1 Description

The LDAP module provides LDAP based user access control for the Dante SOCKS server.

Based on a users LDAP group membership or LDAP attributes, the module can be used to e.g., limit users access to web/ftp sites, or to only allow access for temporary staff to work related web/ftp sites.

If the username contains a domain extension, the module determines the LDAP server in an automatic way using the following method:

1. Extract the domain name from the username e.g., DOMAIN.COM from user@DOMAIN.COM when GSSAPI authentication is used.

2. Perform a DNS SRV record lookup (typically available in a Windows environment with Active Directory),
   (a) without SSL, from ldap.tcp.DOMAIN.COM.
   (b) with SSL, from ldaps.tcp.DOMAIN.COM, or if this entry does not exist, from ldap.tcp.DOMAIN.COM.

3. Perform a DNS A record lookup of DOMAIN.COM.

4. Use /etc/hosts file entry for DOMAIN.COM.

5. Sort entries by weight and priority and remove duplicates.

If the username does not contain any domain extension a pre-configured LDAP URL can be used to verify the LDAP group membership of users.

The module authenticates to the LDAP server using SASL/GSSAPI with the appropriate entry of the GSSAPI or LDAP specific keytab or the username/password provided as part of the LDAP URL.
2 Syntax

The syntax used to control the behavior of the LDAP module is as follows:

ldap.group: <LDAP-GROUP|LDAP-GROUP@|LDAP-GROUP@DOMAIN.COM>

The parameters are defined as follows:

LDAP-GROUP name of LDAP group to be used for any user.

LDAP-GROUP@ name of LDAP group to be used for users who have a domain extension in their username (e.g., user@DOMAIN1.COM).

LDAP-GROUP@DOMAIN.COM name of LDAP group to be used only for users who have a domain extension of DOMAIN.COM in their username.

This statement can be repeated.

ldap.group.hex: <LDAP-GROUP|LDAP-GROUP@|LDAP-GROUP@DOMAIN.COM>

The parameters are defined as follows:

LDAP-GROUP name of LDAP group in hex UTF-8 to be used for any user.

LDAP-GROUP@ name of LDAP group in hex UTF-8 to be used for users who have a domain extension in their username (e.g., user@DOMAIN1.COM).

LDAP-GROUP@DOMAIN.COM name of LDAP group in hex UTF-8 to be used only for users who have a domain extension of DOMAIN.COM (not in UTF-8) in their username.

For a translation of hex UTF-8 see for example http://www.utf8-chartable.de/unicode-utf8-table.pl

This statement can be repeated.

ldap.group.hex.all: <LDAP-GROUP|LDAP-GROUP@|LDAP-GROUP@DOMAIN.COM>

The parameters are defined as follows:

LDAP-GROUP name of LDAP group in hex UTF-8 to be used for any user.

LDAP-GROUP@ name of LDAP group in hex UTF-8 to be used for users who have a domain extension in their username (e.g., user@DOMAIN1.COM).

LDAP-GROUP@DOMAIN.COM name of LDAP group in hex UTF-8 to be used only for users who have a domain extension of DOMAIN.COM in hex UTF-8 in their username.

For a translation of hex UTF-8 see for example http://www.utf8-chartable.de/unicode-utf8-table.pl

This statement can be repeated.

ldap.domain: <DOMAIN>

The parameter is defined as follows:
**DOMAIN** default Kerberos domain to be used for pam/username authentication to emulate a GSSAPI user.

---

**ldap.url:** `<URL>`
The parameter is defined as follows:

**URL** LDAP URL of the following form:

```
ldap(s)://<username>:<password>@<host:port>/<basedn>
```

This statement can be repeated.

---

**ldap.server:** `<server@DOMAIN.COM>`
The parameter is defined as follows:

**server@DOMAIN.COM** LDAP server name of the LDAP server for domain `DOMAIN.COM`. This setting avoids the automated server determination via DNS SRV or A records.

This statement can be repeated.

---

**ldap.basedn:** `<base DN|base DN@DOMAIN.COM>`
The parameters are defined as follows:

**base DN** base DN for LDAP search for any LDAP server.

**base DN@DOMAIN.COM** the base DN for LDAP search for LDAP server for domain `DOMAIN.COM`.

This statement can be repeated.

---

**ldap.basedn.hex:** `<base DN|base DN@DOMAIN.COM>`
The parameters are defined as follows:

**base DN** base DN in hex UTF-8 for LDAP search for any LDAP server.

**base DN@DOMAIN.COM** base DN for LDAP search for LDAP server for domain `DOMAIN.COM`.

This statement can be repeated.

