

Reference Manual

ufdbGuard version 1.12 *the internet filter for the Squid web proxy*

URLfilterDB

Table of Contents

1	Introduction	4
1.1	What is URL Filtering?.....	4
1.2	What is Tunnel Detection?.....	4
1.3	Copyright	4
2	Prerequisites	5
3	Architecture	5
3.1	Dynamic Proxy Tunnel Detection	6
3.2	Enhanced HTTPS Security	6
4	Software Installation	7
4.1	Upgrading from a Previous Version	7
4.2	User Account	7
4.3	Installation Directory	7
4.4	Unpack Software.....	7
4.5	Configure the Software Build	7
4.6	Compile Software	8
4.7	Install Software	8
4.8	Get Daily Updates.....	8
5	Configuration.....	9
5.1	Know your Internet Usage Policy	9
5.2	Automatic Improvements of the URL Database.....	9
5.3	Proxy Tunnel Detection	10
5.4	Control HTTPS Usage	10
5.5	Main Configuration Settings ufdbGuard.....	10
5.6	Ensure Access to Internal and 3 rd Party Websites.....	11
5.7	Configure the Style of Messages	12
5.8	Configure Squid.....	12
5.8.1	Test Mode.....	12
5.8.2	Configuration for 2 servers.....	12
5.9	Monitoring	13
6	Start the URL Filter	13
7	Exception Rules	13
7.1	Allow Extra Sites	13
7.2	Block Extra Sites	14
8	Advanced Options	15
8.1	Blocking Adult Images produced by Search Engines.....	15
8.2	Different Policies for Different Users	15
8.2.1	Policy based on IP Address	15
8.2.2	Policy based on Username.....	16
8.2.3	Policy based on UNIX Groupname	16
8.2.4	Policy based on Domain Name	17
8.3	Multiple ACLs	17
8.4	Whitelisting.....	18
8.5	Extended Logging.....	18

URLfilterDB

8.6	Logfile Rotation.....	18
9	Performance Tuning	19
9.1	Squid performance	19
9.2	Linux 2.6 performance.....	19
10	Integration with 3rd Party Products	20
11	Frequently Asked Questions	21
11.1	URL Categories	22
12	More Information.....	24
13	Product Comparison	24
13.1	Method A: Content Scanning.....	24
13.2	Method B: Artificial Intelligence.....	25
13.3	Method C: Blacklist	25
13.4	Comparison of Methods.....	25

URLfilterDB

1 Introduction

ufdbGuard is a URL filter software suite that can be used with the Squid web proxy to block unwanted web sites on the internet. URLfilterDB is the URL database that ufdbGuard uses to determine which web sites are restricted.

Version 1.10 of ufdbGuard introduced a new unique feature to dynamically detect proxy tunnels. Section 3.1 contains an explanation about proxy tunnels are and why they are considered security threats.

Version 1.12 of ufdbGuard introduced a new feature to enforce SafeSearch on the Google search engine and family filters on other search engines. This feature reduces the number of adult images that a user can see on the internet.

You may register as a trial user at <http://www.urlfilterdb.com> to receive a 60-day trial license for the URL database. The trial license is for an operational URL filter. For evaluation purposes, the test mode of ufdbGuard may be used to verify which sites can be blocked without actually blocking anything. At the end of the trial period, the URL filter will not block any URL any more and allows access to any website.

Starting from release 1.3, a new architecture is used so administrators of previous versions are urged to read this manual carefully.

1.1 What is URL Filtering?

Users browse the internet and often without knowing it, they usually use a web proxy like Squid that is in between a PC and the internet. Squid and ufdbGuard can verify if a user has access to a particular (part of) a website that is visited. Approved websites can be visited without restriction. In case that a (part of) a website is restricted; the web browser displays a message that access is prohibited.

1.2 What is Tunnel Detection?

Users who want to circumvent company internet and security policies, use so-called *tunnels* to break through the firewall that protects your internal network from attacks from the outside. Tunnels can be extremely harmful since in most cases the antivirus protection is circumvented and tunnels are bidirectional, i.e. the firewall is open to attacks from the outside through the tunnel. ufdbGuard detects and blocks these tunnels.

1.3 Copyright

The ufdbGuard software suite is free. To protect the ownership and the freedom of use of ufdbGuard, there is a copyright and you have a license to use and modify the software freely, known as the GPL version 2 license. The license is here: <http://www.gnu.org/licenses/old-licenses/gpl-2.0.html>.

The URLfilterDB database is a commercial product and has a copyright by URLfilterDB. You need a license to use the database and will receive a copy of the Terms of Service for the database when you register as a trial user, buy a subscription for the database or at request.

URLfilterDB

2 Prerequisites

ufdbGuard runs on all flavors of UNIX and is usually installed on the same system where Squid is installed. Squid can be downloaded at <http://www.squid-cache.org>. ufdbGuard needs 20 MB disk space and 80 MB memory. The required CPU power is compared to Squid relatively low, so it can run on one CPU for small to medium sized user groups. For large user groups, it is highly recommended to use a dual-CPU system and to use top-of-the-line dual-core servers for very large environments.

For all systems, ufdbGuard must be compiled with a C compiler. Most Unix distributions come with the free GNU C compiler, `gcc` (see also <http://gcc.gnu.org>) or a native C compiler. In addition, the `wget`, `make`, `lex`, `yacc` and `install` commands are required which are all included in most UNIX distributions.

ufdbGuard uses a compressed database and therefore requires the compression library `bz2`. This library is installed on most UNIX systems. In case that it is not on your system, it can be installed from your UNIX distribution CDs or downloaded from <http://www.bzip.org>. It also uses `openssl` and needs the `openssl` header files which are included in your UNIX distribution (usually the package name is `openssl-devel`).

3 Architecture

Squid is a popular web proxy that is used as an internet cache and is usually part of the internet security solution where users are not given direct access to the internet. Squid is free and can be downloaded from <http://www.squid-cache.org>.

