

# **Installing OSSIM on a Debian GNU/Linux**

**David Gil**

**Stéphane Fournier**

## **Installing OSSIM on a Debian GNU/Linux**

by David Gil and Stéphane Fournier

### Revision History

Revision 0.1 2004-10-21 Revised by: sfournier

Conversion from plain text to Docbook v4.2 and update of debian packages versions

# Table of Contents

<b>1. Introduction.....</b>	<b>1</b>
1.1. Foreword.....	1
1.2. Structure considerations .....	1
1.3. Lazy install.....	1
1.4. Necessary software installation .....	2
<b>2. Databases .....</b>	<b>3</b>
2.1. Ossim database .....	3
2.2. Snort/Acid database.....	4
2.3. Updating from Ossim 0.9.6 .....	4
<b>3. Server .....</b>	<b>5</b>
<b>4. Framework .....</b>	<b>6</b>
4.1. Apache + PHP + ADOdb .....	6
4.2. phpGACL .....	7
4.3. RRDtool .....	8
4.4. MRTG.....	9
4.5. Graphing.....	9
4.6. ACID .....	10
4.6.1. Cache auto update (OPTIONAL) .....	11
4.6.2. Acid backups (OPTIONAL).....	12
4.7. Control Panel .....	12
4.8. Ntop.....	13
4.9. Nmap .....	14
4.10. Nessus client.....	14
4.11. PDF reports .....	15
<b>5. Agents.....</b>	<b>16</b>
5.1. Plugins.....	17
5.1.1. Snort.....	17
5.1.2. Ntop .....	19
5.1.3. tcptrack .....	19
5.1.4. Pads.....	19
5.1.5. pOf.....	19
5.1.6. arpwatch.....	20
5.2. Nessus server .....	20
5.3. OpenNMS.....	20
5.3.1. Java SDK .....	20
5.3.2. Tomcat .....	21
5.3.3. RRDTool, Postgresql and perl .....	21
5.3.4. OpenNMS installation .....	22
5.3.5. Configuration .....	22
<b>6. Tools.....</b>	<b>24</b>
<b>7. TODO .....</b>	<b>25</b>

# Chapter 1. Introduction

## 1.1. Foreword

This document aims to show step by step and in detail the installation of OSSIM on a Debian Sarge GNU/Linux system.

Almost everything exposed in this document can be applied to any other Linux distribution. Hopefully it will be useful to you.

If you have any problems related to the installation in Debian or want to add or correct something in this document, please, feel free to contact us at <dgil at ossim.net> and <gingurz at ossim.net>

You can also contact all the OSSIM developers at <devel at ossim.net>

## 1.2. Structure considerations

Ossim structure can be divided into 4 components, each can be installed on different hosts :

### Databases

ossim, snort/acid and phpgacl.

### Server

it runs the correlation engine.

### Framework

the web interface where you'll interact with Ossim.

### Agents

Agents: they run plugins that sends the results of various tools such as Snort, Pads, Ntop, etc. to the databases to be correlated by the server.

Generally you will have the databases, the server and the framework on the same host, and you can eventually add some sensors, depends on the processor and memory load.

## 1.3. Lazy install

To make installation easier we already built the debian packages of the tools we need to patch (snort, ntop, rrdtool etc.). To use them, simply edit /etc/apt/sources.list and add the following line at the beginning of the file :

```
deb http://www.ossim.net/download/ debian/
```

You can now use apt to download and install the packages.

- dpkg-dev : Package building tools for Debian (>= 1.10.22)
- fakeroot : Gives a fake root environment (>= 1.0.7)

Also, you will need a deb-src line into your /etc/apt/sources.list file in order to download source packages.

```
# apt-get install dpkg-dev fakeroot
# echo "deb-src http://http.us.debian.org/debian sarge main contrib non-free" >> /etc/apt/sources.list
```

## 1.4. Necessary software installation

Download the latest stable version of OSSIM from the <http://www.ossim.net/download.php> (<http://www.ossim.net/download.php>)download section.

Unpack it to '/opt/ossim' (or wherever you want). This directory will be called \$OSSIM\_PATH in the rest of the document.

Eventually, you can delete the unused directories. For example, if you install a host with no agent, you can remove \$OSSIM\_PATH\agent

# Chapter 2. Databases

OSSIM can use any database manager, but for simplicity and efficiency reasons, we are going to install mysql.

Install the following packages and their respective dependencies:

- mysql-server : mysql database server binaries (>= 4.0.20)
- mysql-client : mysql database client binaries (>= 4.0.20)

```
# apt-get install mysql-server mysql-client
```

The initial root password is empty, so anyone can connect as root without a password and be granted all privileges. The first thing you should do is specify a password for the MySQL root user.

```
# mysqladmin -u root password your_password
```

Now, you must use the '-p' option whenever you run mysql.

Networking is disabled by default. Edit the file '/etc/mysql/my.cnf' commenting the line with the 'skip-networking' option. MySQL will be listening on port TCP-3306 after restart.

