LDAP Authentication

- Configuration Examples
- Configuration Options
  - bindPassword
- Bind Mode vs Search Mode

!!! info "Traefik Hub Feature" This middleware is available exclusively in Traefik Hub. Learn more about Traefik Hub's advanced features.

The LDAP Authentication middleware secures your applications by delegating the authentication to an external LDAP server.

The LDAP middleware will look for user credentials in the `Authorization` header of each request. Credentials must be encoded with the following format: `base64(username:password)`.

Configuration Examples

```
apiVersion: traefik.io/v1alpha1
kind: Middleware
metadata:
  name: test-ldap-auth
  namespace: apps
spec:
  plugin:
    ldap:
      url: ldap://ldap.example.org:636
      baseDN: dc=example,dc=org
```

```
apiVersion: traefik.io/v1alpha1
kind: Middleware
metadata:
```

```
  name: test-ldap-auth
  namespace: apps
spec:
  plugin:
    ldap:
      url: ldap://ldap.example.org:636
      baseDN: dc=example,dc=org
      bindDN: cn=binding_user,dc=example,dc=org
      bindPassword: "urn:k8s:secret:my-secret:bindpassword"
```

```
apiVersion: traefik.io/v1alpha1
kind: Middleware
metadata:
  name: test-ldap-auth
  namespace: apps
spec:
  plugin:
    ldap:
      url: ldap://ldap.example.org:636
      baseDN: dc=example,dc=org
      searchFilter: (&(objectClass=inetOrgPerson)(gidNumber=500)(uid=%s))
      forwardUsername: true
      forwardUsernameHeader: Custom-Username-Header-Name
      wwwAuthenticateHeader: true
      wwwAuthenticateHeaderRealm: traefikee
```

Configuration Options

Field	Description	Default	Required
`url`	LDAP server URL. Either the `ldaps` or `ldap` protocol and end with a port (ex: `ldaps://ldap.example.org:636`).	""	Yes
`startTLS`	Enable `StartTLS` request when initializing the connection with the LDAP server.	false	No

Field	Description	Default	Required
`certificateAuthority`	PEM-encoded certificate to use to establish a connection with the LDAP server if the connection uses TLS but that the certificate was signed by a custom Certificate Authority.	""	No
`insecureSkipVerify`	Allow proceeding and operating even for server TLS connections otherwise considered insecure.	false	No
`bindDN`	Domain name to bind to in order to authenticate to the LDAP server when running on search mode. Leaving this empty with search mode means binds are anonymous, which is rarely expected behavior. Not used when running in bind mode.	""	No
`bindPassword`	Password for the `bindDN` used in search mode to authenticate with the LDAP server. More information here	""	No
`connPool`	Pool of connections to the LDAP server (to minimize the impact on the performance).	None	No
`connPool.size`	Number of connections managed by the pool can be customized with the `size` property.	10	No
`connPool.burst`	Ephemeral connections that are opened when the pool is already full. Once the number of connection exceeds `size` + `burst`, a `Too Many Connections` error is returned.	5	No

Field	Description	Default	Required
`connPool.ttl`	Pooled connections are still meant to be short-lived, so they are closed after roughly one minute by default. This behavior can be modified with the `ttl` property.	60s	No
`baseDN`	Base domain name that should be used for bind and search queries.	""	Yes
`attribute`	The attribute used to bind a user. Bind queries use this pattern: `<attr>=<username>,<baseDN>`, where the username is extracted from the request header.	cn	Yes
`forwardUsername`	Forward the username in a specific header, defined using the `forwardUsernameHeader` option.	""	No
`forwardUsernameHeader`	Name of the header to put the username in when forwarding it. This is not used if the `forwardUsername` option is set to `false`.	Username	Yes
`forwardAuthorization`	Enable to forward the authorization header from the request after it has been approved by the middleware.	false	Yes
`searchFilter`	If not empty, the middleware will run in search mode, filtering search results with the given query. Filter queries can use the `%s` placeholder that is replaced by the username provided in the `Authorization` header of the request (for example: `(&(objectClass=inetOrgPerson)(gidNumber=500)(uid=%s))`).	""	No

Field	Description	Default	Required
`wwwAuthenticateHeader`	Allow setting a `WWW-Authenticate` header in the `401 Unauthorized` response. See the WWW-Authenticate header documentation for more information. The `realm` directive of the `WWW-Authenticate` header can be customized with the `wwwAuthenticateHeaderRealm` option.	false	No
`wwwAuthenticateHeaderRealm`	Realm name to set in the `WWW-Authenticate` header. This option is ineffective unless the `wwwAuthenticateHeader` option is set to `true`.	""	No

bindPassword

When setting the `bindPassword`, you can reference a Kubernetes secret from the same namespace as the Middleware using the following URN format:

```
urn:k8s:secret:[secretName]:[key]
```

Bind Mode vs Search Mode

If no filter is specified in its configuration, the middleware runs in the default bind mode, meaning it tries to make a bind request to the LDAP server with the credentials provided in the request headers. If the bind succeeds, the middleware forwards the request, otherwise it returns a `401 Unauthorized` status code.

If a filter query is specified in the middleware configuration, and the Authentication Source referenced has a `bindDN` and a `bindPassword`, then the middleware runs in search mode. In this mode, a search query with the given filter is issued to the LDAP server before trying to bind. If result of this search returns only 1 record, it tries to issue a bind request with this record, otherwise it aborts a `401 Unauthorized` status code.

{% include-markdown "includes/traefik-for-business-applications.md" %}

□□□ #2
□ □□□ □□□ 2026-10-31 14:03:24 UTC
□ □□□ □□□ 2026-10-31 14:03:25 UTC