

IwayPatrol Open Source Edition (OSE-1.00)

Mar. 3, 2009

I. Introduction

IwayPatrol is a web filtering product from iTech, Inc. (www.itech-mke.com) that has been in use by schools, businesses, libraries, ISPs, and municipal governments since 1996. It incorporates a number of techniques for controlling web access including black lists, white lists, content checking, and rating tag checking. A typical installation is a school building with a PC Linux server that handles 50-100,000 web requests a day. This scales up to a large site with multiple Sun servers and traffic approaching 30 million web requests per day.

IwayPatrol OSE is an open source version of the commercial product. It can be downloaded from <http://www.itech-mke.com/ose>. It is distributed under a GPL V3 license. For complete license details see <http://www.gnu.org/licenses/>. With the release of OSE-1.00, Itech is shifting to an open source based business model. It is hoped that OSE will find use in environments that neither need or desire commercial support. Itech will continue to provide a fully supported commercial alternative for those who need it. We look to the open source community, as well as the commercial users, to provide fresh ideas and opportunities for product improvement.

IwayPatrol OSE-1.00 is targeted to run on PC Linux systems. The 1.00 distribution was developed and tested on Ubuntu 8.04 and Fedora 8. The distribution is a source distribution with binary site lists. It requires a C/C++ development environment to build and install it.

How does IwayPatrol OSE differ from the commercial version? 1) Site list encryption has been removed. OSE is not able to read the encrypted bad site lists from the commercial version. 2) There is no login manager. This means that the login and override features of the commercial version do not work in OSE. 3) There is no web based administration component. Administration in OSE is currently done from the command line. 4) The commercial version includes an iTech maintained blacklist of bad sites that is updated daily. 5) The commercial version includes feedback of logs to iTech to check for candidates that need to be added to the bad sites list. Items 1,4 and, 5 are restricted to the commercial version and service and will remain so. Items 2 and 3 are not ready yet for OSE and will probably show up eventually in some form.

II. Site Blacklists

To be effective, a web filter needs a blacklist of sites to block. The commercial version of IwayPatrol provides a blacklist divided into 26 categories with a daily update. This list is not provided with the open source edition but may be purchased by visiting “<http://www.itech-mke.com>”. There are several open source blacklists that can be found by following links at “<http://www.squidguard.com/blacklists>”. IwayPatrol OSE includes two of these open source blacklists, the Toulouse blacklist from “<http://cri.univ-tlse1.fr/blacklists/download>” and the MESD blacklist from “<http://squidguard.mesd.k12.or.us/blacklists.tgz>”. Use of these lists seem not to require any fees or registrations. There are other links from “<http://www.squidguard.com>” that ask for payment for their use.

The open source blacklists have been imported into the IwayPatrol binary sitelist format. The following modifications have been made. 1) The categories of the open source lists have been translated into the iTech category scheme. Sites with categories that don't match anything in the iTech scheme have been skipped. Many of the sites in these two lists appear in multiple categories. The iTech lists format only supports 1 category per site. Which ever entry appears first in the sorted input ended up in the binary file. 2) Sites that are specified as an IP address have been skipped. IwayPatrol uses host names. An IP address is translated via reverse DNS to a host name for blacklist checking. IwayPatrol by default rejects any URL with an IP address that has no reverse DNS entry. 3) URLs with a path have been replicated to match an optional www. at the beginning. So, if there is an entry for “naughty.com/path”, an entry has also been added for “www.naughty.com/path”.

The iTech sitelist category scheme includes 26 categories A-Z. These categories can be enabled or disabled in the config file.

A	Alcohol	N	Social
B	Gambling	O	Tobacco
C	Chat	P	Personal
D	Drugs	Q	
E	Email	R	Sports
F	Jokes	S	Softcore
G	Games	T	Topical (mature subjects like abortion)
H	Hardcore (porn)	U	Untagged
I	Illegal	V	Violence
J	Jobs	W	Weapons
K	Proxy	X	Acheck (adult check)
L	Language	Y	Childporn
M	Streaming	Z	Shopping

III. Installation

First off, this software is not of open source ancestry and doesn't install using the common ./configure, make, make install scenario. The intended target is a current Linux distribution. The install is from source and requires a C/C++ development environment. The following details the procedure:

- 1) Download the IwayPatrol OSE distribution file ose-100.tgz from www.itech-mke.com/ose and place into a work directory.
- 2) tar xzvf ose-100.tgz
(Expand the distribution.)
- 3) cd ose-100
- 4) ./Config
(Run configuration script which generates Make files.)
- 5) ./Build
(Run script that builds the application.)
- 6) ./Install
(Installation script.)
- 7) Edit ose.conf and add ACL's to restrict who can use the web filter. For machines that are visible from the Internet it is generally a bad idea to create an open proxy.
- 8) Add ose_daily script to cron to rotate logs at midnight.
- 9) Add startup script.

