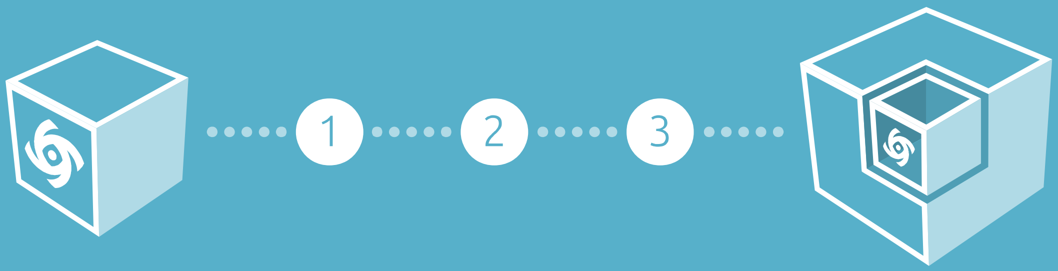
NICOLAS ZIN

OSSEC HOWTO

THE QUICK AND DIRTY WAY



Savoir-faire
LINUX

CC BY-SA (3.0)

TABLE OF CONTENT

PREFACE....................4
About the Author....................4
About Savoir-faire Linux....................4
Credits....................5
1 INTRODUCTION....................8
1.1 If you are in a hurry, and be on track....................9
2 FIRST STANDALONE INSTALLATION....................12
2.1 Installation....................12
2.1.1 Package installation....................12
2.1.2 Manual installation....................13
2.2 Configuring OSSec....................17
2.2.1 Global section....................18
2.2.2 Collector section....................19
2.2.3 Syscheck....................20
2.2.4 Rules....................20
2.2.5 alerts....................21
2.2.5.1 A simple alert example....................21
2.2.5.2 An alert_by_email alert example....................22
3 CLIENT-SERVER CONFIGURATION....................24
3.1 Server installation....................24
3.2 Agent installation....................25
3.3 Deploy “en masse”....................29
3.3.1 First method: ossec-authd....................29
3.3.2 Second method: Automate it....................30
3.3.3 Third method: Deb, rpm and puppet....................31
4 GRAPHICAL USER INTERFACE....................34
4.1 OSSec-wui....................34
4.2 Splunk....................34
5 RULES MANAGEMENT....................40
5.1 A good configuration strategy....................40
5.2 Tweaking alerts....................41
5.2.1 alerts/email_alert_level....................41
5.2.2 email_alerts....................41

5.3 Tweaking rules: rules/local_rules.xml 42
5.3.1 Shut-up a rule you don’t care 43
5.3.2 Variant: shut-up email alert you don’t care 43
5.3.3 Test your rule! 44
5.4 Enable active-response 46
5.4.1 1st example: block ssh bruteforce attempt 47
5.4.2 2nd example: block longer repetitive offender 47
5.4.3 3rd example: block all alerts above level 10 48
5.4.4 4th example: do not receive alerts that are blocked by active-response 48
5.5 Writing your own rules 49
5.5.1 Blocking brute-force attack (frequency and time-frame) 49
5.5.2 Blocking DoS on a web server (change rule level + frequency and timeframe) 50
5.5.3 Blocking on specific hosts (hostname option) 51
5.5.4 Blocking web attack on specific keywords (regex) 51
5.5.5 Summary: avoid the false positive 31151 rule pitfall 58
5.5.6 Receiving active-response actions 59
6 TO GO FURTHER 62
6.1 Geoip support 62
6.2 PCI-DSS coverage 64
6.3 ossim 64
6.4 OSSec and LXC 65
6.5 Snoopy 65

PREFACE

About the Author

Nicolas Zin

With a degree in computer engineering from EFREI (France, 1999) and an Executive MBA (2009) , **Nicolas Zin** worked for over 14 years in Information Technology as a consultant engineer, IT manager and trainer. In April 2010, he joined **Savoir-faire Linux** in Montreal where he operates as a system architect and a project manager for major mandates of the Infrastructure department .

Nicolas is a specialist in databases, computer security and automation of advanced configuration of large parks servers (with Puppet for example). He holds several Redhat certifications (RHCSA, RHCE, RHCI and RHCX) and he courses in project management (PMI) in 2012. In his spare time, he enjoy volleyball and video. He also loves performing at security competitions such as Hackfest and NorthSec where you may often see him.

About Savoir-faire Linux

Savoir-faire Linux is the leading Free/Open Source Software service supplier in Quebec and Canada. Since 1999, the company has developed a unique expertise and brings it to companies and public organizations to meet the challenges of their ever changing information systems.

With a multidisciplinary team of near 80 consultants, it serves today a customer base of over 500 organizations, including Quebec and Canadian government organizations, major international agencies, industry giants and Quebec SMEs / SMIs. Savoir-faire Linux has its headquarters in Montreal and offices in Quebec and Ottawa.

ISO 9001 and ISO 14001 certified, it has a strong presence in the Free Software community. Member of the prestigious “Linux Foundation”, **Savoir-faire Linux** provides major contributions to many open source software projects.

Savoir-faire Linux
7275, Saint-Urbain, Suite 200
Montreal (Quebec), H2R 2Y5, Canada

Tel.: **+1 514 276-5468** – Fax: **+1 514 276-5465**
http://www.savoirfairelinux.com
contact@savoirfairelinux.com

Certified System
ISO 14001
ISO 9001

Credits

- Edition: **Savoir-faire Linux**
- Author: **Nicolas Zin** (nicolas.zin@savoirfairelinux.com)
- Cover and layout: **Michaël Veilleux**
- Reference: **SFL-ED01**
- This work is licensed under a: **CC BY-SA (3.0)**

cc BY SA

INTRODUCTION

CHAPTER 1

1 INTRODUCTION

15 years ago, I tried to installed an IDS: snort. After 2 days, I gave up: it was painful to install, docs were not as prevalent as now, and most importantly, it didn't come with predefined rules. Before beginning to have something working, I had to spend 2 days, to write regular expression and be lucky to catch something.

Days have changed, I guess snort too, but before trying back snort,
I found OSSec.

Initially one of my clients asked me to monitor its infrastructure of more than 60 servers. Basically a centralized syslog server should do the work, but to analyzed so much data, syslog (and my poor eyes) wasn't sufficient. Instead I installed OSSec.

OSSec is called an HIDS: a "Host Intrusion Detection System"; because it is mainly a Log Analyzer but not only. Some key facts about OSSec, it brings:

- log centralization service (a bit like "rsyslog"), but doesn't store logs, just analyze it;
- an analyzer and an alert system (a bit like "logstash");
- an "active-response" mechanism (like "fail2ban");
- a file monitoring system (like "tripwire");
- a small rootkit engine (like "rkhunter");
- in a light product: it is not written in ruby/python or java, but in plain old good C;
- and some potential features, that are beyond this book (OSSec is part of OSSIM: a SIEM, i.e. Security Information and Event Management, used for example to cover PCI-DSS requirements);
- but most important: a lots of rules already written (and some neaty features that respond to real problems) and a very convenient way to extend them!