## 2.1. Ossim database

Now we're going to create the OSSIM database structure.

The first thing you have to do is to edit the database configuration at '/etc/ossim/framework/ossim.conf'.

```
# mkdir /etc/ossim/  
# cp -r $OSSIM_PATH/contrib/debian/framework /etc/ossim/
```

Then adjust these values according to your configuration :

```
ossim_base=ossim  
ossim_user=root  
ossim_pass=your_password  
ossim_host=localhost  
ossim_port=3306
```

Create a new database for OSSIM:

```
# mysql -u root -p  
mysql> create database ossim;  
mysql> exit  
# cd $OSSIM_PATH/db  
# cat create_mysql.sql | mysql -u root ossim -p  
# cat ossim_data.sql snort_nessus.sql realsecure.sql | mysql -u root ossim -p
```

For complete description of Ossim database structure check : `$OSSIM_PATH/doc/ossim_db_structure.txt`

## 2.2. Snort/Acid database

Then we have to create the database needed by Snort and Acid. Note that even you don't use snort in your sensors, you have to create this database as Ossim uses it to store alerts from others plugin.

First you have to get `create_mysql.gz` from snort-mysql install and `create_acid_tbls_mysql.sql` from Acid patched for ossim.

You can get these .sql files via apt-get source:

```
# apt-get source snort-mysql acidlab
# cd acidlab-0.9.6b20/
# patch -p1 < $OSSIM_PATH/contrib/acid.patch
```

Then, you need to create the snort database :

```
# mysql -u root -p
mysql> create database snort;
mysql> exit;
```

Snort tables:

```
# cat $SNORT_SOURCE/contrib/create_mysql | mysql -u root -p snort
```

Acid tables:

```
# cat $ACID_SOURCE/create_acid_tbls_mysql.sql | mysql -u root snort -p
```

If you are going to install snort and acid on the same host, just type (after installing snort/acid, see sections 5.1 and 4.5):

```
# zcat /usr/share/doc/snort-mysql/contrib/create_mysql.gz | mysql -u root -p snort
# cat /usr/share/acidlab/create_acid_tbls_mysql.sql | mysql -u root snort -p
```

## 2.3. Updating from Ossim 0.9.6

If you already installed Ossim 0.9.6, you can get lazy ;) just by running the script :

```
# cat $OSSIM_PATH/db/096-to-097.sql | mysql -u root ossim -p
```

## Chapter 3. Server

Now, we are going to compile the server. You'll need the following packages and their respective dependencies:

- `autoconf` : automatic configure script builder (`>= 2.59`)
- `automake` : A tool for generating GNU Standards-compliant Makefiles (`>= 1.6.3`)
- `gcc` : The GNU C compiler (`>= 3.3.4`)
- `libglib2.0-dev` : Development files for the GLib library (`>= 2.4.7`)
- `libgda2-dev` : Development files for GNOME Data Access lib (`>= 1.0.4`)
- `gda2-mysql` : MySQL backend plugin for GNOME Data Access lib (`>= 1.0.4`)
- `libgnet-dev` : Developer files for GNet network library (`>= 2.0.4`)

```
# apt-get install autoconf automake libglib2.0-dev libgda2-dev gda2-mysql libgnet2.0-dev
```

Follow these steps:

```
# cd $OSSIM_PATH/  
# ./autogen.sh  
# cd src/  
# make  
# cp ossim-server /usr/local/bin/
```

Copy `$OSSIM_PATH/etc/server/` to `/etc/ossim`

```
# cp -r $OSSIM_PATH/etc/server to /etc/ossim
```

Now, you must edit the `'/etc/ossim/server/config.xml'` file and:

- edit sensor entry replacing name, ip (don't use 127.0.0.1) and interface.
- adjust the database configuration (ossimDS and snortDS).

Create the log directory :

```
# mkdir /var/log/ossim
```

Run server:

```
# ossim-server -d -c /etc/ossim/server/config.xml
```



# Chapter 4. Framework

First of all, copy the sample configuration from \$OSSIM\_PATH/etc/framework to /etc/ossim directory, if you hadn't already do it with databases installation :

```
# mkdir /etc/ossim/
# cp -r $OSSIM_PATH/contrib/debian/framework /etc/ossim/
```

Take a first look to '/etc/ossim/framework/ossim.conf' file and try to tune it as much as you can (don't worry, at the end of this section you will have a ossim.conf filled).

The first thing you have to do is to edit the database configuration:

```
ossim_type=mysql
ossim_base=ossim
ossim_user=root
ossim_pass=your_password
ossim_host=localhost
ossim_port=3306
```

## 4.1. Apache + PHP + ADOdb

It is necessary to install the Apache Web server with PHP support. It is also strongly recommended to use SSL with Apache.

ADOdb is a PHP database abstraction layer that is being used by Acid and by the OSSIM PHP code.

