web2ldap and OpenLDAP special features

- Personal info:
  - Michael Ströder
  - Consulting as a freelancer
  - Focus on LDAP-based directories and PKI
  - Author of web2ldap.de
  - Maintainer of python-ldap and author of modules ldif, dsml, ldap.schema etc.

Why this presentation?

- Present web2ldap and some features maybe helpful for others when experimenting with OpenLDAP
- Sometimes standard authors and server-side LDAP developers have a different view than client developers ;-)
  => discussion wanted

History

- Started in diploma thesis back in '98
- In the beginning a simple search and download tool for certificates stored on LDAP server (Umich 3.3)
- Add / modify entries for Netscape Communicator address book
- But time went on...

Utilization by author

- Mainly personal needs trigger incorporation of features
- Sometimes an I-D inspires development
- Personal goals:
  - Avoid using the command-line but without limitation
  - Experiment with special LDAP features
- In consulting projects:
  - Dig unknown servers with sometimes brain-dead configuration (e.g. check schema or PKI data)
  - Demonstrate directory administration use-cases

What it's not!

- A light-weight search interface
- An administration user interface for your secretary to maintain the corporate address book
- An admin interface which does many things automagically (uidNumber assignment, non-directory side-effects)

Installation: Prerequisites

- Runs on Unix or Win32
- At least needs Python 2.3+, python-ldap and pyweblib
- Some more modules for additional features
- See http://www.web2ldap.de/install.html
- Script sbin/checkinst.py displays status of installed modules

Installation: python-ldap

- Wrapper module around OpenLDAP libs => needs OpenLDAP 2.2+ client libs (together with Cyrus SASL, Kerberos, OpenSSL)
- Some tweaking in python-ldap's setup.cfg:
  - Adjust paths (include_dirs and library_dirs)
  - Link against libldap_r if possible!
  - libs = ldap_r lber sasl2 ssl crypto
- python setup.py build, python setup.py install
- Choose right python executable!
- python-ldap for Win32 without SASL support (MingW)

Features: Configuration

- Default configuration convenient for most use-cases and local use => no big obstacles for beginners
- Default configuration is rather strict (regarding security)
- Configuration syntax is Python
- Cascaded configuration avoids having to repeat same configuration items, but somehow difficult to understand
- See http://www.web2ldap.de/web2ldapcnf.html
- People do not read docs anyway, so they stick with default configuration ;-)

Features: Running modes

- Long-running multi-threaded process
- Two modes:
  - Stand-alone based on builtin web server of Python's standard lib
  - Under control of web server through FastCGI / SCGI (e.g. Apache with mod_fastcgi / mod_scgi)
- Which one? As usual it depends...
- Stand-alone mode preferred when using web2ldap as a local client together with SASL/GSSAPI or ldapi:// with SASL/EXTERNAL

Features: LDAP connection handling

- Exactly one LDAP connection held persistent within one web2ldap session
- no special authorization within web2ldap
- user can personally bind to LDAP server, solely the DSA's access control is in effect
- LDAPv3 vs. LDAPv2 automagically chosen
- Automatic reconnect based on bind information cached in RAM, no connection information written to disk
- Following referrals to other server requires user interaction with new bind information

Features: Bind

- Simple bind with user search
- SASL bind
  - Password-based (DIGEST-MD5, etc.)
  - EXTERNAL with StartTLS, ldaps://, ldapi:// (needs configuration)
  - EXTERNAL with ldapi:// (needs stand-alone mode)
  - GSSAPI (needs stand-alone mode and kinit)
- Not trivial to pack all these variants into reasonable user interface
  => not usable for unexperienced users
- „Who Am I?“ extended operation is used to retrieve authz-DN

Features: Schema support

- attributeTypes, dITContentRules, ldapSyntaxes, objectClasses are affecting presentation and input forms
- Schema browser displays all references incl. inheritance
- Additional schema attributes displayed in schema browser: matchingRules, matchingRuleUse, dITStructureRules, nameForms
- Different subschema subentries in DIT is supported
- Schema caching: subschema subentry itself and DN2schema mapping
- Fall-back to local LDIF schema definition (e.g. for LDAPv2)
- Special handling for collective attributes
- Handling attributeTypes, ldapSyntaxes can be overridden by plug-ins

Features: Adding/modifying entries

- Adds / modifys entries obeying attributeTypes, dITContentRules, ldapSyntaxes, objectClasses
- Two steps: 1. Choose object class(es) and 2. edit and submit new entry
- Accepts LDIF input data including :<
- Client-side syntax checking helps finding errors
- Upload of binary attributes (e.g. jpegPhoto, userCertificate;binary) possible (but limitation regarding multi-valued attributes)
- Delta modification
- LDIF templates for new entries (configurable quick list)
- HTML templates for input forms (per object class)

Features: Other DIT modification

- It's possible to delete
  - single entries
  - whole sub-trees or subordinate entries/trees
  - single binary attributes
- Renaming with new superior entry (move)
- Group administration (with arbitrary group-like entries)
- Set password(s)

Features: LDAPv3 extensions

- Manage DSA IT mode (RFC 3296)
- Subentries (RFC 3672 and draft-ietf-ldup-subentry-07)
- Relax Rules Control (draft-zeilenga-ldap-relax)
- StartTLS
- "Who am I?" (RFC 4532)
- Sometimes it's difficult to design an appropriate user interface for behaviour of extended controls
  => normal users never use these features

Features: More helpful things under the hood...

