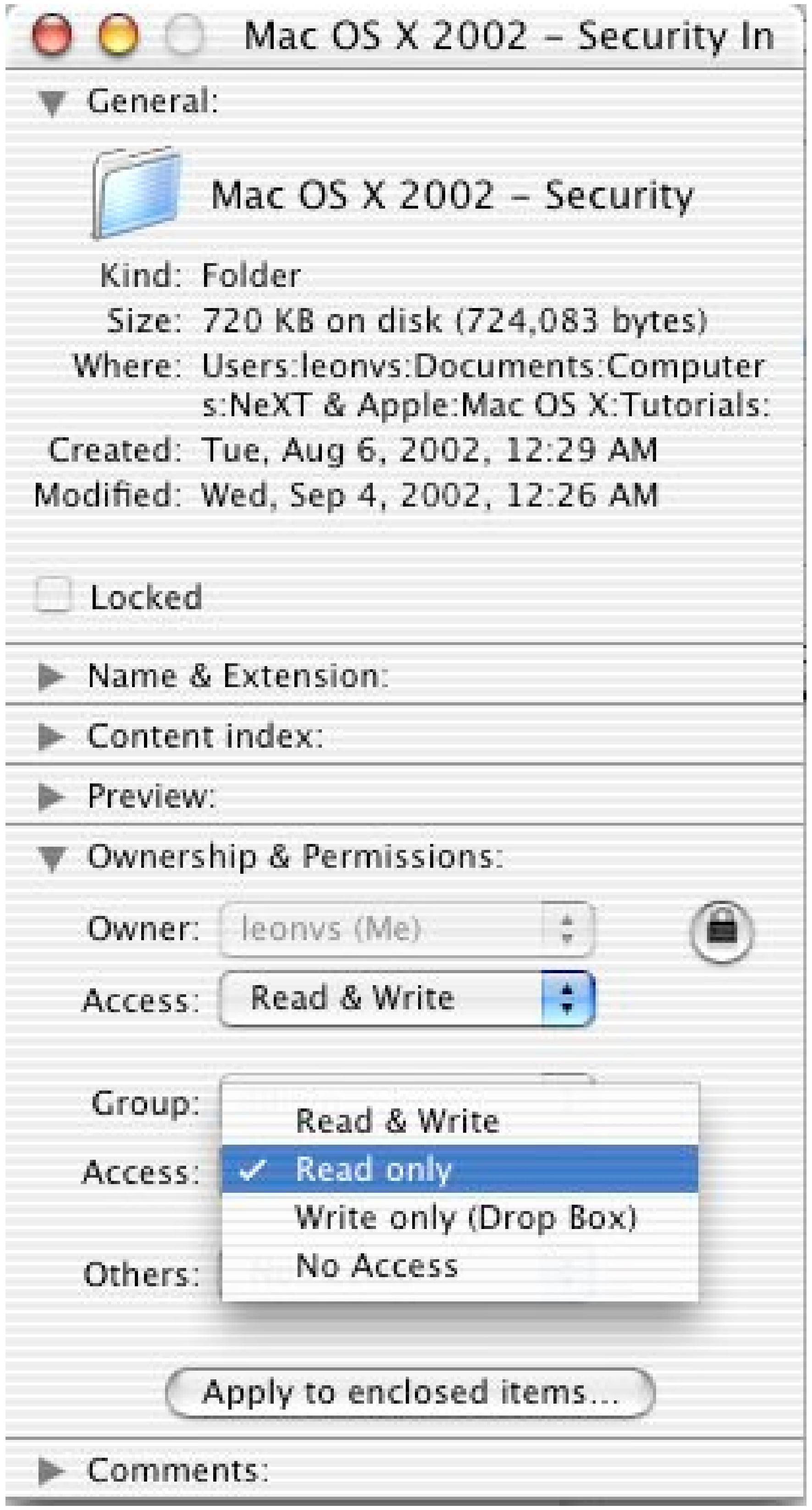
- Attributes on Local Filesystems
- Attributes on File Shares
- Encryption



Local Attributes

- UNIX permissions and ownership
  - Standard user/group ownership, read/write/execute privileges, as well as setuid, setgid, sticky bits
  - Managed with Get Info in the Finder, or `ls -l`/`chown`/`chgrp`/`chmod`
  - Volume permissions
    - If volume permissions are not enabled on an HFS+ volume (such as on removable media volumes), file ownership attributes are ignored
      - Everything assigned to group `unknown`, owner as mount point
    - Turn off/on with Ignore Privileges in Get Info dialog for a volume
    - Can also use `vsdbutil -`*`switch`* `pathname`
      - switch: `-c` to check, `-a` to activate, `-d` to deactivate
    - `/var/db/volinfo.database` lists volume status by UUID

Local Attributes



Mac OS X 2002 – Security In

General:

Mac OS X 2002 – Security

Kind:	Folder
Size:	720 KB on disk (724,083 bytes)
Where:	Users:leonvs:Documents:Computers:NeXT & Apple:Mac OS X:Tutorials:
Created:	Tue, Aug 6, 2002, 12:29 AM
Modified:	Wed, Sep 4, 2002, 12:26 AM

Locked

Name & Extension:

Content index:

Preview:

Ownership & Permissions:

Owner: leonvs (Me)

Access: Read & Write

Group:

Access: [✓ Read only]

Others:

Get Info

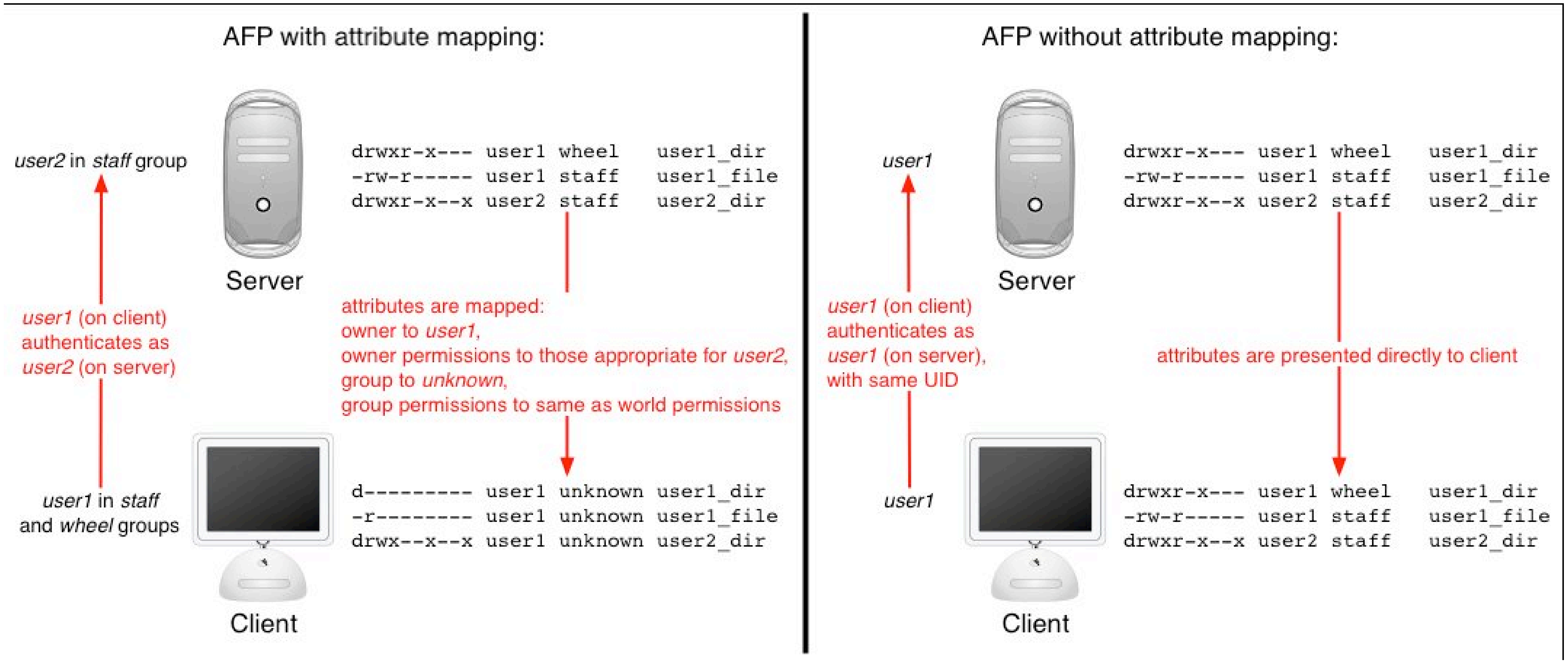
Local Attributes

- BSD flags
  - Immutable and append-only flags, as described in man page for `chflags`
  - HFS Locked attribute mapped to BSD immutable flag
    - Can view and set in Get Info, or with utils included in Dev Tools
      - `/Developer/Tools/GetFileInfo -al` *`filename`*
      - `/Developer/Tools/SetFile -a L` *`filename`*
        - Same as `chflags uchg filename`
  - `sappnd` and `suchg` flags are similar to `appnd` and `uchg`, but can only be unset if the `securelevel` is `0`
    - This is only the case in single-user mode; `securelevel` can never be lowered, only raised
    - `sysctl kern.securelevel` to display current setting

File Share Attributes

- Apple Filesharing Protocol (AFP)
  - Often, AFP client maps server file attributes to values that work better for the client
    - Unlike NFS, AFP is designed to work in environments where user and group IDs don't match between server and client, and where username used for auth may be different than that on client system
    - On mounted share, file owner set to user ID that mounted share, group set to `unknown`; owner permissions set to whatever is appropriate for authenticated user on server, group permissions made to match world permissions
    - Get Info shows actual server attributes, but mapped attributes determine access and are shown at command line
  - Starting with 10.2, if user authenticates with same name used on client, and UIDs match, attribute mapping disabled

File Share Attributes


AFP with attribute mapping:
Server
Client
user2 in *staff* group
user1 (on client) authenticates as *user2* (on server)
user1 in *staff* and *wheel* groups
attributes are mapped:
owner to *user1*,
owner permissions to those appropriate for *user2*,
group to *unknown*,
group permissions to same as world permissions
AFP without attribute mapping:
user1
user1 (on client) authenticates as *user1* (on server), with same UID
user1
attributes are presented directly to client

AFP with attribute mapping — Server:

```
drwxr-x--- user1 wheel   user1_dir
-rw-r----- user1 staff   user1_file
drwxr-x--x user2 staff   user2_dir
```

AFP with attribute mapping — Client:

```
d--------- user1 unknown user1_dir
-r-------- user1 unknown user1_file
drwx--x--x user1 unknown user2_dir
```

AFP without attribute mapping — Server:

```
drwxr-x--- user1 wheel   user1_dir
-rw-r----- user1 staff   user1_file
drwxr-x--x user2 staff   user2_dir
```

AFP without attribute mapping — Client:

```
drwxr-x--- user1 wheel   user1_dir
-rw-r----- user1 staff   user1_file
drwxr-x--x user2 staff   user2_dir
```

File attribute mapping on AFP shares

File Share Attributes

- AFP (cont'd.)
  - Attributes on files created over AFP are set according to method chosen in Workgroup Manager->Sharing->Protocols->Apple File Settings (if you're on OS X Server; otherwise, set `afp_use_parent_privs` property under `/config/SharePoints/share_point` in `local` OD domain)
    - "Use standard UNIX behavior": new items owned by authenticated user, group matches parent folder; permissions determined by default umask of 022, making group collaboration difficult
    - "Inherit permissions from parent": new with 10.2.3; new files owned by authenticated user, new folder owner same as parent folder, group matches parent folder; permissions match parent folder, which enables collaboration in group-writable folders

File Share Attributes

- Server Message Block (SMB)
  - Samba also does attribute mapping
    - Attributes usually derived from mount point, but a plethora of options in `/etc/smb.conf` can alter the behavior (some of which can be set from Workgroup Manager)
      - `writable`, `read list`, `write list`, `create mask`, `directory mask`, `force create mode`, `force directory mode`, `inherit permissions`, `map archive`, `map system`, `map hidden`, `nt acl support`
  - Other file security parameters: `delete readonly`, `dos filemode`, `dos filetimes`, `follow symlinks`, `wide links`
- Network File System (NFS)
  - No attribute mapping; user and group IDs must match between client and server

File Share Attributes

- File Transfer Protocol (FTP)
  - `lukemftpd` (vanilla OS X)
    - Configured in `/etc/ftpd.conf`
      - `umask` determines permissions for new files
        - Default: `guest 077`, `all 027`
      - `modify` and `upload` options can disable certain commands
        - Default: `modify guest off`

File Share Attributes

- FTP (cont'd.)
  - `xftpd` (OS X Server)
    - Based on `wu-ftpd`
    - Configured in `/Library/FTPServer/Configuration/ftpaccess`
      - `defumask` determines permissions for new files
      - Disable access to commands with `chmod`, `delete`, `overwrite`, `rename`, `umask`
- WebDAV
  - Permissions are wide open, but only authenticated users can access WebDAV share mounted in Finder (or directly with `mount_webdav`)

Encryption

- Disk Copy can create password-locked encrypted disk images
  - Disk Utility on Panther
- File Vault (10.3) can encrypt home directories
- Many third-party tools
  - GPGFileTool, PuzzlePalace, etc.