Install the following packages and their respective dependencies:

- apache-ssl : Versatile, high-performance HTTP server with SSL support (>= 1.3.31)
- php4 : Server-side, HTML-embedded scripting language (>= 4.3.9)
- php4-cgi : Server-side, HTML-embedded scripting language (>= 4.3.9)
- libphp-adodb : The 'adodb' database abstraction layer for php (>= 4.52)

```
# apt-get install apache-ssl php4 php4-cgi libphp-adodb
```

- php4-mysql : MySQL module for php4 (>= 4.3.9)
- php4-pgsql : PostgreSQL module for php4 (>= 4.3.8)
- php4-gd2 : GD module (with GD2) for php4 (>= 4.3.2+rc3)
- libphp-plot : The graphic library for php (>= 4.4.6)
- libphp-jpgraph : Object oriented graph library for php4 (>= 1.5.2)
- wwwconfig-common : Debian web auto configuration (>= 0.0.34)

```
# apt-get install php4-mysql php4-pgsql php4-gd2 libphp-plot \
libphp-jpgraph wwwconfig-common
```

And for the directive viewer :

```
# apt-get install php4-domxml php4-xslt
```

Edit the Apache configuration file at `/etc/apache-ssl/modules.conf` and make sure that the php module is loaded. Also, you can use `'dpkg-reconfigure apache'` and mark the module `'mod_php4'` from the configuration list.

Copy the configuration file of ossim for Apache :

```
# cp $OSSIM_PATH/contrib/debian/httpd/ossim.conf /etc/apache-ssl/conf.d/
```

Also check in `'/etc/php4/apache/php.ini'` and `'/etc/php4/cgi/php.ini'` files that the following lines are in:

```
extension=gd.so
extension=mysql.so
extension=pgsql.so
extension=domxml.so
extension=xslt.so
```

Re-run Apache-ssl.

You must edit these framework/ossim.conf variables:

- `adodb_path=/usr/share/adodb/`
- `jpgraph_path=/usr/share/jpgraph/`

## 4.2. phpGACL

phpGACL is used to manage user profiles.

There's a complete installation manual of phpGACL at  
<http://phpgac1.sourceforge.net/demo/phpgac1/docs/manual.html>  
 (<http://phpgac1.sourceforge.net/demo/phpgac1/docs/manual.html>)

Here is a short summary:

Download phpgacl from [https://sourceforge.net/project/showfiles.php?group\\_id=57103](https://sourceforge.net/project/showfiles.php?group_id=57103)  
 ([https://sourceforge.net/project/showfiles.php?group\\_id=57103](https://sourceforge.net/project/showfiles.php?group_id=57103)).

Uncompress and place it into `/var/www/phpgac1` directory.

Edit `phpgac1/gacl.class.php` and set `"db_type"`, `"db_host"`, `"db_user"`, `"db_password"`, and `"db_name"` with the same values that the OSSIM database has (we are going to insert gacl tables into ossim database). Set `"db_table_prefix"` to `"`.

Edit `phpgac1/admin/gacl_admin.inc.php` with *the same db settings*.

Go to <http://yourhost/phpgac1/setup.php>

Create `phpgac/admin/templates_c` directory (IMPORTANT: this directory must be writable by the user the webserver runs as).

phpGACL now is installed. Take a look at [http://yourhost/phpgac/admin/acl\\_admin.php](http://yourhost/phpgac/admin/acl_admin.php)

Remember to setup an ossim user for apache:

```
# htpasswd -c /var/www/ossim-users ossim
```

Now that phpgacl is installed you must run '`http://yourhost/ossim/setup/ossim_acl.php`' script to fill database with default acls.

And you should force to use Ossim pages instead of phpGACL ones with an apache auth directory.

Reade `README.phpgac` for more details.

NOTE: Since you have phpgacl configured, you should can enter to ossim framework login as admin-admin. It's fully recommended that you reset to default values at configuration->main.

## 4.3. RRDtool

In OSSIM, we use the developpement version of rrdtool (1.1.X) in order to get the aberrant behaviour support.

If you use ossim.net as a source for Debian Package (cf Intro) :

```
# apt-get install rrdtool librrd0 librrd0-dev librrdp-perl librrds-perl
```

If you prefer to compile everything, let's go.

Your system must have installed the following packages in order to compile the libraries:

• <code>dpkg-dev</code>	: Package building tools for Debian	(>= 1.10.23)
• <code>fakeroot</code>	: Gives a fake root environment	(>= 1.0.7)
• <code>gcc</code>	: The GNU C compiler	(>= 3.3.4)
• <code>g++</code>	: The GNU C++ compiler	(>= 3.3.4)
• <code>make</code>	: The GNU version of the "make" utility	(>= 3.80)
• <code>libtool</code>	: Generic library support script	(>= 1.5.6)
• <code>automake</code>	: A tool for generating GNU Standards-compliant Makefiles	(>= 1.6.3)
• <code>autoconf</code>	: automatic configure script builder	(>= 2.59)
• <code>cgilib</code>	: Simple CGI Library	(>= 0.5)
• <code>libart-2.0-dev</code>	: Library of functions for 2D graphics - devel	(>= 2.3.16)
• <code>libfreetype6-dev</code>	: FreeType 2 font engine, development files	(>= 2.1.7)

