ENGR3821 Network Engineering

NETENG ASSIGNMENT 2

June 24, 2019

Table of Contents

Disclaimer

This report and its contents are permitted to be redistributed on behalf of the author for the express purpose of peer review and grading only.

Introduction

This report will step through the process required to install, configure, and populate an OpenLDAP server and have an instance of Wiki.js authenticate against it. The authentication mechanism is further hardened with TLS to encrypt credentials as they are queried from the directory.

LDAP Server Setup

For the purpose of this report we will be using OpenLDAP out of familiarity, however there are several free and open-source alternatives out there which will work just fine, such as:

- **ApacheDS** (Apache Licence 2.0)
  A cross-platform LDAP and Kerberos server bundle written in Java and certified as compliant with the LDAPv3 protocol[1].
- **389 Directory Server** (GPLv3.0)
  An enterprise-grade LDAP server designed to be performant and resilient even under heavy load[2].
- **GLAuth** (MIT)
  An authentication-oriented LDAP server written in Go with the express goal of backing authentication mechanisms[3].

Installation and Configuration

1. Install slapd and ldap-utils through apt with: `sudo apt install slapd ldap-utils`
   a. Take note of the administrator password you enter for slapd, you will need it later
2. Enable and start slapd with: `sudo systemctl enable slapd && sudo systemctl start slapd`
   slapd is now installed, but not yet populated
3. Prepare an LDIF file of the users to insert into the directory (a sample file is available in Appendix A)
   NOTE: When setting user passwords, make use of `slappasswd` as it will hash passwords in an LDIF-compatible format[4], increasing security over storing passwords in plaintext
4. Apply the LDIF to the directory with: `ldapadd -cxWD cn=admin,dc=nodomain -f <path_to_ldif>`, replacing `<path_to_ldif>` with the path to your LDIF file
   a. Enter the administrator password from step 1a and press enter
   slapd is now populated

Wiki.js Configuration

5. Log in to the wiki as an administrator
6. Open the administration panel by clicking on the gear icon in the top-right of the web page
7. On the left-hand sidebar, scroll down to ‘Authentication’ and click to open it
8. In the ‘Strategies’ list, scroll until you see ‘LDAP/Active Directory’ and click the text to open it

9. Now there should be a variety of fields which look vaguely similar to those in the LDIF from earlier. From top-to-bottom, fill them out like so:
    a. Set the LDAP URL to ‘ldap://<hostname>’, where <hostname> is the hostname of the LDAP server
    b. Set the admin bind to ‘cn=admin,dc=nodomain’
    c. Set the admin password to the password from step 1a
    d. Set the search base to ‘dc=nodomain’
    e. Set the filter to ‘(mail={{username}})’
    f. Leave the TLS settings turned off for now, we will come back for them later
    g. Set the unique ID field to ‘mail’
    h. Set the email field to ‘mail’
    i. Set the display name field to ‘cn’
    j. Set the avatar picture field to ‘jpegPhoto’
    Now the LDAP connection is configured, but users cannot log in just yet
    k. Enable ‘Allow self-registration’
    NOTE: Disable this once all of the users you wish to allow access have logged in for the first time if you do not wish to allow new users access in future
    l. *(Optional)* If you wish to only allow users with a particular email domain, list them in the ‘Limit to specific email domains’ box
    m. Click into the ‘Assign to group’ box and select which groups LDAP users should be added to when they first log in
10. In the ‘Strategies’ list, check the checkbox next to ‘LDAP/Active Directory’ to enable it as a login provider
11. Scroll to the top of the page and click ‘Apply’ to save your changes

LDAP is now configured as a login provider for Wiki.js!

12. Sign out of the administrator account
13. When at the login page, click the pyramid-shaped icon labelled ‘LDAP/Active Directory’ to use LDAP as the login provider
14. Enter a user’s email and password to test if LDAP is working successfully

TLS Configuration

At the moment, all LDAP data between slapd and Wiki.js is transferred in plain text, allowing anyone who can sniff the packets in transit to see that data too. A screenshot of a packet captured in transit is shown in Figure 1. Configuring TLS between Wiki.js and OpenLDAP will secure data while in transit and is highly recommended.

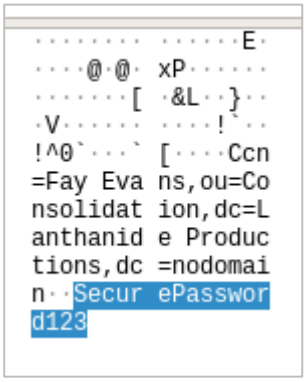

Fig. 1: Screenshot of a captured LDAP packet with a user's password highlighted

Generating and Copying Certificates

To enable TLS between Wiki.js and OpenLDAP you must first generate a set of X.509 certificates. To do this we will be using easy-rsa, a utility provided by OpenVPN to simplify certificate management.