IV. Sample Installation

The following is a sample log of an installation. User commands are highlighted.

```
# tar xzvf ose-100.tgz
ose-100/
ose-100/include/
ose-100/include/strfun.h
ose-100/include/Wtable.h
ose-100/include/Mail.h
ose-100/include/udata.h
ose-100/include/Stream.h
ose-100/include/PtyStream.h
ose-100/include/icra.h
ose-100/include/SockStream.h
ose-100/include/PipeStream.h
ose-100/include/base64.h
ose-100/include/Logger.h
ose-100/include/tld.h
ose-100/include/Env.h
ose-100/include/Keyed_list.h
ose-100/include/Sitelist.h
```

ose-100/include/ScanString.h
ose-100/include/Safe4Kids.h
ose-100/include/acl.h
ose-100/include/smtp.h
ose-100/include/host_access.h
ose-100/include/Url.h
ose-100/include/flock.h
ose-100/include/cpipe.h
ose-100/include/Key.h
ose-100/include/FileStream.h
ose-100/include/Hash.h
ose-100/include/time_mark.h
ose-100/include/popuser.h
ose-100/include/rsac.h
ose-100/include/Sitetag.h
ose-100/include/html.h
ose-100/include/sock.h
ose-100/include/List.h
ose-100/include/uad-1.20.h
ose-100/include/ConfigStream.h
ose-100/include/uad.h
ose-100/include/Ringbuf.h
ose-100/include/SafeSurf.h
ose-100/include/int32.h
ose-100/include/FilterConfig.h
ose-100/include/IP.h
ose-100/include/String.h
ose-100/tools/
ose-100/tools/slu.C
ose-100/tools/TARGETS
ose-100/Config.pl
ose-100/mkdiff
ose-100/Install
ose-100/db/
ose-100/db/db.h
ose-100/db/dbload.c
ose-100/db/dbcopy.c
ose-100/db/sortest.c
ose-100/db/ltest.c
ose-100/db/dbdump.c
ose-100/db/db_dict.c
ose-100/db/db_seq.c
ose-100/db/db_var.c
ose-100/db/sort.c
ose-100/db/readme.doc
ose-100/db/dblib.h
ose-100/db/vtest.c
ose-100/db/fname.c
ose-100/db/lock2.c
ose-100/db/stest.c
ose-100/db/TARGETS
ose-100/db/db.doc
ose-100/db/db_main.c
ose-100/db/dbu.c
ose-100/db/itest.c
ose-100/db/db_ran.c
ose-100/db/lock1.c
ose-100/db/rtest.c
ose-100/db/dtest.c
ose-100/db/db_idx.c
ose-100/db/db_link.c
ose-100/Config
ose-100/ose/
ose-100/ose/ose.C
ose-100/ose/TARGETS
ose-100/fixperl
ose-100/Install.defaults
ose-100/configs/
ose-100/configs/toulouse.dom
ose-100/configs/mesd.url
ose-100/configs/searchwords.txt
ose-100/configs/toulouse.url
ose-100/configs/contentwords.txt

```
ose-100/configs/ose.conf.src
ose-100/configs/mesd.dom
ose-100/License.txt
ose-100/scripts/
ose-100/scripts/ose_start.src
ose-100/scripts/ose_daily.src
ose-100/scripts/ose_check.src
ose-100/scripts/ose_rotate.src
ose-100/scripts/ose_stop.src
ose-100/scripts/ose_restart.src
ose-100/genmakefile
ose-100/lib/
ose-100/lib/FileStream.C
ose-100/lib/strfun.c
ose-100/lib/Env.C
ose-100/lib/rsac.C
ose-100/lib/PipeStream.C
ose-100/lib/base64.c
ose-100/lib/icra.C
ose-100/lib/String.C
ose-100/lib/acl.C
ose-100/lib/SockStream.C
ose-100/lib/Sitelist.C
ose-100/lib/Date.C
ose-100/lib/udata.c
ose-100/lib/html.c
ose-100/lib/sock.c
ose-100/lib/Url.C
ose-100/lib/Sitetag.C
ose-100/lib/flock.c
ose-100/lib/Safe4Kids.C
ose-100/lib/cpipe.c
ose-100/lib/Stream.C
ose-100/lib/ScanString.C
ose-100/lib/tld.C
ose-100/lib/IP.C
ose-100/lib/host_access.c
ose-100/lib/TARGETS
ose-100/lib/FilterConfig.C
ose-100/lib/SiteFilter.C
ose-100/lib/popuser_lib.c
ose-100/lib/Wtable.C
ose-100/lib/Mail.C
ose-100/lib/time_mark.C
ose-100/lib/Logger.C
ose-100/lib/uad_lib.c
ose-100/lib/smtp.C
ose-100/lib/SafeSurf.C
ose-100/lib/Key.C
ose-100/lib/PtyStream.C
ose-100/lib/ConfigStream.C
```

```
# cd ose-100
```

```
# ./Config
```

```
Configuring for LINUX
```

```
configs/...
```

```
db/...
```

```
include/...
```

```
ose/...
```

```
lib/...
```

```
