LDAP103: Authentication

If your find these documents useful and feel the need to express that opinion in a tangible way, consider selecting an item from my Amazon Wish List.

awilliam@whitemice.org

Copyright

If you find this document useful or further it's distribution we would appreciate you letting us know.

Role Playing

- We are a climate control products distribution company called Ragnarok Service & Supply, which owns the DNS domain rss.nw.
  - We have a heterogeneous network
    - Windows 2000, XP, 9x, and Linux workstations.
      - A Samba DC manages authentication for Win32 services.
    - Various network services:
      - Intranet (Apache)
      - Extranet (Apache)
      - Mail
        - IMAP: Cyrus IMAPd
        - SMTP: Sendmail
      - Logon (shell)
        - Hylafax (authenticates to PAM)
      - Jabber IM (using Jabberd2)
      - OpenGroupware

Goals

- A single username and password for each user to all services.
  - Secure IMAP and SMTP services.
    - Use DIGEST MD5 for authentication to these services.
      - With DIGEST MD5 the user's password is never exposed between the client and the server.
  - Support a variety of services.
    - Squid proxy-cache access limitations.
      - NTLM and BASIC support.
    - Apache with .htaccess limitations.

Requirements

- A basically functioning OpenLDAP DSA
  - None of these configurations have been tested on a version prior to 2.2.6 while versions up to 2.2.19 have been tested.
    - Any OpenLDAP 2.1.x or later should work, theoretically.
- Cyrus SASL packages and source code (or source packages)
  - Versions prior to 2.2.14 will not work.
  - It is optimal to have version 2.2.19 or later.

Checking Mechs

- Since we want to support DIGEST MD5 we need to verify what SASL mechanisms our DSA supports
  - What mechanisms are supported depend on how the DSA was compiled and what SASL libraries are installed.
    - $ ldapsearch -x -H ldap://lif/ -LLL -s "base" -b "" supportedSASLMechanisms

      dn:
      supportedSASLMechanisms: DIGEST-MD5
      supportedSASLMechanisms: PLAIN
      supportedSASLMechanisms: GSSAPI
      supportedSASLMechanisms: LOGIN
      supportedSASLMechanisms: ANONYMOUS
    - rpm -qa | grep cyrus-sasl

      cyrus-sasl-plain-2.1.18-29
      cyrus-sasl-digestmd5-2.1.18-29
      cyrus-sasl-devel-2.1.18-33.5
      cyrus-sasl-gssapi-2.1.18-29
      cyrus-sasl-2.1.18-33.5

Configuring for DIGEST MD5

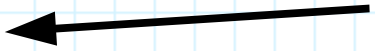
- Most modern authentication mechanisms (DIGEST, Kerberos, CHAP) require storing clear text passwords in the credential database.
  - Edit slapd.n1.conf
    - password-hash {CLEARTEXT}
      - The DSA will now store the contents of the userpassword field as a clear text value.
- The DSA must be able to map usernames to user objects.
  - Edit slapd.conf

```
sasl-regexp
  uid=(.*),cn=DIGEST-MD5,cn=auth
  ldap:///ou=Entities,ou=SAM,dc=rss,dc=nw??sub?(&(uid=$1)(objectclass=account))
sasl-regexp
  uid=(.*),cn=PLAIN,cn=auth
  ldap:///ou=Entities,ou=SAM,dc=rss,dc=nw??sub?(&(uid=$1)(objectclass=account))
```

Testing SASL To The DSA

- You should now be able to perform a SASL bind to the DSA by specifying your username.
  - The OpenLDAP utilities will automatically select the most secure available SASL mechanism.
    - In this case DIGEST MD5.

```
$ ldapsearch -U awilliam -Y digest-md5 -H ldap://lif/ \
      -b dc=rss,dc=nw 'uid=awilliam'
SASL/DIGEST-MD5 authentication started
Please enter your password: *********
SASL username: awilliam
SASL SSF: 128
SASL installing layers
# extended LDIF
...
```



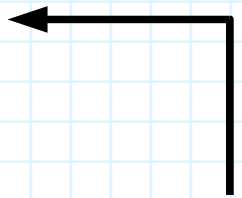
Digest MD5 authentication works!