Then, follow these steps:

```
# cd /tmp
# wget http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/pub/beta/rrdtool-cvs-snap.tar.gz
# tar -xvzf rrdtool-cvs-snap.tar.gz
# cd rrdtool-XXXX-XX-XX
```

```
# dpkg-buildpackage -rfakeroot -uc -b
# cd ..
# dpkg -i librrd0-dev_1.1.0-1_i386.deb librrd0_1.1.0-1_i386.deb \
  librrdp-perl_1.1.0-1_all.deb librrds-perl_1.1.0-1_i386.deb \
  rrdtool_1.1.0-1_i386.deb
```

You must edit these ossim configuration variables:

```
rrdtool_path=/usr/bin/
rrdtool_lib_path=/usr/lib/perl5/
```

## 4.4. MRTG

Once again, you can easily install the patched package if you add OSSIM source of packages in sources.list :

```
# apt-get install mrtg libsnmp-session-perl
```

Otherwise you have to install Mrtg from sources:

- mrtg : Multi Router Traffic Grapher (>= 2.10.13)

```
# apt-get source mrtg
```

You have to patch mrtg in order to use the new rrdtool improvements. You will find this patch at '\$OSSIM\_PATH/contrib/mrtg/mrtg.diff'.

```
# cd mrtg-XXX/bin
# patch -p0 < $OSSIM_PATH/contrib/mrtg/mrtg.diff
```

Compile and install mrtg:

```
# cd ..
# dpkg-buildpackage -rfakeroot -uc -b
# cd ..
# apt-get install libsnmp-session-perl
# dpkg -i mrtg_XXX.deb mrtg-contrib_XXX.deb
```

## 4.5. Graphing

Make the following directories (if they don't exist)

```
$ mkdir $OSSIM_PATH/www/mrtg
$ cd $OSSIM_PATH/www/mrtg
$ mkdir host_qualification net_qualification global_qualification level_qualification
```

Edit /etc/ossim/framework/mrtg-rrd.cfg:

```
WorkDir: $OSSIM_PATH/www/mrtg
```

```

Include: $OSSIM_PATH/mrtg/hosts/host_qualification.cfg
Include: $OSSIM_PATH/mrtg/nets/net_qualification.cfg
Include: $OSSIM_PATH/mrtg/global/global_qualification.cfg

```

Important paths in order to draw graphs are:

```

mrtg_rrd_files_path=$OSSIM_PATH/www/mrtg
rrdpath_host=$OSSIM_PATH/www/mrtg/host_qualification/
rrdpath_net=$OSSIM_PATH/www/mrtg/net_qualification/
rrdpath_global=$OSSIM_PATH/www/mrtg/global_qualification/

```

Copy \$OSSIM\_PATH/include/ossim\_conf.pm in any place where Perl can find it. For example:

```
#ln -s $OSSIM_PATH/include/ossim_conf.pm /usr/lib/perl5/
```

Run the script named 'launch-mrtg':

```

# cd $OSSIM_PATH/mrtg
# ./launch-mrtg

```

If you see any error, you must tune these config files. To execute this script periodically, the best option is to add an entry to your crontab.

```
# cp $OSSIM_PATH/etc/cron.d/ossim /etc/cron.d/
```

or

```

$ crontab -e
0-59/5 * * * * $OSSIM_PATH/mrtg/launch-mrtg

```

Finally, copy the script 'draw\_graph.pl' to your cgi-bin directory.

```
$ cp $OSSIM_PATH/scripts/draw_graph.pl /usr/lib/cgi-bin/
```

NOTE: there are three versions of draw\_graph (normal, combined and fourrier). Choose the one you prefer.

Take a look at the section links of '/etc/ossim/framework/ossim.conf' again ;).

Ah, obtain a .ttf font and put it somewhere visible to ossim. Adjust /etc/ossim/framework/ossim.conf and make sure rrdtool can find its default font. (Look for fonts at \$OSSIM\_PATH/contrib/fonts)

## 4.6. ACID

You can use lazy way if you have modified your sources.list :

```
# apt-get install acidlab acidlab-mysql acidlab-doc
```

Or compile from the source as follow :

Download the latest version of Acid and patch it with the patch that you will find in the OSSIM contrib directory.

```
# apt-get source acidlab
# cd acidlab-XXX
# patch -p1 < $OSSIM_PATH/contrib/acid.patch
# dpkg-buildpackage -rfakeroot -uc -b
# dpkg -i acidlab_XXX.deb acidlab-mysql_XXX.deb acidlab-doc_XXX.deb
```

NOTE: An error occurs when apply the patch. Don't worry about it if you are using MySQL database, since it only affects postgresql database.

You may notice that with official acid sources you'll not see that error.

Check the DB abstraction library variable \$DBlib\_path in /etc/acidlab/acid\_conf.php and set it to "/usr/share/adodb/" and complete the database configuration.