OSSec is mainly useful for 3 things:

- see what is going on;
- stop brute-force attacks (ftp, web, ssh…);
- cover PCI-compliance requirements related to monitoring.

This how-to book is a quick and dirty guide for OSSec, it is not a reference book. If you need more, the author of OSSec (Daniel Sid) wrote a book in 2008 named “OSSec Host-Based Intrusion Detection Guide”.

1.1 If you are in a hurry, and be on track

- Check how to deploy en masse your ossec infrastructure (chapter 3) and don't forget to open UDP port 1514 if you are using a client/server infrastructure;
- Install the graphical user interface (chapter 4);
- Review the good configuration strategy (chapter 5);
- Cherry pick in chapter 5 to tweak your rules, setup active-response for alerts>10, and when happy, raise your alert level to 12.

Happy hacking!

FIRST STANDALONE INSTALLATION

CHAPTER 2

2 FIRST STANDALONE INSTALLATION

The best way to understand a product is to use it, so basically in this chapter I will:

- install a OSSec standalone server;
- let OSSec send me alerts;
- tweaks it to reduce “false positives”.

2.1 Installation

2.1.1 Package installation

For Windows, you can find on OSSec website installation software.

For Unix, and particularly Linux, you will not find OSSec package in standard distribution (at least Debian/Ubuntu and Redhat/CentOS), but you can find it in third party repo:

- for Redhat/Centos, check Atomic repo: http://www.atomicorp.com/channels/atomic/

```
# wget -q -O - https://www.atomicorp.com/installers/atomic |sh
# yum install ossec-hids ossec-hids-server (or ossec-hids-client for the agent)
```

- for Debian/Ubuntu, check my ppa repo: https://launchpad.net/~nicolas-zin/+archive/ossec-ubuntu

```
# apt-get install python-software-properties
# add-apt-repository ppa:nicolas-zin/ossec-ubuntu
# apt-get update
# apt-get install ossec-hids-local
```

For a standalone installation, use the “local” version. More on automatic methods in the chapter “Deploy en masse”.

2.1.2 Manual installation

On Unix, if you want to ensure to have the latest OSSec or do not want to trust third party package maintainer, grab the last version of OSSec at www.ossec.net, (at the time of writing it is ossec-hids-2.7.1), and install it.

Before installing it, 3 specifics details to know:

- it is not a “`./configure;make;make install`”, rather we have to launch “`install.sh`”;
- it will install everything in `/var/ossec` (nothing in `/usr/local`, no log in `/var/log`,...) except creation of 3 users (ossec, ossecm and ossecr) and put an init script. It is done on purpose for security reason (non privileges executable chroot into `/var/ossec`), but it helps for easy removal/update;
- You will need of course development tools:
  - on debian it is “`apt-get install build-essential libssl-dev`”;
  - on Redhat it is “`yum groupinstall 'Development Tools' `“,
    and “`yum install openssl-devel`”.

Let’s begin:

```
# wget http://www.ossec.net/files/ossec-hids-2.7.tar.gz
# tar -xvzf ossec-hids-2.7.tar.gz; cd ossec-hids-2.7
# ./install.sh

 ** Para instalação em português, escolha [br].
 ** 要使用中文进行安装, 请选择 [cn].
 ** Fur eine deutsche Installation wohlen Sie [de].
 ** Για εγκατάσταση στα Ελληνικά, επιλέξτε [el].
 ** For installation in English, choose [en].
 ** Para instalar en Español , eliga [es].
 ** Pour une installation en français, choisissez [fr]
 ** A Magyar nyelvű telepítéshez válassza [hu].
 ** Per l'installazione in Italiano, scegli [it].
 ** 日本語でインストールします. 選択して下さい. [jp].
 ** Voor installatie in het Nederlands, kies [nl].
 ** Aby instalować w języku Polskim, wybierz [pl].
 ** Для инструкций по установке на русском ,введите [ru].
 ** Za instalaciju na srpskom, izaberi [sr].
 ** Türkçe kurulum için seçin [tr].
 (en/br/cn/de/el/es/fr/hu/it/jp/nl/pl/ru/sr/tr) [en]: en

OSSec HIDS v2.7 Installation Script - http://www.ossec.net

 You are about to start the installation process of the OSSec HIDS.
 You must have a C compiler pre-installed in your system.
 If you have any questions or comments, please send an e-mail
 to dcid@ossec.net (or daniel.cid@gmail.com).
 - System: Linux TheShell 3.7-trunk-amd64
 - User: root
 - Host: TheShell

 -- Press ENTER to continue or Ctrl-C to abort. --

1- What kind of installation do you want (server, agent, local, hybrid or help)?
Local
 - Local installation chosen.

2- Setting up the installation environment.
 - Choose where to install the OSSec HIDS [/var/ossec]: /var/ossec
 - Installation will be made at /var/ossec .

3- Configuring the OSSec HIDS.

 3.1- Do you want e-mail notification? (y/n) [y]:
 - What's your e-mail address? nicolas.zin@gmail.com
 - What's your SMTP server ip/host? localhost

 3.2- Do you want to run the integrity check daemon? (y/n) [y]: y
 - Running syscheck (integrity check daemon).

 3.3- Do you want to run the rootkit detection engine? (y/n) [y]: y
 - Running rootcheck (rootkit detection).

 3.4- Active response allows you to execute a specific command based on the
events received. For example, you can block an IP address or disable access for
a specific user.
```

```
 More information at:
 http://www.ossec.net/en/manual.html#active-response

 - Do you want to enable active response? (y/n) [y]: y
 - Active response enabled.
 - By default, we can enable the host-deny and the
 firewall-drop responses. The first one will add a host to the /etc/hosts.deny
and the second one will block the host on iptables (if linux) or on ipfilter (if
Solaris, FreeBSD or NetBSD).
 - They can be used to stop SSHD brute force scans, portscans and some other
forms of attacks. You can also add them to block on snort events, for example.
 - Do you want to enable the firewall-drop response? (y/n) [y]: y
 - firewall-drop enabled (local) for levels >= 6
 - Default white list for the active response:
 - 127.0.0.1
 - 213.186.33.99
 - Do you want to add more IPs to the white list? (y/n)? [n]: n

 3.6- Setting the configuration to analyze the following logs:
 -- /var/log/messages
 -- /var/log/auth.log
 -- /var/log/syslog
 -- /var/log/mail.info
 -- /var/log/dpkg.log
 -- /var/log/apache2/error.log (apache log)
 -- /var/log/apache2/access.log (apache log)

 - If you want to monitor any other file, just change the ossec.conf and add a
new localfile entry. Any questions about the configuration can be answered by
visiting us online at http://www.ossec.net .

 --- Press ENTER to continue ---
```

And *voilà*, it compiles and installs OSSec in `/var/ossec` directory (you will have a summary of what has been done at the end).