Configuring saslauthd

- Many services want to support the PLAIN mechanism.
  - The saslauthd daemon/service provides password authentication for a variety of services.
    - sendmail
      - SMTP
    - Cyrus
      - IMAP
      - POP3
      - SIEVE

- Make sure your version of saslauthd supports the ldap mechanism.
  - Built with --with-ldap

```
ldap_servers: ldap://lif/ ldap://liftrasir/
ldap_use_sasl: yes
ldap_mech: DIGEST-MD5
ldap_auth_method: fastbind
ldap_version: 3
# ldap_realm:
```



/etc/saslauthd.conf

SuSe 9.2's saslauthd

- SuSe 9.2's saslauthd does not include the ldap mechanism.
  - But it is trivial to add the functionality to the package.

```
$ vi cyrus-sasl.spec
....
-Release:      7
+Release:      8ldap
....
=./configure    --libdir=%{_libdir} \
=               --with-plugindir=%{_libdir}/sasl2 \
=               --prefix=/usr \
=               --sysconfdir=/etc \
=               --mandir=%{_mandir} \
=               --with-saslauthd=/var/run/sasl2/ \
=               --enable-pam \
=               --enable-sample \
=               --enable-login \
=               --enable-gssapi \
-              --enable-krb4=no
+               --enable-krb4=no \
+               --with-ldap
=make sasldir=%{_libdir}/sasl2
$ rpmbuild -bb cyrus-sasl.spec
$ rpm -Uvh ../RPMS/i586/cyrus-sasl*
```

Starting saslauthd

- The mechanism used by saslauthd is stored in the the saslauthd file in /etc/sysconfig.
  - SASLAUTHD_AUTHMECH=ldap
- saslauthd is started with
  - /etc/rc.d/saslauthd start
- Set to start at a given run-level
  - chkconfig –level 345 saslauthd on
- You may want to hack the service file to add various directives to fine-tune saslauthd's performance.
  - Number of worker threads can be set via "-n #"
  - Enable caching of authentication credentials with "-c"
    - Time for credentials to live: "-t *{seconds}*"
    - Size of hash table: "-s *{kilobytes}*"

Testing saslauthd

- The Cyrus SASL project includes a utility for directly testing the functionality of saslauthd: testsaslauthd

Building and installing testsaslauthd on SuSe 9.2

```
$ cd /usr/src/packages/BUILD/cyrus-sasl-2.1.19/saslauthd
$ make testsaslauthd
$ mv testsaslauthd /usr/local/bin
```

Testing saslauthd with testsaslauthd

```
$ testsaslauthd -u username -p wrongpassword
0: NO "authentication failed"
$ testsaslauthd -u username -p correctpassword
0: OK "Success."
```

The /etc/sasldb2 file

If /etc/sasldb2 does not exist, create it by adding a bogus user:

```
$ saslpasswd2 username
Password: ********
Again (for verification): ********
```

Set sufficient access permissions so DSA can read the file:

```
$ setfacl -m u:ldap:r /etc/sasldb2
$ setfacl -m g:ldap:r /etc/sasldb2
$ getfacl /etc/sasldb2
getfacl: Removing leading '/' from absolute path names
# file: etc/sasldb2
# owner: root
# group: root
user::rw-
user:ldap:r--
group::r--
group:ldap:r--
mask::r--
other::---
```

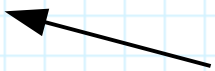
- Traditional use of SASL involved storing user names and secrets in a file: /etc/sasldb2
  - Even though we are not going to use this file, it must exist, and the DSA must be able to read the file.

ldapdb

- Current packages of Cyrus SASL do not support using an LDAP DSA as the credential repository.
  - You can store passwords outside the DSA in the the /etc/sasldb2 file using the SASL password utilities.
    - A pain, since credential information will not be replicated to other hosts.
  - You can install the ldapdb SASL plugin.
    - The ldapdb plugin is located in the contrib directory of the OpenLDAP source code.
      - openldap-2.2.19/contrib/ldapsasl