Edit apache configuration file (/etc/acidlab/apache.conf) and change this line:

```
php_value include_path .
```

to:

```
php_value include_path .:$OSSIM_PATH/include/
```

Make sure you have created the snort database with Acid tables as described in section 2.2.

### 4.6.1. Cache auto update (OPTIONAL)

In order to auto-update the acid cache, edit '/etc/acidlab/acid\_conf.php' and set \$event\_cache\_auto\_update variable to 0.

Copy acid\_update\_db.php in the acidlab directory:

```
# cp $OSSIM_PATH/contrib/acid_update_db.php /usr/share/acidlab/
```

Edit /etc/ossim/framework/ossim.conf and set acid properties:

```
acid_link=http://localhost/acidlab/    # must be in http:// format
                                         # (needed by wget)

acid_path=/usr/share/acidlab/

# acid dir with password (htpasswd)?
acid_user=ossim
acid_pass=ossim
```

Execute the following script, that auto-update the cache:

```
# $OSSIM_PATH/scripts/acid_cache.pl
```

## 4.6.2. Acid backups (OPTIONAL)

In order to make db backups and clean acid database, you must run the script called 'backupdb.pl' placed in \$OSSIM\_PATH/scripts directory.

Create the '/var/lib/ossim/backup' directory and add an entry to your crontab:

```
$ crontab -e
5 0 * * * $OSSIM_PATH/scripts/backupdb.pl
```

Note: the script uses Zlib perl module. Install it:

```
# apt-get install libcompress-zlib-perl
```

And you must include the \$OSSIM\_PATH/include/ossim\_conf.pm in any place in which Perl can find it.

Setup this variables at /etc/ossim/framework/ossim.conf to configure the location of the backups and the days that the data are mantained in the snort database:

```
backup_dir=/var/lib/ossim/backup/
backup_day=7
```

## 4.7. Control Panel

You need to run the script 'control\_panel.py' that you will find at '\$OSSIM\_PATH/scripts/control\_panel.py'. This script updates the metrics database reading the rrd's that mrtg generates.

control\_panel.py is written in python, so you need to install the following packages:

- python : An interactive high-level object-oriented language (>= 2.3.4)
- python-mysqldb : A Python interface for MySQL (>= 1.1.6)

```
# apt-get install python python-mysqldb
```

The database configuration is readed from framework/ossim.conf, so make sure that ossim\_user, ossim\_db, etc. are set up correctly.

Make the script executable:

```
# cp $OSSIM_PATH/scripts/control_panel.py /usr/local/bin/
# chmod +x /usr/local/bin/control_panel.py
```

Type 'control\_panel.py --help' for more help.

## 4.8. Ntop

It's not necessary to install ntop in the same machine where the framework is; you can install ntop in one sensor and use it.

Let's go to install ntop (lazy way):

```
# apt-get install ntop
```

If you want to compile ntop, follow these steps:

You need the following packages:

- ntop : display network usage in top-like format (>= 3.0)
- libgd-dev : GD Graphics Library (>= 1.8.4)

```
# apt-get install libgd-dev
# apt-get source ntop
```

Patch ntop:

```
# cd ntop-XXX/
# patch -p0 < $OSSIM_PATH/contrib/ntop/ntop3-ossim.patch
# aclocal-1.6
# autoheader
# autoconf
# automake-1.6 --add-missing --gnu
```

Edit debian/rules file and remove '--enable-i18n' option from the 'configure-stamp-ntop' section.

Build ntop .deb:

```
$ dpkg-buildpackage -rfakeroot -uc -b
```

NOTE: use -d option to solve circular dependency problems (if needed).

Install it:

```
$ cd ..
$ dpkg -i ntop_XXX.deb
```

You must define the password for the admin user of ntop the first time you start ntop:

```
# ntop -u ntop
>> Please enter the password for the admin user:
# ^C
```



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

Now, the ntop daemon should be listening at port 3000, and 3001 (ssl).

Go to 'http://yourhost:3000/' to see Ntop in action.

Activate the rrdPlugin at Admin->plugins. Click on Host at Data Dump and specify your netmask at Hosts Filter.

Note: The RRD Files Path should be set to /var/lib/ntop/rrd

In order to make ntop to work with ips instead of macs (needed by agent), edit '/etc/default/ntop' file and add "--no-mac" to GETOPT="".

Set ntop variables into framework configuration:

```
ntop_link=https://your_ntop_host:3001/
rrdpath_ntop=/var/lib/ntop/rrd
```

## 4.9. Nmap

The framework has an active service detector that uses nmap. It will be replaced with a passive detector using agents, but in the meantime you need to install nmap.

- nmap : The Network Mapper (>= 3.70)

```
# apt-get install nmap
```

## 4.10. Nessus client

For more information about Ossim and Nessus integration please take a look at README.nessus

You can use Nessus in distributed mode with one nessus server on many sensors or you can have only a central nessus to scan all the hosts.