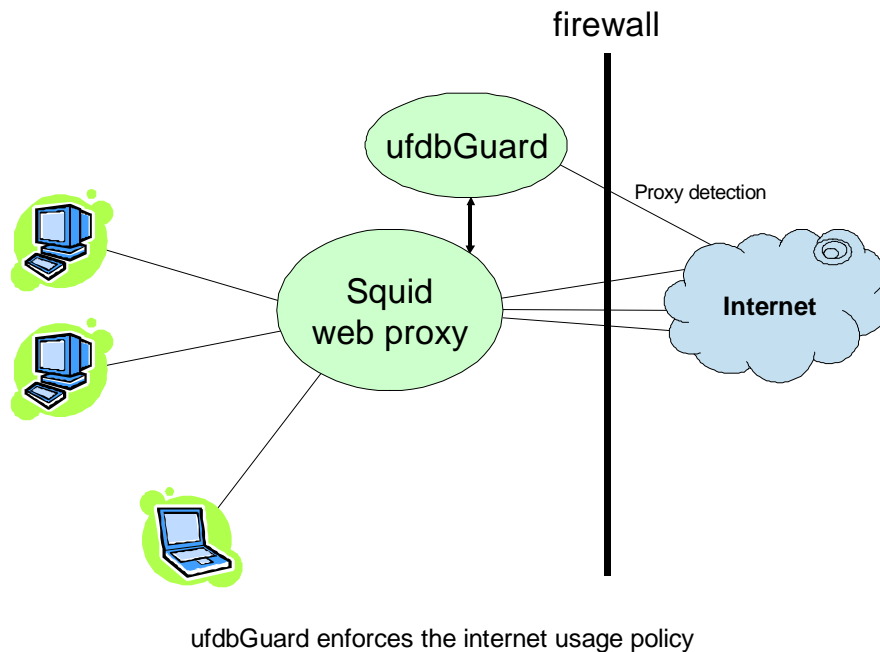
Squid uses child processes called *redirectors* to verify URLs for blocking. Starting with version 1.3, the redirectors are lightweight processes (`ufdbgclient`) that communicate with a multithreaded daemon process (`ufdbguardd`). The daemon has only one copy of the URL database in memory and does the actual URL verification. The daemon sends a reply indicating OK or BLOCKED to the redirector which sends the answer to Squid.

In case that the daemon is not running or when the daemon is loading a new version of the URL database, all URL verifications get an immediate OK and no web site is blocked. A database reload typically takes between 15 and 25 seconds.

The `ufdbguardd` daemon uses socket port 3977 to communicate with `ufdbgclient`. `ufdbguardd` and `ufdbgclient` can be configured to use an alternative port number.

In the next simplified diagram, all pieces are put together: the web browser on a PC connects to the Squid web proxy who uses ufdbGuard to block access to websites.

URLfilterDB



Each request to retrieve a page from a website is first verified with ufdbGuard. It is extremely fast with 50,000 URL verifications/sec so blocking unwanted web sites is very easy.

ufdbGuard can be used with any URL database that either comes in a flat file format or the proprietary .ufdb format. The ufdbGuard software suite comes with a utility ufdbGenTable to generate an .ufdb database file from a flat file.

3.1 Dynamic Proxy Tunnel Detection

The https protocol is encrypted and designed for secure online transactions like online banking. Unfortunately, the https protocol is a security risk since anybody can type "proxy tunnel" at Google and find out how to transfer data to and from any system on the internet *bypassing firewalls and internet access security measurements*. ufdbGuard has a unique protection mechanism that detects abuse of the https protocol and blocks all unauthorized transfers of data with these so-called proxy tunnels.

Proxy tunnels can also be used with reverse port forwarding (using ssh) which means that from any system on the internet an unauthorized connection can be made into the protected network. It also does not matter how good the firewall is! As long as https is allowed and there is not a proper countermeasure against proxy tunnels, a security risk exists.

The ufdbguarddd daemon verifies all websites that are accessed with the https protocol and detects all popular proxy tunnels.

3.2 Enhanced HTTPS Security

As stated earlier, the https protocol is a useful protocol where secrecy is desired. Unfortunately, the https protocol is also used by applications that may introduce security issues. To enhance the security of HTTPS connections on the internet, two configuration options control the use of HTTPS.

URLfilterDB

4 Software Installation

It is assumed that the Squid web proxy is already installed, configured and operational. Please refer to the documentation of Squid to make it operational (see also <http://www.squid-cache.org>).

4.1 Upgrading from a Previous Version

A previous version can simply be overwritten by a new version.

CAVEAT: you must stop Squid and ufdbGuard to overwrite executables, so you must stop them before a 'make install'.

When an upgrade is performed, it is recommended to perform all installation steps including the last step, retrieval of database update in section 4.8.

When an upgrade is performed, the default configuration file (`ufdbGuard.conf`) is not installed to prevent loss of previous settings. It is recommended to read the sections about configuration options and to add settings for the new options to the existing configuration file.

4.2 User Account

The ufdbGuard software suite should be installed with the same user account as the Squid software. Usually the username `squid` is used but it can be any name.

NOTE: the following steps in this section should be done as the aforementioned user.

4.3 Installation Directory

ufdbGuard can be installed in any directory. However, it is logical to put ufdbGuard together with Squid. In this manual, the word `TOPDIR` refers to the top level installation directory for ufdbGuard. Examples of `TOPDIR` are `/local/squid` or `/usr/local/squid` depending on your choice.

4.4 Unpack Software

The tar file that contains the ufdbGuard software suite must be unpacked in a source directory of your choice, e.g. `/local/src`.

Unpack the ufdbGuard tar file in the source directory:

```
$ cd /local/src
$ tar xzf ufdbGuard-latest.tar.gz
```

A directory with the name `ufdbGuard-1.12` contains the software.

4.5 Configure the Software Build

Before ufdbGuard is compiled, it needs to be configured and told what the `TOPDIR` of your choice is:

```
$ cd ufdbGuard-1.12
$ ./configure --prefix=TOPDIR
```

So, assuming that `TOPDIR` is `/usr/local/squid`, the `configure` command becomes:

```
$ ./configure --prefix=/usr/local/squid
```

The command `./configure --help` displays a description on how to configure alternative locations and username.