Maybe for you first install, you would not want to install “active-response”, i.e. just have email alert.

Now, you just have to start it (else use `/var/ossec/bin/ossec-control`):

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

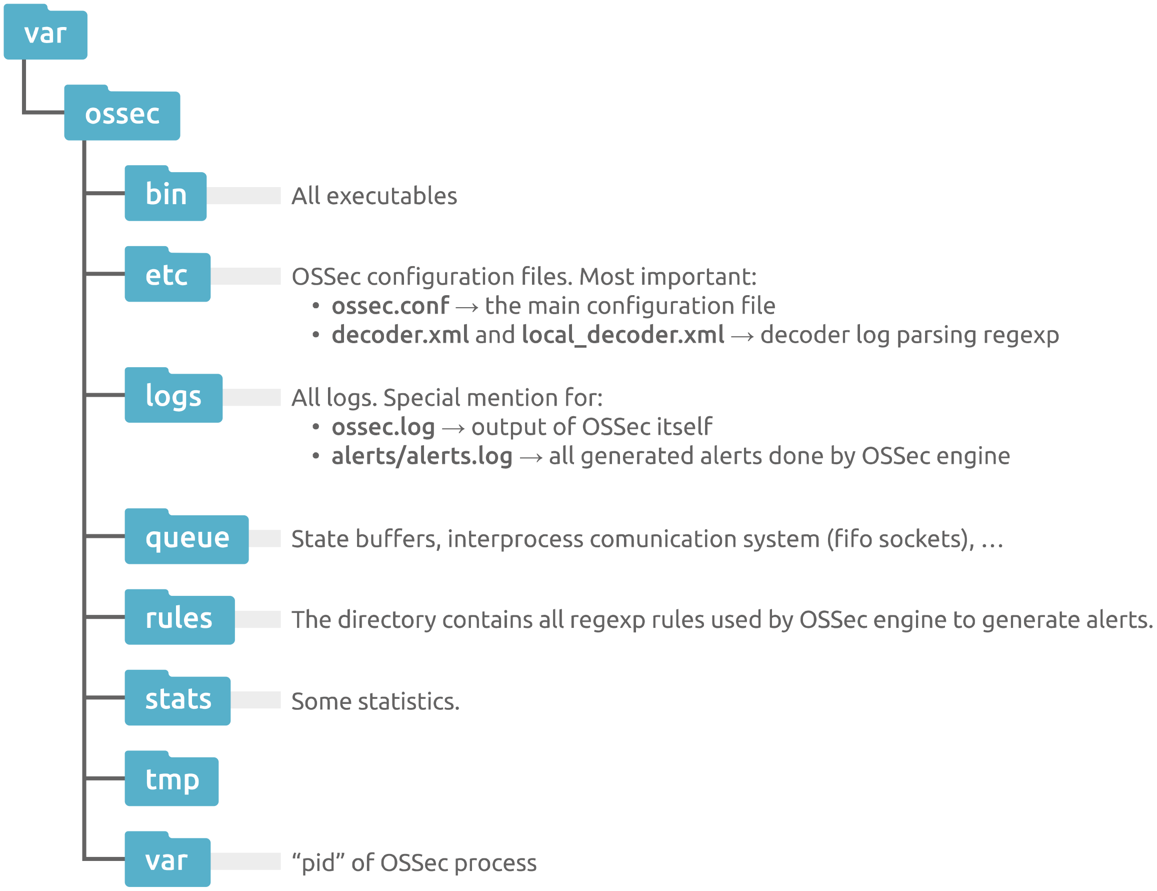
You can monitor it by having a look at `/var/ossec/logs/ossec.log`. And you will quickly receive some alerts. You can check also that OSSec process are running:

```
# ps ax | grep ossec
 4932 ? S 0:00 /var/ossec/bin/ossec-maild
 4937 ? S 0:00 /var/ossec/bin/ossec-execd
 4941 ? S 0:00 /var/ossec/bin/ossec-analysisd
 4947 ? S 0:00 /var/ossec/bin/ossec-logcollector
 4949 ? S 0:00 /var/ossec/bin/ossec-syscheckd
 4953 ? S 0:00 /var/ossec/bin/ossec-monitord
 5123 pts/1 S+ 0:00 grep --color=auto ossec
```

2.2 Configuring OSSec

So we just left everything by default and we started it. Now, we would like to know what is going on and tweak it.

First, a look at the `/var/ossec` directory. It is composed of several subdir; most important sub-directories for the moment are: etc, logs and rules.



All executables

OSSec configuration files. Most important:

- **ossec.conf** → the main configuration file
- **decoder.xml** and **local_decoder.xml** → decoder log parsing regexp

All logs. Special mention for:

- **ossec.log** → output of OSSec itself
- **alerts/alerts.log** → all generated alerts done by OSSec engine

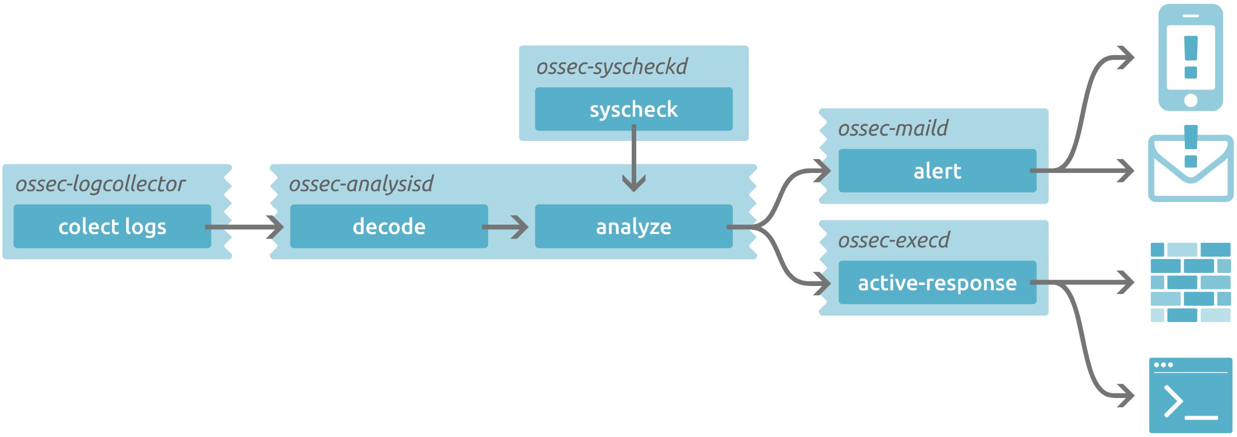
State buffers, interprocess comunication system (fifo sockets), …

The directory contains all regexp rules used by OSSec engine to generate alerts.