Any way you choose you need to have a nessus client on the framework. See 5.2 to install nessus server first.

So you'll need the following package for the client :

- nessus : Remote network security auditor, the client (>= 2.0.10)
- nessus-plugins : Nessus plugins (>= 2.0.12)

```
# apt-get install nessus nessus-plugins
```

Update `/etc/ossim/framework/ossim.conf` with the following lines :

```
nessus_user=ossim
nessus_pass=your_password (see 5.2 for creating users)
nessus_host=yourhost
nessus_port=1241
nessus_path=/usr/bin/nessus
nessus_rpt_path=$OSSIM_PATH/www/vulnmeter/
```

We need to approve the nessus server certificate, to do it just run against each nessus server :

```
# nessus -s -q xxx.xxx.xxx.xxx 1241 ossim your_nessus_pass
```

Where `xxx.xxx.xxx.xxx` is the adress of nessus server.

Follow the on-screen instructions. Personally I set the paranoia to 2.

To update nessus plugins (ie nessus attack scripts) and update ossim database with them try :

```
# nessus-update-plugins
# perl $OSSIM_PATH/scripts/update_nessus_ids.pl
```

It's a good idea to update plugins before each scan. To start a scan run the script :

```
# $OSSIM_PATH/scripts/do_nessus.pl
```

## 4.11. PDF reports

From `fpdf` web page:

"FPDF is a PHP class which allows to generate PDF files with pure PHP, that is to say without using the `PDFlib` library. The advantage is that `PDFlib` requires a fee for a commercial usage. F from `FPDF` stands for Free: you may use it for any kind of usage and modify it to suit your needs".

To Install it on debian :

- `php-fpdf` : PHP class to generate PDF files (`>= 1.52`)

```
# apt-get install php-fpdf
```

More detail on `FPDF` website (<http://www.fpdf.org>) in french, english, spanish, italian and dutch.

# Chapter 5. Agents

Install the following packages and their respective dependencies:

- python : An interactive high-level object-oriented language ( $\geq 2.3.4$ )
- python-dev: Header files and a static library for Python ( $\geq 2.3.4$ )

```
# apt-get install python python-dev
```

You need install python-mysqldb and/or python-psycopg2 depending of the plugins that you activate. For example, the plugin for OpenNMS reads from a PostgreSQL database.

- python-mysqldb : A Python interface for MySQL ( $\geq 1.0.0$ )
- python-psycopg2 : A Python DB-API 2.0 interface to PostgreSQL ( $\geq 2.4.0$ )

```
# apt-get install python-mysqldb python-psycopg2
```

Copy agent configuration file to /etc/ossim/agent directory:

```
# mkdir /etc/ossim/agent
# cp $OSSIM_PATH/etc/agent/config.xml /etc/ossim/agent
```

Edit config.xml (please, read the notes written in the file). In this file you can:

- change the address where the server is listening,
- activate/deactivate watchdog,
- activate/deactivate plugins,
- configure plugins,
- etc.

Install OS-SIM Agent:

```
# cd $OSSIM_PATH/agent
# python setup.py install
```

Run agent this way:

```
# ossim-agent -v
```

to see possible errors.

Type:

```
# ossim-agent --help
```

for more help or consult the man page.

Don't forget to authorize your sensors to access to your database, on your database server type :

```
$ mysql -u root -p
mysql> GRANT INSERT, SELECT on snort.* to root@sensor_ip IDENTIFIED BY 'mysql_password';
```

```
mysql> GRANT INSERT, SELECT on ossim.* to root@sensor_ip IDENTIFIED BY 'mysql_password';
mysql> exit;
```

The next sections will explain some of the plugins that you can use with OSSIM:

## 5.1. Plugins

### 5.1.1. Snort

OSSIM uses Snort as NIDS, and Acid to visualize alerts via Web.

If you use ossim.net as a source for Debian Package (cf Intro) install snort with mysql support:

- snort-mysql : Flexible Network Intrusion Detection System (>= 2.2.0)
- ```
# apt-get install snort-mysql
```

If you want to compile snort, follow these steps:

```
# apt-get source snort-mysql
```

Patch snort with spade patch that you will find at \$OSSIM\_PATH/contrib/spade/Spade-XXXXXX.tgz

Edit path to snort in the spade Makefile.

```
# tar -xvzf Spade-XXXXXX.tgz
# cd Spade-XXXXXX
# vi Makefile <-- edit SNORTBASE variable
# make
```

Patch snort with the OSSIM patch:

(Read this post ([https://sourceforge.net/forum/message.php?msg\\_id=2627915](https://sourceforge.net/forum/message.php?msg_id=2627915))) to know what the patch do)

```
# cd snort-X.X.X
# patch -p0 < $OSSIM_PATH/contrib/snort-2.1-ossim.patch
```

Compile snort:

```
# dpkg-buildpackage -rfakeroot -uc -b
```

Install debs generated:

```
$ cd ..
$ dpkg -i snort-mysql_XXX.deb snort-common_XXX.deb
  snort-rules-default_XXX.deb snort-doc_XXX.deb
```

Installing debs, you will be asked for the database configuration. Use these values:

```
hostname = where_your_db_is
database = snort
user      = root
password = yourMySQLpass
```

If you want to re-configure snort, just type 'dpkg-reconfigure snort-mysql'. You can also edit the configuration files at '/etc/snort/snort.conf'.