Packet Filtering

- `ipfw`
- Graphical Tools

Packet Filtering `ipfw`

- `ipfw` ported from FreeBSD
  - Kernel extension (`/System/Library/Extensions/IPFirewall.kext`) and command-line tool (`ipfw`)
  - Details in `ipfw` and `ipfirewall` man pages
- `ipfw list` displays current list of packet filtering rules
  - `ipfw -N show` includes counters, and number->name translation

Packet Filtering ipfw

- Typical rule set, made by System Preferences with SSH and AFP enabled:

```
02000 allow ip from any to any via lo*
02010 deny ip from 127.0.0.0/8 to any in
02020 deny ip from any to 127.0.0.0/8 in
02030 deny ip from 224.0.0.0/3 to any in
02040 deny tcp from any to 224.0.0.0/3 in
02050 allow tcp from any to any out
02060 allow tcp from any to any established
02070 allow tcp from any to any afpovertcp in
02080 allow tcp from any to any svrloc in
02090 allow tcp from any to any ssh in
12190 deny tcp from any to any
65535 allow ip from any to any
```

- Stateful packet filtering supported with `keep-state` and `check-state`
  - For example, rules `2050` and `2060` above could be replaced with a single rule offering better security:
    - `2050 allow tcp from any to any keep-state out`
  - On outbound connection, creates dynamic rule allowing return connection only from destination host and port

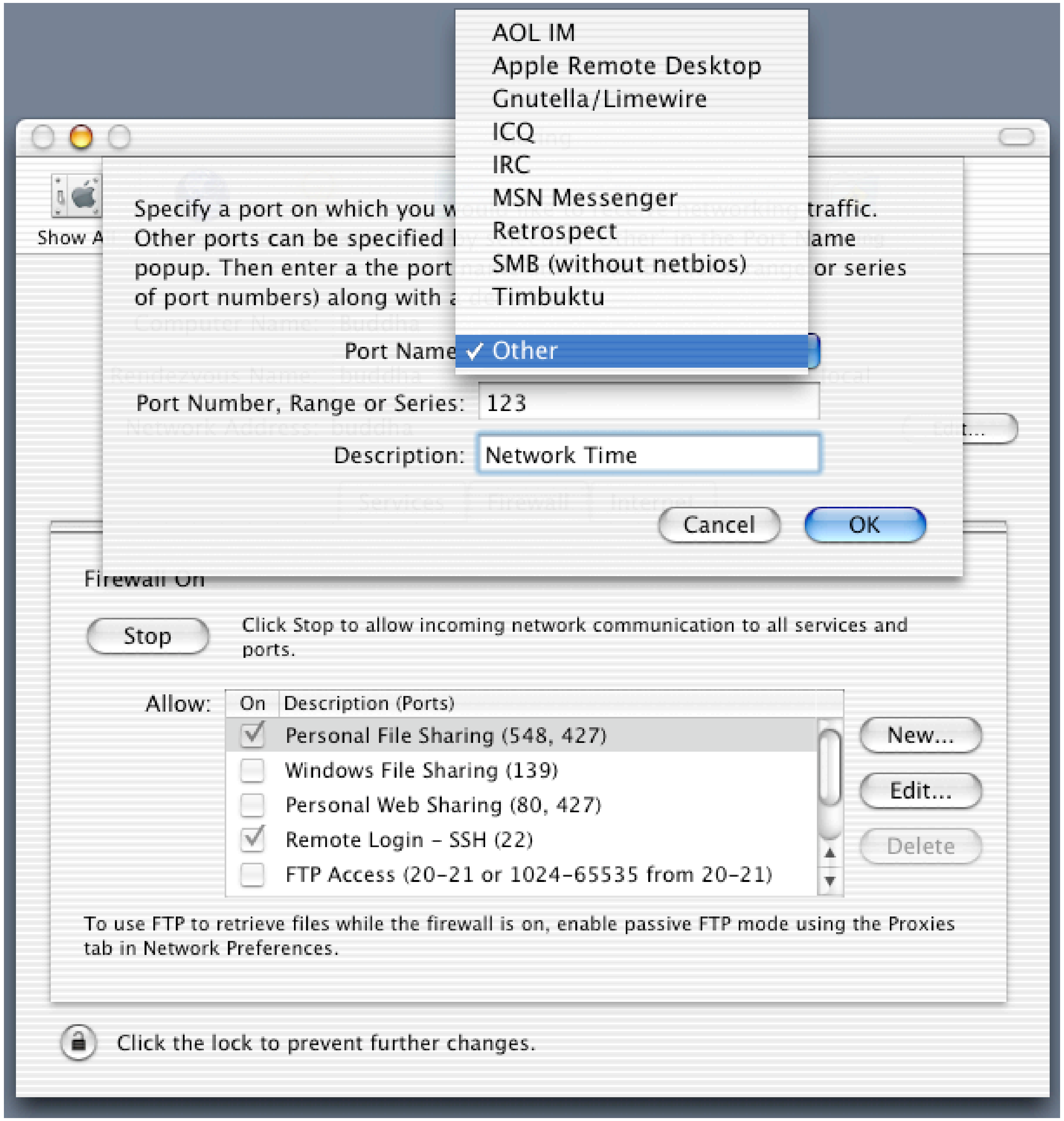
- Related sysctl variables (displayed with sysctl net.inet.ip.fw):

```
net.inet.ip.fw.enable: 1
net.inet.ip.fw.one_pass: 1
net.inet.ip.fw.debug: 1
net.inet.ip.fw.verbose: 0
net.inet.ip.fw.verbose_limit: 0
net.inet.ip.fw.dyn_buckets: 256
net.inet.ip.fw.curr_dyn_buckets: 256
net.inet.ip.fw.dyn_count: 0
net.inet.ip.fw.dyn_max: 1000
net.inet.ip.fw.dyn_ack_lifetime: 300
net.inet.ip.fw.dyn_syn_lifetime: 20
net.inet.ip.fw.dyn_fin_lifetime: 20
net.inet.ip.fw.dyn_rst_lifetime: 5
net.inet.ip.fw.dyn_short_lifetime: 30
```

  - Described in ipfw man page
  - Setting securelevel to 3 (sysctl -w kern.securelevel=3) prohibits rule set changes
    - Note that securelevel cannot be lowered, so to make rule set changes, you'd have to boot into single-user mode

Graphical Tools

- System Preferences->Sharing->Firewall
  - One-button activation
  - Automatically opens ports as services are enabled
  - Filters stored in `/Library/Preferences/com.apple.sharing.firewall.plist`, activated by `NetworkExtensions` startup item
- Server Settings->Network->Firewall
  - Mac OS X Server
  - Clunkier, but more flexible, interface
  - Server Status displays results of `ipfw show`
  - Configuration in `/config/IPFilters` in `local` Open Directory domain
  - Panther: Use Server Admin; config stored in files in `/etc/ipfilter/`



Specify a port on which you w… traffic. Other ports can be specified … ame popup. Then enter a the port … or series of port numbers) along with a …

Port Name: ✓ Other

Port Number, Range or Series: 123

Description: Network Time

Firewall On

Click Stop to allow incoming network communication to all services and ports.

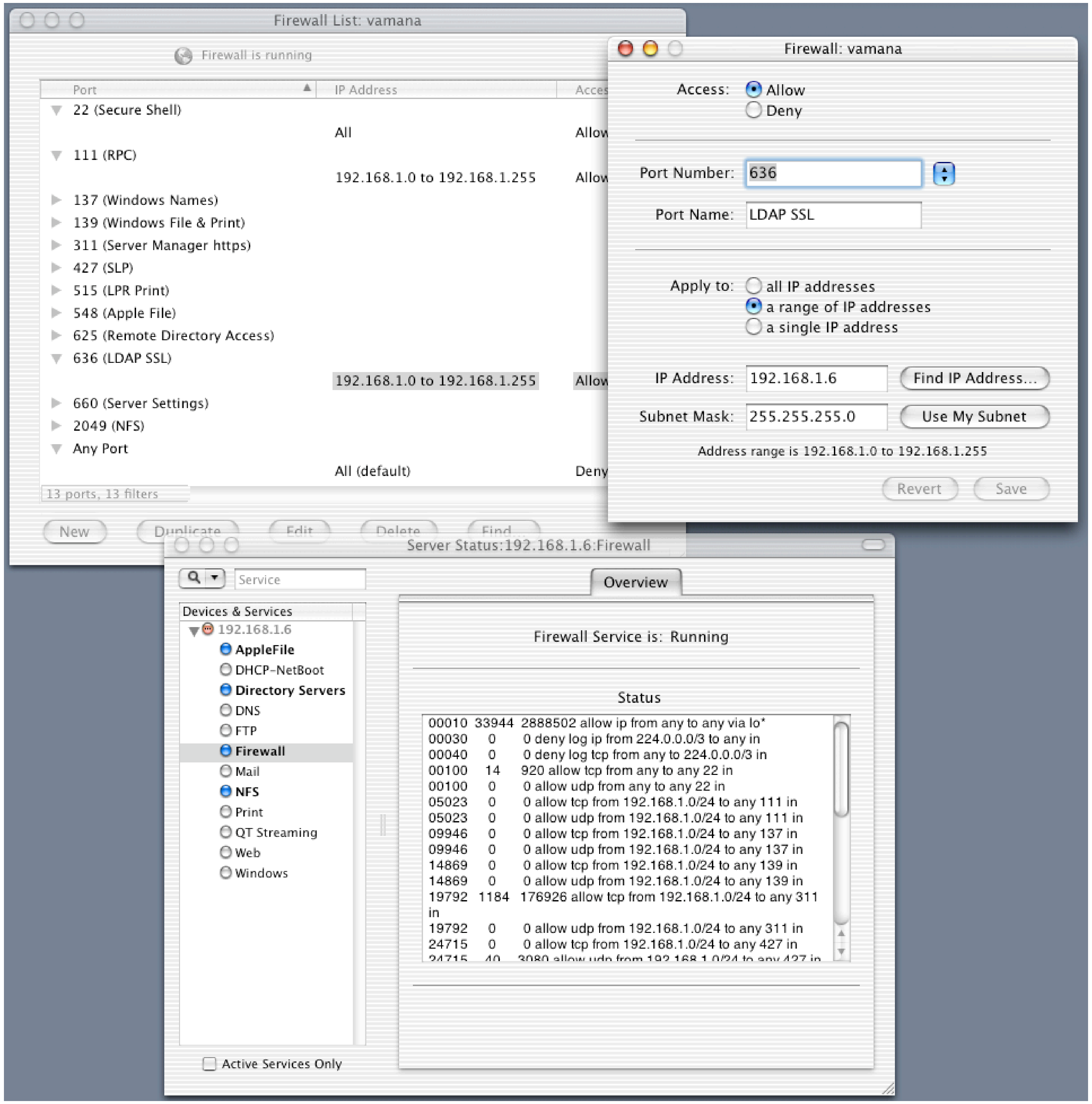
Allow:

On	Description (Ports)
✓	Personal File Sharing (548, 427)
	Windows File Sharing (139)
	Personal Web Sharing (80, 427)
✓	Remote Login - SSH (22)
	FTP Access (20-21 or 1024-65535 from 20-21)

To use FTP to retrieve files while the firewall is on, enable passive FTP mode using the Proxies tab in Network Preferences.

Click the lock to prevent further changes.