UfdbGuard is installed with the assumption that Squid runs with user account `squid`. If `squid` is run with a different user account, you should use the `--with-ufdb-user` option.

URLfilterDB

The most common error is that the development packages for openssl and bzip are not installed. The configure command checks for this and gives appropriate messages. For most operating systems, you can find the packages openssl-devel and bzip2 on your installation media.

4.6 Compile Software

Now compile the software suite:

```
$ cd src
$ make
```

4.7 Install Software

In case that you perform an upgrade and Squid is already running with ufdbgclient redirectors, squid must be stopped to be able to perform a proper installation. After the installation, Squid is started after ufdbGuard.

Stop any running instances of Squid:

```
$ su -
# /etc/init.d/squid stop
```

Stop any running instances of ufdbGuard. I.e.

```
$ su -
# /etc/init.d/ufdb stop
```

The programs can now be installed. Note that you must be root.

```
# cd src
# make install
```

It is recommended to start the ufdbguardd daemon at system boot. The start script is in the `init.d` directory (this may vary depending on the OS). Optionally modify `UFDB_OPTIONS` to use the `-T` option to use the test mode.

4.8 Get Daily Updates

The script `ufdbUpdate` takes care of downloading a new version of the URL database and signaling `ufdbGuard` that a new version is downloaded. Daily updates are available for everyone with a permanent or trial license of URLfilterDB. The `ufdbUpdate` script should be run twice per day by the cron scheduler. There are 3 things to do:

- make sure that the `squid` user is allowed to run crontab jobs (`crontab -l` should give no errors)
- enter the username and password in `ufdbUpdate` that you received when you purchased a license or received a trial license.
- choose an appropriate time at the end of the day and in the early morning to run `ufdbUpdate`

Edit the `ufdbUpdate` script to include the username and password that you received when the (trial) license was received:

```
$ vi /usr/local/squid/bin/ufdbUpdate
...
DOWNLOAD_USER=lic99999
DOWNLOAD_PASSWORD=aa22bb
```

Users that evaluate the URL database may use the `demoXX` username and password.

URLfilterDB

Test the update script with the verbose option:

```
# su - squid
$ /usr/local/squid/bin/ufdbUpdate -v
```

The output should be similar to:

```
new database downloaded:
-rw-r--r-- 1 root root 5121312 Dec 6 18:04 /tmp/urlfilterdb-latest.tar.gz
done.
```

To install the cron job, edit the crontab table of the user `squid` and add the appropriate lines:

```
$ crontab -e
```

To run the URL database update each day at 10:00 PM and 6:15 AM, add these two lines:

```
00 22 * * * /usr/local/squid/bin/ufdbUpdate
15 6 * * * /usr/local/squid/bin/ufdbUpdate
```

At this time, you may want to verify that the Squid housekeeping is also executed. Verify that the crontab has an entry to run “`squid -k rotate`” (see the squid manuals for more details).

5 Configuration

5.1 Know your Internet Usage Policy

Before you start with the configuration of `ufdbGuard`, you need to define an internet usage policy that defines which web site categories are considered unwanted and therefore must be blocked by `ufdbGuard`. The HR department might be a good starting point to find out which categories of the database must be used to block access to parts of the internet.

NOTE: the default configuration of `ufdbGuard` blocks only adult, proxies, gambling and warez!

In addition, there is an option to define categories that are managed by the local administrator: *alwaysallow* and *alwaysdeny* to define URLs that should be allowed or denied according to the local policy. Section 7 explains on how to configure these categories.

It is recommended to start with the URL filter in the test mode. In this mode, sites are not blocked but only logged in the log file. An analysis of the log file may help in defining the appropriate Internet Usage Policy. At last a free tip: to prevent a storm at the helpdesk or system administrator, it is advised to inform users about (change in) the implementation of a URL filter *before* it is implemented.

5.2 Automatic Improvements of the URL Database

Although the URL database is updated daily, it may happen that some web sites are not included in the URL database and therefore users might be able to visit inappropriate websites. `UfdbGuard` has a configuration option to collect a sample of these uncategorized domains and upload them to `URLfilterDB` for inclusion in the database. This option is ON by default. Privacy is guaranteed since no identification of client, source or user is included, just the domain name is registered for categorization.

To *prevent* analysis of uncategorized URLs, the `analyse-uncategorised-urls` configuration option must be set to OFF. It is recommended to leave this option ON to receive more updates in the database. `URLfilterDB` has a policy to categorize URLs within 24 hours so that they are always included in the next day's URL database.

URLfilterDB

5.3 Proxy Tunnel Detection

Proxy tunnels are a security risk and it is strongly recommended to use detection of proxy tunnels. Enable proxy tunnel detection in the configuration file `ufdbGuard.conf` with the following line:

```
check-proxy-tunnel queue-checks
```

This option queues checks for proxy tunnels to be detected a few seconds later. It means that a proxy *can* be used but only for 3 seconds. Alternatively, proxy tunnels can be detected in an aggressive mode, where all https traffic is tested for proxies *before* access is given. The aggressive mode introduces some delays for https traffic and is therefore only recommended in extremely high security environments. If proxy tunnels are allowed, the value for `check-proxy-tunnel` can be `off` or `log-only`. All valid options are:

```
check-proxy-tunnel queue-checks      # recommended
check-proxy-tunnel aggressive
check-proxy-tunnel log-only
check-proxy-tunnel off
```

NOTE: when the aggressive detection method is used, the number of threads in the `ufdbguardd` daemon and the number of redirector processes must be increased to 64. Therefore, change in `squid.conf` the parameter `redirect_children` or `url_rewrite_children` to 64 and restart squid. To start the `ufdbguardd` daemon with 64 worker threads, the command line option `-w 64` must be used. To use this option, the variable `UFDB_OPTION` in `/etc/init.d/ufdb` must be set.

