Reference Manual

ufdbGuard for BIND named
version 1.0.3

Filter Plugin for the BIND named DNS Server

Table of Contents

1 Introduction

The ufdbGuard plugin for the BIND named DNS server is an internet access filter. With the ufdbguard plugin the DNS administrator can set a filtering policy to filter adult content, malware and other URL categories. The filter can be used to effectively block categories of websites and applications that use domainnames.

There are products, most notably web browsers, that may use DNS-over-HTTPS (DoH) which may circumvent the DNS-based filter. The URL database of URLfilterDB include a URL category dnsoverhttps to block the domainnames of these services. The database also include a plaintext file of IPv4 and IPv6 addresses of DoH services which can be used in a firewall.

This product is intended to be used by service providers, mid-size to large organisations and system integrators. Interested parties may register as a trial user at www.urlfilterdb.com to receive a 60-day trial license for the ufdbguard named plugin and the URL database.

1.1 Latest Major Enhancements

The ufdbGuard plugin for named is built on top of the latest version of the ufdbGuard API and hence includes the most recent features.

1.2 Support

The support conditions are negotiated and offered in a quote of URLfilterDB.

The free support conditions for each client of the ufdbGuard plugin for named is

- email support for questions related to the ufdbGuard plugin and the URL database. Answers are on the same business day or morning of the next business day.

Organisations with a large number of users can negotiate

- virtual meetings on business days with support staff. Virtual meetings must be requested at least 2 business days in advance.

Resellers may also have

- 24x7 access to the partner portal where emergency changes to the URL database can be made,
- 12x5 phone support.

On the partner portal of URLfilterDB a client can modify the classification of URLs. This feature is only to be used in emergencies and should not be abused. URLfilterDB reserves the right to undo modifications of a client when it finds that a modification is not according its own classification guidelines. In this case a client may consider using a user-defined blacklist or whitelist.

1.3 Feedback

We welcome feedback from those who test our software. Feel free to send your questions and feedback to the support desk: support@urlfilterdb.com.

2 Prerequisites

The ufdbGuard plugin needs a supported version of BIND named. At this moment the supported versions are 9.16.45 and 9.18.21[1] on RHEL/CentOS/Rocky Linux/AlmaLinux versions 8 and 9, and Ubuntu versions 20.04 LTS and 22.04 LTS. Consult the support desk for support of other versions of named on other Linux or FreeBSD platforms.

The URL database internals on Intel/AMD systems use AVX2 SIMD instructions to improve performance by 20%. This means that the CPU must be an Intel Haswell (introduced in June 2013) or more recent, or AMD EPYC/Ryzen (introduced in March 2017) or more recent. Consult the support desk for support for other CPU platforms or a non-AVX2 version of the ufdbguard plugin. Caveat: virtual machines may not always support AVX2 and BMI1 CPU flags which are both required by the AVX2 version of the plugin.

`named` must be configured as a DNS resolver.

The ufdbGuard plugin for named needs 10 GiB disk space and 4-8 GiB memory. In case that the default log file size is changed, the required disk space may increase. During database downloads at least 1 GiB free file system space is required in `TMPDIR` (default `/tmp`).

Besides this, basic commands like `sh`, `tar`, `gzip`, `socat`, `curl` etc. are used to send commands to the plugin, and download and unpack the URL database.

The URL database is regularly downloaded via the default route to the internet or via a web proxy. The files with uncategorised URLs and statistics are uploaded via the default route to the internet. Both processes connect to `updates.urlfilterdb.com` using port 443. The webservers for `updates.urlfilterdb.com` currently have 2 IPv4 and 2 IPv6 addresses.

3 How it works

3.1 The plugin

The ufdbguard plugin for named is a shared object file that the DNS server dynamically loads. The configuration file `named.conf` contains one or more `plugin query` statements to load the plugin.

The ufdbguard plugin can block all query types except for query type PTR since this query does not have a domainname. All other query types (including ANY) are refused if the queried domainname is in a blocked URL category. When refused, the DNS query return code, or *rcode*, is depending on a configurable parameter set to NXDOMAIN or REFUSED and the DNS client does not receive any records. This mechanism can block access to a website and any application that uses a blocked domainname *if the clients are unable to use other DNS servers* that do not filter. It is therefore required to also block access to other DNS servers to have a fully functioning internet filter system.

The ufdbguard filter plugin configuration is included in the named configuration file `named.conf`. A universal filter policy can be defined for all users or multiple filter policies can be defined where each named view has its own filter policy. See section 5 for a detailed explanation on how to configure the ufdbguard plugin.

[1] the plugin supports 9.16.40 or newer and 9.18.14 or newer. URLfilterDB has a policy to provide a compatible version of the plugin within 24 hours after each release of a patch of BIND named.

The plugin creates one extra thread named “ufdbguard-plugn” which is a housekeeping thread that listens to a command socket and performs database reloads, create upload files, displays plugin status etc. The housekeeper thread is mostly idle. The plugin becomes an integral part of named and the actual filtering is done inside the worker threads of named.