---

**ldap.basedn.hex.all:** `<base DN|base DN@DOMAIN.COM>`
The parameters are defined as follows:

**base DN** base DN in hex UTF-8 for LDAP search for any LDAP server.

**base DN@DOMAIN.COM** base DN for LDAP search for LDAP server for domain `DOMAIN.COM` in hex UTF-8.

This statement can be repeated.

---

**ldap.port:** `<PORT>`
The parameter is defined as follows:
**PORT** LDAP port to be used for automatic LDAP server determination if no SRV DNS records exist.

`ldap.port.ssl: <PORT>`
The parameter is defined as follows:

**PORT** LDAP SSL port to be used for automatic LDAP server determination if no SRV DNS records exist.

`ldap.ssl: no|yes`  
Require SSL for LDAP connection. The default value is no.

`ldap.certcheck: no|yes`  
Require SSL certificate check. The default value is no.

`ldap.certfile: <filename>`  
The parameter is defined as follows:

**filename** OpenLDAP CA certificate file name.

`ldap.certpath: <pathname>`  
The parameter is defined as follows:

**pathname** Sun/Mozilla LDAP SDK certificate database location.

`ldap.debug: <debug level>`  
The parameter is defined as follows:

**debug level** OpenLDAP debug level to set when OpenLDAP is used. The default value is 0.

`ldap.mdepth: <maximal search depth>`  
The parameter is defined as follows:

**maximal search depth** maximal depth of recursive group searches in Active Directory. The default value is 0.

`ldap.keeprealm: no|yes`  
Keep the realm name when comparing username with LDAP user attribute. The default value is no.

`ldap.filter: <filter>`  
The parameter is defined as follows:

**filter** search filter for an OpenLDAP server. The default filter is (memberuid=%s) and assumes a rfc2307bis schema.
ldap.filter.hex: <filter>
The parameter is defined as follows:

**filter** search filter in hex UTF-8 for an OpenLDAP server. The default filter is (memberuid=%s) and assumes a rfc2307bis schema.

ldap.filter.ad: <filter>
The parameter is defined as follows:

**filter** search filter for an Active Directory server. The default filter is (samaccountname=%s).

ldap.filter.ad.hex: <filter>
The parameter is defined as follows:

**filter** search filter in hex UTF-8 for an Active Directory server. The default filter is (samaccountname=%s).

ldap.attribute: <attribute>
The parameter is defined as follows:

**attribute** OpenLDAP server to be matched against the **ldap.group** values to identify the users group membership. The default attribute is **cn**.

ldap.attribute.hex: <attribute>
The parameter is defined as follows:

**attribute** hex UTF-8 for an OpenLDAP server to be matched against the **ldap.group** values to identify the users group membership. The default attribute is **cn**.

ldap.attribute.ad: <attribute>
The parameter is defined as follows:

**attribute** Active Directory server to be matched against the **ldap.group** values to identify the users group membership. The module will search recursively through groups. The default attribute is **memberof**.

ldap.attribute.ad.hex: <attribute>
The parameter is defined as follows:

**attribute** hex UTF-8 for an Active Directory server to be matched against the **ldap.group** values to identify the users group membership. The module will search recursively through groups. The default attribute is **memberof**.

ldap.keytab: <keytab>
The parameter is defined as follows:
keytab file containing the Kerberos principals to authenticate the module to the LDAP servers. The default keytab is `FILE:/etc/sockd.keytab` or the value of `gssapi.keytab` if it is set.

ldap.auto.off: no|yes
Disable automatic determination of LDAP server. The default value is no.

3 Semantics

The LDAP module statements described above are generally only used as a part of Dante socks-rules.

4 Special notes

The Dante server uses a set of timeout values defined in the Dante source code.

The following values are defined in the file `include/sockd.h`:

- **SOCKDLDAPDEADTIME** the time a dead LDAP server should not be retried.
- **SOCKDLDAPSEARCHTIME** the maximal time an LDAP search can take.
- **SOCKDLDAPTIMEOUT** the maximal network connect time for an LDAP connection.

The following value is defined in `include/config.h`:

- **SOCKDLDAPCACHE_TIMEOUT** is the maximal time a LDAP group result is cached.

Should it be necessary to change these values, the above values will need to be redefined and the Dante server recompiled.

4.1 SIGHUP

Sending the server a SIGHUP signal forces a reload of the configuration file. It should be noted that this does not affect current sessions or limits placed on them.

A reload of the configuration file only affects sessions created after the reload. It will not affect any of the existing sessions.

This means that changing e.g., a pass statement to a block statement, does not terminate the session of any existing client. Likewise, changing the limits set in a rule does not change the values for any existing session.