Some statistics.

“pid” of OSSec process

What we are interested for now is in `etc/ossec.conf`, it is the central configuration file used by all executable, that will act on this work-flow:



The `etc/ossec.conf` has 6 sections:

- global (global);
- rules (rules);
- syscheck (syscheck/rootcheck);
- alerts (alert);
- active-response (command/active-response);
- collector (localfile).

2.2.1 Global section

```
<global>
   <email_notification>yes</email_notification>
   <email_to>nicolas.zin@gmail.com</email_to>
   <smtp_server>mymailserver.mycompany.com</smtp_server>
   <email_from>ossecm@mycompany.com</email_from>

   <white_list>127.0.0.1</white_list>
   <white_list>^localhost.localdomain$</white_list>
   <white_list>213.186.33.99</white_list>
</global>
```

Here you find general information we set: where to send notification and which SMTP server. Change it if you don't receive alert, or want to white-list some host/ip.

> *In your `ossec.conf` file you get, maybe you will find 2 `<global>` sections: one for the global info, and a bit later on, one for the whitelist. OSSec aggregates them of course when it parses the config file.*

2.2.2 Collector section

```
<localfile>
   <log_format>syslog</log_format>
   <location>/var/log/dpkg.log</location>
</localfile>

<localfile>
   <log_format>apache</log_format>
   <location>/var/log/nginx/access.log</location>
</localfile>

[…]
```

At the end of the `ossec.conf` file, you will find a list of all files monitored by OSSec.

You will certainly add other files you want to monitor.

You can even add regular expressions. For example if you have split your apache log per virtual-host:

```
<localfile>
   <log_format>apache</log_format>
   <location>/var/log/apache/*_error.log</location>
</localfile>
```

2.2.3 Syscheck

```
<syscheck>
   <frequency>79200</frequency>

   <!-- Directories to check (perform all possible verifications) →
   <directories check_all="yes">/etc,/usr/bin,/usr/sbin</directories>
   […]

   <!-- Files/directories to ignore -->
   <ignore>/etc/mtab</ignore>
   […]
</syscheck>

<rootcheck>
   […]
</rootcheck>
```

You will certainly get quickly alert message from legitimates files that you want to ignore, like "`/dev/blkid`" (that changed each time a block device changed, for example when you add a USB stick). Just put it in the ignore section whatever seems inappropriate.

2.2.4 Rules

You don't need to touch to that (yet), it just specify rules files to load.

Normally, when you will need to write your own rules, you will edit the "`local_rules.xml`".

```
<rules>
   <include>rules_config.xml</include>
   […]
   <include>local_rules.xml</include>
</rules>
```

You can begin to have a look to files in `rules/` directory if you are curious.

2.2.5 alerts

```
<alerts>
   <log_alert_level>1</log_alert_level>
   <email_alert_level>6</email_alert_level>
</alerts>
```

`log_alert_level`*: level at which alert is stored (in log/alerts/alerts.log)*

`email_alert_level`*: level at which alert are sent by email*

By default, alerts are sent by email to the email provided in the “global” section. Alerts have level from 0 (useless) to 16 (critical). Each rule is configured with an alert level. And when a rule is fired, depending on its level, it is sent (or not) by email.

An alert level of 6 is a good one. If you receive too much alerts you can raise this level, but IMHO the best way is not to raise this level yet, but shutoff rules you don’t want, and afterwards raise it.

even with a level of 7, you will received some alerts that are below this threshold. Indeed some rules are marked to be sent whatever the level is (search for “`alert_by_email`*” in the rules files).*

2.2.5.1 A simple alert example

In `rules/pam_rules.xml`, you can find:

```
<rule id="5555" level="3">
   <match>: password changed for</match>
   <description>User changed password.</description>
</rule>
```

What it means:

- if in a log line we find ": password changed for";
- then we will fire an alert of level 3;
- with rule id "5555" stating "User changed password."

Of course you will not see it, because your current global alert level is 7. If you want to see it:

- either you change your global alert level;
- change the level of the rule directly in `rules/pam_rules.xml` (and relaunch ossec);
- or add in the ossec.conf file a specific alert filter.

2.2.5.2 An alert_by_email alert example

In `rules/ossec_rules.xml`, you will find:

```
<rule id="501" level="3">
   <if_sid>500</if_sid>
   <if_fts />
   <options>alert_by_email</options>
   <match>Agent started</match>
   <description>New ossec agent connected.</description>
</rule>
```

There are 2 tags here which I want to put emphasis on:

- `<if_fts />`, means "if first time seen": this rule will generate an alert the first time OSSec see it. It can be useful for example to track ssh login from user not expected to connect via ssh (apache for example);
- `<options>alert_by_email</options>`: it means, "whatever the global level selected, send it by email".

CLIENT-SERVER CONFIGURATION

CHAPTER 3

3 CLIENT-SERVER CONFIGURATION

OSSec is really powerful when you begin to install it on a whole infrastructure. It behave as a classical client/server software, on a OSSec specific protocol on UDP port 1514:

- agents are “passive”, they do not run rules, they send logs to the server, and receive active-response instructions from the server;
- the server gathers everything and triggers rules, alerts and active-response.

By the way, Active Response can be configured to act:

- on the agent where the source alert comes from;
- on the server itself (if you need to feed a database, or trigger an external action);
- on all agents (if you have a web-server farms, you want to stop people attacking you on all web-servers).

3.1 Server installation

Follow the same steps than for a local installation, except you specify server instead of local. (it will ask for “remote syslog enabled”, say “y” if you intend to receive syslog from devices where you cannot install OSSec agents: routers for example)

Open ports to let udp communication on UDP port 1514:

```
iptables -I INPUT 7 -p udp --dport 1514 -s mysubnet/24 -j ACCEPT
iptables -I OUTPUT 7 -p udp --sport 1514 -d mysubnet/24 -j ACCEPT
```

3.2 Agent installation

First compile and install OSSec following the same steps than for a local installation, except you specify “agent” instead.