scripts/...
```

```
tools/...
```

```
Generating Build file...
```

```
# ./Build
```

```
Entering lib ...
```

```
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H sock.c
```

```
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H strfun.c
```

```
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H flock.c
```

```
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H host_access.c
```

```
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H popuser_lib.c
```

```
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H uad_lib.c
```

```

gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H html.c
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H udata.c
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H base64.c
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H cpipe.c
ar -rv itech_clib.a sock.o strfun.o flock.o host_access.o popuser_lib.o uad_lib.o html.o
udata.o base64.o cpipe.o
ar: creating itech_clib.a
a - sock.o
a - strfun.o
a - flock.o
a - host_access.o
a - popuser_lib.o
a - uad_lib.o
a - html.o
a - udata.o
a - base64.o
a - cpipe.o
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H String.C
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H SockStream.C
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H ScanString.C
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H Url.C
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H Sitelist.C
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H Key.C
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H Env.C
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H rsac.C
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H Logger.C
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H FilterConfig.C
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H IP.C
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H acl.C
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H time_mark.C
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H Wtable.C
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H SafeSurf.C
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H Sitetag.C
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H Safe4Kids.C
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H PtyStream.C
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H PipeStream.C
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H smtp.C
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H ConfigStream.C
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H Stream.C
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H FileStream.C
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H Mail.C
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H Date.C
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H icra.C
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H tld.C
ar -rv itech_cpplib.a String.o SockStream.o ScanString.o Url.o Sitelist.o Key.o Env.o
rsac.o Logger.o FilterConfig.o IP.o acl.o time_mark.o Wtable.o SafeSurf.o Sitetag.o
Safe4Kids.o PtyStream.o PipeStream.o smtp.o ConfigStream.o Stream.o FileStream.o Mail.o
Date.o icra.o tld.o
ar: creating itech_cpplib.a
a - String.o
a - SockStream.o
a - ScanString.o
a - Url.o
a - Sitelist.o
a - Key.o
a - Env.o
a - rsac.o
a - Logger.o
a - FilterConfig.o
a - IP.o
a - acl.o
a - time_mark.o
a - Wtable.o
a - SafeSurf.o
a - Sitetag.o
a - Safe4Kids.o
a - PtyStream.o
a - PipeStream.o
a - smtp.o
a - ConfigStream.o
a - Stream.o
a - FileStream.o
a - Mail.o