Mac OS X Sharing Preferences



Firewall List: vamana

Firewall is running

Port	IP Address	Acces
22 (Secure Shell)		
	All	Allow
111 (RPC)		
	192.168.1.0 to 192.168.1.255	Allow
137 (Windows Names)		
139 (Windows File & Print)		
311 (Server Manager https)		
427 (SLP)		
515 (LPR Print)		
548 (Apple File)		
625 (Remote Directory Access)		
636 (LDAP SSL)		
	192.168.1.0 to 192.168.1.255	Allow
660 (Server Settings)		
2049 (NFS)		
Any Port		
	All (default)	Deny

13 ports, 13 filters

Firewall: vamana

Access: [✓ Allow] Deny

Port Number: 636

Port Name: LDAP SSL

Apply to: all IP addresses [✓ a range of IP addresses] a single IP address

IP Address: 192.168.1.6

Subnet Mask: 255.255.255.0

Address range is 192.168.1.0 to 192.168.1.255

Server Status:192.168.1.6:Firewall

Devices & Services: 192.168.1.6 — AppleFile, DHCP-NetBoot, Directory Servers, DNS, FTP, Firewall, Mail, NFS, Print, QT Streaming, Web, Windows

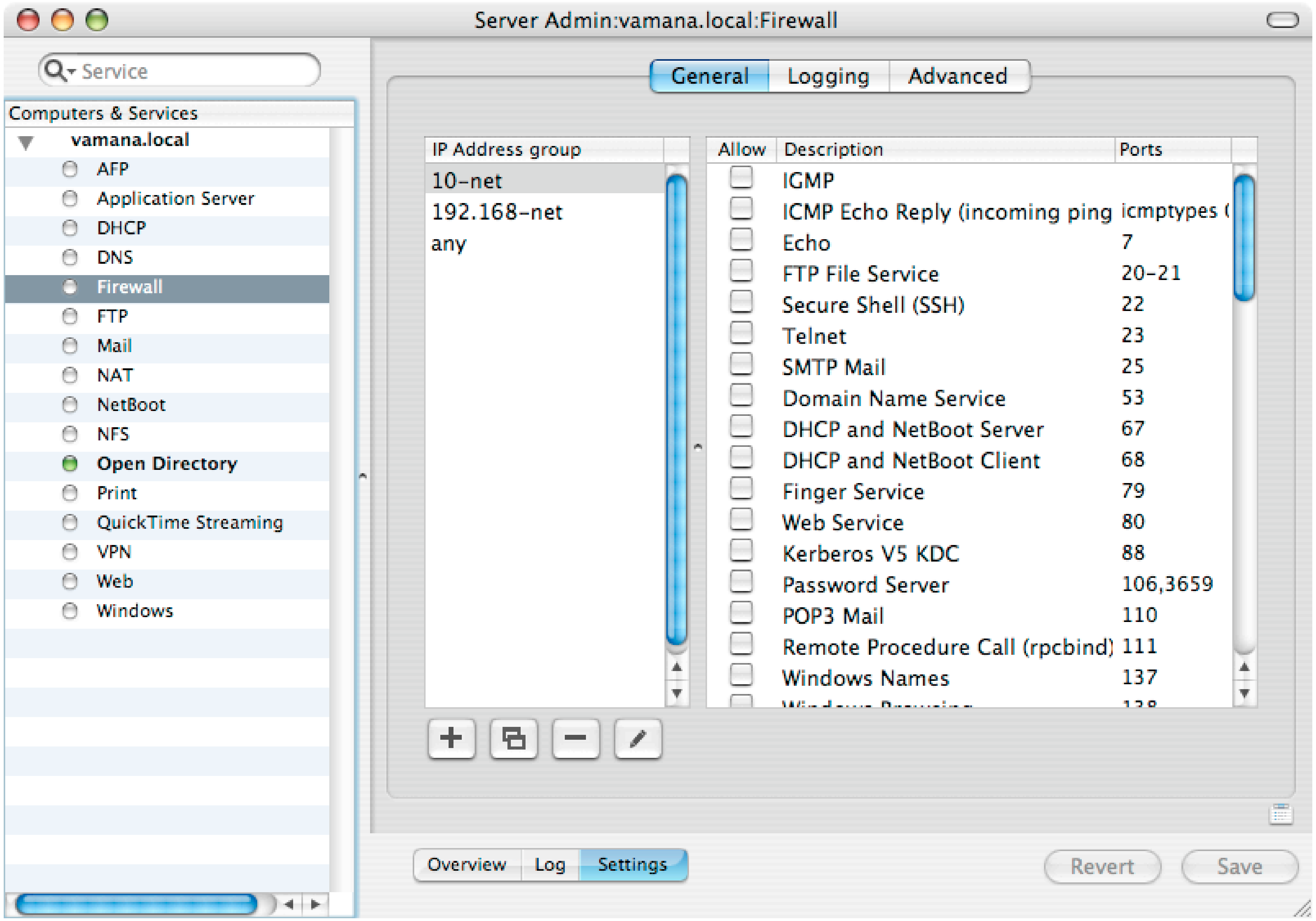
Active Services Only

Firewall Service is: Running

Status

```
00010  33944  2888502 allow ip from any to any via lo*
00030     0      0 deny log ip from 224.0.0.0/3 to any in
00040     0      0 deny log tcp from any to 224.0.0.0/3 in
00100    14    920 allow tcp from any to any 22 in
00100     0      0 allow udp from any to any 22 in
05023     0      0 allow tcp from 192.168.1.0/24 to any 111 in
05023     0      0 allow udp from 192.168.1.0/24 to any 111 in
09946     0      0 allow tcp from 192.168.1.0/24 to any 137 in
09946     0      0 allow udp from 192.168.1.0/24 to any 137 in
14869     0      0 allow tcp from 192.168.1.0/24 to any 139 in
14869     0      0 allow udp from 192.168.1.0/24 to any 139 in
19792  1184  176926 allow tcp from 192.168.1.0/24 to any 311
in
19792     0      0 allow udp from 192.168.1.0/24 to any 311 in
24715     0      0 allow tcp from 192.168.1.0/24 to any 427 in
24715    40   3080 allow udp from 192.168.1.0/24 to any 427 in
```

Mac OS X Server: Server Settings and Server Status



IP Address group
10-net
192.168-net
any

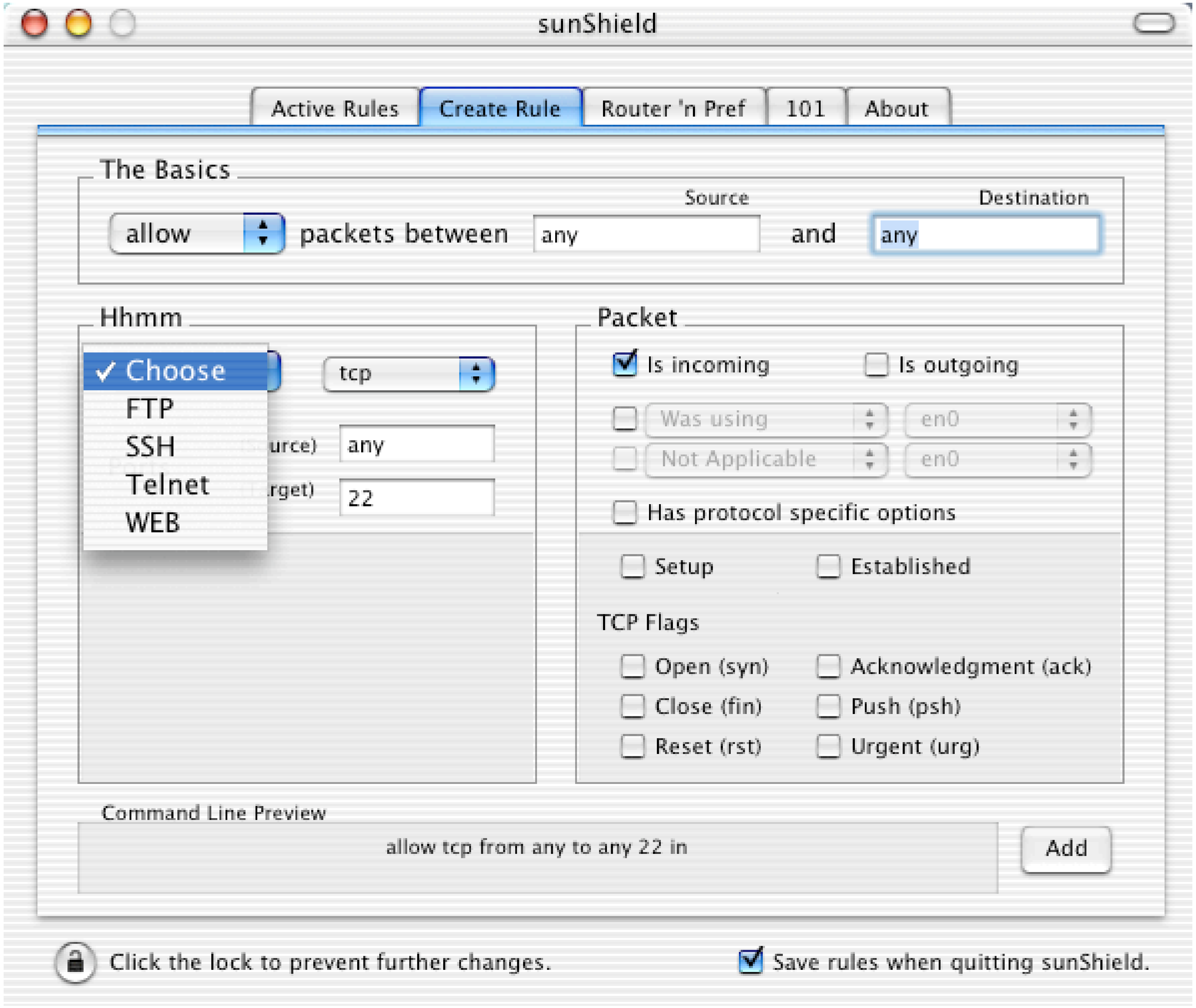
Allow	Description	Ports
☐	IGMP	
☐	ICMP Echo Reply (incoming ping	icmptypes (
☐	Echo	7
☐	FTP File Service	20-21
☐	Secure Shell (SSH)	22
☐	Telnet	23
☐	SMTP Mail	25
☐	Domain Name Service	53
☐	DHCP and NetBoot Server	67
☐	DHCP and NetBoot Client	68
☐	Finger Service	79
☐	Web Service	80
☐	Kerberos V5 KDC	88
☐	Password Server	106,3659
☐	POP3 Mail	110
☐	Remote Procedure Call (rpcbind)	111
☐	Windows Names	137

Panther Server: Server Admin

Graphical Tools

- Third-party `ipfw` front ends
  - sunShield (`http://homepage.mac.com/opalliere/shield_us.html`)
    - Freeware
    - System Preferences pane, startup item
    - Little more control, nice features and interface, but important limitations (no dynamic rules, logging, etc.)
  - Impasse (`http://glu.com/products/impasse/`)
    - Commercial, with trial period
    - System Preferences pane, startup item, daemon
    - Nearly full control, straightforward interface, especially nice log viewer

Graphical Tools



allow packets between any and any

tcp

(Source) any

(Target) 22

[✓ Is incoming] Is outgoing

Was using en0

Not Applicable en0

Has protocol specific options

Setup Established

TCP Flags

Open (syn) Acknowledgment (ack)

Close (fin) Push (psh)

Reset (rst) Urgent (urg)

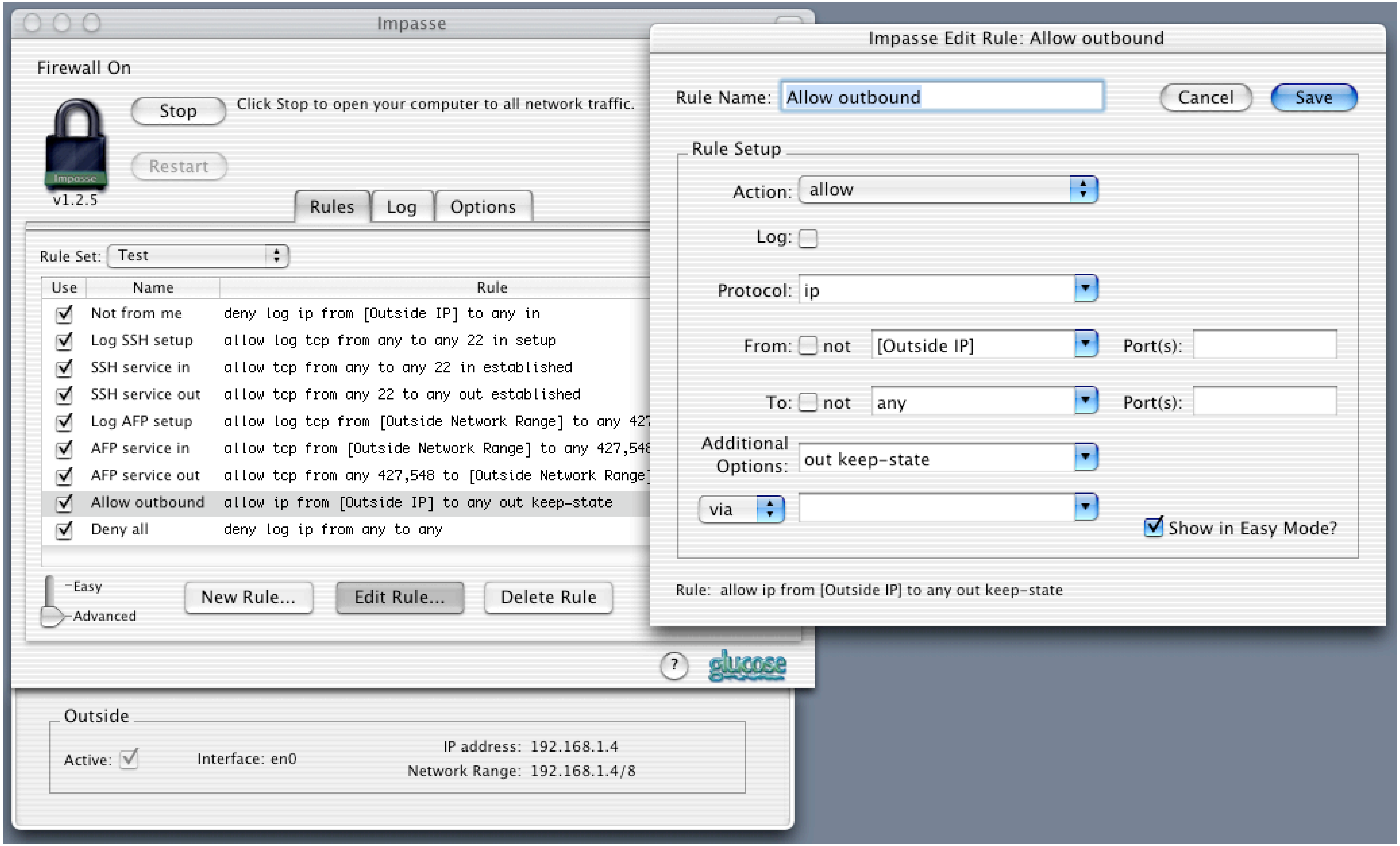
Command Line Preview

```
allow tcp from any to any 22 in
```

Click the lock to prevent further changes.