The instructions are on http://www.ossec.net/doc/manual/agent/agent-management.html:

A) On the server: add the agent to the server “database”

In fact in `/var/ossec/etc/clients.keys`.

```
# /var/ossec/bin/manage_agents

****************************************
* OSSec HIDS v2.5-SNP-100809 Agent manager. *
* The following options are available: *
****************************************
   (A)dd an agent (A).
   (E)xtract key for an agent (E).
   (L)ist already added agents (L).
   (R)emove an agent (R).
   (Q)uit.
Choose your action: A,E,L,R or Q: a

Adding a new agent (use '\q' to return to the main menu).
Please provide the following:
   * A name for the new agent: agent1
   * The IP Address of the new agent: 192.168.2.0/24
   * An ID for the new agent[001]:
```

About the IP: it specifies the IP where the agent come from. You can type its IP, a CIDR range, or “any”, and give an uniq ID.

B) On the server: Now extracts its key:

```
Choose your action: A,E,L,R or Q: e
Available agents:
 ID: 001, Name: agent1, IP: 192.168.2.0/24

Provide the ID of the agent to extract the key (or '\q' to quit): 001
Agent key information for '001' is:
MDAyIGFnZW50MSAxOTIuMTY4LjIuMC8yNCBlNmY3N2RiMTdmMTJjZGRmZjg5YzA4ZDk5MmQ4NDE4MjYw
MjJkN2ZkMzNkYzZiOWE5NWY4MzU5YWRlY2JkY2Rm

** Press ENTER to return to the main menu.
Choose your action: A,E,L,R or Q: q
```

C) On the agent: grab the key and go back to the agent.

On the agent run `/var/ossec/bin/manage_agents`, and type “i”:

```
# /var/ossec/bin/manage_agents
****************************************
* OSSec HIDS v2.5-SNP-100809 Agent manager. *
* The following options are available: *
****************************************
 (I)mport key from the server (I).
 (Q)uit.
Choose your action: I or Q: i
Paste it here (or '\q' to quit): [key extracted via manage_agents on the server]
Agent information:
 ID:001
 Name:agent1
 IP Address:192.168.2.0/24

Confirm adding it?(y/n): y
Added.

** Press ENTER to return to the main menu.
Choose your action: I or Q: q
```

D) On the agent: restart the agent

```
# /var/ossec/bin/ossec-control restart
```

If you have some errors like:

```
root@ossecclient:~/ossec-hids-2.7# /var/ossec/bin/ossec-control restart
ossec-logcollector not running ..
ossec-syscheckd not running ..
ossec-agentd not running ..
ossec-execd not running ..
OSSec HIDS v2.7 Stopped
Starting OSSec HIDS v2.7 (by Trend Micro Inc.)…
Started ossec-execd…
2013/09/29 11:57:30 ossec-config(1230): ERROR: Invalid element in the
configuration: 'client'.
2013/09/29 11:57:30 ossec-config(1202): ERROR: Configuration error at
'/var/ossec/etc/ossec.conf'. Exiting.
2013/09/29 11:57:30 ossec-agentd(1215): ERROR: No client configured. Exiting.
```

Check that in the `/var/ossec/etc/ossec.conf`, you have at least specified where is the server to connect to (see http://www.ossec.net/doc/syntax/head_ossec_config.client.html for more options):

```
<ossec_config>
   <client>
      <server-ip>184.200.172.16</server-ip>
   </client>
<ossec_config>
```

E) On the server: check that the agent manage to connect to the server.

On the server run “`/var/ossec/bin/list_agents -c`”. You should see your new agent:

```
# /var/ossec/bin/list_agents -c
agent1-12.15.18.133 is active.
[…]
```

F) If it doesn’t work

- Have you opened UDP port 1514 on the agent ?
- Check the log on the agent and on the server.

Because you want to control everything from your OSSec server, note that you can define from the OSSec server, specific configuration per agent, especially log file or syscheck integrity files to monitor. Have a look to `<agent_config>` *in OSSec documentation*

3.3 Deploy “en masse”

If you have a lot of agents to deploy, there are ways to do it quicker. The trick is the client keys (`/var/ossec/etc/client.keys`). It i used to encrypt communication between client and server. and they have to be the same.

3.3.1 First method: ossec-authd

OSSec 2.7 comes with a new feature: `ossec-authd` (http://www.ossec.net/doc/programs/ossec-authd.html):

- it is a daemon you run on the server when you deploy your agent;
- it will populate your agents key;
- when you have finished to deploy, you stop it.

A) Create ssl keys

```
# openssl genrsa -out /var/ossec/etc/sslmanager.key 2048
# openssl req -new -x509 -key /var/ossec/etc/sslmanager.key -out
/var/ossec/etc/sslmanager.cert
```

You are about to be asked to enter information that will be incorporated into your certificate request. What you are about to enter is what is called a Distinguished Name or a DN. There are quite a few fields but you can leave some blank.

For some fields there will be a default value, If you enter '`.`', the field will be left blank.

```
Country Name (2 letter code) [AU]:CA
State or Province Name (full name) [Some-State]:Quebec
Locality Name (eg, city) []:Montreal
Organization Name (eg, company) [Internet Widgits Pty Ltd]:My Company
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:mycompany.com

Email Address []:nicolas.zin@gmail.com
```

B) Open port on firewall

By default ossec-auth will listen on port 1515 (you can change that with "`-p`" argument) .

```
iptables -I INPUT 7 -p tcp -m tcp --dport 1515 -s mysubnet/24 -j ACCEPT
```

C) run ossec-auth

```
# /var/ossec/bin/ossec-authd
```

D) agent-auth on agents

```
# /var/ossec/bin/agent-auth -m <ip of ossec server> -A <agent name>
```

3.3.2 Second method: Automate it

If you have a look to `/var/ossec/etc/client.keys`, you will see the format is quite simple:

`<agent id> <agent name> <ip range> <cipher>`

Field	Value
agent id	It should be a uniq id, from 0 to 2048 (can be change in src)
agent name	Whatever you like
ip range	Either ip, or a CIDR (i.e. 192.168.10.0/24 for example) or "any"
cipher	In fact it is 2 md5sum concatenated. You can create your own

So if you want to deploy on scale, you can script it, and put whatever cipher you want. The alone problem its to have unique id.

This is great, but you still need to compile agents on all servers: in default Linux distribution there are no official packages. You can certainly automate having a script that:

- copy `/var/ossec`
- create the correct users (`ossec`, `ossecm`, `ossecr`)
- copy `/etc/ossec-init.conf` and `/etc/init.d/ossec`

3.3.3 Third method: Deb, rpm and puppet

Personally, I tend to use configuration management tools to administer large infrastructure. I try to automate things as far as possible.

To deploy OSSec I:

- wrote deb package, you can find on launchpad: https://launchpad.net/~nicolas-zin/+archive/ossec-ubuntu. I try hard to follow OSSec releases;
- wrote a puppet recipe that can be found on github: https://github.com/nzin/puppet-ossec. This one is lagging a bit.

It is more restrictive than the OSSec generic way (works just for Linux), you want probably not want to install a security tools that you have not compiled, these are not perfect (the puppet recipe can be lagging), but was useful few times for me, and so maybe can be for you.