The plugin filters at a late stage in the process of name resolving and silently skips filtering logic if named finds an error with the domainname lookup (e.g. NXDOMAIN, SERVFAIL etc.)

The plugin has very low performance impact since it can do 550,000 domainname lookups/second on a single thread of a modern CPU. During a URL database reload DNS queries are interrupted for a tiny fraction of a second, almost always less than 0.0001 second on a modern CPU.

The URL database is a collection of URL tables and DNS administrators can add user-defined URL tables and extend the URL database. User-defined whitelist categories can be created and configured so it is possible to prevent blocking a set of user-defined domainnames.

Note that the URL database is *not* a relational database; it is read-only and is only suitable to query the URL categories of domainnames. To achieve high query performance, the entire URL database is loaded into memory.

3.2 Applications

The most popular application that does DNS lookups is the web browser. Web browsers that use a DNS server that filters some URL categories may receive valid DNS responses with rcode set to NXDOMAIN or REFUSED (configurable with parameter `block-reply-code`). At this moment all tested web browsers display an error message where it tells the end user that the DNS server sent an NXDOMAIN reply which is a sign for the end user that access to the blocked website is not possible, but the browser does not display any further explanation of the reason why.

Of course all other applications that do domainname lookups which are blocked by the DNS server also receive a DNS reply with rcode NXDOMAIN or REFUSED.

3.3 Access to 3rd party DNS Servers

Browsers and other applications may want to circumvent a DNS server that filters requests. Therefore it is recommended that

1. all DNS servers within an organization are configured for filtering
2. the firewall is configured to block access to DNS servers outside the organization. The URL database contains a text file with IP addresses of known DNS servers that can be used for circumvention.
3. the URL category *dnsoverhttps* is configured to be blocked by the plugin.

4 Processes

The ufdbguard plugin is integral part of named and has no daemon processes of its own. The plugin creates an extra thread inside named which can be seen with the `ps` command and the appropriate flags. This thread is referred to as *the housekeeper*.

4.1 named

The plugin is part of the named process and inside the worker threads of named the actual filtering takes place.

The process of reloading the URL database does not disturb DNS server performance and is done in 3 steps.

1. the new URL database is loaded into memory – the currently loaded URL database is still being used.
2. the new database is activated. This usually takes less than 100 *micro*seconds. In this period on a very busy name server some queries may not be filtered, i.e. the DNS server sends a few unfiltered responses.
3. the previous in-memory database is deallocated.

In step 2 it is possible that on high volume DNS servers a few responses are not filtered. This is by design since sustained performance is considered more important than a few responses that with a high probability were allowed anyway, are allowed without being filtered.

4.2 ufdbBindUpdate

`ufdbBindUpdate` is a script that downloads a fresh copy of the URL database and sends a *reload* command to the command socket of the plugin.

The `ufdbBindUpdate` script accepts the following command line option:

`-v` be verbose.

It is recommended to have a crontab job to download a fresh URL database 1-4 times per day. Note that the plugin refuses to load the URL database if it is older than 28 days.

`ufdbBindUpdate` uses the binary executable `ufdbBinddl` which does the actual download. `ufdbBinddl` is used instead of `wget` or `curl` since it has support for an encrypted password.

See section 5.6 for more information.

4.3 ufdbBindUpload

The ufdbguard plugin maintains a list of uncategorised URLs which, after being uploaded to `updates.urlfilterdb.com`, are analysed and used to extend the URL database. The ufdbguard plugin can produce *upload files*. The upload files can be uploaded with the script `ufdbBindUpload`.

Upload files are human-readable ASCII files that contain statistics like total number of domainname filter queries, number of occurrences of NXDOMAIN, number of blocked queries, CPU type etc. and a list of domainnames that are uncategorised (i.e. do not yet exist in the URL database). URLfilterDB explicitly states that it has no interest in having in-depth knowledge about which domainnames are processed by the DNS server and the upload files do not include such data.

It is recommended to have a crontab job that runs `ufdbBindUpload` 10-30 minutes after the crontab job(s) for `ufdbBindUpdate`.

On sites with very large volumes, it is recommended to create an upload file every 2 hours.

4.4 Default crontab

The plugin package installs a default crontab in `/etc/cron.d/ufdbguard-bind` where there is a schedule for `ufdbBindUpdate` and `ufdbBindUpload`. Its is recommended to review the cron job schedule.

5 Software Installation

5.1 Choose a named version

The internal data structures of bind vary from version to version and the ufdbguard plugin accesses these data structures. Therefore it is of critical importance that the correct plugin version is used for any particular version of named. Since patches may introduce data structure changes the plugin always verifies that the plugin is suitable for the version of named that is in use.

When the plugin is loaded it does a version verification. The plugin fails to load if the version of named is not exactly the same as the last part of the plugin version. The plugin version always has six numbers where the the first three refer to the plugin and the last three refer to the named version. So the first release of the plugin has version number 1.0.0.9.18.14 which is compatible with named 9.18.14. The plugin version 1.0.3 is available for BIND named versions 9.16.45+ and 9.18.21+. The DNS administrator has to choose which version of named will be installed and also install a compatible plugin.

