Proxy Integration

A **web proxy** sits between the web browsers of end users and the Internet. Instead of attempting to fetch a web page from its original location on the Internet, web browsers configured to use a proxy ask the proxy to retrieve that page. In many cases, the proxy will simply fetch the page from its original location and pass it on to the browser that requested it. If the proxy is part of a larger hierarchy of proxies, it can also ask neighboring proxies if they can fulfill the request. Many proxy software implementations, such as Squid, are also **caches**. After receiving a web page that was requested by an end user (either by fetching it from the web or a neighboring proxy), the **proxy cache** (also called **caching proxy**) stores the resulting web page before delivering it to the user's browser. This means that the next time that same page is requested the proxy can serve up this local copy instead of having to fetch the content again from a remote site. *LOCKSS Boxes are proxy caches, although not exactly in the typical sense described above*. A LOCKSS Box only caches content it has been set up to preserve. The LOCKSS Box does not cache content only when it is visited by a user's web browser; it pro-actively fetches and stores all the content it is set up to preserve ahead of time. When the LOCKSS Box receives a proxy request for content not cached, it forwards this request on to the publisher's site. Even if the LOCKSS Box does have the content, a request is still sent to the publisher's site using the "if-modified-since" method to verify content hosted by the publisher has not been updated. If the publisher responds with newer content, this new content is served to the user. If the content is unchanged (the publisher responds with a 304 Not Modified header), the publisher DNS lookup fails, doesn't respond within a limited time, or responds with an error, LOCKSS serves the locally cached content.

LOCKSS Boxes usually don't have the horsepower to proxy all requests for an institution; some means must be employed to route only appropriate requests to it. The mechanism for this depends on the institution's existing cache/proxy architecture. LOCKSS can currently produce PAC files to configure individual browsers or a fragment of an EZproxy config file. LOCKSS also implements ICP, a communication mechanism used by proxy caches like Squid to ask each other whether they have a given piece of content and to respond to requests for content.

Configuring the Proxy

CONTROLLING ACCESS TO CONTENT

From the web user interface, navigate to Content Access Control and enter the list of IP addresses that should be allowed access to content preserved on this LOCKSS box. To be allowed access, an IP address must match some entry on the allow list, and not match any entry on the deny list. We strongly advise that the range of allowed IP addresses be kept to the minimum appropriate.

ENABLING THE PROXY

Navigate to Content Access Options/Content Server Options and ensure that "Enable content proxy" is checked,

and an appropriate port entered.

Integrating with PAC files

LOCKSS will generate a PAC file that can direct browsers to fetch preserved content from the LOCKSS box. This is the appropriate method for integrating LOCKSS into environments that have no existing proxy infrastructure. It's also the easiest way to access the LOCKSS proxy for short-term testing or demonstration.

In order to use the PAC file, each client browser will need to configured for use. For institutions already using PAC files, this may be transparent as the original PAC file can simply be replaced. Where a PAC file is not in use, manual intervention by users to configure their browsers is required. If you do not wish to impose such requirements on your users, consider using one of the alternative proxy integration approaches.

The LOCKSS Box locally stores material originally residing on a publisher site. The range of local content, and thus the set of designated sites from whom content is collected, depends on the Archival Units the box is configured to preserve. The LOCKSS PAC file will direct a web browser to route to the LOCKSS proxy requests for any content from that set of designated sites. If the LOCKSS Box does not respond, the browser will fetch directly from the publisher (daemon 1.22 and later). The browser will not route requests to the LOCKSS Box for other sites.

CREATING A LOCKSS PAC FILE

1. In the LOCKSS web interface, navigate to "**Proxy Info**."
2. Select "**PAC file**." The system will generate an up to date PAC file for you to save locally.
3. Post this file on a web server accessible by your institution's users.
    - Full instructions on how to do so are outside the scope of this guidance; for an introduction on how this works see the Wikipedia entry on PAC files.
4. Have users enter the URL for the above file in the "automatic proxy configuration" box in their web browser preferences.

INTEGRATING WITH EXISTING PAC FILES

If your institution already uses a PAC file, LOCKSS can generate a PAC file that combines the rules from it with the LOCKSS-specific rules, so your users will have a single PAC file that performs both functions.