GRAPHICAL USER INTERFACE

CHAPTER 4

4 GRAPHICAL USER INTERFACE

Before modifying your OSSec configuration, maybe you want to see what is going on.

4.1 OSSec-wui

- http://www.ossec.net/wiki/index.php/OSSecWUI:Install

This is a small PHP web interface you have to install on your OSSec server that will dig through your alerts for statistics. But it is quite limited. You can check what are the most frequent alerts for a particular day, but you can get for example the “top ten alerts above level 7 during the last 10 days”

4.2 Splunk

There is a page related to splunk on ossec (http://www.ossec.net/?p=402), but not up to date.

In this quick how-to, I will install splunk on the same server as OSSec. It is possible to have ossec on one server sending info via syslog to splunk on another, which is out of scope here.

A) Download application

Go to http://www.splunk.com/download, and download.

- Also go to http://splunk-base.splunk.com/apps/22285/splunk-for-ossec-splunk-v4-version, register and download the app
- Now you should have something similar to:
    - *splunk-5.0.2-149561-linux-2.6-x86_64.rpm*
    - *reporting-and-management-for-ossec_1189.tgz*

B) Install splunk

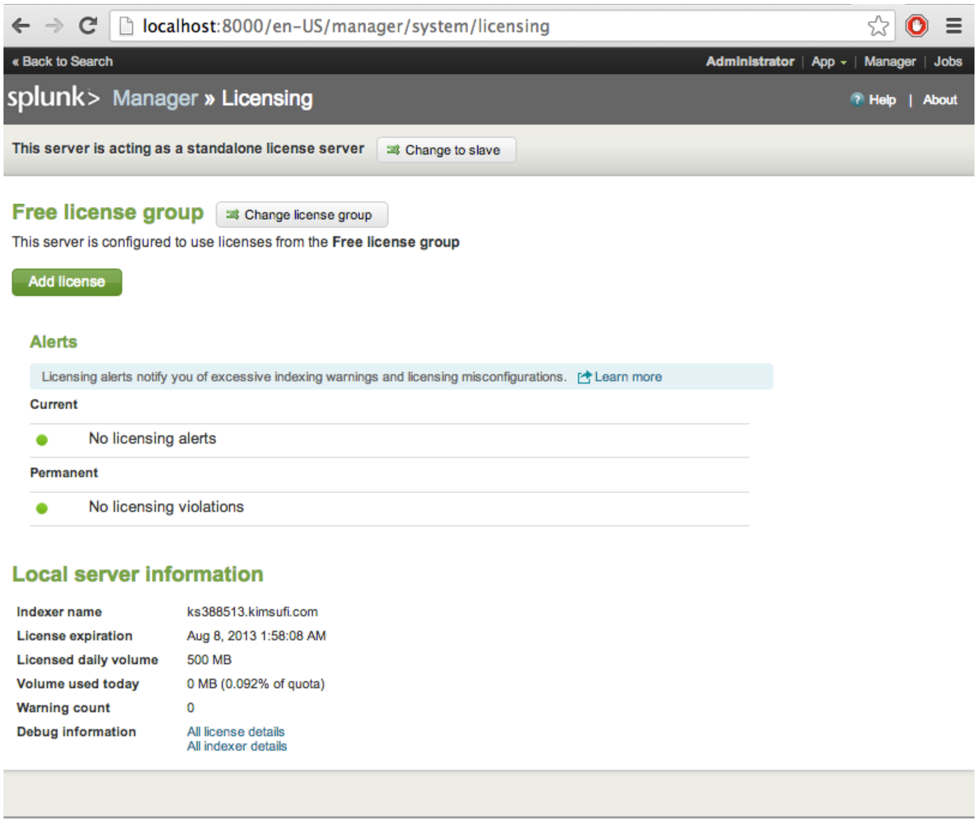
On a terminal:

```
# rpm -i ./splunk-5.0.2-149561-linux-2.6-x86_64.rpm
# cd /opt/splunk/etc/apps/
# tar -xvzf /root/reporting-and-management-for-ossec_1189.tgz
# chown -R splunk:splunk ossec
# /opt/splunk/bin/splunk enable boot-start
# /opt/splunk/bin/splunk start
```

C) Configure it

Now, go with a browser to http://splunkserver:8000:

- change password (by default it is **admin/changme**);
- for sake of soul, change the licence. Click on:
  - Settings; (in the upper right corner)
  - Licensing;
  - “Change license group”;
  - Click on “Free license”;
  - Save ;
  - Restart now.


localhost:8000/en-US/manager/system/licensing
« Back to Search
Administrator | App | Manager | Jobs
splunk> Manager » Licensing
Help | About
Change to slave
Change license group
Add license
Learn more

This server is acting as a standalone license server

Free license group

This server is configured to use licenses from the **Free license group**

Alerts

Licensing alerts notify you of excessive indexing warnings and licensing misconfigurations.