5.4 Control HTTPS Usage

Most websites that use HTTPS for legitimate business reasons use an SSL certificate that is signed by a well-known certificate authority and a fully qualified domain name in the URL for maximum security and a clear identification of the website, while most websites that use HTTPS for other reasons, have self-signed SSL certificates and IP addresses instead of domain names. Access to HTTPS websites can be controlled with 2 options.

The following options are default in `ufdbGuard.conf`:

```
enforce-https-with-hostname on
enforce-https-official-certificate on
```

It is recommended to keep these options on. In case that a legitimate website uses an IP address in the URL or an SSL certificate that is not signed by a trusted authority, it is recommended to add this site to the locally trusted websites (see section 7.1).

Setting the above two options is not enough since there is not yet a category defined for it. The default configuration file has a category “security” that includes these options:

```
category security
{
    domainlist security/domains
    option      enforce-https-with-hostname
    option      enforce-https-official-certificate
    redirect    ...
}
```

Finally, include the category security into the ACLs (see section 5.5) to block https abuse.

5.5 Main Configuration Settings `ufdbGuard`

Use your favorite editor to edit the configuration file and define which categories must be blocked.

```
$ vi /usr/local/squid/etc/ufdbGuard.conf
```

URLfilterDB

There are 4 sections with a comment 'EDIT THE NEXT LINE'. Find each section and change the configuration where needed.

The first section defines the locations of the log directory and the blacklist database directory. This section does not have to be changed if the initial choice of these locations during the configuration of ufdbGuard (see section 4.5) has not changed. The section looks like this:

```
# EDIT THE NEXT LINE FOR LOCAL CONFIGURATION
dbhome /usr/local/squid/blacklists
logdir /usr/local/squid/logs
```

The second section defines the IP address range of your local network. The section looks like this:

```
src allSystems {
    ip 10.0.0.0/8
}
```

The appropriate network subnet must be entered in this section. 10.0.0.0/8 and 192.168.0.0/16 are the most common values for this. Consult your network administrator for assistance.

The third and fourth sections are close to each other and define the list of categories to be blocked (one list for the systems with IP address defined in `allSystems` and one list for all other systems). Change the list of categories to be blocked. The default list of blocked categories contains the categories `security`, `adult`, `proxies`, `gambling` and `warez`. The third section looks like this:

```
acl {
    allSystems {
        # EDIT THE NEXT LINE
        pass !security !adult !proxies !gambling !warez any
    }
}
```

To block a category, it needs to be present with an exclamation mark (!) that is used as a blocking indicator. So to block the `adult` category, `!adult` must be present in the line that starts with 'pass'. If you prefer to allow gambling, the definition "`!gambling`" must be removed.

At a site that only blocks security, adult and proxies, the section looks like this:

```
acl {
    allSystems {
        # EDIT THE NEXT LINE
        pass !security !adult !proxies any
    }
}
```

The fourth section is very similar to the third section and defines which categories to block for computer systems that are not part of `allSystems`.

5.6 Ensure Access to Internal and 3rd Party Websites

The domain name of your company may be included in the URL database of URLfilterDB. To ensure access to all own websites, the category `alwaysallow` should be configured (see section 7.1).

To prevent unhappy users, one should carefully examine which sites of 3rd parties are used for daily activities and make sure that these sites can be accessed without restriction. Therefore, it is recommended to run a few days in test mode (see section 5.8.1) and add sites of important 3rd parties to the category `alwaysallow` (see section 7.1).

URLfilterDB

5.7 Configure the Style of Messages

When a URL is prohibited to be visited, a message is displayed that access is forbidden. The size and background color can be set by the administrator.

The options for the background color are: orange (default), white, black, grey and red. The options for the font size of the message are: normal (default) and small.

To change the style of a message, edit the configuration file and change the default settings of all `redirect` rules.

```
$ vi /usr/local/squid/etc/squid.conf
...
redirect http://www.urlfilterdb.com/cgi-bin/URLblocked.cgi?admin=%A&color=orange&size=n
ormal&clientaddr=%a&clientname=%n&clientuser=%i&clientgroup=%s&targetgroup=%t&url=%u
...
```

Substitute the default value for a new value.

5.8 Configure Squid

To let Squid use the URL filter (in Squid terminology the URL filter is a *redirector* or a *URL rewriter*), the following 2 parameters must be added to the squid configuration file.

```
$ vi /usr/local/squid/etc/squid.conf
```

Add the following 2 lines for Squid 2.5:

```
redirect_program /usr/local/squid/bin/ufdbgclient -l /usr/local/squid/logs
redirect_children 16
```

Add the following 2 lines for Squid 2.6:

```
url_rewrite_program /usr/local/squid/bin/ufdbgclient -l /usr/local/squid/logs
url_rewrite_children 16
```

NOTE: by default, the `ufdbguardd` cannot support more than 32 `ufdbgclient` processes so do not use more than 32 `redirect_children`. Please contact the support desk in case you think more children are required.

5.8.1 Test Mode

`ufdbGuard` has a test mode where internet access is not blocked and where the log file contains lines which web sites would have been blocked in normal mode.

In case that you want to use the test mode, add the `-T` option for `ufdbguardd`. The `/etc/init.d/ufdb` (might be elsewhere for your OS) file should then have a line like this:

```
UFDB_OPTIONS="-T"
```

In test mode, the log file of `ufdbguardd` contains lines like this:

```
TEST BLOCK adult www.sex.com
```

5.8.2 Configuration for 2 servers

In case that `ufdbguardd` runs on a different system then where Squid runs, you can specify the server name and port number with the following options for `ufdbgclient`:

```
-S servername -p 3977
```

The `squid.conf` file should then have a line like this:

```
redirect_program /local/squid/bin/ufdbgclient -S urlchecker01 -p 3977
```

URLfilterDB

5.9 Monitoring

ufdbGuard uses 2 log files and the system log to write messages to. The location of the log files depends on the choice that was made during the configuration phase (see section 4.5). The names of the log files are `ufdbguardd.log` and `ufdbgclient.log`. The location and name of the system log is OS-dependant and is usually `/var/log/messages` or `/var/adm/syslog.log`.

It is recommended that the log files are regularly inspected, either manually or automatically.