Building ldapdb

```
$ rpm --install cyrus-sasl-2.1.19-7.src.rpm
$ cd /usr/src/packages/SPECS
$ rpmbuild -bb cyrus-sasl.spec
$ cd {...}/openldap-2.2.19/contrib/ldapsasl
$ cp ldapdb.c /usr/src/packages/BUILD/cyrus-sasl-
2.1.19/plugins/ldapdb.c
$ cd /usr/src/packages/BUILD/cyrus-sasl-2.1.19/plugins/
$ make ldapdb.lo
$ /bin/sh ../libtool --mode=link gcc  -Wall -W -g -O2 -L/
usr/local/lib -Wl,-rpath,/usr/local/lib -module -export-dynamic
-rpath /usr/lib/sasl2 -o libldapdb.la  -version-info 2:4:0 ldapdb.lo
-lldap -llber -lssl -lcrypto
$ cp .libs/libldapdb* /usr/lib/sasl2
```



Your SASL plugin directory.

SASL ProxyAuthz

- The LDAP backend for SASL (ldapdb) must bind to the DSA as an entity with permission to authenticate users.
  - This is called Authentication by Proxy
  - You can use authentication proxies to exert control over what realms and services can authenticate as various entities within the Dit.

Create the proxy authentication entity:

```
dn: uid=saslauthzproxy,ou=System,ou=Entities,ou=SAM,dc=rss,dc=nw
objectClass: top
objectClass: account
objectClass: simpleSecurityObject
ou: SASL
userPassword: WallaWallaBoomBang
uid: saslauthzproxy
```

Add Proxy Auth Rights

- Modify the SASL proxy account object to contain the appropriate saslAuthzTo value.
  - saslAuthzTo is an operational attribute, it cannot be added via 3rd party LDAP clients such as GQ.

```
$ cat setupProxy.ldif
dn: uid=saslauthzproxy,ou=System,ou=Entities,ou=SAM,dc=rss,dc=nw
changetype: modify
add: saslAuthzTo
saslAuthzTo: ldap:///ou=Entities,ou=SAM,dc=rss,dc=nw??sub?(objectclass=account)
$ ldapmodify -Y digest-md5 -U adam -W -f setupProxy.ldif
Enter LDAP Password:
SASL/DIGEST-MD5 authentication started
SASL username: adam
SASL SSF: 128
SASL installing layers
modifying entry "uid=saslauthzproxy,ou=System,ou=Entities,ou=SAM,dc=rss,dc=nw
```

SaslAuthzTo is “operational”

```
$ ldapsearch -LLL -Y digest-md5 -U adam uid=saslauthzproxy
SASL/DIGEST-MD5 authentication started
Please enter your password:
....
dn: uid=saslauthzproxy,ou=System,ou=Entities,ou=SAM,dc=rss,dc=nw
objectClass: top
objectClass: account
objectClass: simpleSecurityObject
ou: SASL
uid: rss-sasl
$ ldapsearch -LLL -Y digest-md5 -U adam uid=saslauthzproxy +
SASL/DIGEST-MD5 authentication started
Please enter your password:
...
dn: uid=saslauthzproxy,ou=System,ou=Entities,ou=SAM,dc=rss,dc=nw
...
saslAuthzTo: ldap:///ou=Entities,ou=SAM,dc=rss,dc=nw??sub?(objectclass=account)
...
```

- You don't see the SaslAuthzTo attribute.
- You see the SaslAuthzTo attribute because you asked for operational attributes.

Test The saslAuthzTo user

- Perform a SASL bind to the DSA as the SASL Proxy user.
  - If this doesn't work, what follows won't work.

```
$ ldapsearch -Y digest-md5 -U saslauthzproxy uid=awilliam
SASL/DIGEST-MD5 authentication started
Please enter your password: *******
SASL username: saslauthzproxy
SASL SSF: 128
SASL installing layers
...
dn: cn=Adam Williams,ou=People,ou=Entities,ou=SAM,dc=rss,dc=nw
homeDirectory: /home/awilliam
```

sasl-authz-policy

- Add the following like to /etc/openldap/slapd.conf
  - sasl-authz-policy to
    - This enables utilization of SASL proxy via the saslAuthzTo attribute.
- Restart the DSA
  - /etc/rc.d/ldap restart

