Apache Authentication, Authorization, and Access Control Concepts

Version 2.2

Overview

The Apache web server software has a respectable history relative to providing and supporting authentication, authorization, and access control (AAA) services since v1.3. The most recent versions now arguably have very mature facilities for providing these services and can leverage most LDAP compliant directories to further enhance AAA security and flexibility.

First, a very brief primer of the referenced services:

Service	Description
Authentication	Being able to prove your identity using set criteria or credentials
Authorization	Being able to access resources based on your identity
Access Control	Being able to define how resources can be accessed

As with most services made available through Apache AAA services are provided by loading and configuring the appropriate modules at Apache server runtime. With previous versions of Apache it was not unusual to load one or more modules to provide each service required. For example the "mod_auth" module, "mod_access" module, and their respective configuration file directives provide some basic AAA services. However additional modules were required to integrate LDAP connectivity with those services.

The most recent Apache implementations do simplify this model by integrating authentication, authorization, and sometimes even LDAP connectivity services within the same module. Often these new module types support and often extend legacy AAA module capabilities. Subsequently allowing them to interoperate to provide very flexible and hardened security for web server AAA services. Such is the case with the "mod_authnz_ldap" module available with the v2.1 Apache release. Those details will be expanded on later in this section.

First a very brief primer of the referenced modules for Novell OES Linux servers:

Available in OES 1 SP2

Module	Apache version	Description
mod_auth	up to v2.0.x	Provides basic authentication using file stores
mod_access	up to v2.0.x	Provides access control using host addresses
mod_auth_ldap	v2.0.41 or better	Provides basic authentication using LDAP stores
mod_ldap	v2.0.41 or better	Provides LDAP connection configuration

Available in OES 2

Module	Apache version	Description
mod_auth_basic	v2.1 or better	Provides basic authentication using several identity store types
mod_authz_host	v2.1 or better	Provides access control using host addresses
mod_authz_user	v2.1 or better	Provides user authorization services
mod_ldap	v2.0.41 or better	Provides LDAP connection configuration
mod_authnz_ldap	v2.1 or better	Provides authentication and authorization services using LDAP directories. Mod_authnz_ldap can also extend other authentication modules to support LDAP directories and provide authorization services to them

First a legacy example using Apache v2.0.59 on OES 1 SP2 Linux.

Modules loaded:
mod_auth_ldap
mod_ldap

Using directives in a site specific .conf file placed in the **/etc/apache2/conf.d** directory to secure some web content:

Line 1: <Directory "/srv/www/htdocs/securedsite1">
Line 2: AuthType Basic
Line 3: AuthName "Secured web site"
Line 4: Require valid-user
Line 5: AuthLDAPEnabled On
Line 6: LDAPTrustedCAType DER_FILE
Line 7: LDAPTrustedCA /etc/apache2/certs/ldap_trusted_root.der
Line 8: AuthLDAPAuthoritative On
Line 9: AuthLDAPURL ldaps://www.mycompany.com/o=corp?uid?sub?
Line 10: Satisfy All
Line 11:</Directory>

This example explained:

Line 1: Defines the directory accessible to Apache used for the site content
Line 2: Enables the "Basic" authentication method (UserID/Password) for directories
Line 3: Configures the authentication realm (used to optimize authentication)
Line 4: Requires a valid user successfully authenticate using the defined store
Line 5: Enables LDAP directories as user authentication stores
Line 6: Specifies the type of certificate to use for secure LDAP connectivity
Line 7: Specifies the file system location of public key certificate of the target LDAP server
Line 8: Disables Apache's ability to use other authentication facilities for this site
Line 9: Specifies the LDAP connection type, LDAP server, target attribute, and search scope used to locate LDAP users. Specifically it searches at and below the "corp" branch for "uid" attributes that match the provided user name (multiple LDAP servers and scopes can be specified)
Line 10: Specifies that all authentication and host address criteria must be met to authorize the user
Line 11: Ends the site configuration declarations

Secure LDAP connectivity is always recommended when authentication services are involved. Optionally additional directives could be used to enable TLS connectivity for LDAP searches, binds, and compares over ports 389 or 636. If TLS is not supported or available, SSL could be used over port 636.

This example authenticates a user against a LDAP directory with a UserID and password pair which has to exist in the directory. Authorization occurs if the previous bind and authentication operation using that credential pair was successful. Authorization can also be delegated to other modules if configured. This is a very simple authentication and authorization model that doesn't offer much in the way of flexibility and scalability and would certainly limit its use to the most basic web application models.

Next, an example using Apache v2.2 on OES 2 Linux.

Modules loaded:
mod_auth_basic
mod_authz_user
mod_authnz_ldap
mod_ldap

Using directives in a site specific .conf file placed in the **/etc/apache2/conf.d** directory to secure some web content:

Line 1: <Directory "/srv/www/htdocs/securedsite2">
Line 2: AuthType Basic
Line 3: AuthName "Secured web site 2"
Line 4: AuthBasicProvider ldap
Line 5: AuthzLDAPAuthoritative Off
Line 6: Require valid-user
Line 7: AuthLDAPUrl ldap://192.168.2.120/o=corp?uid?sub?
Line 8: Satisfy All
Line 9: </Directory>

Additional directives are configured in the **/etc/apache2/httpd.conf** file to configure the secure LDAP connectivity for the site authentication. This particular site is configured to use TLS over port 389.