1. Get a recent copy of easy-rsa with: `wget https://github.com/OpenVPN/easy-rsa/releases/download/v3.0.6/EasyRSA-unix-v3.0.6.tgz`
2. Extract the archive with: `tar -xf EasyRSA-unix-v3.0.6.tgz`
3. `cd` into the EasyRSA-unix-... directory
4. Run `./easyrsa init-pki` to set up the work directory
5. Run `./easyrsa build-ca`
    a. When prompted for a CA passphrase, make sure you use a strong password, as this key will be used to sign new certificates
    b. Enter a name for your certificate authority (this can be anything you wish)
6. Run `./easyrsa gen-req <hostname> nopass`, where `<hostname>` is the hostname of the LDAP server
    a. Enter the hostname of the LDAP server
    NOTE: The hostname must be *exactly the same* as the LDAP server or else clients will have issues connecting
7. Run `./easyrsa sign-req server <hostname>`, where `<hostname>` is the hostname of the LDAP server
    a. Confirm that the details displayed match the details entered in step 6 and type 'yes'
    b. Enter the CA passphrase

    Certificates have now been generated and signed, but are not yet installed
8. Make a certificate directory for the LDAP server with: `sudo mkdir /etc/ldap/certs/`

9. Copy the server certificate, key, and CA certificate with: `sudo cp pki/issued/<hostname>.crt pki/private/<hostname>.key pki/ca.crt /etc/ldap/certs/`
10. Set the permissions and owner for both the certs directory and its contents with: `sudo chown -R openldap:openldap /etc/ldap/certs/ && sudo chmod 600 /etc/ldap/certs/* && sudo chmod 700 /etc/ldap/certs/`
11. Copy the CA certificate to Wiki.js with: `cp pki/ca.crt <wikijs>/`, where `<wikijs>` is the path to your Wiki.js installation

    Certificates are now installed, but are not yet used by either slapd or Wiki.js

Configuring slapd

12. Create a text file called ‘tls.ldif’ containing the following:

```
dn: cn=config
changetype: modify
add: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ldap/certs/ca.crt
-
add: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ldap/certs/<hostname>.key
-
add: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ldap/certs/<hostname>.crt
```

    Remember to replace `<hostname>` with the hostname of the LDAP server!
13. Install the LDIF file with: `sudo ldapmodify -Y EXTERNAL -H ldapi:/// -f ./tls.ldif`
14. Open `/etc/default/slapd` with a text editor
    a. Set ‘SLAPD_SERVICES’ to ‘ldapi:/// ldaps:///’
15. Restart slapd with: `sudo systemctl restart slapd`

    slapd is now configured to use TLS

Configuring Wiki.js

16. Log in to the wiki as an administrator
17. Open the administration panel by clicking on the gear icon in the top-right of the web page
18. On the left-hand sidebar, scroll down to ‘Authentication’ and click to open it
19. In the ‘Strategies’ list, scroll until you see ‘LDAP/Active Directory’ and click the text to open it

a. Set ‘LDAP URL’ to use ‘ldaps://’ instead of ‘ldap://’
   NOTE: Make sure that the LDAP URL matches the hostname that was embedded in the LDAP server’s certificate. If it does not match, authentication will fail!
b. Scroll until you see ‘Use TLS’, toggle it on
c. Enter the path to the ‘ca.crt’ we copied in step 11

20. Scroll to the top of the page and click ‘Apply’ to save your changes
21. Log out of the administrator account

Wiki.js is now configured to use TLS with LDAP!

Where Wiki.js would normally be communicating with slapd in plain text, as shown in Figure 1, it is now communicating exclusively through TLS layered over LDAP, as shown in Figure 2 below.