Testing SASL Proxy Auth

- Verify that proxy authentication is working.
  - The password is that of the "saslproxyauthz" user, and it should be able to assume the identity of any user matching the defined rule.
  - If this doesn't work, what follows will not work.

another user

```
$ ldapwhoami -U saslproxyauthz -Y digest-md5 -X u:awilliam  -W
Enter LDAP Password: ********
SASL/DIGEST-MD5 authentication started
SASL username: u:awilliam
SASL SSF: 128
SASL installing layers
dn:cn=adam williams,ou=people,ou=entities,ou=sam,dc=rss,dc=nw
```

Guard saslAuthzTo

- Since the saslAuthzTo attribute influences how the DSA performs authentication, access to the attribute should be extremely restricted.
  - Users do no even need read access to the attribute since it is only used internally by the DSA.
    - Provide “auth” access to all connections.
    - Provide write access only to trusted entities
    - Provide read access to replicants

```
access to attrs=saslAuthzTo
    by group/groupOfUniqueNames/uniqueMember="cn=DSA
Administrators,ou=Groups,ou=Access Control,dc=whitemice,dc=org" write
    by group/groupOfUniqueNames/uniqueMember="cn=Full SyncRepl
Consumers,ou=Groups,ou=Access Control,dc=whitemice,dc=org" read
    by * auth
```

Postfix (authenticated SMTP)

•Edit the Postfix's main.cf configuration file and verify or set the following options:

/etc/postfic/main.cf
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
smtpd_sasl_local_domain =
broken_sasl_auth_clients = yes

•Support applications (M$-Outlook) developed by people who can't spell "R.F.C."

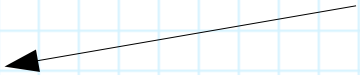
•There also needs to be a smtpd.conf file, this should be provided by postfix package.

/usr/lib/sasl2/smtpd.conf
pwcheck_method: saslauthd
mech_list: plain login

•The default SASL smtpd.conf file, supports PLAIN authentication via saslauthd.

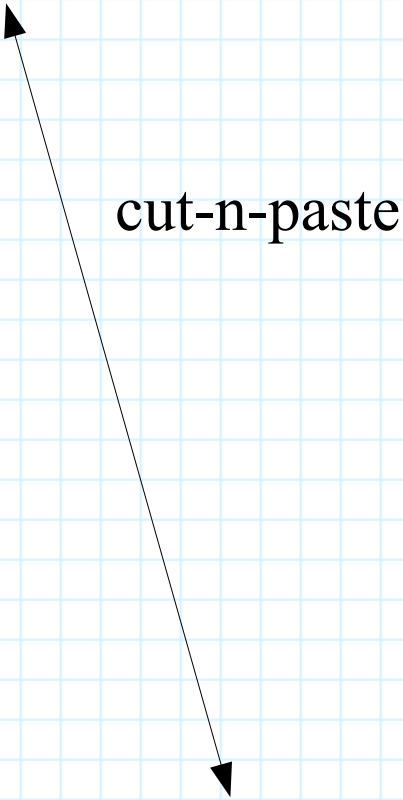
Testing PLAIN SMTP Auth

```
$ /etc/rc.d/postfix restart
$ telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 lif.rss.nw ESMTP Postfix
EHLO example.com
250-lif.rss.nw
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-AUTH LOGIN PLAIN
250-AUTH=LOGIN PLAIN
250 8BITMIME
```



Testing PLAIN SMTP Auth

```
$ perl -MMIME::Base64 -e 'print encode_base64
  ("username\0username\0password");'
YWRhbQBLaXNzQX4uaWUAS2lzc0Fubmll
$ telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 lif.rss.nw ESMTP Postfix
EHLO example.com
250-lif.rss.nw
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-AUTH LOGIN PLAIN
250-AUTH=LOGIN PLAIN
250 8BITMIME
AUTH PLAIN YWRhbQBLaXNzQX4uaWUAS2lzc0Fubmll
235 Authentication successful
```



Setting up 'real' SASL

- To implement DIGEST-MD5 we need to use the ldapdb SASL plugin that we built earlier.
  - Change the contents of the smtpd.conf file and restart the MTA.
    - /etc/rc.d/postfix restart

/usr/lib/sasl2/smtpd.conf

```
ldapdb_uri: ldapi://
ldapdb_id: saslproxyauthz
ldapdb_pw: Bingo
ldapdb_mech: DIGEST-MD5
mech_list: digest-md5
#ldapdb_starttls: [ try | demand ]
```

- ldapi:// for domain socket connections, otherwise use a ldap:// URL.
- List of mechanisms to advertise to the client.