6 Start the URL Filter

To start the URL filter daemon:

```
# /etc/init.d/ufdb start
```

Now you can restart squid to use the URL filter:

```
# /etc/init.d/squid reload
```

or

```
# /etc/init.d/squid stop
```

```
# /etc/init.d/squid start
```

7 Exception Rules

In cases where exceptions to the categories of URLfilterDB are desired, an administrator can define 2 extra categories that are managed by the administrator and never by URLfilterDB.

7.1 Allow Extra Sites

A common case is that you want to ensure access to your own websites and websites of 3rd parties that are used for normal activities. To grant users access to the company websites, the URL `yourcompany.com` needs to be added to the category *alwaysallow*.

For universities that want to allow access to all other universities in the United States, a simple `edu` in the *alwaysallow* list. In the UK, only `ac.uk` needs to be included!

Edit the file that contains the extra sites that should always be allowed:

```
$ cd /usr/local/squid
$ vi blacklists/alwaysallow/domains
```

Add the appropriate URLs and always remove a leading `www.`:

```
yourcompany.com
news.google.com
google.com/news
```

Additional domains can be added according to the local internet usage policy. For example, if news should be blocked but access to CNN allowed, then `cnn.com` should be added also. Alternatively, when news should be blocked but Google news allowed, `news.google.com` and `google.com/news` should be added.

ufdbGuard only uses proprietary database files, so generate an `.ufdb` database file from the ASCII file with `ufdbGenTable`:

```
$ cd /usr/local/squid
$ bin/ufdbGenTable -n -t alwaysallow -d blacklists/alwaysallow/domains
```

URLfilterDB

The above command generates the file `blacklists/alwaysallow/domains.ufdb` and should be invoked each time that the domains file is changed. Then activate the category by editing the `ufdbGuard.conf` file and uncomment the category definition for *alwaysallow*. The configuration file should have the following lines:

```
category alwaysallow
{
    domainlist alwaysallow/domains
    redirect ...
}
```

Also, add the category *alwaysallow* to the ACL `allSystems`. The ACL should then start with

```
pass alwaysallow !adult ...
```

Finally restart the `ufdbguardd` daemon:

```
# /etc/init.d/ufdb restart
```

7.2 Block Extra Sites

In case that you want to block access to a site that is not in any category, you can add this site to the category *alwaysdeny*. For example, `google.com` is not in any category but if you like to block access to this popular search engine, `google.com` can be included in the *alwaysdeny* category. Analogous to the *alwaysallow* category, the domain (without leading `www.`) must be added to the category domains file, and the ACL `allSystems` must be extended.

Edit the file that contains the extra sites that should always be blocked:

```
$ cd /usr/local/squid
$ vi blacklists/alwaysdeny/domains
```

Add the appropriate URLs (always remove a leading `www.`):

```
google.com
```

`ufdbGuard` only uses proprietary database files, so generate an `.ufdb` file from the ASCII file with `ufdbGenTable`:

```
$ cd /usr/local/squid
$ bin/ufdbGenTable -n -t alwaysdeny -d blacklists/alwaysdeny/domains
```

The above command generates the file `blacklists/alwaysdeny/domains.ufdb` and should be invoked each time when the domains file is changed. Then activate the category by editing the `ufdbGuard.conf` file and uncomment the category definition for *alwaysdeny*. The configuration file should have the following lines:

```
category alwaysdeny
{
    domainlist alwaysdeny/domains
    redirect ...
}
```

Also, add the category *alwaysdeny* to the ACL `allSystems`. The ACL should then start with

```
pass alwaysallow !alwaysdeny !adult ...
```

URLfilterDB

Finally restart the ufdbguardd daemon:

```
# /etc/init.d/ufdb restart
```

8 Advanced Options

8.1 Blocking Adult Images produced by Search Engines

Search engines like Google, Yahoo and MSN have a capability to search for images and allow users to view adult images that can not be blocked in a simple way since the images come from Google, Yahoo and MSN themselves and in general one would not like to block all images from the search engines.

Google and Yahoo offer a *safesearch* feature which blocks most adult images. UfdbGuard has the configuration parameter *safe-search* that enforces the safesearch policies of the search engines. The default value for the parameter is ON.

The safe-search feature enforces safe searches for the following search engines: A9, Alltheweb, Ask, Dogpile, Foxnews, Google, Infospace, Live, Lycos, Metacrawler, Metaspy, MSN, Webcrawler, and Yahoo.

8.2 Different Policies for Different Users

Suppose that ufdbGuard is used in a bank. The internet usage policy could be defined as: block sex, chat, dating, entertainment, finance, news, webmail. This policy can be appropriate for most users but not for staff working in a dealing room where access to news and finance-related sites is required. This section explains how to achieve this.

Always be careful with the order of rules and make sure that the more privileged user (groups) is defined first in the ACL.

8.2.1 Policy based on IP Address

If the dealing room has a separate IP subnet, the *dealingroom* policy can be defined in the following way in the ufdbGuard configuration file.

```
src dealingroom {
    ip 10.4.0.0/16      # in this example the dealingroom uses this subnet
}

acl {
    # more privileged users first
    dealingroom {
        pass !security !adult !dating any
    }
    allSystems {
        pass !security !adult !chat !dating !entertain !news !webmail any
    }
    default {
        pass none
        # the following redirect is for the pseudo category 'none'
        redirect http://www.urlfilterdb.com/cgi-bin/URLblocked.pl?...
    }
}
```

URLfilterDB

8.2.2 Policy based on Username

Internet access policies can also be based on a list of usernames. In this example, a list of usernames is used to use the dealingroom policy. The `src` definition defines which users are members of the group *dealingroom*.

```
src dealingroom {
    userlist /usr/local/squid/etc/dealers
}
```

The file `/usr/local/squid/etc/dealers` should contain the usernames of all dealers. The final policy definition is the same as a policy based on IP address:

```
acl {
    # more privileged users first
    dealingroom {
        pass !security !adult !dating any
    }
    allSystems {
        pass !security !adult !chat !dating !entertain !news !webmail any
    }
    default {
        pass none
        redirect http://www.urlfilterdb.com/cgi-bin/URLblocked.pl?...
    }
}
```

CAVEAT: additional configuration is required to make Squid able to find out which users are using Squid. You may configure squid to use user authentication or you have to install `identd` on all PCs to support this feature. Please read the Squid documentation for more information and do not forget to use `acl foo ident REQUIRED`.