Current

● No licensing alerts

Permanent

● No licensing violations

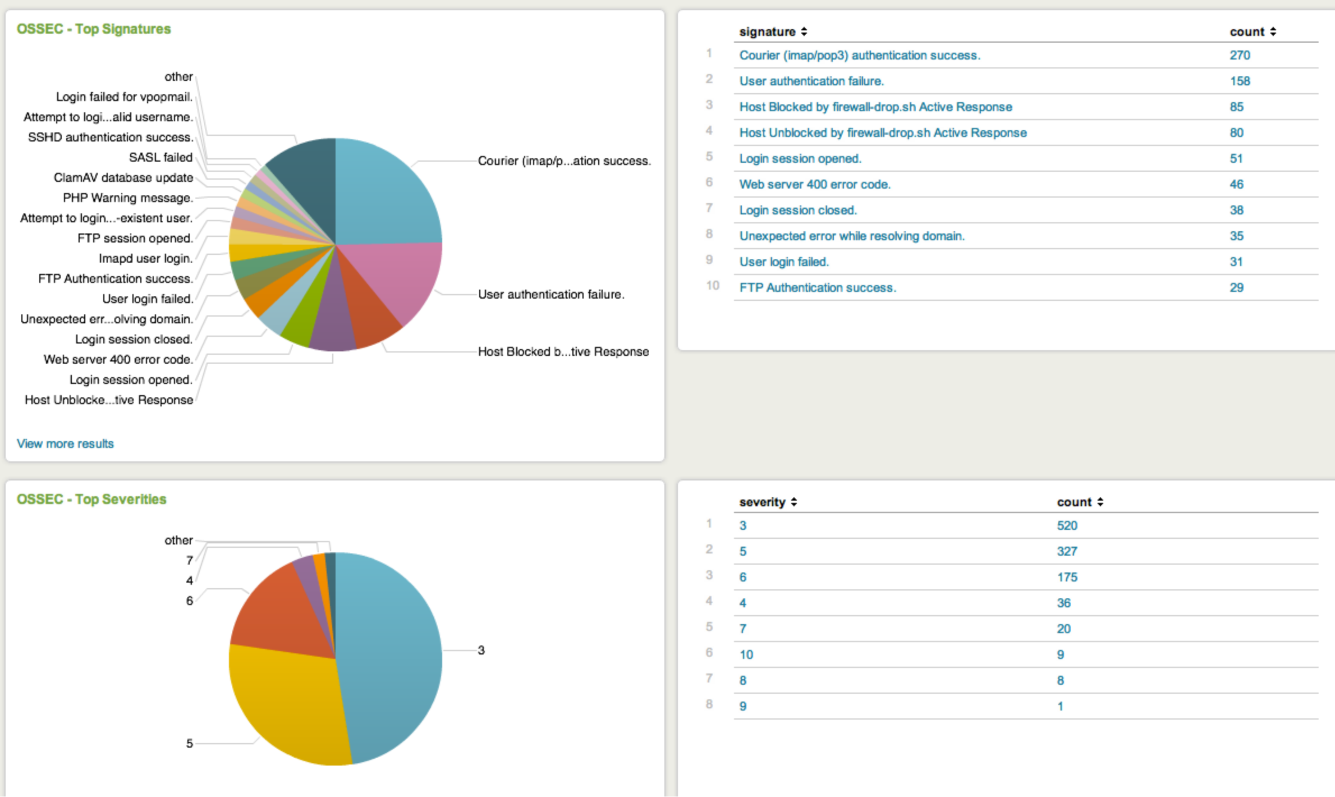
Local server information

Indexer name	ks388513.kimsufi.com
License expiration	Aug 8, 2013 1:58:08 AM
Licensed daily volume	500 MB
Volume used today	0 MB (0.092% of quota)
Warning count	0
Debug information	All license details All indexer details

D) Use it

- Go back to the main page again;
- Go into App, **Splunk for OSSec**;

- In “Search&Reports” you can find preconfigured reports, like these one:


OSSEC - Top Signatures
other
Login failed for vpopmail.
Attempt to logi...alid username.
SSHD authentication success.
SASL failed
ClamAV database update
PHP Warning message.
Attempt to login...-existent user.
FTP session opened.
Imapd user login.
FTP Authentication success.
User login failed.
Unexpected err...olving domain.
Login session closed.
Web server 400 error code.
Login session opened.
Host Unblocke...tive Response
Courier (imap/p...ation success.
User authentication failure.
Host Blocked b...tive Response
View more results
OSSEC - Top Severities
other
7
4
6
5
3

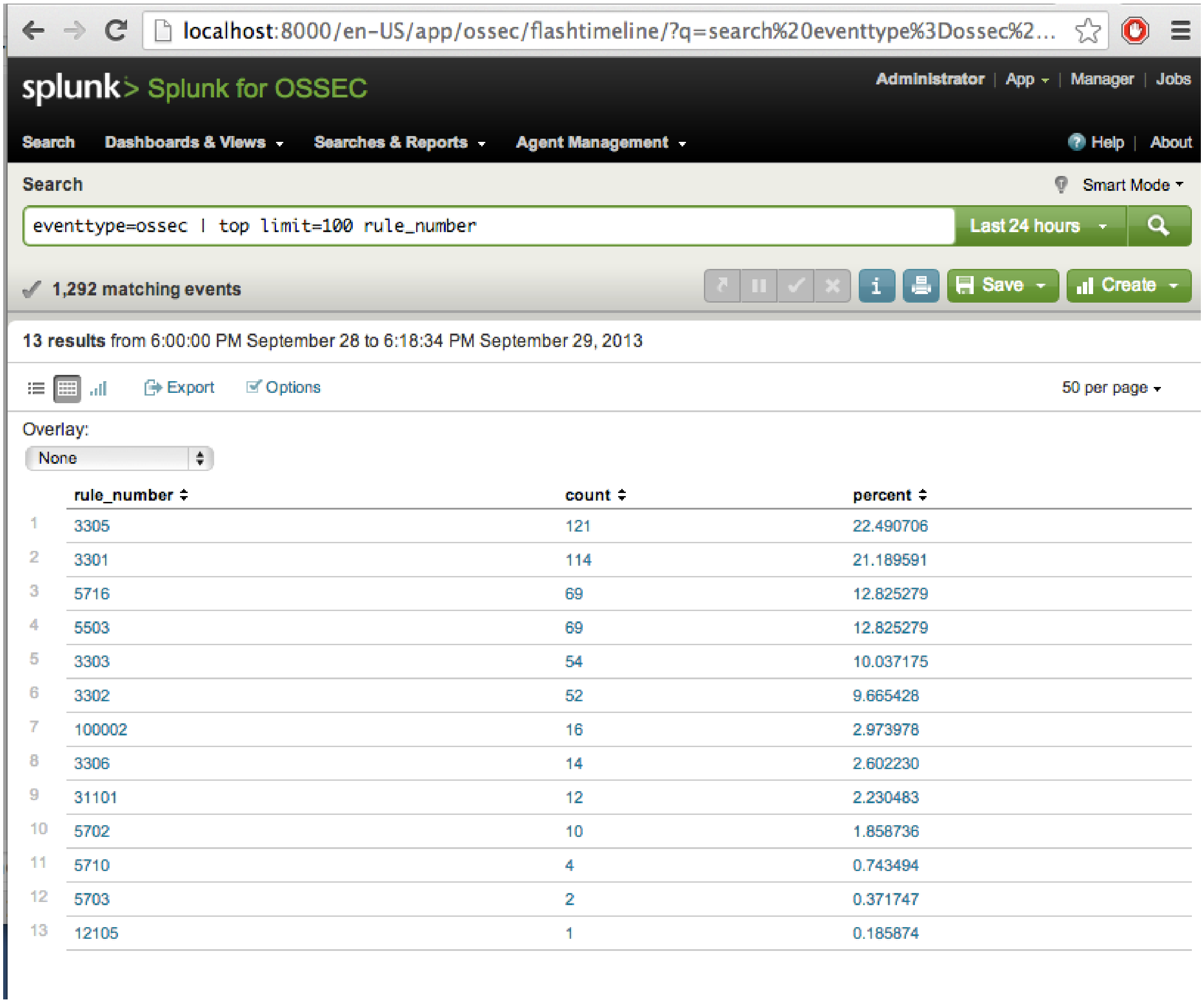
	signature ⬍	count ⬍
1	Courier (imap/pop3) authentication success.	270
2	User authentication failure.	158
3	Host Blocked by firewall-drop.sh Active Response	85
4	Host Unblocked by firewall-drop.sh Active Response	80
5	Login session opened.	51
6	Web server 400 error code.	46
7	Login session closed.	38
8	Unexpected error while resolving domain.	35
9	User login failed.	31
10	FTP Authentication success.	29

	severity ⬍	count ⬍
1	3	520
2	5	327
3	6	175
4	4	36
5	7	20
6	10	9
7	8	8
8	9	1

After that you will be able to dig, and create your own report, by mastering the search bar and its configuration.