Cyrus IMAPd SASL

- Once the ldapdb is installed and SASL proxy is configured, setting up SASL authentication in Cyrus IMAPd is as simple as editing the configuration file.
  - Restart the Cyrus IMAPd service
    - /etc/rc.d/cyrus restart

/etc/imapd.conf

```
sasl_pwcheck_method: auxprop
sasl_auxprop_plugin: ldapdb
sasl_mech_list:  digest-md5
sasl_ldapdb_uri: ldapi:///
sasl_ldapdb_id: saslproxyauthz
sasl_ldapdb_pw: Bingo
sasl_ldapdb_mech: DIGEST-MD5
# imapd.conf will also have other contents
....
```

- ldapi:// for domain socket connections, otherwise use a ldap:// URL.

Squid

- Squid can use NTLM to perform authentication to a Samba DC
  - The Samba DC in turn can use an LDAP SAM.
    - That scenario is really a Samba configuration issue.
- Squid can also perform BASIC authentication to an LDAP DSA.
  - A BASIC/LDAP 'helper' is included in the Squid distribution
    - This helper uses lookups for the user DN and simple binds to perform authentication, not SASL.

'configure' options

```
--enable-basic-auth-helpers=LDAP,NCSA,PAM,SMB \
--enable-ntlm-auth-helpers=SMB \
--enable-auth="ntlm basic"
```

Authentication Helpers

- Squid communicates with its helpers by starting them as child processes and writing to their standard in and reading the results from their standard out.
  - Helpers are configured using auth_param directives in the form of auth_param *{scheme} {option} {value}*
  - The options and values supported depend on how squid was compiled or packaged.

```
auth_param basic program /usr/lib/squid/squid_ldap_auth -b "dc=rss,dc=nw" -h
localhost -f "(&(objectclass=posixAccount)(uid=%s))" -s sub
auth_param basic children 5
auth_param basic Realm Squid proxy-caching web server
```

Helper program

Keep 5 helpers of this type running.

Prompt title seen by users.

squid_ldap_auth

- The helper for direct authentication against an LDAP server is squid_ldap_auth.
  - squid_ldap_auth *-b basedn {options...}*
  - Options:
    - *-b {search base}* This is the only required parameter.
    - *-f {filter}*
    - *-u {Username attribute}*
    - *-s { base | one | sub }*
      - Controls the scope of the search.
    - *-D {bind as DN}*
    - *-w {bind password}*
    - -h *{LDAP Server}* Defaults to “localhost”
    - *-p {port}*
    - *-P*
      - Use persistent connections, otherwise each search is a new connection.
    - *-R* Disables following referrals
    - *-a { never | always | search | find }*
      - Determines how aliases are processed

squid_ldap_match

- An additional helper for performing group membership matching is also available.
  - http://marasystems.com/download/LDAP_Group/

```
external_acl_type ldap_group concurrency=10 %LOGIN /
usr/lib/squid/squid_ldap_match -b "ou=Groups,ou=SAM,dc=rss,dc=nw" -f "(&
(objectclass=posixGroup)(memberuid=%u)(cn=%g))" -s sub -P -h localhost -S

acl ldap_internet external ldap_group internet
. . .
http_access allow ldap_internet
. . .
```

ACL is created.

At this point in the ACL stack any request by anyone in the group internet is approved.

- For additional information on external_acl_type see the Squid documentation:
  - http://www.squid-cache.org/

Building squid_ldap_match

- Recent SuSE RPMs include the squid_ldap_match helper.
- Otherwise the squid_ldap_match ACL helper is distributed as a simple C file, that must be compiled.
  - gcc -llber -lldap -o squid_ldap_match squid_ldap_match.c
  - Move the resulting squid_ldap_match executable to an appropriate directory and make sure that the security context of the squid process has sufficient privileges to execute the file.
    - RedHat RPMs place the helpers in /usr/lib/squid.
    - SuSe RPMs place the helpers in /usr/sbin.