8.2.3 Policy based on UNIX Groupname

Internet access policies can also be based on a UNIX group. In this example, a groupname is used to use the dealingroom policy. In this case, the `src` definition defines that the group `dealer` represents the dealingroom.

```
src dealingroom {
    unix group dealer
}
```

The group `dealer` must be a valid UNIX group name. The final policy definition is the same as a policy based on IP address:

URLfilterDB

```
acl {
  # more privileged users first
  dealingroom {
    pass !security !adult !dating any
  }
  allSystems {
    pass !security !adult !chat !dating !entertain !news !webmail any
  }
  default {
    pass none
    redirect http://www.urlfilterdb.com/cgi-bin/URLblocked.pl?...
  }
}
```

CAVEAT: additional configuration is required to make Squid able to find out which users are using Squid. You may configure squid to use user authentication or you have to install `identd` on all PCs to support this feature. Please read the Squid documentation for more information and do not forget to use `acl foo ident REQUIRED`.

8.2.4 Policy based on Domain Name

Internet access policies can also be based on the domain name of a PC. In this example, the PCs in the dealingroom have a unique domain name that can be used to define a policy.

Let's assume that the dealingroom PCs have a name like `pc31.dealingroom.bank.com`, and then the `src` definition for the `dealingroom` group is as follows:

```
src dealingroom {
  domain dealingroom.bank.com
}
```

NOTE: To use this feature, the reverse name lookup must be enabled in `squid.conf`:

```
log_fqdn on
```

8.3 Multiple ACLs

In case that there are exceptions for (groups) of users, multiple ACLs will be used (see also the examples for a dealingroom in the previous sections).

The order of ACLs is important since the first ACL that matches a URL/user/IP will be used. The following example demonstrates this. Suppose there is an administration PC that should have access to all websites and that the PC has a fixed IP address: `10.2.3.4`. The configuration file should have the ACL for the administration PC *in front of* the ACL `allSystems`. Otherwise, the administration PC is considered part of `allSystems`!

```
src adminpc {
  ip 10.2.3.4
}
src allSystems {
  ip 10.0.0.0/8
}
```

URLfilterDB

```
acl {
  adminpc {
    pass any
  }
  allSystems {
    pass !security !adult !chat !dating !entertain !news !webmail any
  }
  default {
    pass none
    redirect http://www.urlfiterdb.com/cgi-bin/URLblocked.pl?...
  }
}
```

Note that the extra `redirect` statement is required because the pseudo-category 'none' is used.

8.4 Whitelisting

Whitelisting is used in case that users are only allowed to visit a predefined set of websites. In this case, the ACL for `allSystems` contains the categories *alwaysallow* and *none*. The ACL in the configuration file looks like this:

```
acl {
  allSystems {
    pass alwaysallow none
    # the following redirect is for the pseudo category 'none'
    redirect http://www.urlfiterdb.com/cgi-bin/URLblocked.pl?...
  }
  default {
    pass none
    redirect http://www.urlfiterdb.com/cgi-bin/URLblocked.pl?...
  }
}
```

Note that the extra `redirect` statement is required because the pseudo-category 'none' is used.

8.5 Extended Logging

`ufdbGuard` has 2 options for extended logging. The keyword `logblock` followed by `on` or `off` tells `ufdbGuard` whether to register blocked URLs in its logfile. The keyword `logall` followed by `on` or `off` tells `ufdbGuard` whether to register *all* URL verifications in its logfile.

Warning: `logall on` requires many resources and has a performance impact on `ufdbGuard`. Use this option with care and for short periods only.

8.6 Logfile Rotation

`ufdbGuard` rotates its logfile automatically whenever it grows beyond 200 MB. When the logfile is rotated, the file `ufdbguardd.log` to `ufdbguard.log.1` and recreates `ufdbguardd.log`. A maximum of 8 logfiles are kept.

On receipt of the `USR1` signal, `ufdbguardd` also rotates the logfile. The most convenient way to do this is to use `/etc/init.d/ufdb rotatelog`, which takes care of sending the `USR1` signal to the right process.

URLfilterDB

The maximum logfile size is configurable with a parameter in the configuration file. The following line sets the maximum size to 50 MB.

```
max-logfile-size 50000000
```

9 Performance Tuning

9.1 Squid performance

The following is recommended to improve the performance of Squid:

- use Linux 2.6 and use the `noatime` mount option for the file system with the cache. If `reiserfs` is used, also use the `notail` mount option.
- use squid 2.6 and configure it to use `epoll` (`--enable-epoll`).
- use a proper number of open files during the configuration phase of squid because at the time that configure is run, it is determined what the maximum number of open files can be during the runtime of squid. You may need to use the `ulimit` command and configure the kernel. There is no official guideline for a proper value for the number of open files, but you may find that $100+2.5*NUSERS$ with a minimum of 1024, is appropriate.
- use a moderately sized cache; a very large cache might have a slightly larger cache hit ratio, but the housekeeping of the cache requires more memory and CPU resources. Note that a larger number of cached objects also requires more physical memory for the index.
- use more than one disk for the cache, use one cache directory per disk and do not stripe.
- on multi-processor systems based on Intel, disable hyperthreading; use only real CPU cores.
- do not forget to run “`squid -k rotate`” from cron as user `squid` because it also does important housekeeping of the cache.
- visit <http://wiki.squid-cache.org> for more tips.

9.2 Linux 2.6 performance

For large sites that have both Squid and URLfilterDB running on a system with Linux 2.6 and with 2 or more real CPU cores, additional performance can be gained by optimizing the CPU cache efficiency. Since squid is a very CPU intensive application, we reserve CPU 0 for squid. By configuring `ufdbGuard` to use the other CPUs, squid will automatically have exclusive access to CPU 0. On a system with Xeon processors and hyperthreading enabled, we reserve CPUs 0 and 1 for Squid to give it exclusive use of the memory cache of the Xeon processor. The remaining CPUs can be used by `ufdbguardd` and the process is bound to these remaining CPUs by including the `cpus` keyword in the `ufdbGuard.conf` file.