- LDAP URL support (think of bookmarks)
- Automatic SRV _ldap._tcp lookup on noSuchObject
- OID registry for displaying details of attributes in root DSE
- honors attributes hasSubordinates, numSubordinates and subordinateCount if available
- export as LDIF, DSMLv1, vCard
- client-hashed passwords (OpenLDAP, Samba)
- displaying PKI data (X.509 certs and CRLs)
- ...time is running...more upon request... :-)

Usage with OpenLDAP: SASL/EXTERNAL with ldapi://

- Convenient connect as local client to local server
- Only meaningful if web2ldap is started in stand-alone mode as local user
- slapd.conf: authz-regexp maps Unix login to bind-DN
- User is not required to enter login data
- Effective authz-DN displayed as result of „Who Am I?“
- <web2ldap-URL>?ldapi://<Unix Domain Socket>/...
- Bind with SASL/EXTERNAL
- LDAP URL ext. x-saslmech=EXTERNAL for bookmarks

Usage with OpenLDAP: Add referral entry

- Connect as authorized admin to server
- [ConnInfo]: Set „Manage DSA IT“ to „Enabled“
- Push button „Change options“
- Browse to referral entry (not displayed as search continuation)
- Add/modify the referral entry and its attribute ref
- Similar for enabling draft-zeilenga-ldap-relax: [ConnInfo]: „Relax Rules“

Usage with OpenLDAP: back-config (1)

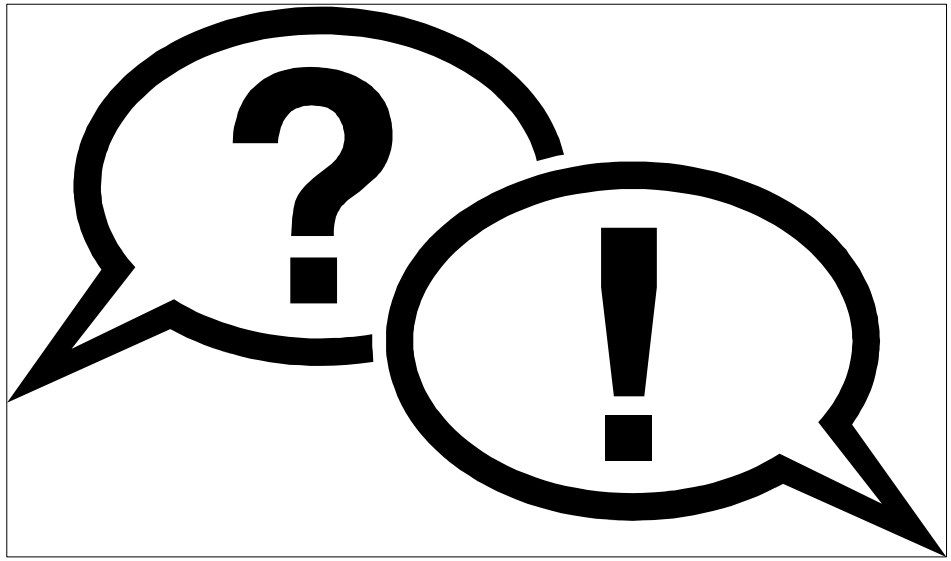
- Hopefully makes configuration through back-config a little bit easier than through command-line
- Let's be honest: Not really foolproof...
- X-ORDERED not yet supported (looks complicated)
- Use-case: Create a backend
- Configuration section needed for cn=config:
  - LDIF templates form a quick-list
  - HTML templates for textual explanation

Usage with OpenLDAP: back-config (2)

Example shortened, use example in source distribution:

```
'ldap:///cn=config': Web2LDAPConfig(
  input_template={
    'olcIncludeFile':'inputform_olcIncludeFile.html',
    'olcBdbConfig':'inputform_olcBdbConfig.html',
    'olcHdbConfig':'inputform_olcHdbConfig.html',
  },
  addform_entry_templates={
    'Adress book (back-bdb)':'add_olcBdbConfig_AddressBook.ldif',
    'Unix Users (back-hdb)':'add_olcHdbConfig_UnixUsers.ldif',
    'Schema config':'add_olcSchemaConfig.ldif',
  }),
```

Discussion



- Questions?
- Suggestions for improvement?
- Feature requests?