Squid configuration

```
auth_param ntlm program /usr/lib/squid/ntlm_auth backbone/liftrasir
auth_param ntlm children 5
auth_param ntlm max_challenge_reuses 0
auth_param ntlm max_challenge_lifetime 2 minutes

auth_param basic program /usr/sbin/squid_ldap_auth -b "ou=Entities,ou=SAM,dc=rss,dc=nw"
-h littleboy -f "(&(objectclass=account)(uid=%s))" -s sub
auth_param basic children 5
auth_param basic Realm Squid proxy-caching web server
. . .
external_acl_type ldap_group concurrency=10 %LOGIN /usr/sbin/squid_ldap_match -b
"ou=Groups,ou=SAM,dc=rss,dc=nw" -f "(&(objectclass=posixGroup)(memberuid=%u)(cn=%
g))" -s sub -P -h liftrasir -S
. . .
acl ldap_internet external ldap_group internet
acl public_sites url_regexp “"/etc/squid/public_sites.text"
acl banned_sites url_regexp “/etc/squid/banned_sites.text”
. . .
http_access deny banned_sites
http_access allow public_sites
http_access allow ldap_internet
. . .
```

NTLM is enabled against the Samba host liftrasir in the domain backbone.

Basic authentication is enabled for clients that don't support NTLM, this is performed directly against the DSA,

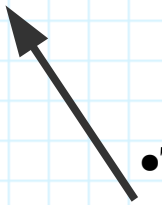
Define ACL for users who are members of the group “internet”.

Create ACL stack. All users are permitted to access sites matching the public sites ACL. All attempts to view banned sites are denied. User who are members of the group internet are permitted to view all other sites.

Apache

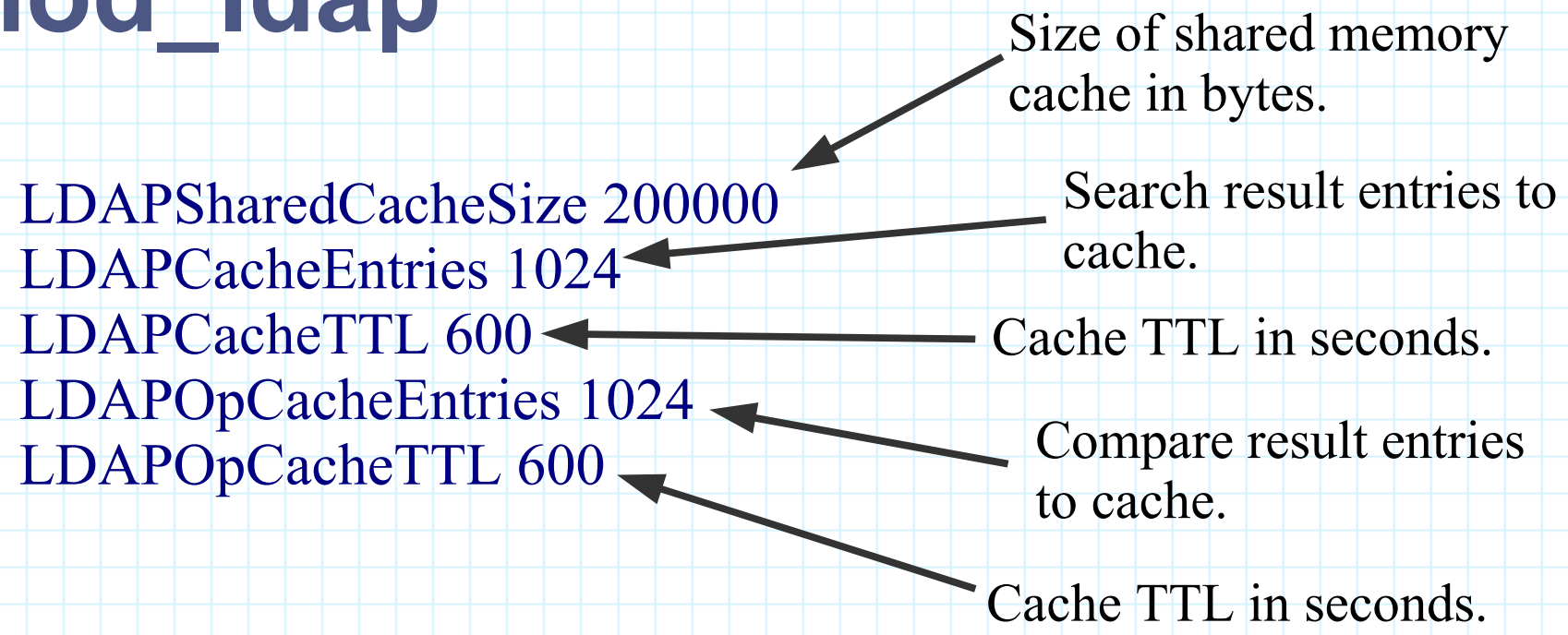
- Authentication in Apache is performed via mod_auth_ldap
  - mod_auth_ldap itself uses Apache's mod_ldap
    - mod_ldap includes aggressive caching for minimizing the load on the DSA and increasing performance.
      - mod_ldap provides a content handler that allows the administrator to monitor cache performance.

```
<Location /server/cache-info>
 SetHandler ldap-status
</Location>
```