By separating the processes over different CPUs with their own memory caches, the caches are used in an optimal way.

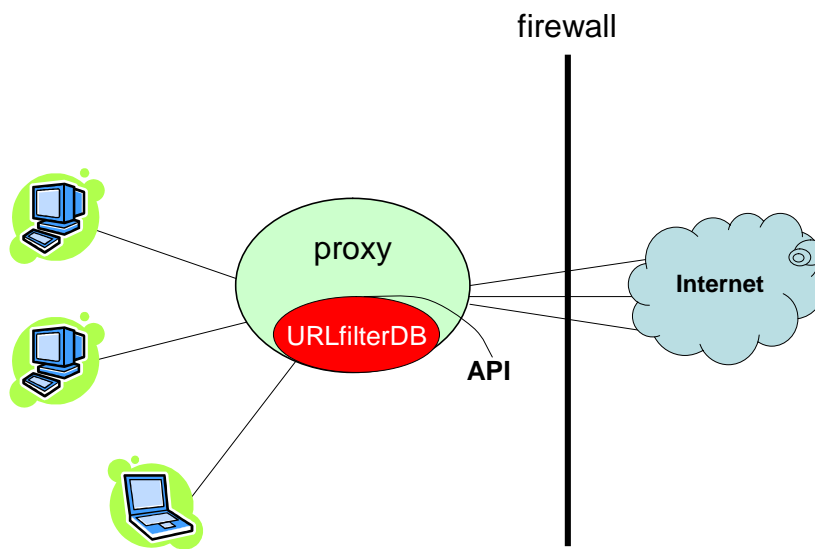
URLfilterDB

<i>system</i>	<i>for optimal performance, use</i>
2 simple CPUs	cpus 1
2 Xeon CPUs without hyperthreading	cpus 1
2 Xeon CPUs with hyperthreading	cpus 2,3
4 simple CPUs	cpus 1,2,3
1 dual core	cpus 1
2 dual core	cpus 2,3
1 quad core	cpus 2,3

10 Integration with 3rd Party Products

ufdbGuard also has an API that allows it to be easily integrated with 3rd party products. With the use of the API, you have access to the core functions of ufdbGuard and can perform URL verifications from any program written in C with the same speed: 50,000 URL verifications per second on an Intel 2.8 GHz CPU (Intel Xeon with 512 KB cache). On a more recent Intel CPU the performance reaches easily 50,000 URL verifications per second (Intel Core 2 Duo with 4 MB cache at 2.4 GHz).

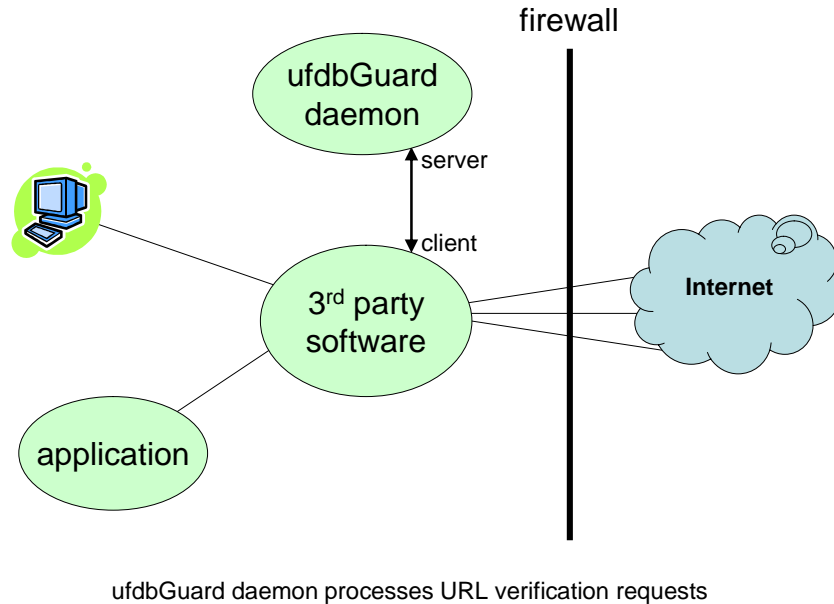
The following diagram gives a simplified overview of how a 3rd party product could operate. In this example, a proxy is linked with the URLfilterDB library for extremely fast URL verifications due to the lack of any interprocess communication overhead.



the URLfilterDB library is part of the web proxy

URLfilterDB

Alternatively, the next diagram shows any 3rd party can copy the simple code of ufdbgclient and incorporate this in 3rd party software and act as a client to the ufdbguardd daemon.



Please contact the support desk for additional information and licensing.

11 Frequently Asked Questions

ufdbGuard does not block

The most common reasons are either a configuration error (check ufdbguardd.log) or that the daemon is not active (verify with `ps -ef | grep ufdbguardd`).

Another common reason is that your trial license has expired. Verify the log file for warning messages.

I want to allow access to site myfavoritesite.com

See section 7.1. Do not forget to run ufdbgGenTable after each change.

What does it cost?

The URL filter software is free. Prices for a subscription to the URL database are on the website and you can request a quote online. The prices are in EURO and we use a fair exchange rate for countries that prefer to be billed in US dollars.

Can I use ufdbgGuard with a free database?

The ufdbgGuard software is free and can be used with free URL databases if the database is in .ufdb format or in an ASCII format.

URLfilterDB

Which URLs are in the database?

URLfilterDB does not disclose the content of their database. You are encouraged to test the database yourself to find out its effectiveness yourself. Please refer to section 11.1 for a more detailed description of what you may find in each URL category.

Any other question

Please contact the support staff at support@urlfilterdb.com to answer any other question.

11.1 URL Categories

The URL database of URLfilterDB uses the following URL categories.