The version of named can be retrieved with the command `named -v`. Note that only ISC bind is supported and distro versions of named are *not* supported since distros do not include all fixes and do not use a unique named version.

Within a few hours after each patch release of ISC bind, URLfilterDB creates new packages for the plugin and makes them available on the software repositories.

5.2 Upgrading from a Previous Version

A previous version can simply be overwritten by a new version. When an upgrade is performed, it is recommended to perform all installation steps including the last step, retrieval of database update in section 5.6.

NOTE: when the plugin version (1.x.x) changes, it is necessary to edit named.conf and edit the plugin statement to reflect the new plugin version.

5.3 Install Software

The installation of the software must be done as `root`.

The software can be downloaded from the repositories `rhel-repo.urlfilterdb.com` and `ubuntu-repo.urlfilterdb.com`.

BIND named must be installed before the ufdbGuard plugin can be installed. As explained earlier the plugin is only compatible with a version of BIND named that is compiled from unmodified sources as released by ISC. There are three options to install a compatible version of BIND named.

1. install the package released by ISC on COPR or Launchpad.
   See https://www.isc.org/blogs/bind-9-packages/, or
2. install the package from the software repository of URLfilterDB, or

3. download the sources, compile and install them.

5.3.1 Ubuntu

On Ubuntu 20.04 (focal) and 22.04 (jammy) the package repository can be added with the following commands.
NOTE: replace ***DIST*** by `focal` or `jammy`.

```
$ sudo su -
# cd /etc/apt/trusted.gpg.d
# wget https://ubuntu-repo.urlfilterdb.com/urlfilterdb-pubkey.asc
# cd /etc/apt/sources.list.d
# echo "deb https://ubuntu-repo.urlfilterdb.com/ DIST main" > urlfilterdb.list
```

Optionally install BIND named 9.18 or 9.16 (replace `918` by `916` below) from the URLfilterDB repository:

```
# apt update
# apt install urlfilterdb-isc-bind-918
```

Install the ufdbGuard plugin for named 9.18 or 9.16 (replace `918` by `916` below):

```
# apt install ufdbguard-bind918
```

5.3.2 RHEL

On RHEL and compatible distros the package repository can be added with the following commands.
NOTE: replace ***N*** by `8` or `9` in the following commands.

```
$ su -
# dnf config-manager --add-repo http://rhel-repo.urlfilterdb.com/rhel-N.repo
# /etc/pki/rpm-gpg
# wget http://rhel-repo.urlfilterdb.com/urlfilterdb-pubkey.asc
```

The EPEL repository is required. If EPEL is not already used, enable it with the following command.

```
# dnf -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-N.noarch.rpm
```

Optionally install BIND named 9.18 or 9.16 (replace `918` by `916` below) from the URLfilterDB repository:

```
# dnf install urlfilterdb-isc-bind-918
```

Install the ufdbGuard plugin for named 9.18 or 9.16 (replace `918` by `916` below):

```
# dnf install ufdbguard-bind918
```

5.4 File Locations

The ufdbGuard plugin for BIND files and its locations on Linux systems are in the following table where *TOP* is either `/opt/urlfilterdb/plugin-bind-916` or `/opt/urlfilterdb/plugin-bind-918`. To not interfere with other named packages, almost all files are installed under *TOP*.

plugin	*TOP*/lib/plugin-*version*.so
ufdbBindUpdate	*TOP*/sbin
ufdbBindUpload	*TOP*/sbin
ufdbBindGenTable	*TOP*/sbin
ufdbguard command socket	*TOP*/socket/commands
ufdbguard-bind.conf configuration file	*TOP*/etc
urlfilterdb-isc-bind-918.service	/lib/systemd/system
named (service config)	*TOP*/etc/sysconfig or *TOP*/etc/default
ufdbguard-bind	/etc/cron.d
ufdbguard.clientid	/etc/ufdbguard.clientid
URL database files	*TOP*/db
license status file	*TOP*/db/license-status
log files	*TOP*/log
upload files	*TOP*/upload

ufdbguard-bind.conf: this configuration file is used by ufdbBindUpdate, ufdbBindUpload and ufdbBinddl[2].

log files: log files are rotated and the maximum size of an individual log file is set with the option max-logfile-size (default is 1 GB). The maximum number of log files (including perpetual rotation) that the plugin creates is 9.

upload files: the plugin generates files with statistics and uncategorised URLs when it receives the command to do so via the command socket. The DNS administrator is responsible for scheduling with little delay the upload of these files to updates.urlfilterdb.com using the ufdbBindUpload script. The ufdbguard-bind cron template in /etc/cron.d contains a reasonable default.

ufdbguard command socket: the filename is a configurable parameter of the plugin but can be set only once and not changed during a reload. If one chooses to change the filename, bind must be restarted or follow this procedure: remove all plugins from named.conf – bind reconfig – add plugins to named.conf – bind reconfig.

ufdbguard.clientid: resellers that use a generic username for the license must include a unique client identifier in the file ufdbguard.clientid. The first line must contain the unique identifier without quotes or trailing spaces.