[✓ Save rules when quitting sunShield.]

sunShield



Firewall On

Click Stop to open your computer to all network traffic.

v1.2.5

Rule Set: Test

Use	Name	Rule
✓	Not from me	deny log ip from [Outside IP] to any in
✓	Log SSH setup	allow log tcp from any to any 22 in setup
✓	SSH service in	allow tcp from any to any 22 in established
✓	SSH service out	allow tcp from any 22 to any out established
✓	Log AFP setup	allow log tcp from [Outside Network Range] to any 427
✓	AFP service in	allow tcp from [Outside Network Range] to any 427,548
✓	AFP service out	allow tcp from any 427,548 to [Outside Network Range]
✓	Allow outbound	allow ip from [Outside IP] to any out keep-state
✓	Deny all	deny log ip from any to any

Outside

Active: ✓ Interface: en0 IP address: 192.168.1.4 Network Range: 192.168.1.4/8

Impasse Edit Rule: Allow outbound

Rule Name: Allow outbound

Rule Setup

Action: allow

Log: ☐

Protocol: ip

From: ☐ not [Outside IP] Port(s):

To: ☐ not any Port(s):

Additional Options: out keep-state

via

✓ Show in Easy Mode?

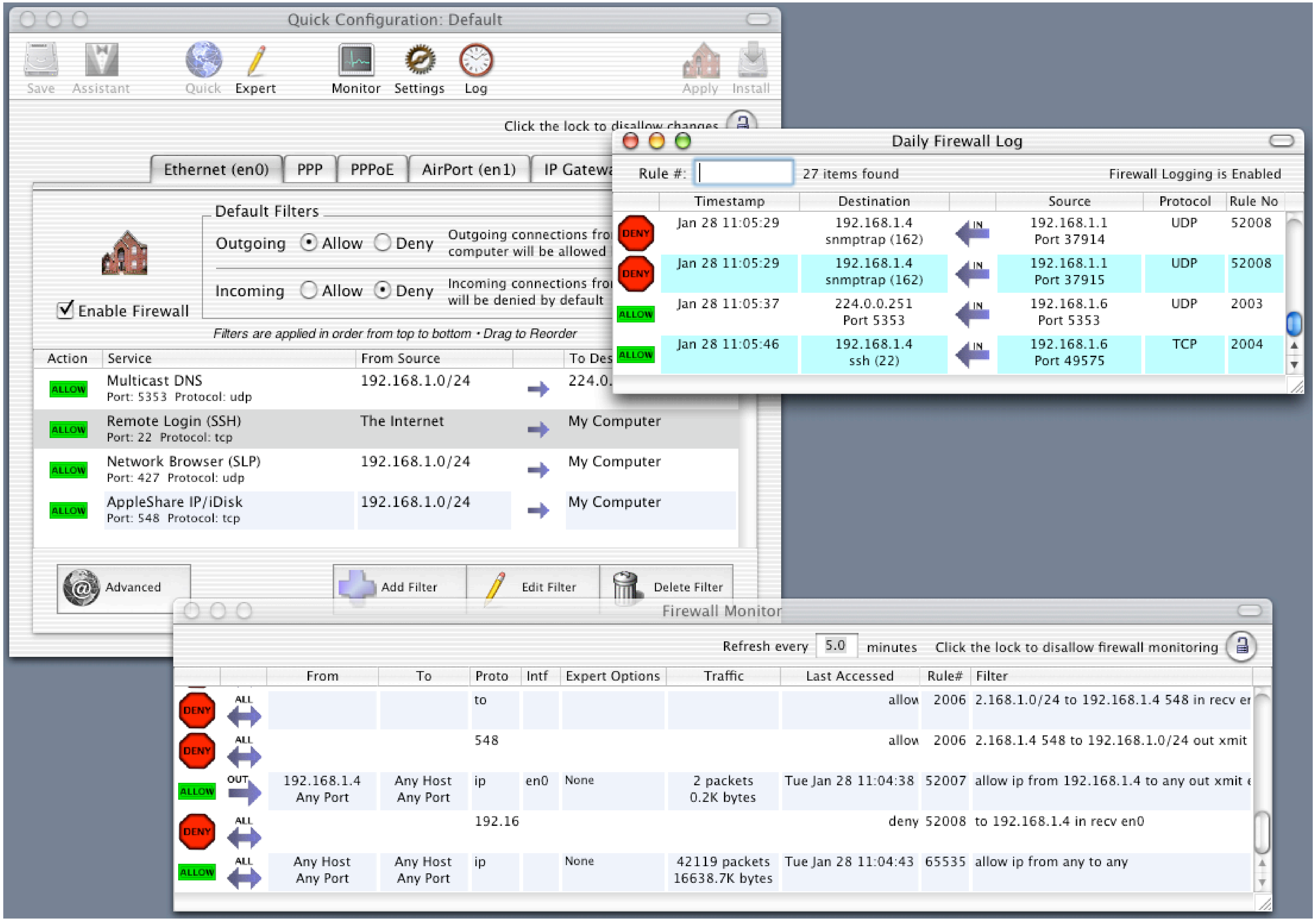
Rule: allow ip from [Outside IP] to any out keep-state

Impasse

Graphical Tools

- Third-party `ipfw` front ends (cont'd.)
  - BrickHouse (`http://personalpages.tds.net/~brian_hill/brickhouse.html`)
    - Shareware
    - Application, startup item
    - Lots of configurability
- Third-party firewall software not based on `ipfw`
  - Firewalk X (`http://www.pliris-soft.com/products/firewalkx/`)
  - NetBarrier (`http://www.intego.com/netbarrier/`)
  - Norton Personal Firewall (`http://www.symantec.com/sabu/nis/npf_mac/`)
  - IPNetSentryX (`http://www.sustworks.com/site/prod_sentryx_overview.html`)

Graphical Tools



Click the lock to disallow changes

Default Filters

Outgoing [✓ Allow] Deny — Outgoing connections fro computer will be allowed

Incoming Allow [✓ Deny] — Incoming connections fro will be denied by default

[✓ Enable Firewall]

Filters are applied in order from top to bottom • Drag to Reorder

Action	Service	From Source		To Des
ALLOW	Multicast DNS Port: 5353 Protocol: udp	192.168.1.0/24	→	224.0.
ALLOW	Remote Login (SSH) Port: 22 Protocol: tcp	The Internet	→	My Computer
ALLOW	Network Browser (SLP) Port: 427 Protocol: udp	192.168.1.0/24	→	My Computer
ALLOW	AppleShare IP/iDisk Port: 548 Protocol: tcp	192.168.1.0/24	→	My Computer

Rule #: 27 items found Firewall Logging is Enabled

	Timestamp	Destination		Source	Protocol	Rule No
DENY	Jan 28 11:05:29	192.168.1.4 snmptrap (162)	IN	192.168.1.1 Port 37914	UDP	52008
DENY	Jan 28 11:05:29	192.168.1.4 snmptrap (162)	IN	192.168.1.1 Port 37915	UDP	52008
ALLOW	Jan 28 11:05:37	224.0.0.251 Port 5353	IN	192.168.1.6 Port 5353	UDP	2003
ALLOW	Jan 28 11:05:46	192.168.1.4 ssh (22)	IN	192.168.1.6 Port 49575	TCP	2004

Refresh every 5.0 minutes Click the lock to disallow firewall monitoring

		From	To	Proto	Intf	Expert Options	Traffic	Last Accessed	Rule#	Filter
DENY	ALL			to				allow	2006	2.168.1.0/24 to 192.168.1.4 548 in recv er
DENY	ALL			548				allow	2006	2.168.1.4 548 to 192.168.1.0/24 out xmit
ALLOW	OUT	192.168.1.4 Any Port	Any Host Any Port	ip	en0	None	2 packets 0.2K bytes	Tue Jan 28 11:04:38	52007	allow ip from 192.168.1.4 to any out xmit e
DENY	ALL			192.16				deny	52008	to 192.168.1.4 in recv en0
ALLOW	ALL	Any Host Any Port	Any Host Any Port	ip		None	42119 packets 16638.7K bytes	Tue Jan 28 11:04:43	65535	allow ip from any to any

BrickHouse

Network Access Restrictions

- MAC Address Restrictions
- General Access Restrictions
- Apple Mail Server
- File Sharing
- Other Open Source Products

MAC Addresses

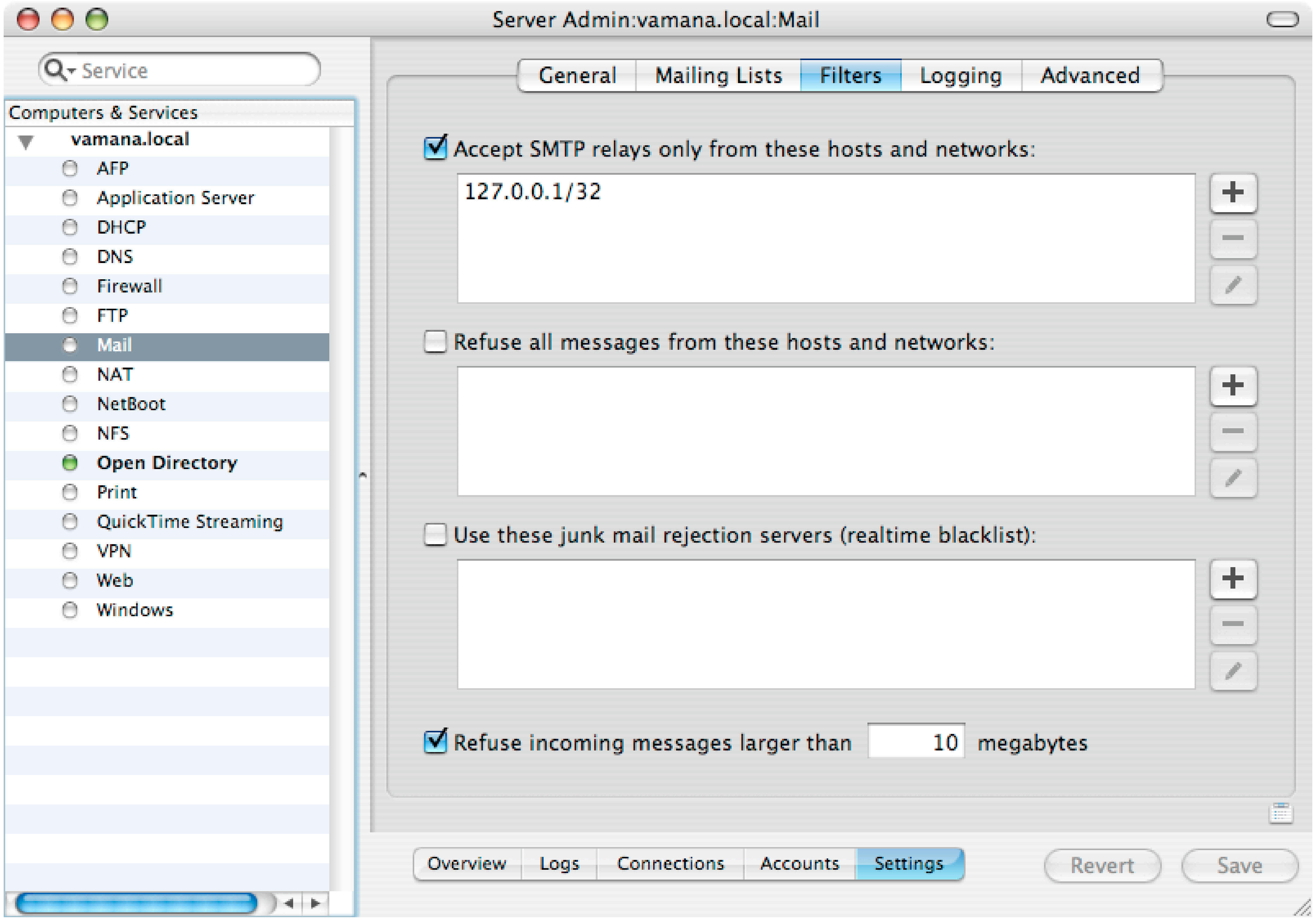
- Mac OS X Server DHCP server
  - Prevent obtaining IP addresses from DHCP by listing allowed MACs under Filter preferences for DHCP/NetBoot in Server Settings
    - Panther: Under NetBoot in Server Admin (now separate from DHCP)
      - Not sure whether these restrictions apply only to NetBoot clients or to DHCP generally
  - Configuration stored in OD under `/config/dhcp`
- AirPort Base Station
  - Limit all wireless access to select list of MAC addresses under Access Control in AirPort Admin Utility
- NB: Nowadays, MAC addresses are easily reset with commonly available software