- The URL http://{servername}/cache-info will provide LDAP cache performance statistics.

mod_ldap



- The behavior of the mod_ldap module can be adjusted via directives in the Apache configuration.

- For detailed configuration information:
  - http://httpd.apache.org/docs-2.0/mod/mod_ldap.html

mod_auth_ldap

- LDAP authentication is enabled for a directory within the Apache configuration,
  - The AuthLDAPURL defines how users are verified
    - ldap://*host*:*port*/basedn?attribute?scope?filter
      - Unused values can be left off.

- Scopes of one and sub are supported.
  - Default is “sub”

```
<VirtualHost *:80>
    ...
    <Directory /var/www/html/rss_nw>
    ...
    AuthType Basic
    AuthName "intranet"
    AuthLDAPURL "ldap://liftrasir:389/ou=Customers,dc=rss,dc=nw?uid"
    </Directory>
</VirtualHost>
```

Apache .htaccess files

```
<Files documents.php>
AuthType Basic
AuthName "intranet"
require valid-user
</Files>
```

- Require the user to be validated, but we don't care who they are.

```
<Files awilliam.php>
AuthType Basic
AuthName "intranet"
require user awilliam
</Files>
```

- Require a specific user.

```
<Files addevent.php>
AuthType Basic
AuthName "intranet"
require group cn=calendar,ou=Intranet,ou=Access Control,dc=rss,dc=nw
</Files>
```

- Require user to be a member of the specified group.

Apache .htaccess files

```
<Files misales.php>
AuthType Basic
AuthName "intranet"
require ldap-attribute st="Michigan" employeeType=Sales
</Files>
```

- Require a user with the provided properties.

```
<Files complex.php>
AuthType Basic
AuthName "intranet"
AuthLDAPURL ldap://liftrasir/ou=Customers,dc=rss,dc=nw?uid??(|(&
(st=Indiana)(custid=7*)))(uid=jmanager))
require valid-user
</Files>
```

- Override AuthLDAPURL and specify an entirely new filter.

mod_auth_ldap Directives

- AuthLDAPAuthoritative
  - Prevent attempting via other authentication methods if LDAP authentication fails.
    - Default on
      - Turn off to make M$-FrontPage happy
- AuthLDAPBindDN
  - DN to bind to DSA as for user search.
    - Also set AuthLDAPBindPassword
- AuthLDAPCharsetConfig
  - Set the charset used conversion.
    - LDAP is all in UTF-8
- AuthLDAPCompareDNOnServer
  - Compare Dns locally or ask DSA to do it.
    - Default is on (DSA performs comparisons)

mod_auth_ldap Directives

- AuthLDAPDereferenceAliases
  - Chase LDAP aliases
    - Default is always
      - Possible values are never, searching, finding, and always.
- AuthLDAPEnabled
  - Enabled or disable LDAP authentication.
- AuthLDAPFrontPageHack
  - Make FrontPage happy.
- AuthLDAPGroupAttribute
  - Attribute of group objects that contain members
- AuthLDAPGroupAttributeIsDN
  - Group attribute contains username or DN
- AuthLDAPRemoteUserIsDN
  - If set REMOTE_USER will be the DN of the authenticated user, rather than the username.