The ufdbguard plugin will be part of the named process and the URL database must be readable by the user that bind uses (user bind on Ubuntu and user named on RHEL). The directories for the log files and the upload files that the plugin produces must be owned by the same user.

[2] ufdbBinddl is a helper program that performs the actual download of the URL database from the servers of URLfilterDB.

5.5 Security

5.5.1 Linux

If the system uses selinux or apparmor, `named` must be given access to the files and directories listed in section 5.4. Most file locations are configurable with parameters so one should be careful to specify all correct locations in configuration files of selinux and apparmor.

5.5.2 User Account

The ufdbguard plugin will be part of the named process and the URL database must be readable by the user that named uses (`bind` or `named`). The directories for the log files and the upload files that the plugin produces must be owned by the same user.

5.6 Get Daily Updates

The script `ufdbBindUpdate` takes care of downloading a new version of the URL database. It is the responsibility of the DNS administrator to integrate `ufdbBindUpdate` in its processes. After `ufdbBindUpdate` has downloaded a fresh URL database, the housekeeper thread receives a reload command via the command socket to load the new URL database.

During the reload the housekeeper creates an upload file which must be uploaded with `ufdbBindUpload`.

The `ufdbBindUpdate` script needs the username and password that you received when the (trial) license was received which can be defined in a system configuration file which resides in *`TOP`*`/etc`.

```
$ vi TOP/etc/ufdbguard-bind.conf
...
DOWNLOAD_USER=lic99999
DOWNLOAD_PASSWORD=aa22bbppxx
```

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

Test the `ufdbBindUpdate` script with the verbose option:

```
$ ufdbBindUpdate -v
```

The output should be similar to:

```
reading parameters from /etc/sysconfig/ufdbguard-bind ...
ufdbBindDBdl: going to download
https://updates.urlfilterdb.com/licensed/databases/ufdbguard-dns/3.1/blacklists-latest.
tar.gz to /tmp/urlfilterdb-latest.tar.gz
ufdbBindDBdl: downloaded URL database is stored in /tmp/urlfilterdb-latest.tar.gz
(430475815 bytes)
ufdbBindDBdl: downloaded license status is stored in
/opt/urlfilterdb/plugin-bind-918/db/3.1/license-status (30 bytes)
-rw-rw-r-- 1 bind bind 430475815 Apr 10 16:36 /tmp/urlfilterdb-latest.tar.gz
/tmp/urlfilterdb-latest.tar.gz successfully unpacked in /opt/urlfilterdb/plugin-bind-
918/db
license status: OK: expiry date is 01-01-2028
Uncategorised URLs saved in
/opt/urlfilterdb/plugin-bind-918/upload/ufdb.upload.1108586.0000001
URL database reload initiated
```

5.6.1 ufdbBindUpdate

`ufdbBindUpdate` downloads the URL database and needs access to the servers of URLfilterDB. Firewall rules may need to be modified to provide access to all IP addresses of `updates.urlfilterdb.com`.

A proxy can be used to download the URL database: modify *`TOP`*`/etc/ufdbguard-bind` and assign appropriate values to the variables `http_proxy` and `https_proxy`. If the proxy requires authentication, also assign appropriate values to `PROXY_USER` and `PROXY_PASSWORD`.

The URL database resides in a compressed tar file which is downloaded to the directory `$TMPDIR` (default is `/tmp`) and then untarred in the directory `$BLACKLIST_DIR`. Both variables can be set in *`TOP`*`/etc/ufdbguard-bind`.

5.6.2 Exit Codes of ufdbBindUpdate

The `ufdbBindUpdate` script has exit code 0 when the download is successful and a non-zero code in case of an issue. In case of a non-zero exit code, the script prints the exit code and explanation on stdout. In case of an error, the script also logs a message to the system log. The script uses the exit codes in the following table.

exit code	*explanation*
0	all is well
1	ufdbBindDBdl option error
2	ufdbBindDBdl cannot retrieve password from <file>
3	ufdbBindDBdl cannot create database file <file>
4	ufdbBindDBdl TLS/SSL or download error
5	ufdbBindDBdl could not sync database to disk
6	downloaded database is too small
7	ufdbBindDBdl cannot create license status file <file>
8	ufdbBindDBdl cannot download license status
9	downloaded license status file is too small
11	The configuration file *`TOP`*`/etc/ufdbguard-bind` cannot be read
12	The variable `DOWNLOAD_USER` is not set.
13	The download failed with unknown reason
14	The temporary directory (`TMPDIR`) does not exist or is not writable
15	The URL database top directory (`BLACKLIST_DIR`) does not exist or is not writable
16	Cannot remove temporary download file – check permissions of directory `$TMPDIR`

exit code	*explanation*
17	The variable `PLUGIN_COMMAND_SOCKET` is not set
18	The plugin command socket does not exist or is not accessible. Verify if named is running
19	The plugin did not respond well to the `reload` command
22	license expiration warning: less than 2 months to renew license
23	license expired: a license renewal is required immediately
61-80	exit code of `tar` + 60. `tar` unpacks the downloaded URL database. There may be an issue with file system space.
127	Could not find ufdbBindDBdl in PATH

In case of an error, it is advised to run `ufdbBindUpdate -v` from the command line to have more feedback about what is going wrong.