Copy \$OSSIM\_PATH/contrib/spade.conf.sample to /etc/snort/spade.conf and configure it. Make sure that the lines 'var HOME\_NET' and 'var EXTERNAL\_NET' of the snort configuration file have valid values.

Example:

```
var HOME_NET [192.168.1.0/24]
var EXTERNAL_NET !$HOME_NET
```

Make sure that snort logs alerts to syslog, uncommenting the 'output alert\_syslog: LOG\_AUTH LOG\_ALERT' line from the snort configuration file.

Add 'logfile' argument to the config file at 'output database' entry, so snort will log alerts on fast format (needed by agent):

```
"output database: alert, mysql, user=root dbname=snort host=yourdbhost logfile=fast.log"
```

Uncomment the following rules:

```
# include $RULE_PATH/web-attacks.rules
# include $RULE_PATH/backdoor.rules
# include $RULE_PATH/shellcode.rules
# include $RULE_PATH/policy.rules
# include $RULE_PATH/porn.rules
# include $RULE_PATH/info.rules
# include $RULE_PATH/icmp-info.rules
# include $RULE_PATH/virus.rules
# include $RULE_PATH/chat.rules
# include $RULE_PATH/multimedia.rules
# include $RULE_PATH/p2p.rules
```

You can make a request to the web server \*from other machine\* in order to verify that snort is installed correctly:

```
$ telnet OSSIM_host 80
GET /cmd.exe HTTP/1.1
```

Checking the file 'var/log/auth.log' and '/var/log/snort/fast.log' you should see an alert of type 'WEB-IIS cmd.exe access'.

Check out <http://www.bleedingsnort.com> for up-to-date, bleeding edge snort rules. The false positive rate is extremely low for little tested signatures and they are being very useful to us.

```
# cd /etc/snort/rules/
# wget http://www.bleedingsnort.com/bleeding-all.rules
# echo "include \${RULE_PATH}/bleeding-all.rules" >> /etc/snort/snort.conf
```

### 5.1.2. Ntop

See section 4.6.

TODO: rrd\_plugin

### 5.1.3. tcptrack

It's quite easy, you install the source package, patch it, recompile it and install it :

```
# apt-get source tcptrack
# cd tcptrack-x.x.x
# patch -p1 < ${OSSIM_PATH}/tcptrack-1.1.3-ossim.patch
# dpkg-buildpackage -rfakeroot -uc -b
# cd ..
# dpkg -i tcptrack_1.1.3-1_i386.deb
```

A good way to run tcptrack is:

```
# tcptrack -i $your_interface -P 4003 -D 2> /var/log/tcptrack.log
```

### 5.1.4. Pads

Passive Asset Detection System (<http://passive.sourceforge.net/>)

There is no pads package for Debian. You have to download/compile yourself. Follow the steps described at web site (<http://passive.sourceforge.net/download.php>).

A good way to run pads is:

```
# /usr/local/bin/pads -i $your_interface -D -w /var/log/assets.csv
```

### 5.1.5. p0f

It's very simple:

```
# apt-get install p0f
```

A good way to run p0f is:

```
# p0f -i $your_interface -lUNtd -o /var/log/p0f.log
```

### 5.1.6. arpwatch

It's very simple:

```
# apt-get install arpwatch
```

A good way to run arpwatch is:

```
# arpwatch -d -i $your_interface -f /var/lib/arpwatch/arp.dat >> \
/var/log/arpwatch.log &
```

Take a look at /etc/arpwatch.conf for the Debian-specific way to watch multiple interfaces. For other distributions you will need the patch provided by OS-SIM sources.

## 5.2. Nessus server

So you'll need the following package for the client :

- `nessusd` : Remote network security auditor, the server (>= 2.0.12)

```
# apt-get install nessusd
```

For communication with client Nessus need a certificate to create that is generate during the install process. Then you need to create a user to allow your nessus client to connect :

```
# nessus-adduser
```

You will have something like that :

```
Login          : ossim
Password       : nessus
DN             :
Rules          :
```

## 5.3. OpenNMS

Installing OpenNMS on Debian Sarge should look tricky at first glance. Here's a small workaround to succeed it.