1. In the LOCKSS web user interface, navigate to "**Proxy Info**".
2. Select "**Combined PAC file**".
3. Enter the existing PAC file information in one of the following ways:
    - Enter its URL in the box marked 'Enter the URL of a remote PAC file:', or
    - Make a local copy of the PAC file and upload it by clicking on the "Browse" button, or
    - Copy the text of the PAC file in the box marked 'enter PAC file contents here'
4. Click the button "Generate Combined PAC."
5. Save the resulting PAC file on a web server accessible to your users, and direct your users to alter their existing settings to use this file.

FREQUENTLY ASKED QUESTIONS

1. What happens if the LOCKSS Box is down?
    - In the event that a LOCKSS Box does not respond to a request for journal content, the PAC file contains some simple logic which directs the browser to attempt a direct connection (or, if using a combined PAC file, use the existing institutional proxy).

TO DO LIST

We plan to add features to make this easier. They include:

1. Scripts to automatically generate combined PAC files on a regular basis.
2. Ability to generate a single PAC file for a set of LOCKSS Boxes.

Integrating with EZproxy

EZproxy is a popular proxying solution for libraries to use to provide off-site access to their licensed content. Unlike traditional proxies, you do not need to use any special browser configuration or routing rules to install it. It is a rewriting proxy, dynamically altering the URLs in a page to point to itself; to paraphrase the example used in the Useful Utilities URL rewriting section a title available at http://www.somedb.com/index.html will be made available through EZproxy at http://www.somedb.com.ezproxy.yourlib.org/index.html. If you are unfamiliar with EZproxy but wish to learn more, please view the more detailed overview from the Useful Utilities site.

You can integrate EZproxy with your LOCKSS Box to access the content you are preserving, for both on-site and off-site users. Note, of course, that both sets of users will have to have to access the journals via the EZproxy base URL as described above.

STAND ALONE INSTALL

Follow these instructions if you wish to set up a minimal EZproxy system in order to serve LOCKSS preserved content. This, for example, may be appropriate if you wish to set up a proxy for on-site access to content.

LOCKSS understands the format of EZproxy configuration files and can generate a suitable fragment that will cause EZproxy to forward selected requests to the LOCKSS Box.

1. Ensure you have a sufficiently recent copy of EZproxy – Release 2.2a or later. EZproxy is available at http://www.usefulutilities.com/download/
2. In the LOCKSS Box's admin UI, go to Proxy Info, click on "EZproxy config fragment".
3. Use that fragment as the start of a configuration file. You will need to add access control to the beginning to limit access to the appropriate users. See the Useful Utilities site for instructions on how to do this.
4. If the machine you are using is behind a firewall or campus proxy, you may need to route outbound requests for non-LOCKSS journals through this using the `Proxy` directive. See the recent changes on the Useful Utilities site for how to do this. Note, when declaring your proxy, do not use a `http://` prefix. In addition, your EZproxy setup may require specific ports opened in the campus firewall (for more information, see the FAQ on firewalls).
5. Restart EZproxy.

INTEGRATE WITH AN EXISTING INSTALLATION

If you are already using EZproxy to provide access to off-site users, you can integrate that server with your LOCKSS Box as well.

1. Ensure you have a sufficiently recent copy of EZproxy — Release 2.2a or later. EZproxy is available at http://www.usefulutilities.com/download/
2. In the LOCKSS Box's admin UI, go to Proxy Info, click on "EZproxy config fragment".
3. Add that fragment to your existing ezproxy.cfg. You generally want to add the fragment to the end, but this may vary depending on your specific configuration.
4. Restart EZproxy.

FREQUENTLY ASKED QUESTIONS

1. What happens if the LOCKSS Box is down?
    - In it's current release, EZproxy does not fail gracefully if the LOCKSS system fails to respond to a query. The EZproxy developers are aware of this issue and looking for colleagues running LOCKSS to test an improved version. If you are willing to work with us to test this, please contact us.

Integrating with ICP

The LOCKSS proxy features built-in support for the Internet Cache Protocol (ICP), a standard communication protocol used by proxies to exchange hints about the availability of web content with neighboring proxies. It integrates therefore very well with other ICP-capable proxies at your institution.

ICP is supported by several proxying solutions, including Squid. See the section on integrating with Squid for details.

ENABLING ICP IN THE PLATFORM

The LOCKSS Platform enforces packet filters, including for ICP. A UDP port is open for ICP only if you chose to allow ICP during configuration, and if you did so, access to this port is permitted only from the subnets you designated as ICP subnets.

If you have disallowed ICP in the Platform, you will need to do a configuration reboot; see the instructions in Build A LOCKSS Box. During the part about ICP, make sure you allow ICP by answering **Y**.