5.6.3 License Status

The content of `license-status` file (see 5.4 for the location) is one of:

OK: expiry date is <date>
WARNING: license expires in <n> days
EXPIRED: license expired <n> day(s) ago
EXPIRED: license terminated since <date>

When the license is two months before its termination date, the `ufdbBindUpdate` script also warns with a different exit code (see previous section).

5.7 Upload Files

Upload files contain uncategorised URLs and statistics. Whenever the housekeeper receives a reload command, it also generates an upload file.

On servers with a very high number of queries it is recommended to create upload files more frequently than the frequency of database reloads. This can be done by sending the command 'upload' to the command socket of the housekeeper.

The `ufdbBindUpload` script uploads all upload files that it finds. ufdbBindUpload can use a proxy. It uses the same configuration file as `ufdbBindUpdate`: *`TOP`*`/etc/ufdbguard-bind`.

6 Configuration

The ufdbguard plugin receives configuration parameters from bind/named which are inside the `plugin query` block.

The bind configuration file, usually named.conf, may contain one or more plugin statements. In case that all users of the DNS server use the same filter policy, there must be one plugin query statement at the

global level. On the other hand, if different filter policies are used for different groups of clients, bind views can be used where each view has an optional query plugin.

Example of a single policy for all users:

```
options {
  recursion  yes;
  ...
};
plugin query "/opt/urlfilterdb/plugin-bind-918/ufdbguard-1.0.0.so" {
  categories "adult malware dnsoverhttps";
};
```

Example of two policies for two groups of clients:

```
options {
  recursion  yes;
  ...
};
view "students" {
  match-clients     { 10.1.0.0/16; };
  max-cache-size    64m;
  ...
  # views can share a cache if they have equal cache policy parameters,
  # see attach-cache in the ARM.
  plugin query "/opt/urlfilterdb/plugin-bind-918/ufdbguard-1.0.0.so" {
    categories   "malware adult qmovies dnsoverhttps +proxies/teamviewer proxies";
  };
};
view "staff" {
  match-clients     { 10.0.0.0/8; 127.0.0.1; };
  max-cache-size    32m;
  ...
  plugin query "/opt/urlfilterdb/plugin-bind-918/ufdbguard-1.0.0.so" {
    categories  "+mywhitelist malware adult qmovies";
  };
};
```

6.1 Parameters

This section discusses all parameters that can be used to configure the ufdbguard plugin. Parameters of the plugin follow the same syntax as named parameters and should always be terminated with a semicolon. In the parameter section lines starting with a hash (#) are interpreted as comments.

Define which URL categories are to be blocked in the current view:

```
   categories  "list of categories";
```

The list of blocked categories is a quoted string where each category name is separated by whitespace. The list may contain user-defined categories. The default behavior is to block a category if it occurs in the list of categories. The list of categories can also contain one or more whitelists which prevent blocking domainnames if it occurs in the whitelist. A whitelist category must be prefixed with a plus sign (+). DNS administrators can create their own URL categories and mix them with the URL categories of the URL

database of URLfilterDB. See section 10 for a list of all URL categories. See section 8.1 on how to create a URL table.

The `categories` parameter is the only mandatory parameter. All other parameters are optional.

Define where the logfiles `ufdbguard.log[.N]` will be created.

```
log-directory "directory";
```

Define the maximum log file size. Must be in the range 2 MB – 10 GB. Default is 1 GB. When the log file reaches the maximum log file size, the log file will be rotated (rename ufdbguard.log to ufdbguard.log.1) where a maximum of 8 rotated log files are kept available. The size is an unsigned integer followed by an optional M (megabytes) or G (gigabytes).

```
max-logfile-size NNN[M|G];
```

Define the directory where the URL database resides:

```
database-directory "directory";
```

Define which DNS reply code to use in a query for a blocked domain (default is “NXDOMAIN”):

```
block-reply-code "NXDOMAIN|REFUSED";
```

Define the debug level of the ufdbguard plugin (default is `0`):

```
debug-level NNN;
```

Specify if all blocked domainnames must be logged (default is `no`):

```
log-block yes|no;
```

Specify if Linux may be hinted to use transparent hugepages using `madvise(2)`. Hugepages are beneficial for large data blobs (like URL tables) in memory and improve overall system performance. (default is `yes`):

```
madvise-hugepages yes;
```

Verify the current Linux configuration for transparent hugepages with the following command:

```
cat /sys/kernel/mm/transparent_hugepage/enabled
```

The default on most systems is “`always [madvise] never`” which is fine for madvise-hugepages.

Specify the directory where upload files are stored:

```
upload-file-directory "directory";
```

Specify the name of the UNIX socket where the plugin receives commands:

```
plugin-command-socket "socket-filename";
```

Specify if *this instance* of the plugin must be disabled:

```
disable yes|no;
```

Disabled plugin instances still have parameters verified for validity where any error is downgraded to a warning.