```

```
a - Date.o
a - icra.o
a - tld.o
Leaving lib ...
```

```
Entering db ...
```

```
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H db_main.c
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H db_seq.c
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H db_ran.c
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H db_idx.c
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H db_var.c
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H fname.c
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H sort.c
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H db_dict.c
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H db_link.c
ar -rv libdb.a db_main.o db_seq.o db_ran.o db_idx.o db_var.o fname.o sort.o db_dict.o
db_link.o
ar: creating libdb.a
a - db_main.o
a - db_seq.o
a - db_ran.o
a - db_idx.o
a - db_var.o
a - fname.o
a - sort.o
a - db_dict.o
a - db_link.o
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H dbcop.c
gcc dbcop.o ../lib/itech_clib.a ../db/libdb.a -lresolv -lcrypt -lgdbm -o dbcop
strip dbcop
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H dbdump.c
gcc dbdump.o ../lib/itech_clib.a ../db/libdb.a -lresolv -lcrypt -lgdbm -o dbdump
strip dbdump
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H dbload.c
gcc dbload.o ../lib/itech_clib.a ../db/libdb.a -lresolv -lcrypt -lgdbm -o dbload
strip dbload
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H dbu.c
gcc dbu.o ../lib/itech_clib.a ../db/libdb.a -lresolv -lcrypt -lgdbm -o dbu
strip dbu
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H dtest.c
gcc dtest.o ../lib/itech_clib.a ../db/libdb.a -lresolv -lcrypt -lgdbm -o dtest
strip dtest
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H itest.c
gcc itest.o ../lib/itech_clib.a ../db/libdb.a -lresolv -lcrypt -lgdbm -o itest
strip itest
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H lock1.c
gcc lock1.o ../lib/itech_clib.a ../db/libdb.a -lresolv -lcrypt -lgdbm -o lock1
strip lock1
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H lock2.c
gcc lock2.o ../lib/itech_clib.a ../db/libdb.a -lresolv -lcrypt -lgdbm -o lock2
strip lock2
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H ltest.c
gcc ltest.o ../lib/itech_clib.a ../db/libdb.a -lresolv -lcrypt -lgdbm -o ltest
strip ltest
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H rtest.c
gcc rtest.o ../lib/itech_clib.a ../db/libdb.a -lresolv -lcrypt -lgdbm -o rtest
strip rtest
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H sortest.c
gcc sortest.o ../lib/itech_clib.a ../db/libdb.a -lresolv -lcrypt -lgdbm -o sortest
strip sortest
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H stest.c
gcc stest.o ../lib/itech_clib.a ../db/libdb.a -lresolv -lcrypt -lgdbm -o stest
strip stest
gcc -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H vtest.c
gcc vtest.o ../lib/itech_clib.a ../db/libdb.a -lresolv -lcrypt -lgdbm -o vtest
strip vtest
Leaving db ...
```

```
Entering ose ...
```

```
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H ose100.C
g++ ose100.o ../lib/itech_cpplib.a ../lib/itech_clib.a ../db/libdb.a -lresolv -lcrypt -
lgdbm -o ose100
ose.o: In function `init(int, char**)':
```

```
strip ose100
Leaving ose ...
```

```
Entering tools ...
g++ -c -I../include -I../db -DLINUX -DUSE_FLOCK -DNEED_CRYPT_H slu.C
g++ slu.o ../lib/itech_cpplib.a ../lib/itech_clib.a ../db/libdb.a -lresolv -lcrypt
-lgdbm -o slu
strip slu
Leaving tools ...
```

./Install

```
-----
Iwaypatrol Open Source Edition (OSE-1.00) - Installer
-----
```

```
Site Name [Iway Patrol OSE-1.00] ?
Installation path [/usr/local/itech] ? /usr/local/ose
Runtime group ID [itech] ? nobody
Runtime user ID [itech] ? nobody
Server host name [] ? proxy.itech-mke.com
Server IP Address [] ? 192.168.1.100
Server TCP Port [8080] ?
??? Path '/usr/local/ose' does not exist
Do you wish to create it <Y|N> [Y] ?
```

```
??? Path '/usr/local/ose' created.
??? Directory '/usr/local/ose/logs' created.
```

```
??? Copy binaries
.. tools/slu
.. ose/ose
```

```
??? Copy scripts
.. ose_check
.. ose_daily
.. ose_restart
.. ose_rotate
.. ose_start
.. ose_stop
```

```
??? Copy config files
.. ose.conf
.. contentwords.txt
.. searchwords.txt
.. mesd.dom
.. mesd.url
.. toulouse.dom
.. toulouse.url
.. License.txt
.. local-goodsites.txt
.. local-badsites.txt
.. iplist.txt
```

V. Running the Server

The destination directory contains several scripts to start (*ose_start*), stop (*ose_stop*), and restart (*ose_restart*) the web filter. The actual binary program that runs is named *ose*. It is controlled by a configuration file named *ose.conf*. As of v1.00, OSE does not include web administration. The filter is configured by editing the config file. If the filter is running, a restart is needed for config changes to take effect.