ENABLING ICP IN THE PROXY

To enable the LOCKSS proxy's ICP server:

1. In the LOCKSS web user interface, select the "Proxy Options" screen, then click on the "Proxy Options" command at the center of the screen.
2. Check the box next to "Enable ICP server".
    - If you see the message "The platform is configured to disable the ICP server", your LOCKSS Box is not set up to run ICP because ICP was disallowed in the LOCKSS Platform during configuration. You will need to run through a configuration reboot. See the section on enabling ICP in the Platform for details.
3. Enter a port number in the corresponding text field.

- Typically ICP servers run on port 3130.
- Make sure that this port number is the one that was opened for ICP in the LOCKSS Platform during configuration. See the section on enabling ICP in the Platform for details.

4. Click on the “Update Proxy” button.

The LOCKSS Box will then have an ICP server running on the specified port, to which you point an ICP-capable proxy. See how to do this with Squid.

Integrating with Squid

Squid is a widely deployed web proxy cache solution released as a free, open source software package. Squid supports ICP and features fine-grained configuration options per domain name that makes integrating LOCKSS boxes easy. The LOCKSS web user interface can generate configuration information for Squid proxies. The recommended method is to use a `dstdomain` file.

As a **prerequisite**, the LOCKSS proxy’s ICP server must be enabled. See the section on integrating with ICP for details.

For better illustration, we will use a running **example** to demonstrate what the configuration parameters will look like. We will assume that there is a LOCKSS Box with host name `lockss.myuniversity.edu`, running its proxy on port 8080 and its ICP server on port 3130.

USING A SQUID `DSTDOMAIN` FILE

This is the **recommended** configuration method. (This configuration method is well-suited to automated Squid configuration; the `dstdomain` list for each LOCKSS Box, which changes over time as more content is added to it, can be kept updated by a script which periodically downloads it and saves it where your Squid process can access it.)

In the LOCKSS web user interface, go to the “Proxy Info” screen, and click on “Generate a `dstdomain` file for Squid”. You will obtain something similar to this:

```
# LOCKSS dstdomain file for Squid
# Generated Fri Feb 02 19:56:42 GMT 2007 by LOCKSS cache lockss.myuniversity.edu

# Suggested file name: lockss-myuniversity-edu-domains.txt

# Suggested use in Squid config file:
#
#    # Edit the path accordingly
#    acl lockss-myuniversity-edu-domains dstdomain "/the/path/to/lockss-myuniversit
#
#    # If you already have "acl XYZ src 0.0.0.0/0.0.0.0" or equivalent elsewhere,
#    # comment out this next line and replace "deny anyone" by "deny XYZ" in
#    # "cache_peer_access lockss.myuniversity.edu deny anyone" (see below).
#    acl anyone src 0.0.0.0/0.0.0.0
```

```
#
#    cache_peer lockss.myuniversity.edu parent 8080 3130 proxy-only
#
#    cache_peer_access lockss.myuniversity.edu allow lockss-myuniversity-edu-domain
#
#    # If you already have "acl XYZ src 0.0.0.0/0.0.0.0" or equivalent elsewhere,
#    # replace "deny anyone" by "deny XYZ" (see above).
#    cache_peer_access lockss.myuniversity.edu deny anyone
#

# Antimicrobial Agents and Chemotherapy Volume 50
aac.asm.org

# Administration & Society Volume 38
aas.sagepub.com
```

(and many more such lines for each domain).

- If the first suggested `cache_peer` directive carries the message “The platform on lockss.myuniversity.edu is configured to disallow ICP. To enable ICP you must perform a platform reconfiguration reboot”, the LOCKSS Box is not set up to run ICP. You will need to run through a configuration reboot. See the section on enabling ICP in the Platform.
- If the first suggested `cache_peer` directive carries the message “The ICP server is not running on lockss.myuniversity.edu. Replace “???” by the ICP port after setting it up”, the Platform allows ICP but the ICP server is not running. See the section on enabling ICP.

To start using your LOCKSS Box with your Squid proxy:

1. Save this file where your Squid process can access it.
    - The generated file will suggest a name (in the “Suggested file name” comment at the top). The suggested name contains the host name of your LOCKSS Box where the dots have been replaced by dashes; in our example the suggested file name will be `lockss-myuniversity-edu-domains.txt`. This naming scheme is useful if you have more than one proxy at your institution.
2. Use the Squid configuration directives suggested in comments at the top of the file to add the necessary information to your Squid configuration file.
    - The meaning of the five suggested directives is explained in detail below.
3. Restart Squid.