Network Access General

- TCP Wrapper offers general access restrictions
  - Configured in `/etc/hosts.allow` and `/etc/hosts.deny`
  - Mac OS X software built with TCP Wrapper support includes OpenSSH, UCD SNMP, and `xinetd`
- `xinetd` is more secure alternative to `inetd`
  - Configured in `/etc/xinetd.conf` and service files in `/etc/xinetd.d/`
    - `only_from`, `no_access`, `access_times`
  - IP-based ACLs supplemented by TCP Wrapper, unless `flags = NOLIBWRAP` specified

Apple Mail Server

- Apple Mail Server (AMS) is derived from AppleShare IP (network software for the legacy Mac OS), and is bundled with Mac OS X Server
- Configured from Server Settings
    - Config stored in `/config/AppleMailServer` in local OD domain
- SMTP relay restrictions
    - Under Incoming Mail preferences, "Allow SMTP relay for: only hosts in this list:" prevents use as open relay
        - By default, lists only IP addresses associated with server's loopback and Ethernet interfaces
    - Under Filter preferences, can reject mail from servers listed explicitly, on a blackhole list, or with mismatched hostname/IP
    - Panther: Server software has changed (from AMS to Postfix), but GUI options are similar



[✓] Accept SMTP relays only from these hosts and networks:

127.0.0.1/32

[ ] Refuse all messages from these hosts and networks:

[ ] Use these junk mail rejection servers (realtime blacklist):

[✓] Refuse incoming messages larger than 10 megabytes

Panther Server SMTP relay restrictions

File Sharing

- Samba
  - Configured in `/etc/smb.conf`
    - `hosts allow`, `hosts deny`
- NFS
  - Restrict access to specified hosts or network addresses on Mac OS X Server with Workgroup Manager, under NFS Export Settings
    - Configuration stored in `clients` property under `/exports/`*`share_point`* in `local` OD database

File Sharing

- FTP
  - `lukemftpd` (vanilla OS X)
    - Configured in `/etc/ftpusers`
      - `user_pat[:group_pat][@src_pat] [directive [class]]`
        - `*_pat`: `?` for single-character, `*` for multi-char replacement
        - `directive` defaults to `deny`
        - If no rule matches, access permitted
      - Examples
        - `*:ftpusers allow`
          - Anyone in `ftpusers` group allowed
        - `*@192.168.1.0/24 allow`
          - Anyone from local network allowed

- FTP (cont'd.)
  - `lukemftpd` (cont'd.)
    - Test access rules with:
      - `ftpd -C` *`username`*`; echo $?`
      - Prints `0` if *`username`* would be allowed, `1` if not
  - `xftpd` (OS X Server)
    - Based on `wu-ftpd`
    - Configured in `/Library/FTPServer/Configuration/ftphosts`
      - `{ allow | deny }` *`username source`* `[`*`source`*`...]`
        - *`source`* is hostname, IP address, or subnet address

- FTP (cont'd.)
  - `xftpd` (cont'd.)
    - Configured in `/Library/FTPServer/Configuration/ftpaccess`
      - `deny` *`source file`*
        - If client address maches *`source`*, display contents of *`file`* and deny access
      - `dns refuse_no_reverse` *`file`*, `dns refuse_mismatch` *`file`*
        - Reject clients with mismatched or no reverse lookup, after displaying contents of *`file`*

Open Source

- Apache
  - Configured in `/etc/httpd/httpd.conf`
    - `Allow from`, `Deny from`, `Order`, `HostnameLookups Double`, `Satisfy`, `Options`, `AllowOverride`, `AccessFileName`
    - Also, `/etc/servermgrd/servermgrd.conf` on Mac OS X Server, for the Server Manager Daemon (an instance of Apache)
- Common UNIX Printing System (CUPS)
  - Configured in `/etc/cups/cupsd.conf`
  - `Allow from`, `Deny from`, `Order`, `BrowseAllow from`, `BrowseDeny from`, `BrowseOrder`, `HostnameLookups Double`, `Satisfy`

Open Source

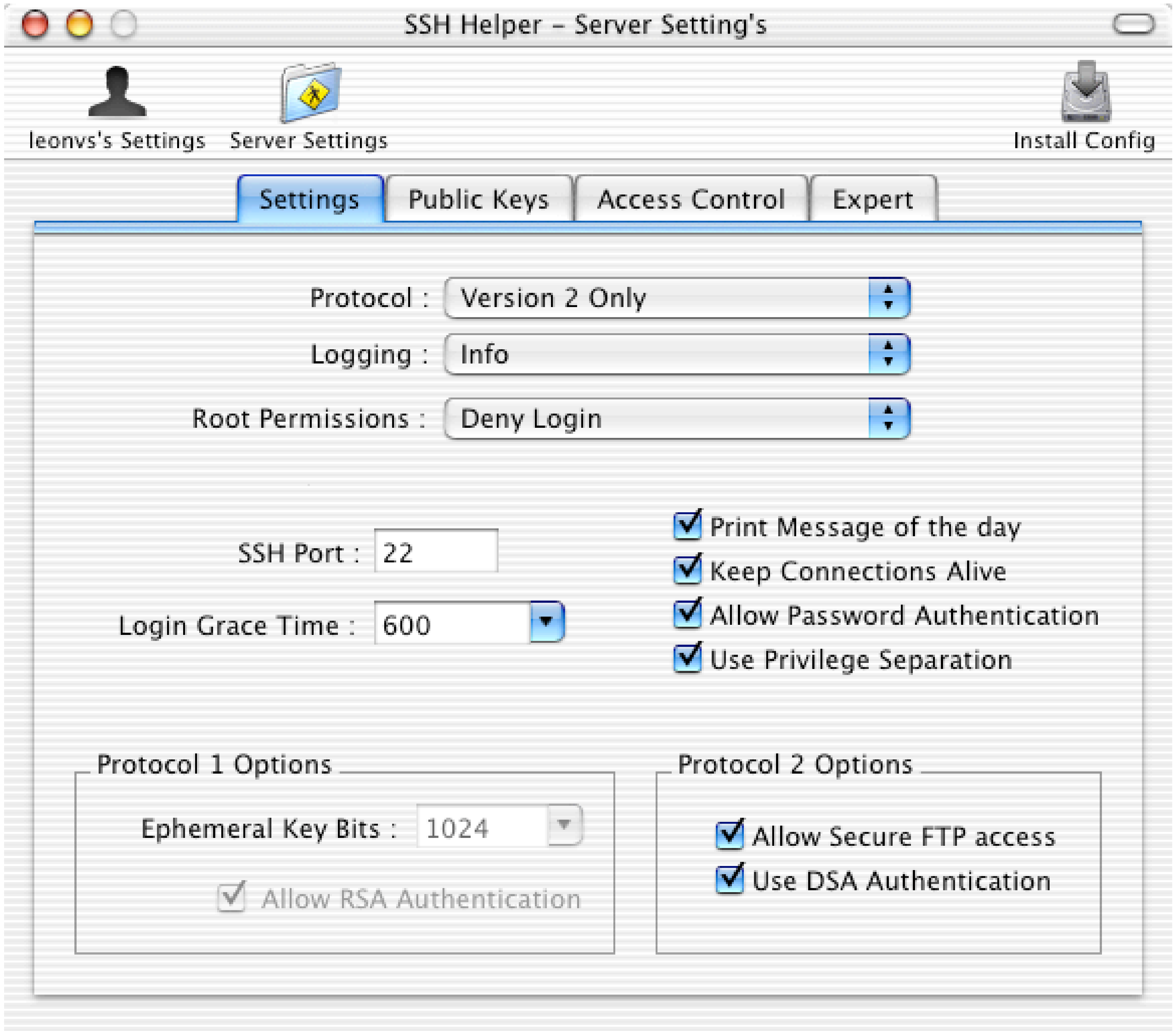
- BIND
  - Configured in `/etc/named.conf`
    - `acl` statements, `allow-*` and `blackhole options`
- NTP
  - Configured in `/etc/ntp.conf`
    - `restrict` *`source_address`* `[mask` *`netmask`*`] [`*`flag`*`]...`
    - Flags include `ignore`, `noserve`, `noquery`
  - View with `ntpdc -c reslist hostname`

Encrypting Network Transmissions

- SSH
- SSL/TLS
- IPsec
- PPTP
- WEP

Network Encryption SSH

- Secure Shell
- OpenSSH ships with Mac OS X
- Enable from System Preferences, under Sharing, as "Remote Login"
- AFP encryption runs over an SSH tunnel
- SSH Helper (`http://www.gideonsoftworks.com/sshhelper.html`)
  - Freeware
  - Manages `sshd_config`, `authorized_keys`, etc.



Protocol : Version 2 Only

Logging : Info

Root Permissions : Deny Login

SSH Port : 22

Login Grace Time : 600

- [x] Print Message of the day
- [x] Keep Connections Alive
- [x] Allow Password Authentication
- [x] Use Privilege Separation

Protocol 1 Options

Ephemeral Key Bits : 1024

- [x] Allow RSA Authentication

Protocol 2 Options

- [x] Allow Secure FTP access
- [x] Use DSA Authentication

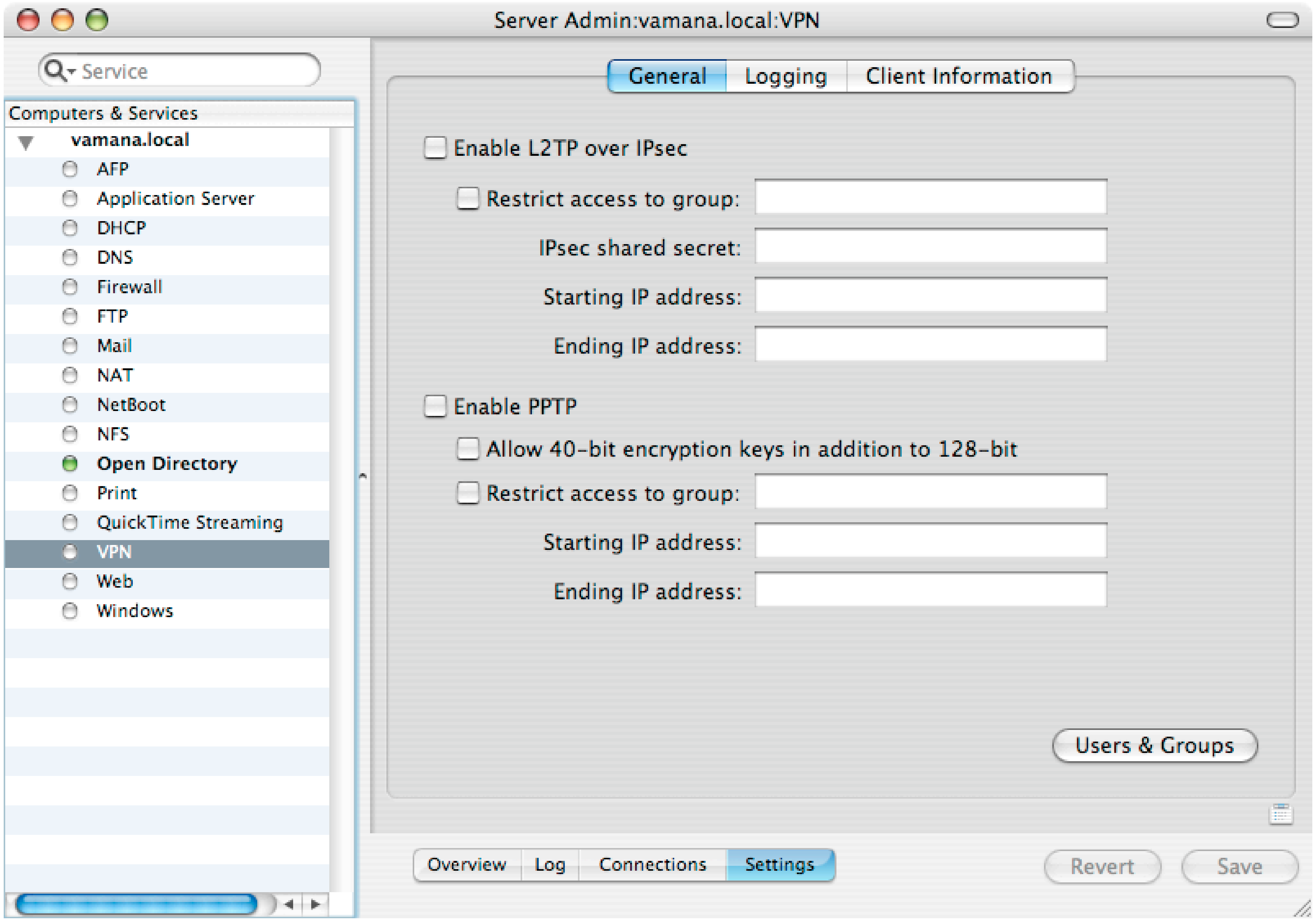
SSH Helper

X Network Encryption SSL/TLS

- Secure Sockets Layer/Transport Layer Security
- Many services can be set up to use SSL (most based on OpenSSL)
  - OpenLDAP
  - Apache, WebDAV (Finder as client doesn't work with SSL)
  - CUPS (but it breaks printing if you're not careful, and even if you are, the CUPS SSL implementation appears to be very unstable)
  - Samba (but you'll have to compile your own)
    - Panther: Samba is compiled with SSL support
  - Sendmail, Postfix, Cyrus IMAP
  - Server Manager Daemon
  - QuickTime Streaming Server admin web interface
  - Apple Mail Server IMAP (based on CDSA, not OpenSSL)