The files *local-goodsites.txt* and *local-badsites.txt* are the domain/url whitelist and blacklist files resp. They are text files with one domain/url per line and may be maintained with a text editor. The intent is to allow local changes to the site lists without having to modify the binary site list files. A restart is needed for any changes to take effect.

The *searchwords.txt* and *contentwords.txt* files are used for checking of search arguments and document content resp. They are text files with one word pattern per line and may be maintained with a text editor. A word pattern that starts with an up arrow (^) is restricted to matching the beginning of a word. A pattern that ends with a dollar sign (\$) must match the end of a word. A space in a pattern will match any number of spaces or other non-word characters like dashes, underscores, etc. Pattern matching is case insensitive. A restart is needed for any changes to take effect.

When a request contains an IP address instead of a host name, a reverse DNS lookup is done against that IP address and the resulting host name is looked up in the site lists. If there is no reverse DNS for an IP, access is denied. The file *iplist.txt* is a file that contains a list of IP addresses, one per line, for which access is allowed even though there is no reverse DNS. This is a text file and can be edited with a text editor. A netmask can be used to specify an entire network as in 192.168.1.0/24. A range of addresses may also be specified as in 192.168.1.200-192.168.1.220. A restart is need for any changes to take effect.

The filter runs as a collection of processes. The original process is the master. Sub-processes are spawned to handle user requests. The config file specifies the minimum (MinProcCnt) and maximum (MaxProcCnt) processes to run. The number of subprocesses is controled by the master process based on the load.

The filter may be used for clients that explicitly have their browsers configured to go through a proxy. Firewall rules should be in place to block any out going port 80 (http) traffic that does not come from the proxy. It may also be used transparently. In transparent mode, the proxy server must act as a gateway to the Internet with traffic forced to flow through it. Firewall rules on the proxy are used to redirect out going port 80 (http) traffic to the local proxy server port.

VI. Server Config File Format

Intro

The iTech web filter is configured with a config file. The standard file is named *ose.conf* and is located in the iTech filter directory. The filter directory can be chosen arbitrarily at installation time. For the purposes of this document, the filter directory is assumed to be */usr/local/itech*. The web filter needs to be restarted (*ose_restart*) for the changes to take effect.

Format

The file is free format with one command per line. A pound sign (#) marks the beginning of a comment which extends to the end of the line. Blank lines are ignored. The file is divided into sections. The first section contains global parameters that apply to all filter levels. That is followed by one section per filter level. Each filter level section contains parameters that are specific to that filter level. The **filter** command marks the beginning of a filter level section. The command keywords will be shown as highlighted. Case does not matter. The capitalization below is entirely optional.

Global Section

AccessLogFile <file-name>

Specify the name of the access log. Suggested name is */usr/local/itech/logs/access.log*. If there is no **AccessLogFile** command, then there will be no access log generated.

ACL Allow <ip> <filter#>

ACL Deny <ip> <filter#>

Specify an access control list of what hosts and networks can connect to the server. The **allow** and **deny** keywords specify the type of access. If the IP address includes a trailing net mask size (e.g., 10.1.0.0/16) it is treated as a network. Otherwise it is treated as a single host. The optional filter level number on the end, sets the filter level for matching clients. When a client attempts to connect to the server, its IP address is compared to the IP in each ACL in order. The first matching command is applied. If no match is found, the default access is **allow**.

ContentBufferSize <#bytes>

This is used to specify how big the content checking buffer should be. When content checking is enabled, the filter will read up to this much of a reply and

check it for bad words before sending it to the client. Only one buffer full of reply is checked. The default value is 20000.

DefaultFilter *<filter#>*

Set the default filter level for clients.

DocumentRoot *<path>*

The web filter contains a built-in web server for handling such tasks as displaying error messages and handling login and override forms. This specifies the document path that it uses.

ErrorLogFile *<file-name>*

Specify the name of the error log. Suggested name is */usr/local/itech/logs/error.log*. If there is no **ErrorLogFile** command, then there will be no error log generated.

ErrorPicURL *<url>*

Set the URL for the picture that appears on the error message screen. The default is */usr/local/itech/cop.gif*.

Group *<name>*

Specify what group the server should run as.