No.	Time	Source	Destination	Protocol	Length	Info
4	0.000076828	127.0.0.1	127.0.0.1	HTTP	987	POST /graphql HTTP/1.1 (application/json)
5	0.000080657	127.0.0.1	127.0.0.1	TCP	66	3000 → 60266 [ACK] Seq=1 Ack=922 Win=45568 Len=0 TSval=2552926384 TSecr=2552926384
6	0.003256592	127.0.0.1	127.0.1.1	TCP	74	33304 → 636 [SYN] Seq=0 Win=43690 Len=0 MSS=65495 SACK_PERM=1 TSval=2324705374 TSecr=0 WS=1…
7	0.003265401	127.0.1.1	127.0.0.1	TCP	74	636 → 33304 [SYN, ACK] Seq=0 Ack=1 Win=43690 Len=0 MSS=65495 SACK_PERM=1 TSval=3927265488 T…
8	0.003271354	127.0.0.1	127.0.1.1	TCP	66	33304 → 636 [ACK] Seq=1 Ack=1 Win=43776 Len=0 TSval=2324705374 TSecr=3927265488
9	0.003449996	127.0.0.1	127.0.1.1	TLSv1.2	316	Client Hello
10	0.003455223	127.0.1.1	127.0.0.1	TCP	66	636 → 33304 [ACK] Seq=1 Ack=251 Win=44800 Len=0 TSval=3927265488 TSecr=2324705374
11	0.004922670	127.0.1.1	127.0.0.1	TLSv1.2	162	Server Hello
12	0.004927775	127.0.1.1	127.0.0.1	TLSv1.2	1763	Certificate
13	0.004930478	127.0.1.1	127.0.0.1	TLSv1.2	404	Server Key Exchange
14	0.004932586	127.0.1.1	127.0.0.1	TLSv1.2	75	Server Hello Done
15	0.008250091	127.0.0.1	127.0.1.1	TCP	66	33304 → 636 [ACK] Seq=251 Ack=97 Win=43776 Len=0 TSval=2324705379 TSecr=3927265490
16	0.008257719	127.0.0.1	127.0.1.1	TCP	66	33304 → 636 [ACK] Seq=251 Ack=1794 Win=174720 Len=0 TSval=2324705379 TSecr=3927265490
17	0.008259911	127.0.0.1	127.0.1.1	TCP	66	33304 → 636 [ACK] Seq=251 Ack=2132 Win=178176 Len=0 TSval=2324705379 TSecr=3927265490
18	0.008261930	127.0.0.1	127.0.1.1	TCP	66	33304 → 636 [ACK] Seq=251 Ack=2141 Win=178176 Len=0 TSval=2324705379 TSecr=3927265490
19	0.009177140	127.0.0.1	127.0.1.1	TLSv1.2	192	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
20	0.009245255	127.0.0.1	127.0.1.1	TCP	74	33306 → 636 [SYN] Seq=0 Win=43690 Len=0 MSS=65495 SACK_PERM=1 TSval=2324705380 TSecr=0 WS=1…
21	0.009252413	127.0.1.1	127.0.0.1	TCP	74	636 → 33306 [SYN, ACK] Seq=0 Ack=1 Win=43690 Len=0 MSS=65495 SACK_PERM=1 TSval=3927265494 T…
22	0.009258078	127.0.0.1	127.0.1.1	TCP	66	33306 → 636 [ACK] Seq=1 Ack=1 Win=43776 Len=0 TSval=2324705380 TSecr=3927265494
23	0.009408187	127.0.0.1	127.0.1.1	TLSv1.2	316	Client Hello
24	0.009411455	127.0.1.1	127.0.0.1	TCP	66	636 → 33306 [ACK] Seq=1 Ack=251 Win=44800 Len=0 TSval=3927265494 TSecr=2324705380

Figure 2: Packets captured while Wiki.js was opening a TLS session with slapd

Future Work

Although the connection between the wiki and LDAP server is now secure, the connection between clients and the wiki is still unsecured. Acquiring a certificate through a service such as LetsEncrypt and configuring either the wiki or a forward proxy capable of SSL termination to serve content only over HTTPS would prevent eavesdropping on the client-wiki path.

Appendices

Appendix A: Sample LDIF file

Sample LDIF can be found here: https://pastebin.com/H0RL3wVg
Each of the users are assigned the password 'SecurePassword123' hashed with SHA1.

Bibliography

[1] Apache Software Foundation, "Welcome to ApacheDS," *Apache Directory*, 2018. [Online]. Available: http://directory.apache.org/apacheds/. [Accessed: 24-Jun-2019].

[2] Red Hat, Inc., "389 Directory Server," *389 Directory Server*, 2019. [Online]. Available: http://www.port389.org/. [Accessed: 24-Jun-2019].

[3] B. Yanke and kost, "GLAuth: LDAP authentication server for developers," *GitHub*, 26-Dec-2018. [Online]. Available: https://github.com/glauth/glauth. [Accessed: 24-Jun-2019].

[4] OpenLDAP Foundation, *slappasswd(8C) Linux User's Manual*. 2018.