Network Encryption IPsec

- IP Security
- Based on KAME, same as used on other BSD platforms
- Provides authentication and encryption at a low layer of the TCP/IP stack, so it's transparent to applications
  - Unlike SSL, services don't need to be set up individually, or even know it's going on
- No built-in GUI
  - Panther has GUI (client in Internet Connect, server in Server Admin)
  - Managed with `setkey`, `racoon`, and `racoonctl`, and some `sysctl` variables: `net.inet.ipsec.*`, `net.inet6.ipsec6.*`, `net.key.*`
- Third-party GUIs
  - VPN Tracker (`http://www.equinux.com/us/products/vpntracker/`)
  - VaporSec (`http://www.afp548.com/Software/VaporSec/`), free


Server Admin:vamana.local:VPN

General | Logging | Client Information

☐ Enable L2TP over IPsec

☐ Restrict access to group:

IPsec shared secret:

Starting IP address:

Ending IP address:

☐ Enable PPTP

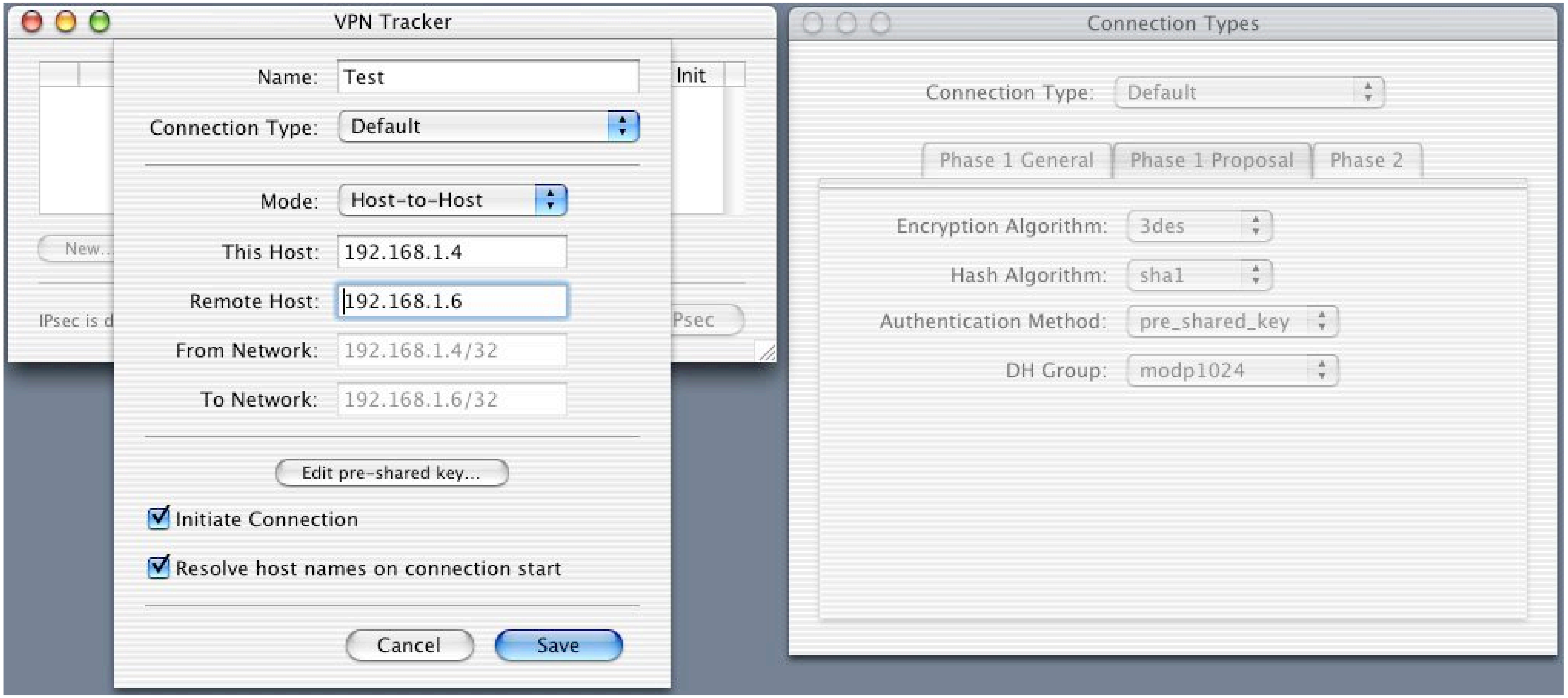
☐ Allow 40-bit encryption keys in addition to 128-bit

☐ Restrict access to group:

Starting IP address:

Ending IP address:

Panther Server Admin VPN settings



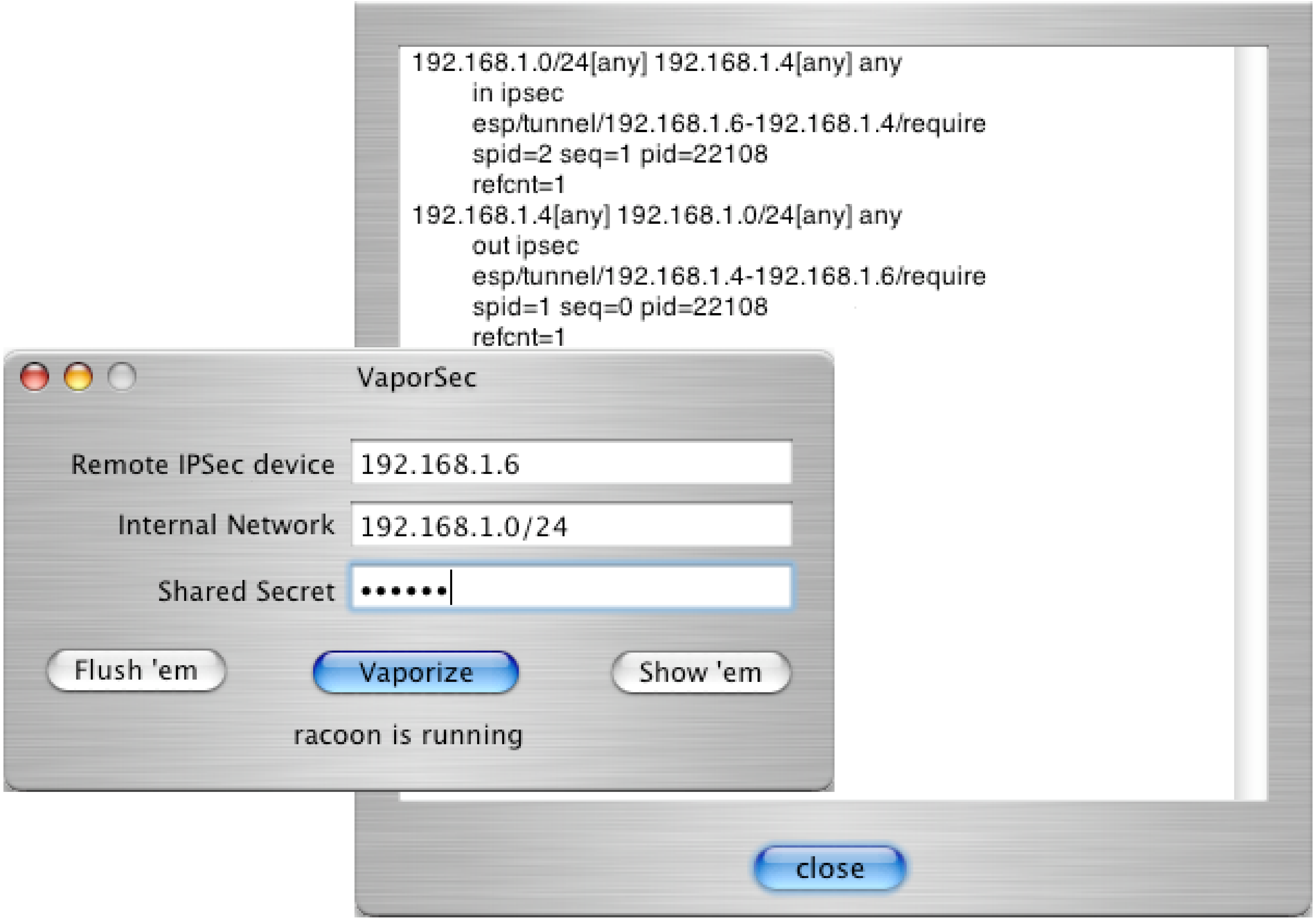
Name:	Test
Connection Type:	Default
Mode:	Host-to-Host
This Host:	192.168.1.4
Remote Host:	192.168.1.6
From Network:	192.168.1.4/32
To Network:	192.168.1.6/32

☑ Initiate Connection

☑ Resolve host names on connection start

Connection Type:	Default
Encryption Algorithm:	3des
Hash Algorithm:	sha1
Authentication Method:	pre_shared_key
DH Group:	modp1024

VPN Tracker



```
192.168.1.0/24[any] 192.168.1.4[any] any
        in ipsec
        esp/tunnel/192.168.1.6-192.168.1.4/require
        spid=2 seq=1 pid=22108
        refcnt=1
192.168.1.4[any] 192.168.1.0/24[any] any
        out ipsec
        esp/tunnel/192.168.1.4-192.168.1.6/require
        spid=1 seq=0 pid=22108
        refcnt=1
```

VaporSec

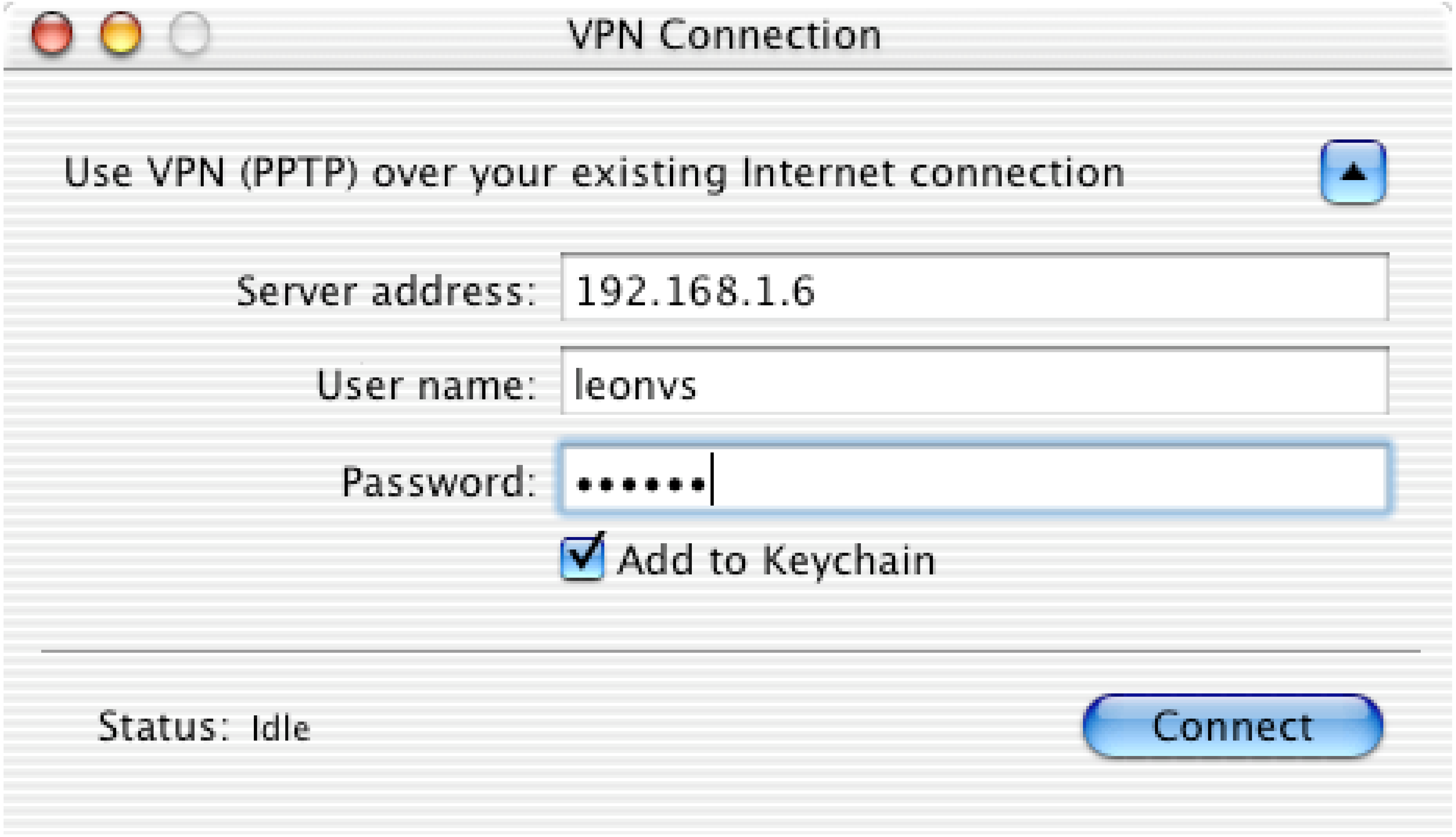
Remote IPSec device: 192.168.1.6
Internal Network: 192.168.1.0/24
Shared Secret: ••••••