6.2 BIND Logging

The ufdbguard plugin writes almost all log messages to its own log file. However, during plugin initialisation, it uses the native bind log files. At this time, it uses the bind category “queries” for logging which one may configure in `named.conf` as follows.

```
logging {
  channel queries_log {
    file "queries.log" versions 5 size 5m;
    severity info;
    print-time yes;
  };

  channel default_log {
    file "default.log" versions 5 size 25m;
    severity dynamic;
    print-category yes;
    print-time yes;
  };

  category queries { queries_log; }; # some ufdbguard messages go here
  category default { default_log; };
};
```

6.3 Firewall

To prevent that the DNS filter is not circumvented, the firewall should block DoH and DoT requests from DNS clients.

DoH uses port 443 for which no generic firewall rule is possible for obvious reasons. The URL database contains a plaintext file (`iplist`) with IPv4 and IPv6 addresses of known DoH providers which can be used to block DoH requests from clients. It is recommended to create firewall rules for all provided IP addresses to prevent circumvention of the DNS filter.

DoT uses port 853 which can be blocked with a single firewall rule.

6.4 Tasks

On a DNS server system, the following tasks must be implemented.

1. run `ufdbBindUpdate` 1-4 times per 24 hours. This script downloads a new version of the URL database and sends a reload command to the ufdbguard.
2. run `ufdbBindUpload` 10-30 minutes after each time `ufdbBindUpdate` runs.

The plugin package installs a simple schedule for these tasks in `/etc/cron.d/ufdbguard-bind`.

7 Monitoring

The ufdbguard plugin can be monitored to be informed if the plugin operational.

The plugin writes informational messages, warnings (prefixed by “WARNING:”), errors (prefixed by "ERROR:") and critical errors (prefixed by "CRITICAL ERROR:") to the log files. Log files are automatically rotated (maximum 8 files with additional suffix `.1` ... `.8`) when the log file reaches the maximum configured log file size (see `maximum log file size` parameter).

Critical errors are written to the log file as well as the `syslog` with priority *alert*.

The health status of the plugin can be queried by sending `status` command on the command socket.

A monitoring agent can monitor the ufdbguard plugin log files, monitor the system log and regularly query the health status by sending a status command to the command socket of the plugin.

7.1 Plugin commands

The ufdbguard plugin listens to a UNIX command socket. `ufdbBindUpload` and `ufdbBindUpdate` send commands to this socket using the `socat` utility. The DNS administrator can also send commands to this socket.

BEWARE: the plugin accepts commands from a *single* source at any time, so the administrator should not hold the command socket open for an extended period of time since that prohibits the correct functioning of `ufdbBindUpload` and `ufdbBindUpdate`. The plugin closes any connection on the command socket when it is idle for 5 minutes.

A DNS administrator can send a single command to the command socket like this

`echo` *`command`* `| socat stdio unix-connect:`*`TOP`*`/socket/commands`

Or open an interactive session with

`socat stdio unix-connect:`*`TOP`*`/socket/commands`

The commands that the plugin supports are in the next table.

`status`	display the database status, license status and plugin statistics per view
`verbose-status`	display the database status and plugin statistics per view including number of domain matches for each category
`reload`	load a new URL database (`upload` is part of a `reload`). DNS queries are interrupted for a tiny fraction of a second, almost always less than 0.0001 second on an appropriately sized server.
`upload`	generate an upload file
`debug [0-3]`	without parameter this command shows the current debug level. When a numeric parameter is used, the debug level is set. The size of the debug information in the log file increases with the debug level.
`exit`	terminate interactive session

8 User-defined URL tables

In cases where additions or exceptions to the categories of URLfilterDB are desired, an administrator can define user-defined URL categories.

8.1 Create a URL Table

In this section a URL table called *alwaysallow* is created which contains a small set of URLs.

Edit the file that contains the URLs of the new category:

```
$ cd /opt/urlfilterdb/plugin-bind-918/db
$ mkdir alwaysallow
$ vi alwaysallow/domains
```

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

```
yourcompany.com
example.com
example.net
```

In case that many URLs have a leading `www.`, one may use the `-W` option to remove the `www.` prefix automatically.

The ufdbGuard plugin only uses proprietary database files, so generate a `.ufdb` database file from the ASCII files with `ufdbBindGenTable`:

```
$ cd /var/ufdbguard/db
$ ufdbBindGenTable -W -n -t alwaysallow -d alwaysallow/domains
```

The above command generates the file `alwaysallow/domains.ufdb` and should be invoked each time that the domains file is changed. The `-W` option removes the initial `www.` from all URLs which is highly recommended since the plugin always removes `www.` from URLs before it does a database query. The `-n` option specifies that the URL table does not have to be encrypted.

`ufdbBindGenTable` does a syntax check for the validity of the URLs.

Then configure the category by editing the `named.conf` file and include the category *alwaysallow*. See section 6.1 on how to configure the categories.