After a reload of the configuration file, old sessions will continue to operate in a separate space, using the old configuration, while new sessions will use the new configuration.
5 Examples

This section shows several examples of how it is possible to use the LDAP module. The first examples require GSSAPI user authentication. For Windows clients, the OpenText (formerly Hummingbird) client can be used (see http://connectivity.opentext.com/products/socks-client.aspx).

5.1 Limiting access to web/http

The rules below shows how one can limit access to web sites from clients on the 10.0.0.0/8 network to members of the SOCKSALLOW LDAP group.

```plaintext
# client-rule, no ldap statements.
client pass {
    from: 10.0.0.0/8 to: 0.0.0.0/0
    gssapi.enctype: clear integrity confidentiality
}

# socks-rule, including a ldap statement.
socks pass {
    from: 10.0.0.0/8 to: 0.0.0.0/0 port = http
    command: connect
    ldap.group: SOCKS_ALLOW
}
```

For an OpenLDAP server with a rfc2307bis schema or an Active Directory server, User1 and User2 will be allowed, whereas User3 would be refused access (see Appendix for additional details).

5.2 Limiting access to SSL VPNs

The next rules, if placed before other general rules, shows how one can limit access for temporary staff on the 10.0.0.0/8 network to only a specific work related site, while permanent staff from the same 10.0.0.0/8 network can access any site.

```plaintext
client pass {
    from: 10.0.0.0/8 to: 0.0.0.0/0
    gssapi.enctype: clear integrity confidentiality
}

socks pass {
    from: 10.0.0.0/8 to: sslvpn.example.com port = 443
    command: connect
    ldap.group: Temporary
    ldap.filter: (uid=%s)
    ldap.attribute: employeeType
}

socks pass {
    from: 10.0.0.0/8 to: 0.0.0.0/0
    command: connect
    ldap.group: Permanent
}
ldap.filter: (uid=%s)
ldap.attribute: employeeType
)

Assuming the OpenLDAP configuration in the Appendix example is used, the temporary user User3 is only allowed to connect to sslvpn.example.com on port 443 whereas the permanent users User1 and User2 can connect to any secure website via https.

5.3 Limiting ftp to company employees only

The next rules shows how one can limit access to ftp sites to company employees on the 10.0.0.0/8 network.

Note that this example will only work for active ftp, because there are no fixed port numbers for passive ftp.

client pass {
    from: 10.0.0.0/8 to: 0.0.0.0/0
    gssapi.enctype: clear integrity confidentiality
}

socks pass {
    from: 0.0.0.0/0 port = ftp-data to: 10.0.0.0/8
    command: bindreply
    ldap.group: MyCompany
    ldap.keeprealm: yes
    ldap.filter.ad: (userprincipalname=%s)
    ldap.attribute.ad: company
}

socks pass {
    from: 10.0.0.0/8 to: 0.0.0.0/0 port = ftp
    command: connect
    ldap.group: MyCompany
    ldap.keeprealm: yes
    ldap.filter.ad: (userprincipalname=%s)
    ldap.attribute.ad: company
}

Assuming the Active Directory example in the Appendix is used, User3 is only allowed to connect to ftp data whereas the users User1 and User2 are not allowed.
5.4 Using an LDAP URL to determine LDAP group membership

The next rule shows how one can limit access to ftp sites for company employees on the 10.0.0.0/8 network without requiring GSSAPI authentication. An LDAP URL with a directly specified username (here user) and password (here pass) is used for authentication.

client pass {
    from: 10.0.0.0/8 to: 0.0.0.0/0
}

socks pass {
    from: 0.0.0.0/0 port = ftp-data to: 10.0.0.0/8
    command: bindreply
    ldap.group: SOCKS_ALLOW
    ldap.auto.off: yes
    ldap.url: ldap://user:pass@ldap1.example.com:389/OU=SALES,DC=MYCOMPANY,DC=COM
    ldap.url: ldap://user:pass@ldap2.example.com:389/OU=SALES,DC=MYCOMPANY,DC=COM
}

socks pass {
    from: 10.0.0.0/8 to: 0.0.0.0/0 port = ftp
    command: connect
    ldap.group: SOCKS_ALLOW
    ldap.auto.off: yes
    ldap.url: ldap://user:pass@ldap1.example.com:389/OU=SALES,DC=MYCOMPANY,DC=COM
    ldap.url: ldap://user:pass@ldap2.example.com:389/OU=SALES,DC=MYCOMPANY,DC=COM
}
6 Appendix

6.1 OpenLDAP
6.2 Active Directory