racoon is running

VaporSec

PPTP

- Point-to-Point Tunneling Protocol
- Primarily used for personal VPNs by Windows clients
- Configured in Internet Connect application
- On Mac OS X Server, `vpnd` provides a PPTP front end to the PPP daemon
  - Details in man page


VPN Connection

Use VPN (PPTP) over your existing Internet connection

Server address: 192.168.1.6

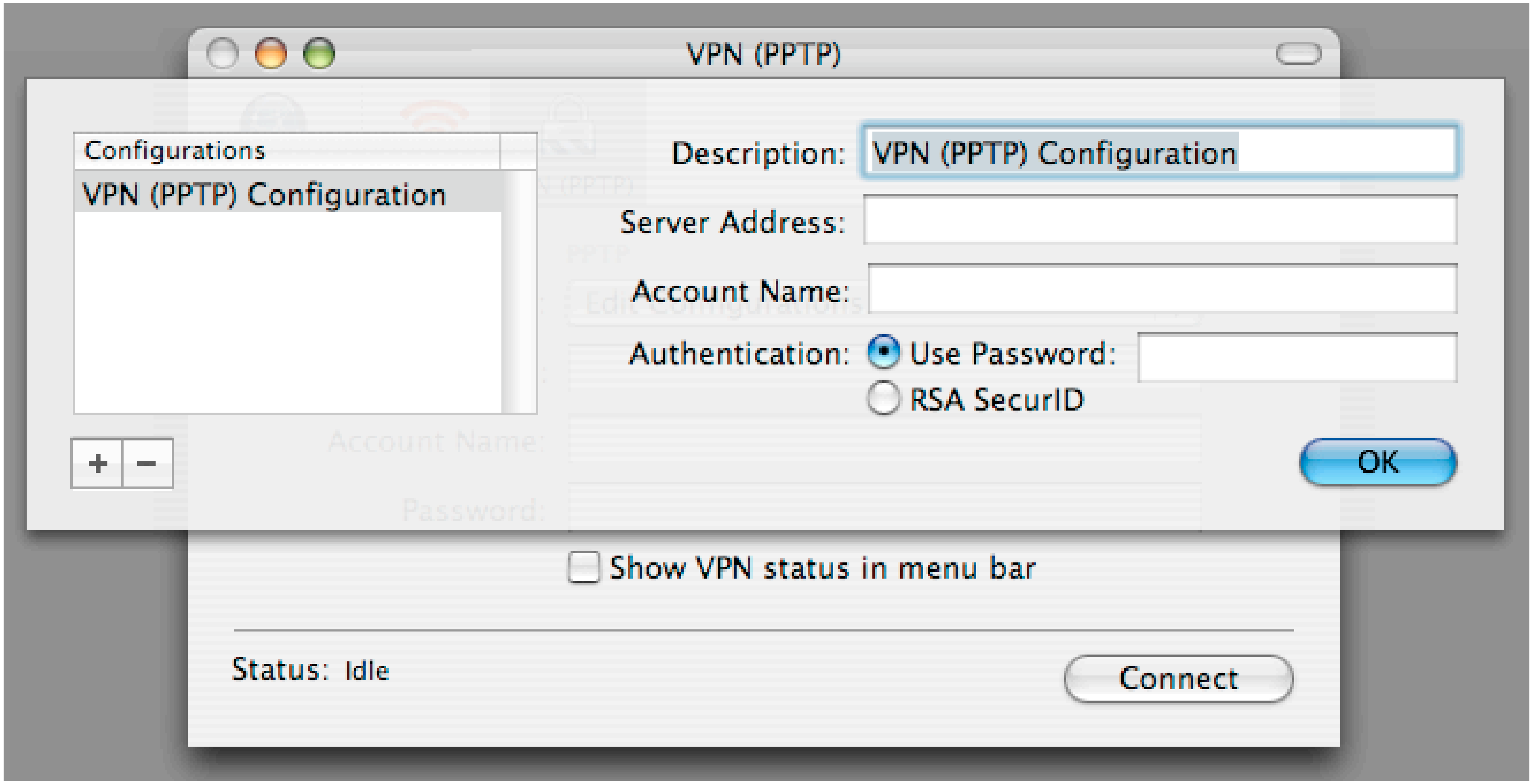
User name: leonvs

Password: ••••••

[✓] Add to Keychain

Status: Idle

Setting up PPTP in Internet Connect



VPN (PPTP) Configuration

Description: VPN (PPTP) Configuration

Server Address:

Account Name:

Authentication: [✓ Use Password:] RSA SecurID

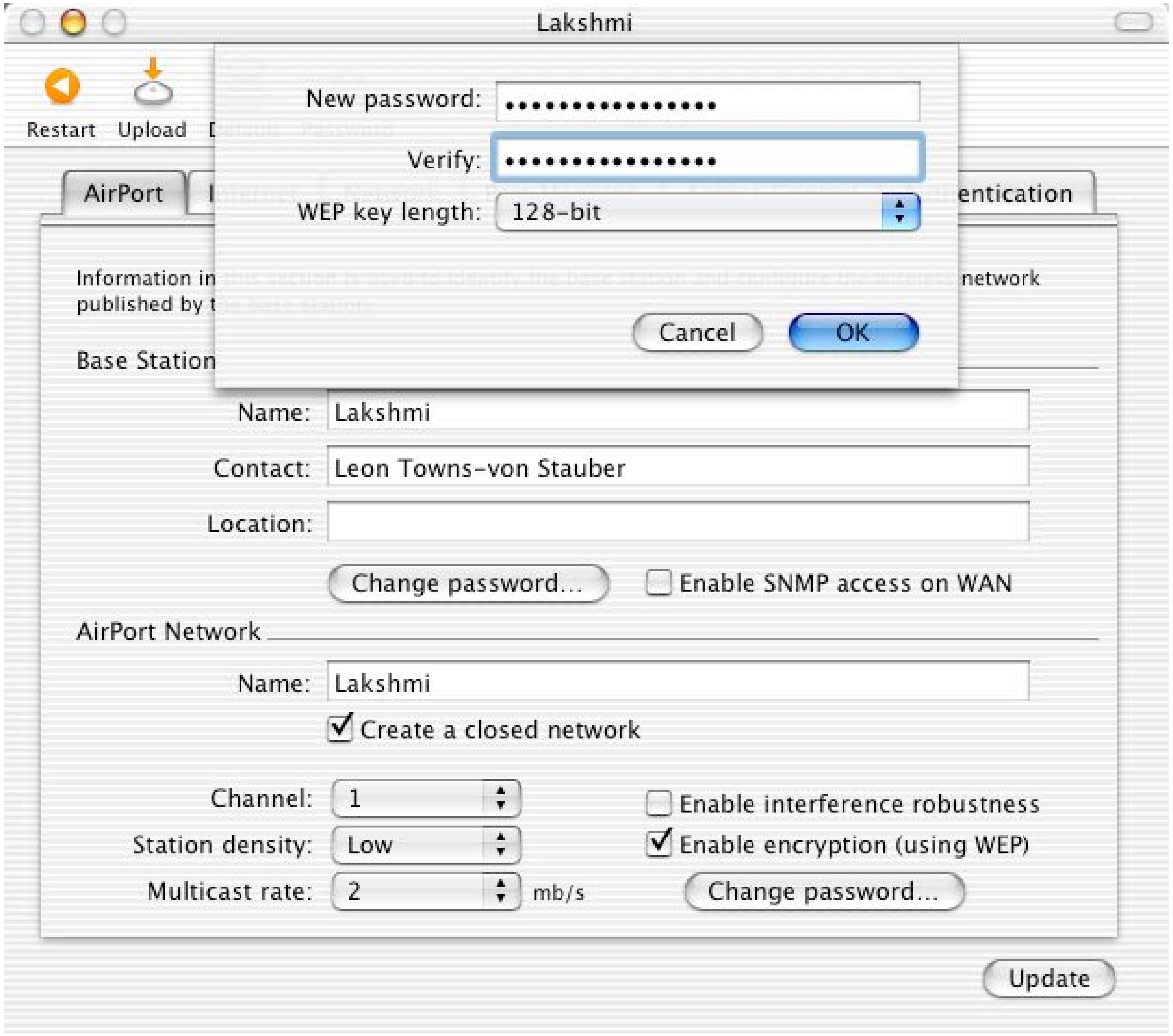
[ ] Show VPN status in menu bar

Status: Idle

Setting up PPTP in Panther’s Internet Connect

Network Encryption WEP

- Wired Equivalent Privacy
- Used to protect Wi-Fi (802.11) connections
  - AirPort is Apple's brand name for 802.11
- Minimal encryption; soon to be supplanted by stronger crypto algorithms
- Use a strong 128-bit password, and create a closed network
- Consider IPsec for additional security



New password: ••••••••••••••••

Verify: ••••••••••••••••

WEP key length: 128-bit

Information in network published by t

Base Station

Name: Lakshmi

Contact: Leon Towns-von Stauber

Location:

Change password...

Enable SNMP access on WAN

AirPort Network

Name: Lakshmi

✓ Create a closed network

Channel: 1

Station density: Low

Multicast rate: 2 mb/s

Enable interference robustness

✓ Enable encryption (using WEP)

Change password...