Finally send a reload command to the plugin to reload the URL database.

```
# echo reload | socat stdio unix-connect:/var/ufdbguard/commands
```

User-generated database files are accepted by the plugin even if they are older than 28 days. The maximum age requirement of 28 days only holds for URL tables generated by URLfilterDB.

User-generated database files must follow the database file naming convention and resides under the same top level database directory as all regular database files.

8.2 How URLs are Matched against the URL Database

The URL database lookup uses an algorithm to match a URL against the entries in the tables of the URL database. The algorithm uses the following logic. The following logic is based on functionality of the ufdbGuard API which is capable of dealing with full URLs. Since a DNS server only deals with domain names, the logic about matching a URL part is not applicable for the plugin.

1. If a URL table contains an entry with a domainname example.com it matches all URLs that contain example.com including subdomains, and matches URLs like example.com/foo.html, www.example.com and secure.example.com.
2. If a URL table contains an entry with a domainname with a "pipe tag", e.g. |.example.com, it matches all URLs that contain the domain example.com but not subdomains (*). This entry matches URLs like example.com/foo and www.example.com.

3. If a URL table contains an entry with a domainname and a path, e.g. "example.com/foobar" it matches all URLs that have the domain example.com (but not subdomains) and have a URL path that *starts* with the given URL path, so its matches www.example.com/foobar.html and does *not* match sub.example.com/foobar.
4. If a URL table contains an entry with a domainname, a path and a pipe tag, e.g. example.com/foobar|, it matches all URLs that have the domain example.com (but not subdomains) and have a URL path *equal* to the given path, so it matches www.example.com/foobar and does *not* match www.example.com/foobar.html.
5. If a URL table has an entry with parameters, the URL is matched if it contains all parameters of the table entry *in any order*. For example, if a table contains example.com/watch?p1=foo, the URLs www.example.com/watch?p1=foo and www.example.com/watch?p0=x&p1=foo&p2=bar are matched.

(*) "www" and "www0"..."www99" are not considered subdomains.

9 Performance Tuning

9.1 Use Hugepages

Total performance is increased by a few percent if the application can use hugepages.

All version of the ufdbguard plugin have support for database format 3.1. When this database format is used, the plugin may hint Linux to use transparent hugepages using the OS call `madvise(2)` for each table that occupies more than 1.6 MB memory. The default configuration of Linux is to transparently try to use hugepages when advised by `madvise(2)`.

10 URL Categories

The URL database of URLfilterDB uses the following URL categories. Some categories have subcategories. URLs in a subcategory are also in the parent category.

Ads

Websites with advertisements, traffic trackers, user behaviour analysis and web page counters.

P2P

P2P stands for point-to-point file sharing. The P2P category contains websites that can be used directly or indirectly to upload, download and share files. Most P2P sites have copies of movies, adult content, malware, warez and entertainment, and much of this content violates copyright.

Proxies

Sites that can be used to download content of other sites, URL rewriting sites and VPNs. Proxies are commonly used in an attempt to circumvent a URL filter and it is recommended to always block proxies.

Adult

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

Malware

Websites that contain or redirect to viruses or malware.
NOTE: this URL category is *not* a replacement for an antivirus tool.

Warez

Websites with illegal software, illegal software codes, hacker's sites, warez and cracks.

Toolbars

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

Illegal

Websites explaining how to perform Illegal activities.

Arms

Sites with firearms and toys that look like firearms.

Violence

Websites about violent behavior.

Gambling

Websites offering gambling opportunities.

Drugs

Websites about hard drugs.

Webmail

Email accessible with a web browser. Webmail of business sites is not included while webmail of ISPs is included in this category.

Dating

Websites about love, dating, romantic poetry, and friendship.

Chat

Websites to use IRC and chat. Subcategories exist for AIM, Ebuddy, Facebook Chat, Google Talk, MSN Messenger, Oovoo, Skype and Yahoo Chat.

AI Chat

Websites where people can talk with an AI bot.
Chatbots for education, business and customer support etc. are not included.

Forum

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

Private

Blogs and sites of private persons.

Webtv

sites with a audiovisual streams or television-like streams.

Webradio

sites with a music streams or radio-like streams.

Dailymotion

videos of dailymotion

Vimeo

videos of Vimeo

Youtube

videos of Youtube

Audio-Video

Audio and video streams.

Sports

Websites related to sports including sports sections of news sites, fans of sports, sites about actively doing a sport.

Finance

Websites of banks, insurance companies, stock markets and stock brokers.

Jobs

Websites about and for job applications.

Games

Websites to play games and information about gaming.

Entertainment

Entertainment, lifestyle, hobby, arts, museums, fashion, electronic cards, magazines, horoscopes, desktop wallpapers, clip art, photos, portals, events, fan sites, baby-related, child sites, other sites for interest of private persons that are not related to business.

Food

Websites of restaurants and sites with recipes. Fast food chains, however, are part of the category *shops*.

Religion

Websites related to any religion.

Shops

Websites with shops, price comparisons, and auctions aimed at consumers (b2b is excluded).