CA Certificate for Apache TLS/SSL connectivity over LDAP
Line 1: LDAPTrustedGlobalCert CA_DER /etc/apache2/certs/ ldap_trusted_root.der
Line 2: LDAPTrustedMode TLS

This example explained:

Line 1: Defines the directory accessible to Apache used for the site content
Line 2: Enables the “Basic” authentication method (UserID/Password) for directories
Line 3: Configures the authentication realm (used to optimize authentication)
Line 4: Enables LDAP directories as user authentication store (mod_authnz_ldap still handles authorization)
Line 5: Enables Apache’s ability to use authentication facilities provided by other modules for this site (mod_auth_basic in this case)
Line 6: Requires a valid user successfully authenticate using the defined store
Line 7: Specifies the LDAP connection port, ldap server, target attribute, and search scope used to locate LDAP users. Specifically it searches at and below the “corp” branch for “uid” attributes that match the provided user name (multiple LDAP servers and scopes can be specified)
Line 8: Specifies that all authentication and host address criteria must be met to authorize the user
Line 9: Ends the site configuration declarations

CA Certificate for Apache TLS/SSL connectivity over LDAP
Line 1: Specifies the file system location and the type of certificate to use for secure LDAP connectivity
Line 2: Configures the LDAP connections initiated by Apache to use TLS

This example authenticates a user against a LDAP directory with a UserID and password pair, which has to exist in the directory. Authorization occurs if the previous bind and authentication operation using that credential pair was successful.

This may seem the same as the previous example and effectively it is. UserID and password pairs are very easily used as both authentication and authorization credentials. In this case "mod_authnz_ldap" is still used and simply provides the authorization services to the "mod_auth_basic" module if the previous bind and authentication operation it performed was successful. This module configuration does have the ability to extend the UserID and password pair authentication and authorization model with enhanced functionality. However the real benefits of utilizing the mod_authnz_ldap model are explained in the next example.

Another example using Apache v2.2 on OES 2 Linux.

Modules loaded:
mod_auth_basic
mod_authnz_ldap
mod_ldap

Using directives in a site specific .conf file placed in the **/etc/apache2/conf.d** directory to secure some web content:

Line 1: <Directory "/srv/www/htdocs/securedsite3">
Line 2: AuthType Basic
Line 3: AuthName "Secured web site 3"
Line 4: AuthBasicProvider ldap
Line 5: AuthzLDAPAuthoritative On
Line 6: AuthLDAPGroupAttribute member
Line 7: Require ldap-attribute employeeStatus=Active
Line 8: Require ldap-group cn=LDAP_GROUP_G,ou=CONTAINER,o=ORGANIZATION
Line 9: AuthLDAPUrl ldap://192.168.2.120/o=dv?uid?sub?
Line 10: Satisfy All
Line 11:</Directory>

Additional directives are configured in the **/etc/apache2/httpd.conf** file to configure the secure LDAP connectivity for the site authentication. This particular site is configured to use TLS over port 389.

Line 1: LDAPTrustedGlobalCert CA_DER /etc/apache2/certs/ ldap_trusted_root.der
Line 2: LDAPTrustedMode TLS

This example explained:

Line 1: Defines the directory accessible to Apache used for the site content
Line 2: Enables the “Basic” authentication method (UserID/Password) for directories
Line 3: Configures the authentication realm (used to optimize authentication)
Line 4: Enables LDAP directories as user authentication store (mod_authnz_ldap still handles authorization)
Line 5: Disables Apache’s ability to use authentication facilities provided by other modules for this site (mod_authnz_ldap in this case)
Line 6: Specifies the attribute used for LDAP compare operations for group memberships to authorize users
Line 7: Specifies any valid attribute used for LDAP compare operations to authorize users
Line 8: Specifies the LDAP group(s) whose membership is used to authorize users
Line 9: Specifies the LDAP connection port, LDAP server, target attribute, and search scope used to locate LDAP users. Specifically it searches at and below the “corp” branch for “uid” attributes that match the provided user name (multiple LDAP servers and scopes can be specified)
Line 10: Specifies that all authentication and host address criteria must be met to authorize the

user

Line 11: Ends the site configuration declarations

Line 1: Specifies the file system location and the type of certificate to use for secure LDAP connectivity

Line 2: Configures the LDAP connections initiated by Apache to use TLS

This example authenticates a user against a LDAP directory with a UserID and password pair, which has to exist in the directory. Authorization occurs if the user is a member of the specified LDAP group and has an LDAP attribute called “employeeStatus” with a value of “Active”.

Its' not hard to see how the last AAA example is the most flexible, secure, and has the potential to be used in a role based provisioning model. Authorizing users by UserID and password credentials alone is often insufficient for most information systems today. Even the most basic business systems deployed and managed today utilize, either directly or indirectly, the inherent role based capabilities present in most open source and commercially available directories. This mature AAA scalability and flexibility makes web platforms that support these features very desirable for the small business, enterprise, and federations to deploy their applications on.