The five suggested Squid directives (using our example) are:

```
acl lockss-myuniversity-edu-domains dstdomain "/the/path/to/lockss-myuniversity-edu
acl anyone src 0.0.0.0/0.0.0.0
cache_peer lockss.myuniversity.edu parent 8080 3130 proxy-only
cache_peer_access lockss.myuniversity.edu allow lockss-myuniversity-edu-domains
```

```
cache_peer_access lockss.myuniversity.edu deny anyone
```

Here is a more detailed explanation of their meaning:

- The first directive creates a `dstdomain` access control list (ACL) made of all the domains in the `dstdomain` file generated by the LOCKSS Box.
- The third directive designates the LOCKSS Box as a parent cache, and gives the LOCKSS Box's proxy port and ICP port respectively.
- The fourth and fifth directives instruct the Squid cache to include the LOCKSS Box when fulfilling requests for URLs from domains in the `dstdomain` ACL and to exclude the LOCKSS Box from consideration otherwise, respectively.
- The second directive defines an ACL that matches all requests, if you do not already have one defined in your Squid configuration file. If you already have one, do not include the second directive (for instance, comment it out by adding `#` at the beginning of it, or remove it altogether), and replace `anyone` in the fifth directive by the name of your existing ACL.

USING A SQUID CONFIGURATION FRAGMENT

This configuration method is **not recommended** and would only be useful if your automated Squid configuration process assembles a configuration file for Squid from fragments and templates. If it does, it would presumably be amenable to inserting the required directives described above for a `dstdomain` file.

In the event the auto-configuration is not amenable to a `dstdomain` file, the LOCKSS web user interface provides you with an alternative option, "Generate a configuration fragment for Squid". The result is a portion of the main Squid configuration file similar to what you would write for the `dstdomain` file, except that the `dstdomain` ACL (named `lockss-myuniversity-edu-domains` in the example above) needs to be declared one domain at a time. See the instructions at the end of the generated file (after all the individual domain declarations) for more details.

FREQUENTLY ASKED QUESTIONS

1. What happens if the LOCKSS Box is down?
    - We advise that Squid is configured so that if the LOCKSS Box fails to respond to an ICP query within a brief interval, the content will be requested directly from the publisher. This configuration directive is included within the Squid fragment produced by the LOCKSS Box.

Integrating with multiple systems at once (e.g. EZproxy and Squid)

Integrating multiple proxy systems can be achieved, but will depend on the precise configuration currently in use within your institution. At this stage, it is more productive to discuss this in detail to ensure this works as expected. Please contact us to begin this discussion.

Verifying Proxy Operation

Of course, the mere fact that pages display normally does not provide any evidence that the proxy configuration is working. In order to determine that the browser actually fetches from the proxy, and that the proxy responds appropriately, it is necessary to inspect the headers of the HTTP response. This is not difficult, but is not

supported by most browsers by default.

- A Mozilla add-on is available (at http://livehttpheaders.mozdev.org/) which adds HTTP header info to the View / Page Info menu item. After fetching a cached page, do View / Page Info, click the Headers tab and look for a Via: or X-LOCKSS header (below).
- On any Unix or Linux system, `wget -S` can be used to fetch a page and display the HTTP response headers. The environment variable `http_proxy` (lower case!) controls proxying for wget.
- A packet capture (ethernet sniffer) tool can be used to inspect the HTTP transaction or to show that the traffic went through the cache.

For example, using wget and assuming that the cache *lockss.lib.edu* is configured to preserve ALR 2003, the headers resulting from:

```
export http_proxy=lockss.lib.edu:8080
wget -S http://absinthe-literary-review.com/
```

should look something like:

```
HTTP/1.1 200 OK
Date: Wed, 15 Sep 2004 00:17:55 GMT
Server: Apache/1.3.27 (Unix) ...
  ...
Via: 1.1 (LOCKSS/jetty)
```

The Via: header indicates that the LOCKSS Box forwarded the request and returned the result from ALR. (This page is not part of the ALR 2003 AU, so this request will always return content from the publisher, not the LOCKSS Box.)

A page that is in the LOCKSS Box (and is up-to-date) will produce a response like this:

```
wget -S http://absinthe-literary-review.com/archives03.htm
```

```
HTTP/1.1 200 OK
Date: Wed, 15 Sep 2004 01:00:24 GMT
Server: Jetty/4.2.17 (OpenBSD/3.5 i386 java/1.3.1_11)
X-LOCKSS: from-cache
  ...
```