AirPort Base Station settings

Password Protection

- Locking User Access
- Open Firmware
- Password Shadowing
- Setting Password Requirements

Locking User Access

- Set screen lock under System Preferences->Screen Effects->Activation
    - Panther: System Preferences->Security->"Require password to wake this computer from sleep or screen saver"
    - Logged-in user's name and password, or admin user's name and password, unlocks screen saver
- Lock Terminal or console sessions with `lock` (see man page)
    - E.g., `lock -p -t 10000000`

Password Protection Open Firmware

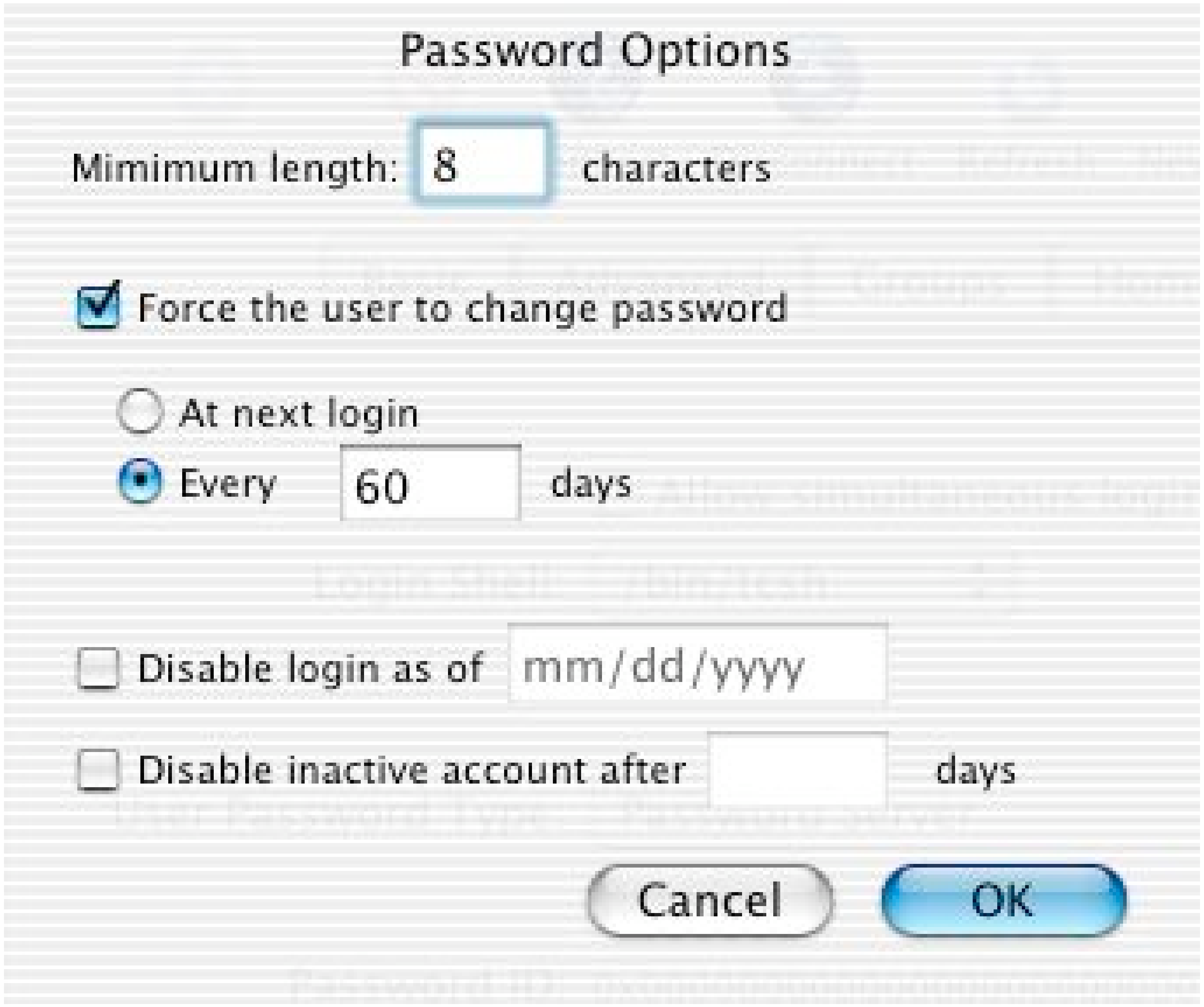
- Setting Open Firmware password disables key sequences that allow booting from an alternate device
  - Open Firmware Password application at `http://www.apple.com/downloads/macosx/apple/openfirmwarepassword.html`
    - Sets `security-password` OF variable, and sets `security-mode` to `command`
    - Can do the same with `nvram` CLI tool
- Setting `security-mode` to `full` prevents booting without someone available to enter the password

Password Shadowing

- It doesn't work, so don't bother trying
  - Access to authentication data mediated by `root`-owned processes (primarily `lookupd` and `DirectoryService`), so everyone can access it
    - The best you can do is set up shadowing, and make the workarounds non-obvious (i.e., "security through obscurity")
- Besides, it involves keeping auth data in flat files, which means you lose the convenience of using a directory service
- Use Password Server
- Panther shadows passwords by default!
  - `ShadowHash authentication_authority`
  - Hash is encrypted, and stored in `/var/db/shadow/hash/`, readable only by `root`

Password Requirements

- By default, `/usr/bin/passwd` requires at least 5 characters in password
  - Set `security_options` Open Directory property to increase that to 8 characters, with at least 1 non-alpha
    - `sudo nicl` *`domain`* `-create security_options secure_passwords`
    - Only affects `passwd` CLI tool
- Password Server lets you set minimum length (in Workgroup Manager, under Advanced->Options... when a user account is selected)
- Panther Server includes new options


Password Options

Mimimum length: 8 characters

[✓] Force the user to change password

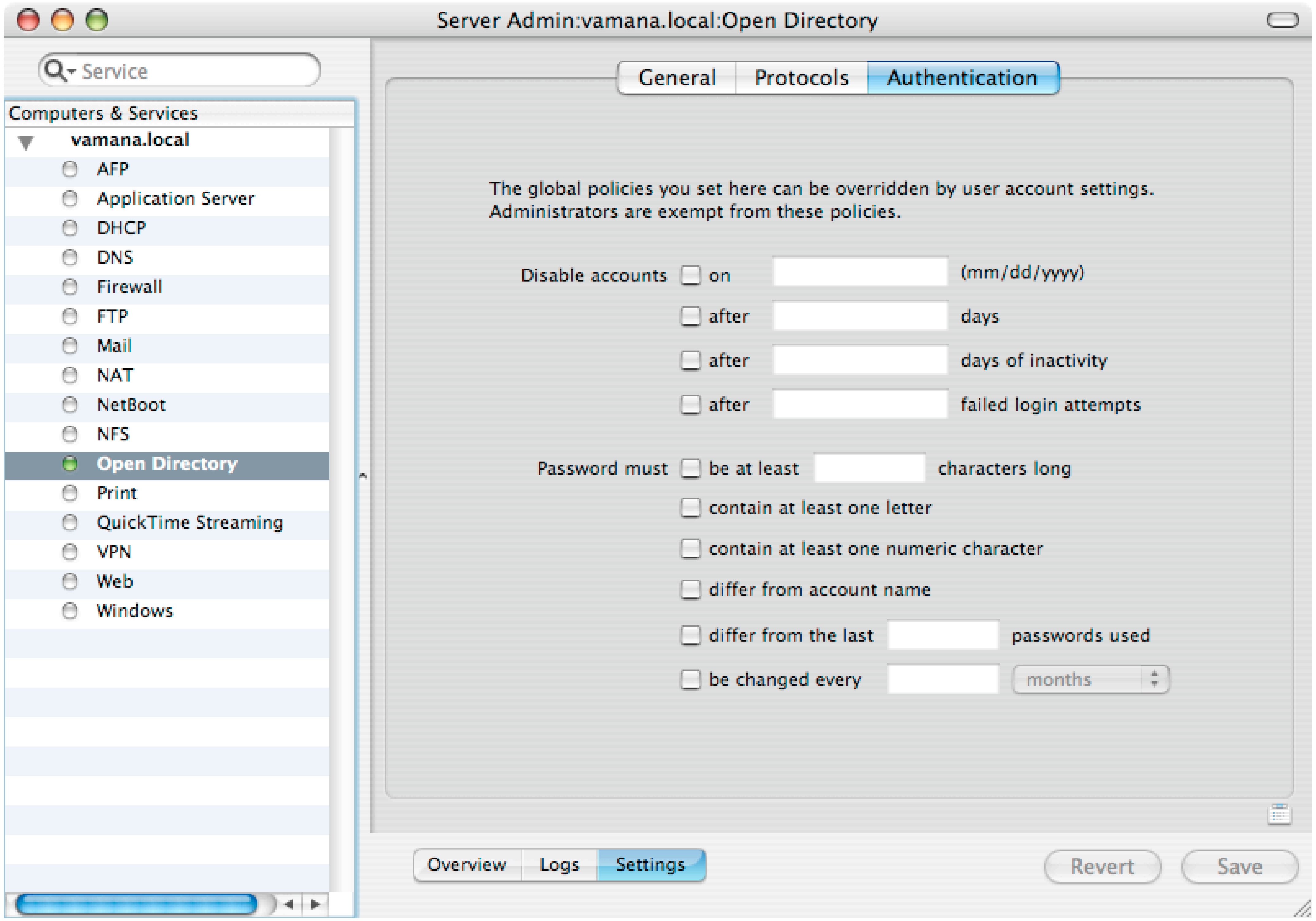
○ At next login

● Every 60 days

☐ Disable login as of mm/dd/yyyy

☐ Disable inactive account after days

Password Server options



The global policies you set here can be overridden by user account settings. Administrators are exempt from these policies.

Disable accounts ☐ on ____ (mm/dd/yyyy)
☐ after ____ days
☐ after ____ days of inactivity
☐ after ____ failed login attempts

Password must ☐ be at least ____ characters long
☐ contain at least one letter
☐ contain at least one numeric character
☐ differ from account name
☐ differ from the last ____ passwords used
☐ be changed every ____ months

Panther Server password options

Miscellany

- Login Options
- Rendezvous
- Process Accounting
- Virus Protection
- STOS and SE-Darwin

Login Options

- Bad ideas for security in System Preferences->Accounts
- Under Users, Set Auto Login... triggers automatic login after bootup, to emulate a single-user personal computer
  - Username saved as value of `autoLoginUser` in `/Library/Preferences/com.apple.loginwindow.plist`
  - Encrypted password apparently saved in `/etc/kcpassword`
- Under Login Options, can choose to have list of system accounts displayed at login (instead of simple username/password dialog)
- Can display a password hint after series of unsuccessful login attempts
  - Consider `cron` job to wipe out `hint` and `_writers_hint` properties from user account entries in Open Directory

Rendezvous

- Rendezvous is Apple's implementation of Zeroconf (`http://www.zeroconf.org/`)
- Automatic link-local addressing
  - Allows any device to acquire valid IP
  - Not a big deal, though: open DHCP does the same, and it's easy to find a valid IP once you have physical access to the network anyway

Rendezvous

- Multicast DNS (mDNS), DNS Service Discovery (DNS-SD)
  - Rogue responder easier to set up than with regular DNS (no need to hijack a DNS server's IP address), could serve bogus and misleading data (e.g., `mail`)
  - `mDNSResponder` may have undiscovered bugs; unlikely, due to open source and compactness
  - Disabling
    - Client: Remove or rename `/etc/resolver/`
    - Server: Remove or rename `mDNSResponder` startup item or executable
    - Watch out for OS updates

Process Accounting

- Process accounting functional as of 10.2
- Enable with:
    - `mkdir /var/account`
    - `touch /var/account/acct`
    - `chmod 0600 /var/account/acct`
    - `accton /var/account/acct`
- Every process logged to `/var/account/acct`, viewable with `lastcomm`
- Accounting log supposed to be rotated by `/etc/periodic/daily/500.daily` `cron` job, but the `sa` command doesn't exist on Mac OS X
    - Replace `sa` line with `cat /dev/null > acct`

Virus Protection

- No Mac OS X-specific malware to date
  - Still, a possibility for the future
  - And Mac OS X can act as a carrier
- Commercial anti-virus products
  - Norton AntiVirus (`http://www.symantec.com/nav/nav_mac/`)
  - VirusBarrier (`http://www.intego.com/virusbarrier/`)
  - Anti-Virus (`http://www.sophos.com/products/software/antivirus/savmac.html`)

STOS and SE-Darwin

- The Secure Trusted Operating System (STOS) Consortium (`http://www.stosdarwin.org/`) brings together representatives from the U.S. federal government, academia, and private industry to work on advanced security capabilities using the Darwin kernel as a starting point
- SE-Darwin meant to conform to Common Criteria
  - Similar to other government-sponsored programs working with open-source OSes, such as Security-enhanced Linux
- Also working on other projects:
  - Apache SSL module using CDSA instead of OpenSSL
  - PGP implementation using CDSA
  - OpenSSH on CDSA
  - Honeypots on Darwin

Resources

- Web Sites
- Mailing Lists
- Books

Web Sites

- Apple's Mac OS X site
  - `http://www.apple.com/macosx/`
- Apple's Security Updates page
  - `http://www.apple.com/support/security/security_updates.html`
- MacSecurity.org
  - `http://www.macsecurity.org/`
- SANS Reading Room: Apple Issues
  - `http://rr.sans.org/mac/mac_list.php`
- Mac OS X Hints
  - `http://www.macosxhints.com/`
- Occam's Razor Apple/NeXT page
  - `http://www.occam.com/links/apple.html`

Mailing Lists

- security-announce (Apple)
  - `http://lists.apple.com/mailman/listinfo/security-announce/`
- MacSec (MacSecurity.org)
  - `http://www.macsecurity.org/mailman/listinfo/macsec/`
- MacOSX-admin (Omni Group)
  - `http://www.omnigroup.com/developer/mailinglists/macosx-admin/`
- macos-x-server (Apple)
  - `http://lists.apple.com/mailman/listinfo/macos-x-server/`

X Resources

Books

- Practical Unix & Internet Security
  - Simson Garfinkel, Gene Spafford, Alan Schwartz
- Mac OS X in a Nutshell
  - Jason McIntosh, Chuck Toporek, Chris Stone

X Closing Remarks

- This talk has focused on security issues, both risks and opportunities, mostly specific to Mac OS X
  - But remember that Mac OS X is UNIX, and similar considerations apply as to any other UNIX platform
- Evaluation forms
  - Some considerations
    - Level of detail, pacing, slides
    - Content (things you'd have liked to see, or liked to see gone)
    - Compatibility with SA tutorial (if you took that)
- BoF: Tuesday 9 PM
- Slides available at `http://www.occam.com/osx/`
- Q & A