LOCKSS Network Administration

Contents

LOCKSS network overview

Daemon operation overview

Configuration

At startup the daemon reads configuration information from several sources:

- Platform/host configuration, normally created on the local disk by installation scripts (e.g., ~lcap/local.txt). Contains host info (fqdn, IP address), machine configuration (available disks), local network configuration (mail & name servers)
- Network-wide configuration, created by the network admin, normally residing in one or more files on a central server. Contains daemon tuning parameters, title (AU) and titleset definitions, seed lists of peer identities, etc.
- A collection of files maintained by the daemon to store local configuration info entered into the web UI.

All config files and URLs are checked for changes periodically (org.lockss.config.reloadInterval (http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.config.reloadInterval)) and reloaded if changed. Some parameters require a daemon restart to take effect. This is noted in the parameter documentation (http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html) .

Config file format

Logging

To configure the daemon to log to a *nix syslog daemon on the same or another host, set

```
org.lockss.log.targets = org.lockss.util.SyslogTarget
org.lockss.log.syslog.host = loghost-fqdn
```

to direct to a nonstandard syslog port, set

```
org.lockss.log.syslog.port = port
```

Log messages are sent to the "user" syslog facility at levels: CRIT, WARNING, NOTICE, INFO and DEBUG. NOTICE is recommended for normal use.

Keystores

Cryptographic keys and certificates are used by various daemon components (currently the admin UI and LCAP communication, when using SSL. The keystore used to verify plugin signatures has not yet been integrated into this mechanism.) Keys and certificates are loaded from one ore more keystore files, configured by setting parameters below org.lockss.keyMgr.keystore.*id* , where *id* is a unique identifier.

org.lockss.keyMgr.keystore.*id*.name
(http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.keyMgr.keystore.%3cid%3e.name)
Name by which daemon component(s) refer to this keystore. Required.

org.lockss.keyMgr.keystore.*id*.file
(http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.keyMgr.keystore.%3cid%3e.file)
org.lockss.keyMgr.keystore.*id*.resource
(http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.keyMgr.keystore.%3cid%3e.resource)
org.lockss.keyMgr.keystore.*id*.url
(http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.keyMgr.keystore.%3cid%3e.file)
Keystore location, either a local filename, a Java resource name or a URL. Exactly one of these must be set.

org.lockss.keyMgr.keystore.*id*.password
(http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.keyMgr.keystore.%3cid%3e.password)
Keystore password. Optional. Password is not required to read keys from a keystore, but if supplied it must be correct.

org.lockss.keyMgr.keystore.*id*.type
(http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.keyMgr.keystore.%3cid%3e.type)
Need not be set if keystore is of a standard type (JKS, JCEKS)

org.lockss.keyMgr.keystore.*id*.provider
(http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.keyMgr.keystore.%3cid%3e.provider)
Need not be set if using standard keystore implementations (Sun, SunJCE)

org.lockss.keyMgr.keystore.*id*.keyPassword
(http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.keyMgr.keystore.%3cid%3e.keyPassword)
org.lockss.keyMgr.keystore.*id*.keyPasswordFile
(http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.keyMgr.keystore.%3cid%3e.keyPasswordFile)
Private key password. One of these must be set; if the latter, the file will be read, overwritten and deleted during daemon startup. It is assumed that the startup script, running as root, will store the password in this file immediately before starting the daemon. This provides some extra security by not leaving the private password visible to non-root users except for a short startup window.

org.lockss.keyMgr.keystore.*id*.create
(http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.keyMgr.keystore.%3cid%3e.create)
If true, and the keystore file does not exist at startup, one will be created, containing a newly generated private key and a self-signed certificate.

org.lockss.keyMgr.defaultKeyStoreType (http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.keyMgr.defaultKeyStoreType) [JCEKS]
Default type for newly created keystores.

Externally generated keystores

In many situations it will be more appropriate to use a keystore created by external tools than one created the daemon (see above). One such procedure is here: http://docs.codehaus.org/display/JETTY/How+to+configure+SSL . Other methods that produce equivalent keystores may be used.

The keystore created by the daemon will contain a self-signed certificate, which will cause browsers to display a security alert and require several confirmations before being accepted. An externally created keystore containing a key and certificate issued by a recognized Certificate Authority will be accepted by browsers with no warnings.

For LCAP SSL, the keystore must contain both a private key and certificates for all the other boxes in the PLN. Generating this sort of keystore requires external tools.

Admin UI

The admin UI listens for HTTP connections on port org.lockss.ui.port (http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.ui.port) (normally 8081). By default, HTTP basic authentication is used, and the only username/password combinations accepted are those supplied by the platform configuration and the network-wide configuration (optional monitoring/debug user and statically configured users).

The UI may be configured to require HTTPS (SSL) encrypted connections, use form-based authentication in place of basic authentication, to allow admin users to create additional user accounts and assign them privileges (account management), and to enforce password quality and rotation requirements.

