

SQUID/ICAP mime-type filter

Buyer introduction:

The buyer is a German internet service provider currently building a hosted web security solution. This is the first of several similar projects. All will address different filtering needs but will be connected to a SQUID via ICAP and configured on a per user basis via MySQL.

Short description:

We're looking for a SQUID 3 content-filter extension. It should be connected via ICAP to the squid proxy server and is therefore an ICAP server. This ICAP server should be able to detect the mime-type of a request and should translate this mime-type into a category learned from configuration files. With the help of a MySQL Database the ICAP server has to determine if the category is allowed for a specific user and send a HTTP redirect header if not.

This functionality doesn't have to come from scratch – GPL code is allowed (some interesting projects listed below), nevertheless LGPL or similar is preferred.

Long description:

In this setup the SQUID proxy server version 3 will function as an ICAP client and will feed the requested ICAP server all data that is coming from the web. The ICAP server should be written in C and should be multi-threaded (see the mentioned c-icap project). During startup the ICAP server will process three configuration files and establish one MySQL connection. One configuration file will tell the server how long it should cache specific user settings, categories and exceptions, which URL to include in the redirect header and what values to use to connect to the MySQL server (hostname, username and password). The second and third configuration file will provide a mapping between file endings and categories as well as content/mime-types and categories. The number of file endings, mime-types and categories can be indefinite. After the configuration files are read these values should be stored in memory. If the mentioned MySQL connection gets lost it should be re-established.

When data is forwarded to the ICAP server the server will take a look at the ICAP header and the listed recipient IP as well as the requested URL. With the help of this IP-Address an individual user can be identified. A MySQL Query is used to determine the user and to check to which extend the content/mime-type filtering is enabled for the specific user. A basic filtering just takes into account the filename and the content-type set by the web server whereas a more advanced filtering will try to guess the mime-type with real content. If the filtering is enabled another query is used to fetch the allowed categories as well as possible URL exceptions. All values that are fetched via MySQL will be cached for a specific amount of time configured in one of the configuration files (e.g. 300 seconds).

If filtering is enabled for the specific user the retrieved user specific list of exceptions (e.g. <http://download.microsoft.com> or somepage.com) is tested against the requested URL found in the ICAP encapsulation. If an exception can be found in the URL-String no further check is performed. Otherwise the HTTP header will be examined. The filename and the claimed content-type are

extracted. Afterwards the file ending is extracted from the filename. This file ending is used to determine a category. If it's not possible to translate the file ending into a category this will be tried with the content-type. If that's also not possible the category is 'Unknown'. Afterwards a check is performed if the resulting category is allowed. If the category is not in the array of allowed categories the ICAP server will send a modified response containing a HTTP redirect with the URL specified in one of the configuration files. The URL as well as the detected category are passed as URL-encoded variables.

If advanced filtering for the user is set another check will be performed. The body or to be more precise the first bytes of the body (e.g. the first 200 bytes) are taken to guess the mime-type by content and not just by file ending or a content-type set by a foreign web server. The detection of the real mime-type relies on the "magic numbers" at the beginning of the file. Libmagic can be used to achieve this (see "man libmagic" or try "file -i <FILENAME>" to see this library in action). After the mime-type has been determined it is again translated into a category and that category is checked against the array of allowed categories. If the detected category is not among the allowed ones the same redirect as above is send.

The source code should be well documented and performance/throughput is very important.

MySQL database structure:

users <ul style="list-style-type: none">• user_id• ip_address	categories <ul style="list-style-type: none">• category_id• category_name	allowed_categories <ul style="list-style-type: none">• user_id• category_id
user_settings <ul style="list-style-type: none">• user_id• filter_value	User_exceptions <ul style="list-style-type: none">• user_id• exception_value	

Values for filter value:
0 = none
1 = basic
2 = advanced

Configuration files:

Main configuration file:

```
cacheTime = 300;
redirectURL = http://www.someURL.com/block.php

mysqlHost = <Some Host or IP>
mysqlUser = <Username>
mysqlPassword = <Password>
```

Categories and file endings:

One category by line; file endings separated by comma

textandhtml = htm, html, txt, conf, ...

pictures = jpeg, jpg, ico, gif, ...

video = mpg, mpeg, avi, ...

.
. .
.

Categories and content-/mime-types:

One category by line; content-/mime-types separated by comma

textandhtml = text/html, text/xml, text/plain,...

pictures = image/jpeg, image/gif, image/png, ...

video = video/mp4, video/mpeg, video/ogg, ...

.
. .
.

Possible interesting projects:

C-ICAP:

<http://c-icap.sourceforge.net/>

Mime-Type guessing:

libmagic

MySQL:

libmysqlclient