Travel

Websites about travel agencies, airliners, tourism sites, hotels, holiday resorts.

News

Websites providing news and opinions.

External Applications

Free web-based document editors, spreadsheet applications, desktops, groupware, etc. where "internal" documents can be stored on external servers.

Social Networks

Sites that focuses on building and reflecting of social networks or social relations among people. Subcategories exist for Badoo, Facebook and Twitter.

DNSoverHTTPS

IP addresses and domainnames of services for DNS lookups over HTTPS. This category has also a text file iplist with all IP addresses which can be used to configure a firewall.

Alternate DNS

There is a collection of alternative DNS systems with alternative TLDs like .coin, .libre, .bazar and .geek. See https://www.opennic.org for more information.
The DNS servers use ports 53, 443 and alternate ports like 8443, 5335 and 5353. This category has also a text file iplist with all IP addresses which can be used to configure a firewall.

Dynaddress

Websites with a dynamic IP address.

Extappl

Websites that deal with off-line documents and data.

Education

Websites of schools, universities and educational institutes.

Health

Websites of doctors, clinics, diseases and other health-related sites.

Qmovies

Websites which contain or link to movies with probable copyright infringement.

Searchengine

URLs used by search engines. Note that `google.com` is not included but `google.com/search` is.

Parkeddomain

Websites that are parked. Usually parked domains are expired domains or domains for sale and managed by domain brokers. Many parked sites have ads and some domain brokers use ad brokers that redirect users to adult, gambling or scam sites.

Checked

URLs that are verified by URLfilterDB not to be part of any other category. This category contains business sites, governmental sites and useful sites for the general public. This URL category is also used by the ufdbGuard API to track uncategorized URLs and should always be loaded.

The classification rules for URL database are based on *user intent* and classification is from the point of view of a business. So, a website that has a business use, is by default part of the category "checked". For example, access to a website to sell equipment for building constructors is in category "checked" and hence is not part of the category "shops". Also governmental sites, and all sites for basic human needs like electricity and water are in the URL category "checked".

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 (although may be included in both categories), and a forum about games is classified as games.

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.

URLs may be part of one or more categories, e.g. `www.usatoday.com` is *news* while `www.usatoday.com/sport` is both *news* and *sports*.

11 Error Messages

The error messages in the following sections may contain variable texts where `%s` represents character string, `%d` represents an integer and `<file>` represents a string with a filename.

Many errors listed in the next sections will never occur or will occur very rarely. To better understand error messages, it is always recommended to try to correlate errors of the ufdbguard plugin with errors and messages from the direct environment of the plugin and DNS server. The direct environment includes but is not limited to file storage, virtual machine management and network management.

11.1 Logfile Messages

The log file rotates automatically when it reaches the maximum size defined by the `max-logfile-size` parameter. When the logfile rotates, it is renamed to `ufdbguard.log.1` ... `ufdbguard.log.8` and a new file `ufdbguard.log` is created.

The logfile contains informational messages, errors and fatal errors.

With the exception of a few informational messages all lines in the logfile are prefixed with a date-time stamp followed by the message. An error message starts with the text "ERROR:" and a fatal error messages starts with the text "FATAL ERROR:". The message may contain multiple lines. In case of a multi-lines message, the second and following lines have an indentation.

Examples:

```
FATAL ERROR: *mandatory* parameter 'categories' not found
FATAL ERROR: parameter log-directory is not a quoted string
parameter <parameter> is set to "<value>"
```

11.2 Errors in the System Log

In case that the plugin has problems with its log file a message is sent to the system log.
Examples:

```
cannot open log file <file> - <reason>
ufdbguard plugin cannot rotate its log files !
lseek failed on log file: <reason>
```

11.3 Fatal Errors in the Download Logfile

The `ufdbBinUpdate` script calls the `ufdbBindDBdl` program to download a URL database. `ufdbBindDBdl` produces the logfile `ufdbBindDBdl.log` which may contain the following fatal error messages. The following messages are fatal errors without the date-time prefix.

```
password has less than 6 characters
cannot open password file <file>: <error>
<file> has no password
fwrite(%d,%d) returned %d - error writing output file: errno %d %s
cannot write to database file <file>: %s
the database file could not (entirely) be saved on disk due to %d write error(s)
timeout downloading the database: %s
libcurl could not download the database: %d %s
URL database download failed with HTTP status code 401 Unauthorized. Doublecheck the
username and password settings.
URL database download failed with HTTP status code %d
failed to flush output file buffers - errno %d %s
URL database file <file> has only %d bytes and is assumed to be corrupt
fsync for URL database file <file> (%d bytes) failed: %s
cannot write to license status file <file>
cannot download the license status file: %s
license status download failed with HTTP status code %d
license status file <file> has only %d byte(s) and is assumed to be corrupt
```

12 Copyright

13 Privacy Policy

The privacy policy of URLfilterDB is stated on the website: www.urlfilterdb.com/privacystatement.html.

14 More Information

The support desk can answer all questions. Feel free to send an email to support@urlfilterdb.com to ask a question or send your feedback.