OpenGroupware

- OpenGroupware can authenticate user access utilizing an LDAP directory.
  - The requisite records are automatically created in the database the first time a user logs in.
  - By default OpenGroupware uses LDAPv2.
    - If a bind username and password are set Ogo uses LDAPv3

```
$ su – opengroupware
$ Defaults write NSGlobalDomain  LSAuthLDAPServer '"lif.rss.nw"'
$ Defaults write NSGlobalDomain \
  LSAuthLDAPServerRoot '"ou=Entities,ou=SAM,dc=rss,dc=nw'
$ Defaults write NSGlobalDomain LSAuthLDAPServerPort 389
$ Defaults write NSGlobalDomain LSUseLowercaseLogin "Yes"
$ Defaults write NSGlobalDomain LDAPInitialBindSpecific "YES"
$ Defaults write NSGlobalDomain LDAPInitialBindDN   \
      “cn=OpenGroupwareAuthenticator,ou=System,ou=Entities,ou=SAM...”
$ Defaults write NSGlobalDomain LDAPInitialBindPW “G8i-1cfhKrbr”
$ Defaults write NSGlobalDomain LDAPLoginAttributeName "cn"
```

Jabber

/opt/jabberd2/etc/c2s.xml

```
...
<!-- LDAP module configuration -->
      <ldap>
            <host>lif.rss.nw</host>
            <port>389</port>
            <--
                  <v3/><starttls/><ssl/>
            -->
            <binddn>uid=jabberd,ou=System Accounts,....</binddn>
            <bindpw>********</bindpw>
            <uidattr>uid</uidattr>
            <basedn>ou=Entities,ou=SAM,dc=rss,dc=nw</basedn>
      </ldap>
</authreg>
...
```

PAM

- We covered PAM authentication in LDAP102
  - To play nicely with other authentication mechanisms we want to ensure that we are using:
    - pam_password exop
      - This allows the DSA to select how the password is hashed.
  - It may be useful to specify a bind DN rather than having NSS operate anonymously
    - binddn uid=nss,ou=System,ou=Entities,ou=SAM,dc=rss,dc=nw
    - bindpw secret
      - This is not a secure connection, but does allow the administrator to customize the limits enforced against NSS
        - Limit should be greater than the number of groups, users, or IP services defined; which ever is the greatest number.

passwd

- The passwd command can be enabled in an LDAP network.
  - rootbinddn
    - This setting in /etc/ldap.conf makes binds when utilities are used as root use this alternate identity.
      - Password is stored in /etc/ldap.secret
        - Which should only be readable by root.
  - If the user object is a shadowAccount object then shadow attributes are updates as well.
- The smbk5pwd overlay can be used to extend the functionality of the change-password-extended-operation.
  - With smbk5pwd installed smbntpassword, sambalmpassword, and Hiemdal Kerberos attributes are updated as well as userpassword when the password is changed.

General Tips

- Passwords
  - You can make and set 'random' passwords with the ldappasswd command.
- Performance
  - Anonymous binds are faster than authenticated binds.
  - Use the domain socket for communication whenever possible.
  - Use the nscd name service cache on LDAP NSS clients.
  - Authentication uses the “uid” attribute constantly.
    - Place the ACL for “uid” near the top of your ACL stack.
    - Index “uid” for equality and presence.

Links

- mod_auth_ldap
  - http://httpd.apache.org/docs-2.0/mod/mod_auth_ldap.html
- mod_ldap
  - http://httpd.apache.org/docs-2.0/mod/mod_ldap.html
- OpenLDAP Administrator's Guide
  - Using SASL
    - http://www.openldap.org/doc/admin22/sasl.html
- LDAP and Jabberd2
    - http://jabberd.jabberstudio.org/2/docs/section04_7.html
- Postfix & Digest-MD5
  - http://www.billy.demon.nl/Postfix-SASL-authentication.html
- OpenLDAP overlays
  - http://www.symas.com/techtips/introtooverlays.html