LoginURL *<file-name>*

Specify the name of the login screen. By default this is */usr/local/itech/login.html*. Although this suggests that an arbitrary URL may be used, it is not the case at this time. **(OSE v1.00 does not support logins.)**

Option *Login*

Specify that clients are required to log in. The filter level in a matching ACL will override this requirements for clients that match on that ACL. **(OSE v1.00 does not support logins.)**

PidFile *<file-name>*

Specify the name of the PID file for the server. The suggested name is */usr/local/itech/logs/ose.pid*. The pid file is used by the *ose_stop* and *ose_restart* scripts. The command needs to be included, for these scripts to work properly.

ServerName <name>

Specify the name of the server. The web filter has the ability to serve a set of local web pages. Typically these are login pages and error messages. The web filter matches the http host header against the ServerName to decide which requests are to be handled locally. This command can be repeated multiple times with different names or IP addresses.

ServerPort <port-number>

This command specifies which tcp port the server is going to listen for connections on. This can be repeated multiple times to have the server listen on more than one port.

TraceLevel <0 or 1>

Although not necessarily part of a production server, a trace log file can be set up to log information for requests from a given IP address or to a selected url destination. This information can be quite useful for debugging issues with sites that don't seem to work through the filter. A value of 0 disables logging to the trace log. A value of 1 enables logging.

TraceLogFile <file-name>

Specify the name of the trace log file. This file can be used for debugging or to track traffic from a specific IP client address or to a specific url.

TraceSRC <ip-address>

Specify that any requests that originate from ip-address should be traced. TraceLogFile must be defined and TraceLevel must be 1.

TraceURL <url>

Any requests to url should be traced. TraceLogFile must be defined and TraceLevel must be 1.

UAD <host> <port>

Specify the host and port number of the authorization server (uad). Typically the port number is 1600. A uad server is required if the filter is configured for either logins or overrides. **(This is not supported in OSE v1.00.)**

UADCacheTime <seconds>

Specify the amount of time a UAD lookup may be cached for. This reduces the load on uad and the latency for requests within the cache window. A client can generate quite a few web requests in a few seconds. (This is not supported in OSE v1.00.)

User <name>

Specify what user the server should run as. This normally should be *itech* and most definitely should not be *root*.

Filter Section

Filter <Level#> <Name>

This command marks the beginning of the filter section for level *filter#* with name *name*.

AllowSites <file_name>

Specify the name of an “allow only” site list. A URL is blocked unless it is in the site list. If the file name has a *.txt* extension, it is assumed to be a text file with one domain/url per line. If the file name has no extension, it is assumed to be a binary site list which consists of two files, *file_name.dom* and *file_name.url*.

It is generally better to specify urls in most general applicable form. *Badsite.com* is usually better than www.badsite.com. The first form will match any domain name that ends with *badsite.com*. The second form will only match *www.badsite.com*. If the url includes a path, then the host part of the url must match exactly. So to specify www.site.com/path, both www.site.com/path and *site.com/path* probably need to be included. Do not prefix entries with *http://*. The site list does not store protocol or port numbers.

BadSites <file_name>

Specify the name of a bad site file. If the file name has a *.txt* extension, it is assumed to be a text file with one domain/url per line. If the file name has no extension, it is assumed to be a binary site list which consists of two files, *file_name.dom* and *file_name.url*. Multiple **BadSites** can be specified and will be checked in order. Local overrides should be listed first.

It is generally better to specify urls in most general applicable form. *Badsite.com* is usually better than www.badsite.com. The first form will

match any domain name that ends with *badsite.com*. The second form will only match *www.badsite.com*. If the url includes a path, then the host part of the url must match exactly. So to specify www.site.com/path, both www.site.com/path and site.com/path probably need to be included. Do not prefix entries with <http://>. The site list does not store protocol or port numbers.

BadWordLimit <n,m>

Set the match limit for content checking to n bad words of which atleast m are unique. This is to avoid false hits. A page with 10 or 15 instances from the content words list has a very high probability of being undesirable.

ContentWordList <file_name>

Specify the filename of the word list used for content checking. The list may either be a text list (.txt) or precompiled binary file (.bin). The iTech default list is */usr/local/itech/contentwords.txt*.

GoodSites <file_name>

Specify the name of the good sites files. If the file name has a *.txt* extension, it is assumed to be a text file with one domain/url per line. If the file name has no extension, it is assumed to be a binary site list which consists of two files, *file_name.dom* and *file_name.url*. A good site file is a white list. If a url matches an entry in a good sites file it is allowed. Any further checking, such as content checking, is skipped. Multiple **GoodSites** can be specified and will be checked in order. Local overrides should be listed first.

