

openLI

REST API Authentication

OpenLI Training: Chapter Twenty

Shane Alcock

University of Waikato

New Zealand

shane.alcock@waikato.ac.nz

Authentication

- Default REST API accepts requests from anyone
 - Not ideal for LI infrastructure!

- Methods to restrict access
 - Network topology
 - Firewalling
 - Listen only on localhost

Authentication

- OpenLI supports HTTP authentication methods
 - API keys
 - Digest

- Strongly recommended for your production deployment!

Authentication Database

- SQLite3 database
 - Stores usernames, digest hashes and API keys
 - Located on the provisioner host
 - Created when you install the provisioner package
 - Encrypted using sqlcipher

Authentication Database

- Database file
 - `/var/lib/openli/provauth.db`
- Passphrase for the database
 - `/etc/openli/provauthdb.phrase`
 - Record passphrase somewhere safe
 - Delete this file afterwards asap

Authentication Configuration

- Edit the provisioner configuration file

```
/home/openli-prov# vim /etc/openli/provisioner-config.yaml
```

Authentication Configuration

- Enable the following configuration options
 - restauthdb: the path to the database file
 - restauthkey: the passphrase for the database

```
restauthdb: /var/lib/openli/provauth.db  
restauthkey: <passphrase>          <-- insert your passphrase here
```

Authentication Configuration

- Restart the provisioner

```
/home/openli-prov# stop_provisioner.sh
```

```
/home/openli-prov# service openli-provisioner start
```


Authentication Configuration

- Check the provisioner logs

```
/home/openli-prov# less /var/log/openli/provisioner.log
```

```
...
```

```
openliprovisioner[1331]: OpenLI provisioner: Authentication enabled for the REST API (using  
DB /var/lib/openli/provauth.db)
```

```
...
```

Authentication Configuration

- Try to use the REST API without authenticating

```
/home/openli-prov# curl -k -X GET https://172.19.0.3:8080/ipintercept
```

```
<html><body>Authentication failed</body></html>
```

Adding a User

- OpenLI provides a script for adding authenticated users
 - Installed into /usr/sbin

```
/home/openli-prov# /usr/sbin/openli-prov-adduser.sh <db passphrase> <username> <password>  
/var/lib/openli/provauth.db
```

- <db passphrase> is the passphrase for the encrypted database
- <username> is the new user's username
- <password> is the password for the new user
 - This is NOT stored in the database -- just a derived hash

Adding a User

- A random API key will be generated when the user is added

```
/home/openli-prov# /usr/sbin/openli-prov-adduser.sh XXXXXXXXXXXX salcock testpwd  
/var/lib/openli/provauth.db
```

```
Successfully added new user salcock -- API key is HEqpcyWgGVvWOSrA3fkDmKvm3JQVisLu
```

Authenticating with an API Key

- Add an `X-API-KEY` HTTP header to your requests
 - Set the value to your assigned API key

Authenticating with an API Key

- Example using curl:

```
/home/openli-prov# curl -k -X GET -H "X-API-KEY: HEqpcyWgGVvWOSrA3fkDmKvm3JQVisLu"  
https://172.19.0.3:8080/ipintercept
```

```
[ { "liid": "STATIC002", "authcc": "NZ", "delivcc": "NZ", "agencyid": "mocklea",  
  "mediator": 1, "user": "salcock", "accesstype": "fiber", "radiusident": "any", "staticips":  
  [ { "iprange": "10.1.18.217\32", "sessionid": 101 } ] }, { "liid": "RADIUS003", "authcc":  
  "NZ", "delivcc": "NZ", "agencyid": "mocklea", "mediator": 1, "user": "b4CPidYn7u8Vesbo",  
  "accesstype": "xDSL", "radiusident": "user" }, { "liid": "NZP_20211010", "authcc": "NZ",  
  "delivcc": "NZ", "agencyid": "mocklea", "mediator": 1, "user": "2On5uRWxvQDeBBepKBu",  
  "accesstype": "wifi", "radiusident": "any", "vendmirrorid": 500 } ]
```

Authenticating with Digest Auth

- Using curl
 - Specify digest auth mode using --digest
 - Specify username and password using -u

Authenticating with Digest Auth

- Using curl:

```
/home/openli-prov# curl -k -X GET --digest -u salcock:testpwd  
https://172.19.0.3:8080/ipintercept
```

```
[ { "liid": "STATIC002", "authcc": "NZ", "delivcc": "NZ", "agencyid": "mocklea",  
  "mediator": 1, "user": "salcock", "accesstype": "fiber", "radiusident": "any", "staticips":  
  [ { "iprange": "10.1.18.217\$/32", "sessionid": 101 } ] }, { "liid": "RADIUS003", "authcc":  
  "NZ", "delivcc": "NZ", "agencyid": "mocklea", "mediator": 1, "user": "b4CPidYn7u8Vesbo",  
  "accesstype": "xDSL", "radiusident": "user" }, { "liid": "NZP_20211010", "authcc": "NZ",  
  "delivcc": "NZ", "agencyid": "mocklea", "mediator": 1, "user": "2On5uRWxvQDeBBepKBu",  
  "accesstype": "wifi", "radiusident": "any", "vendmirrorid": 500 } ]
```


Removing Users

- Command line to remove a user from the database
 - Run this on the provisioner

```
/home/openli-prov# sqlcipher /var/lib/openli/provauth.db "PRAGMA key=<passphrase>; DELETE  
FROM authcreds WHERE username=`<user to delete>`;"
```

Next Time

- Enabling RabbitMQ for buffering on the collector