Ads

Advertisements, traffic trackers and web page counters.

Proxies

Sites that can be used to download content of other sites. Proxies are used in an attempt to circumvent a URL filter. URL rewriting sites.

Adult

Sites suitable for adults only (not only sexual content).

Warez

Illegal software, illegal software codes, hacker's sites.

Toolbars

A toolbar is an extension to a web browser that may violate your privacy or make private files public.

Illegal

Illegal acts

Violence

Violent behavior, aggressive sales of arms.

Gambling

Gambling.

Drugs

Hard drugs, soft drugs, penis enlargement drugs, medicines.

Webmail

Private email accessible with a web browser.

Dating

Love, dating, romantic poetry, and friend sites.

URLfilterDB

Chat

IRC and chat.

Forum

Sites where people exchange non-business information in a forum.

Private

Blogs and sites of private persons, private web disk, and private file stores.

Audio-Video

Audio and video streams (a TV station is usually in the entertainment category).

Sports

Sports.

Finance

Banks, insurance companies, stock markets, stock brokers

Jobs

Job applications

Games

Games and sites about games

Entertainment

Entertainment, lifestyle, hobby, arts, museums, food, fashion, electronic cards, magazines, horoscopes, desktop wallpapers, clip art, photos, portals, events, fan sites, meditation, baby-related, child sites, file sharing, religion, non-business private interest.

Shops

Shops, price comparisons, and auctions aimed at consumers (b2b is excluded).

Travel

Travel agencies, airliners, tourism sites, hotels, holiday resorts.

News

News and opinions.

Checked

URLs that are verified not to be part of any category and hence always allowed by the URL filter. This “hidden” category is used to track uncategorized URLs.

URLs may be part of 2 categories, e.g. www.usatoday.com is news while www.usatoday.com/sport is both news and sport. The nature of the content is more important than the strict definition, so an advertisement with a nude person is classified as adult rather than advertisement, and a forum about games is classified as games rather than a forum.

URLfilterDB

The general impression is also taken into account when a site is categorized. For example, most buyers at ebay.com are consumers rather than business users and therefore ebay.com is considered a shop for consumers and part of the shops category.

12 More Information

More information can be found on the internet at the following addresses.

URLfilterDB	www.urlfilterdb.com
Squid	www.squid-cache.org
Redhat Linux	www.redhat.com
SuSE Linux	www.suse.com
wget	www.gnu.org/software/wget/wget.html
Bzip	www.bzip.org
Lex, yacc	directory.fsf.org

13 Product Comparison

This section contains a comparison of internet filtering methods.

3 filtering methods block unwanted web content:

1. **content scanning:** block web pages if it contains (a set of) "bad" words
2. **artificial intelligence:** an improved version of *content scanning*
3. **blacklist:** block sites based on a list of categorized websites

The following criteria are used for the comparison of the filtering methods:

- user experience: the method must be sufficiently fast for the individual user.
- wrong blocking I: a site about *breast cancer* should not be blocked as if it was a site about sex. This is called *overblocking* since the site should not have been blocked.
- wrong blocking II: a site with sexual content should be blocked. If the filter fails, it is called *underblocking*.
- block https: the *https* protocol is an encrypted protocol intended for security and privacy. Because the protocol uses encryption, the content cannot be scanned for words or phrases.
- infrastructure costs: the components are bandwidth usage and computing power.

13.1 Method A: Content Scanning

When web pages are scanned for content, they are first downloaded which costs time and bandwidth. Then the content is scanned for bad words like "breast", "s*x", "s*ck", "f*ck", etc. Depending on the vendor, one or more words trigger the blocking mechanism and unwanted content can be blocked. The theory looks nice...

URLfilterDB

In practice, however, many sites are blocked because of word combinations like "I don't like sex", "breast cancer" etc. which is called *overblocking*. On the other hand, sites with sexual content that only have pictures (text can also appear in a picture), are not blocked because they don't contain any of the bad words, which is called *underblocking*.

The time that it takes to scan and guess the type of content of a web page varies per page (some pages on the internet are very large). This method is sufficiently fast for an individual user. However, for 250 or more users, a very fast computer system is required for the proxy server.

13.2 Method B: Artificial Intelligence

When web pages are blocked based on artificial intelligence (AI), they are also downloaded first and then scanned, so this method also consumes bandwidth and time for the download process. The various AI methods are more complex versions of method A. To reduce the failures caused by underblocking and overblocking, all words in the web page are rated and some word combinations are rated. Some products try to find out if a picture contains nudity by looking at colors and claim a high level of correctness. This improvement of correctness of blocking comes with a large cost: much CPU power. Therefore, for 100 or more users, a very fast computer system is required for the proxy server.

13.3 Method C: Blacklist

When web pages are blocked with the use of a blacklist, they are not downloaded to make a decision about blocking it or not. Instead, the URL filter module of the proxy server makes a simple decision based on the URL: www.sex.com is blocked and www.google.com is not. The URL filter makes this decision based on a URL database that is often referred to as a blacklist.

This method is fast since blocked sites are not downloaded and the URL filter `ufdbGuard` does 25,000 URL verifications/sec on a single 2.8 GHz Intel Xeon CPU with 512 KB cache. More modern CPUs like an Intel dual-core Conroe CPU with 4 MB cache have a performance of nearly 50,000 URL verifications/sec.

`ufdbGuard` also features dynamic detection of https proxy tunnels and hence increases the security on your network.

13.4 Comparison of Methods

The next table shows the pros and cons of all methods.

method	user experience	correct blocking	block https	scalability beyond 500 users	infrastructure costs
content scanning	±	-	--	-	±
artificial intelligence	±	±	--	--	-
blacklist (<code>ufdbGuard</code>)	++	+	++	++	+

No method is perfect and will never be. This is due to the large amount of websites on the internet that simply cannot be rated and categorized perfectly. We believe that this imperfection is not an issue as long as a method blocks 99% of unwanted content and does not block wanted content. `ufdbGuard` has a feature to recognize URLs that are not yet part of the URL database and uploads these URLs to be included in the next day's database.