It is generally better to specify urls in most general applicable form. *Badsite.com* is usually better than www.badsite.com. The first form will match any domain name that ends with *badsite.com*. The second form will only match *www.badsite.com*. If the url includes a path, then the host part of the url must match exactly. So to specify www.site.com/path, both www.site.com/path and site.com/path probably need to be included. Do not prefix entries with <http://>. The site list does not store protocol or port numbers.

Grade <grade_level>

Set the grade level for site list checks. When the grade level is set and a requested URL matches an entry in the bad site list, a comparison is made against the grade level of the bad site entry. If the grade level of the filter equals or exceeds the grade level of the site list entry, the requested is permitted. Otherwise it is denied. The following grade levels may be specified:

- **GradeSchool, GS, Elementary**
- **MiddleSchool, MS, JuniorHigh**
- **HighSchool, HS, Secondary**

(None of the site lists provided with OSE v1.00 contain useful grade level.)

ICRA (<label>)

Set a ICRA label threshold and enable ICRA label checking.

IPFile <file_name>

Specify the name of the IP file. This file contains a list of ip addresses (hosts and networks) that are exempt from reverse DNS checking. The iTech standard file is */usr/local/itech/iplist.txt*.

Option CheckDNS

Option CheckDNSforHTML

Option DontCheckDNS

When a request url contains an ip-address instead of a host name, a reverse DNS lookup is done and the resulting host name is checked against the site lists. If there is no reverse DNS the default behavior is to reject the request. The CheckDNS option explicitly requests the default behavior. The CheckDNSforHTML option only does the reverse DNS check for html documents. The DontCheckDNS option disable this checking completely. The checking can be overridden on a per ip basis by adding the exceptions to the iplist.txt file.

Option CheckEmbeddedURLs

Option WarnEmbeddedURLs

Check for urls of bad sites embedded in the current request as a query string or part of the path. The intent is to stop proxy bypass schemes that pass the url of the bad site as part of the url. The CheckEmbeddedURLs version will result in a request being blocked. The WarnEmbeddedURLs version will just log an error.

Option CheckFormURLs

Option WarnFormURLs

Check for bad site urls in the form content of a request. The intent is to stop proxy bypass schemes that pass the url of the bad site in a form. The CheckFormURLs

version will block the request if an offending url is found. The WarnFormURLs variant will just log an error message.

Option CheckMetaTags

Enable the content checking of the meta tags portion of the html reply. In particular, the scanning of the keywords meta tag is useful.

Option CheckOverrides

When access to a web page is denied, give the user the option of supplying an override id and password to continue. If the id and password are validated, the user goes into an unfiltered override mode for five minutes. After five minutes they drop back to their default filter level. (Does not apply to OSE-1.00)

Option CheckRequest

Check client requests (query strings and form data) for bad content. If any matching bad words are found the request is denied and a error is returned.

Option CheckWebContent

Option WarnWebContent

Check web pages (text/html and text/plain) against the bad word list. If more than **BadWordLimit** words are matched, the request is denied and an error message is returned. The check is done against the first 20000 characters of the page (Config default). The WarnWebContent variant performs the content check but does not reject requests that fail. Instead it logs an error message. This provides feedback for sites that may need to be added to the black list.

Option ClientKeepAlive

Enable client keepalive and allow multiple requests per connection for improved performance and better response for the client. The downside is more filter processes on the system. When a filter completes a request it will stay connected to the client waiting for more requests.

Option FullWebAccess

Allow full web access. Don't do any site list, content, or label checking regardless of what other options are set. The would be set for an adult unfiltered or override filter level.

Option HTTPDirect

Option HTTPSDirect

The default behavior of the filter is to forward requests to a squid caching proxy server. HTTPDirect instructs the filter to connect directly to destination server for HTTP traffic. HTTPSDirect specifies the same for HTTPS traffic.

Option NoWebAccess

Block all web access regardless of what other options are set. This also takes precedence over the **FullWebAccess** option should they both be set.

OutGoingIP <ip-address>

Use ip-address as the source ip address for requests going out through this filter. The ip-address must be bound by the server.