### 5.3.1. Java SDK

OpenNMS is programmed in Java, so the first task is to install the Java SDK. You can follow the instruction given by the Debian FAQ (<http://www.debian.org/doc/manuals/debian-java-faq/ch11.html#s11.2>)

```
# mkdir /usr/local/src/java/1.4.2 && cd /usr/local/src/java/1.4.2
```

Then download the `j2sdk-1_4_2_05-linux-i586.bin` from Java web site (<http://java.sun.com>). You have to go to the web site and accept the policy of sun in order to get an url to download the file. When download is completed :

```
# chmod u+x j2sdk-1_4_2_05-linux-i586.bin
# ./j2sdk-1_4_2_05-linux-i586.bin
# mv j2sdk1.4.2_05/ /usr/local/lib/
# ln -s /usr/local/lib/j2sdk1.4.2_05/ /usr/local/lib/j2sdk
# apt-get install java-common equivs
# mkdir -p /usr/local/source/java/pkg
# cd /usr/local/source/java/pkg
# equivs-build java-compiler-dummy.control
# equivs-build java-virtual-machine-dummy.control
# equivs-build java1-runtime-dummy.control
# equivs-build java2-compiler-dummy.control
# equivs-build java2-runtime-dummy.control
# dpkg -i java-compiler-dummy_1.0_all.deb \
  java-virtual-machine-dummy_1.0_all.deb \
  java2-compiler-dummy_1.0_all.deb \
  java2-runtime-dummy_1.0_all.deb
# update-alternatives --verbose --install /usr/bin/java java \
  /usr/local/lib/j2sdk/jre/bin/java 500 --slave /usr/share/man/man1/java.1 \
  java.1 /usr/local/lib/j2sdk/man/man1/java.1
```

To check if java is correctly installed, type :

```
# java -version
```

### 5.3.2. Tomcat

OpenNMS needs at least Tomcat 4. This package is still in the unstable branch of Debian. So you need to add it to `/etc/apt/sources.list` file, for example :

```
deb ftp://ftp.debian.org/debian unstable contrib
deb ftp://ftp.debian.org/debian unstable main
```

Then update tree and install tomcat package.

```
apt-get update
apt-get install -t unstable tomcat4
```



### 5.3.3. RRDTool, Postgresql and perl

Make sure to have at least PostgreSQL 7.1, RRDTool 1.0.28, and pgsq module for perl :

```
# apt-get install postgresql rrdtool libdbd-pg-perl
```

### 5.3.4. OpenNMS installation

You have to get the debian package from OpenNMS server, so add the following line to your /etc/apt/sources.list :

```
deb http://debian.opennms.org/ debian/opennms unstable
```

The simply get the packages :

```
# apt-get install opennms
```

Note that the installation fix the following

- Postgresql user : opennms
- Postgresql password : opennms
- Database name : opennms

### 5.3.5. Configuration

Change this lines in /etc/default/tomcat4 :

```
JAVA_HOME=/usr/local/lib/j2sdk
TOMCAT4_USER=tomcat4
```

In /etc/postgresql/pg\_hba.conf change

```
host opennms opennms 127.0.0.1 255.0.0.0 password
```

by

```
host opennms opennms 127.0.0.1 255.0.0.0 trust
```

Finally in /etc/default/opennms :

```
JAVA_HOME=/usr/local/lib/j2sdk
```

Now you can start opennms and tomcat

```
# /etc/init.d/opennms start
# /etc/init.d/tomcat4 start
```

You can see OpenNMS in action at [http://your\\_host:8180/opennms/](http://your_host:8180/opennms/) Note that it uses port 8180 on Debian instead of the classical 8080. The default user and password are admin / admin.

## Chapter 6. Tools

To start everything :

```
# cp $OSSIM_PATH/contrib/debian/init.d/ossim /etc/init.d/
```

and then :

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

You can also use option stop and restart (self-explaining options). If you want to run ossim on startup:

```
# update-rc.d ossim defaults
```

# Chapter 7. TODO

```
renum.pl
create_sidmap.pl
chkconfig.pl (need install libcompress-zlib-perl)
test-directive.pl ?
Is mac.pl os.pl netbios.pl services.pl get_date.pl get_rrd_valur.pl still in
    use ?
```

dramatically improvement of the packet capture speed?  
[http://www.ntop.org/PF\\_RING.html](http://www.ntop.org/PF_RING.html)

**Experimental:** Actually the script `draw_fournier.pl` has some functions in test in order to draw and tune rrd files of NTop. In the header of the script there's a description of the different paramaters that you can pass to it. For example to see a rrd, use a link like this :

```
http://ossim\_server/cgi-bin/draw\_fournier.pl?what=anomaly&ip=xxx.xxx.xxx.xxx&start=now-12h&end=now&file=
```

Take care to allow apache user to read the rrd file (you can set this in ntop rrd\_plugin configuration page).

To tune a Holt-Winter parameter try something like this :

```
http://ossim\_server/cgi-bin/draw\_fournier.pl?what=tune&hwparam=alpha&ip=xxx.xxx.xxx.xxx&start=now-12h&er=
```

Now you have to grant write access to rrd file to the Apache user. The Holt-Winter parameters are : alpha, beta, gamma, gamma-deviation, deltapos, deltaneg, threshold and window\_length.

You can also try [http://ossim\\_server/report/anomalies.php](http://ossim_server/report/anomalies.php)