HTTPS

To require users to connect to the UI with HTTPS, either use one of the SSL-enabled account policies below, or set org.lockss.ui.useSsl (http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.ui.useSsl) to true, and org.lockss.ui.sslKeystoreName (http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.ui.sslKeystoreName) to the name of a keystore configured with org.lockss.keyMgr.keystore.*id*. For example:

```
<lockss-config>
  <property name="org.lockss">
    ...
    <property name="keyMgr.keystore.1">
      <property name="name" value="ks1" />
      <property name="file" value="/etc/lockss/keystore" />
      <property name="password" value="..." />
      <property name="keyPassword" value="..." />
    </property>

    <property name="ui.useSsl" value="true" />
    <property name="ui.sslKeystoreName" value="ks1" />
    ...
  </property>
</lockss-config>
```

If org.lockss.ui.sslRedirFromPort (http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.ui.sslRedirFromPort) is set, http: connections to that port will be redirected to https: connections on the SSL port. (*Eg*, setting org.lockss.ui.sslRedirFromPort (http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.ui.sslRedirFromPort) to 80 and org.lockss.ui.useSsl (http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.ui.useSsl) to 443 results in the standrd redirect default-port http: to default-port https: behavior.

User Account Policies

The simplest way to enable HTTPS and user account control is to select one of several pre-configured accounting policies, by setting org.lockss.accounts.policy (http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.accounts.policy) to one of the following values.

org.lockss.accounts.policy (http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.accounts.policy) values

Compat
: The default. Basic authentication, no SSL, no account management.

Basic
: Basic authentication, no SSL, account management enabled.

Form
: Form authentication, no SSL, account management enabled.

SSL
: Form authentication, SSL, account management enabled.

LC
: Form authentication, SSL, account management enabled, Library of Congress rules for password quality and rotation, inactivity timeout, etc.

For finer-grained control over password requirements, use the Form policy in conjunction with org.lockss.accounts.minPasswordLength (http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.accounts.minPasswordLength) , org.lockss.accounts.minPasswordChangeInterval (http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.accounts.minPasswordChangeInterval) , org.lockss.accounts.maxPasswordChangeInterval (http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.accounts.maxPasswordChangeInterval) , org.lockss.accounts.passwordChangeReminderInterval (http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.accounts.passwordChangeReminderInterval) , org.lockss.accounts.inactivityLogout (http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.accounts.inactivityLogout) , org.lockss.accounts.passwordHistorySize (http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.accounts.passwordHistorySize) , org.lockss.accounts. (http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.accounts.) , org.lockss.accounts. (http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.accounts.) , org.lockss.accounts.maxFailedAttempts (http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.accounts.maxFailedAttempts) , org.lockss.accounts.failedAttemptWindow (http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.accounts.failedAttemptWindow) , org.lockss.accounts.failedAttemptResetInterval (http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.accounts.failedAttemptResetInterval) .

User Privileges

Users may be assigned any of the following privileges:

- User Admin - may create and delete user accounts, change passwords and privileges, and control which IP addresses or subnets are allowed access to the admin UI.
- Access Admin - may control which IP addresses or subnets are allowed access to content preserved on the box.
- Collection Admin - may select content (AUs) to be collected and preserved, and control whether that content should be collected via a proxy.
- Debug - will see links to additional status tables and information useful for diagnosing system problems.

Branding

To display a message or banner on the login page, set org.lockss.ui.loginBanner (http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.ui.loginBanner) to a string. The string may contain HTML markup.

To display a logo image on each UI page, set org.lockss.ui.logo.img (http://www.lockss.org/lockssdoc/gamma/daemon/paramdoc.html#org.lockss.ui.logo.img) to the URL of an image. The image will be displayed alongside or below the LOCKSS logo

Serving content to users

Preserved content can be made available to users via either a proxy server, or a direct content server.

Proxy

Proxying provides transparent access to preserved content. It requires some browser setup or some institution-wide setup, then content is accessed at its original URLs. See Proxy Integration for instructions to enable the proxy server and integrate it into your environment.

Content Server

The direct content server provides access to preserved content at URLs that point directly to the LOCKSS box.

To enable the content server, navigate to Content Access Options/Content Server Options and ensure that "Enable content server" is checked, and an appropriate port entered. See Controlling Access to Content to control who may access the preserved content.

Now access the page at `http://LOCKSS-box:port/` , where *LOCKSS-box* is the hostname of your LOCKSS box and *port* is the port you selected above. A list of AUs preserved on the box will be presented, with links to the manifest page(s) for each. Following these links will lead to copies of the content, with internal links rewritten to point back to the LOCKSS box.

Integration with SFX and other link resolvers will be available in early 2011.

Polling

LCAP

- SSL

Retrieved from "http://lockss.org/lockss/LOCKSS_Network_Administration"