OverrideURL <url>

The URL of the screen that is displayed when the user selects the override button in response to a blocked site message. **(Does not apply to OSE-1.00)**

ProxySRC <ip-address>

noProxySRC <ip-address>

This can be used to change the current proxy behavior for traffic originating from ip-address. If traffic for the current filter is being forwarded to a proxy, noProxySRC can be used to disable that behavior for ip-address. If traffic is not going to a proxy, ProxySRC can be used to force traffic for an ip-address through a proxy.

ProxyURL <url>

noProxyURL <url>

This can be used to change the current proxy behavior for requests going to url. If traffic for the current filter is being forwarded to a proxy, noProxyURL can be used to disable that behavior for url. If traffic is not going to a proxy, ProxyURL can be used to force traffic for a url through a proxy. Sometimes web sites that don't seem to work very well will work better if passed through a squid proxy. This bit of a bandaid can keep users happy while you try to figure out the latest bit of IIS weirdness.

RSAC (<label>)

Set an RSAC label threshold and enable RSAC label checking.

SafeForKids (<label>)

Set a SafeFor Kids label threshold and enable SafeForKids label checking.

SafeSurf (<label>)

Set a SafeSurf label threshold and enable SafeSurf label checking.

SearchWordList <file_name>

Specify the filename of the word list used for request checking. The file may either be a text list (.txt) or precompiled binary file (.bin). The iTech default list is */usr/local/itech/searchwords.txt*.

UAD <host> <port>

Set the host and port of the UAD server to be used for override requests. This has no bearing on login requests since they are made before a filter level is determined.

Unblock <category> <category> ...

Unblock the specified categories from the sitelist. Available categories are:

- **Acheck**
- **Alcohol**
- **Chat**
- **ChildPorn**
- **Drugs**
- **Email**
- **Gambling**
- **Games**
- **Hardcore**
- **Illegal**
- **Jobs**
- **Jokes**
- **Language**
- **Personal**
- **Proxy**

- **Shopping**
- **Social**
- **Softcore**
- **Sports**
- **Streaming**
- **Tobacco**
- **Topical**
- **Weapons**
- **Violence**

Webproxy *<host:port>*

Specify host and port number for the web proxy. If multiple web proxies are specified, the filter selects one based on a hash of the client IP address. This is to ensure that a stream of requests from the same client go through the same proxy to preserve any session semantics.

VII. Sample Config File – ose.conf

```
#
# ose.conf - iTech Filter Config
#

# This file is part of the Iwaypatrol - Open Source Edition (OSE) web filter.
# Copyright (C) 2008 iTech, Inc / Ken Harris chief developer
# Project website is "http://www.itech-mke.com/ose".
#

# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#

# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#

# You should have received a copy of the GNU General Public License
# along with this program. If not, see <http://www.gnu.org/licenses/>.

ServerName proxy
ServerName 10.10.1.103
ServerPort 8080
User nobody
Group nobody
DocPath /usr/local/ose
#uad localhost 1600
#LoginURL login.html
ErrorPicURL http://10.10.1.103:8080/cop.gif
AccessLogFile /usr/local/ose/logs/access.log
ErrorLogFile /usr/local/ose/logs/error.log
PIDFile /usr/local/ose/logs/ose.pid
LockFile /usr/local/ose/logs/ose.lck
MinProcCnt 10
MaxProcCnt 300
BusyThreshold 90
MaxConnects 100
#option logins
option verbose
DefaultFilter 1
```

Filter 1 Full-access
Option FullWebAccess
Option ClientKeepAlive
Option HTTPDirect
Option HTTPSDirect

Filter 2 Filtered
grade 3
Goodsites /usr/local/ose/local-goodsites.txt
Badsites /usr/local/ose/local-badsites.txt
Badsites /usr/local/ose/mesd
Badsites /usr/local/ose/toulouse
IPfile /usr/local/ose/iplist.txt
SearchWordList /usr/local/ose/searchwords.txt
ContentWordList /usr/local/ose/contentwords.txt
RSAC (12 n1 s1 v2)
SafeSurf (SS~~000 4)
BadWordLimit 5,2
Option CheckRequest
Option CheckEmbeddedURLs
Option CheckMetaTags
Option CheckWebContent
Option WarnWebContent
Option ClientKeepAlive
Option HTTPDirect
Unblock Games
Unblock Jobs
Unblock Streaming
Unblock Sports
Unblock Shopping