For example, you have the top 100 of all rules:

```
eventtype=ossec | top limit=100 rule_number
```



```
eventtype=ossec | top limit=100 rule_number
```

✓ 1,292 matching events

13 results from 6:00:00 PM September 28 to 6:18:34 PM September 29, 2013

	rule_number	count	percent
1	3305	121	22.490706
2	3301	114	21.189591
3	5716	69	12.825279
4	5503	69	12.825279
5	3303	54	10.037175
6	3302	52	9.665428
7	100002	16	2.973978
8	3306	14	2.602230
9	31101	12	2.230483
10	5702	10	1.858736
11	5710	4	0.743494
12	5703	2	0.371747
13	12105	1	0.185874

If you want the top 100 for rules above 5, just complete with:

```
eventtype=ossec severity>5 | top limit=100 rule_number
```

Go to splunk website for more information, but the interface is very intuitive.

RULES MANAGEMENT

CHAPTER 5

5 RULES MANAGEMENT

So you have now a basic working client-server architecture, you see what is going on on your network. But :

- you receive a lot of alerts;
- you want to trigger active-response on specific rules;
- you want to write you own rules.

5.1 A good configuration strategy

The best approach so far I have found is the following:

Setting	Comment
`log_alert_level = 5`	If you put splunk as a web interface, you don't want splunk to show you alerts below 5
`block all alerts >= 10`	via active-response, you can stop all alerts above a certain threshold. Rules coming with OSSec with level 10+ are significant threat. (read next chapters to see how to setup active-response)
`email_alert = 6`, when installing	Before raising it to 12 at the end, you want to receive more alerts to tweak ossec to your local environment
`email_alert = 12`, when switching to production	Because `alerts >= 10` are blocked, you will check your web interface times to times, and only receive worrisome alerts

The alone problem you will have with this setup, is with rule 31151.

The 31151 rule is raised when several 4xx HTTP error are reported coming from the same IP (see `rules/web_rules.xml` for details). Unfortunately, if you have a quite large web-server infrastructure, you will have lot of false positive (/favicon.ico, but more importantly web application poorly written…), especially if you are a hosting provider with a Cpanel© like offer.

Personally, I override this rule, to downgrade its rule level, and I write a new rule to catch "sensitive URI" (`/phpmyadmin`, `/wp-login`, …). See next chapters to see how to do it, and if you are lazy, go directly to "Summary: avoid the false positive 31151 rule pitfall"

5.2 Tweaking alerts

5.2.1 alerts/email_alert_level

In `etc/ossec.conf`, raise the `email_alert_level` (in alerts block), if you think you receive non significant enough alerts: level is from 1 (insignificant) to 14 (critical). Personally, I leave it to 6, works on tweaking rules, and after that raise it to 12.

Beware: you can still receive alert below the email_alert_level threshold. Some rules have the `<options>alert_by_email</options>` tag which will send you email whatever the email alert threshold is.

See “shut-up email alert you don’t care” to stop receiving them.

5.2.2 email_alerts

You can also add other email recipient than the general one, to receive alert for specific rules, or a subset. For example:

```
<email_alerts>
   <level>5</level>
   <email_to>support@mycompany.com</email_to>
   <rule_id>2501,2502,100003,9951,9952,9953,40111,100002,100012</rule_id>
<!--
   <group>authentication_failed,attacks</group>
-->
</email_alerts>
```

5.3 Tweaking rules: rules/local_rules.xml

But soon you will need to write rules:

- To do that edit only the `rules/local_rules.xml` files. OSSec suppose that when upgrading to a new version it will overrides all rule present in rules directory except `local_rules.xml` file.
- when writing a rule, use id between 100,000 and 119,999.

When editing the `rules/local_rules.xml` file, the general structure is:

```
<var name="A_VARIABLE">…content…</var>

<group name="local,syslog,">
   <rule id="100001" level="12">
      <!-- rule definition -->
   </rule>
   <rule id="100002" level="8">
      <!-- rule definition -->
   </rule>
</group>

<group name="mygroup,web,">
   <rule id="100101" level="0">
      <!-- rule definition -->
   </rule>
   <rule id="100102" level="2">
      <!-- rule definition -->
   </rule>
</group>
```

Let’s beginning with a simple group (mygroup) and simple rules.

5.3.1 Shut-up a rule you don't care

Let say you have a rule (for example 31101, which is level 5) and receive some alerts that you don't care or you can't fix, but don't want to raise the global alert threshold. You can shut-up the rule by writing into rules/local_rules.xml

```
<group name="mygroup">
   <rule id="100002" level="0">
      <if_sid>31101</if_sid>
   </rule>
</group>
```

The trick is this rule override the 31101, and change the level to 0, so at the end, OSSec will not trigger 31101, but 100004, with a level of 0, which is below your `email_alert_level`.

5.3.2 Variant: shut-up email alert you don't care

```
<group name="mygroup">
   <rule id="100002" level="0">
      <if_sid>31101</if_sid>
   </rule>
   <rule id="100004" level="2">
      <if_sid>1002</if_sid>
      <description>do not send by email</description>
   </rule>
</group>
```

Same thing here, the 1002 rule has the tag `<options>alert_by_email</options>` which will send you email whatever the email alert threshold is.

The rule 1002 will send you an email, when it see in the log some keywords: denied, refused, unauthorized, fatal, failed … Sounds great, except on hosting website companies when you got all day long PHP error in the apache logs you cannot fix:-)

5.3.3 Test your rule!

You wrote a first rule, and you want to deploy. OSSec allows you to test it easily before deploying it.

First grab a log you want to test for example this log gathered in an HTTP acces.log:

```
184.121.127.168 - - [21/May/2013:13:10:08 -0400] "GET /member/404.php HTTP/1.1"
404 11736 "http://myserver.com/contact/alignement.php?id=591" "Mozilla/5.0
(iPhone; CPU iPhone OS 6_0_1 like Mac OS X) AppleWebKit/536.26 (KHTML, like
Gecko) Version/6.0 Mobile/10A523 Safari/8536.25"
```

Execute `bin/ossec-logtest` and paste your log:

```
# /var/ossec/bin/ossec-logtest
2013/05/23 19:14:42 ossec-testrule: INFO: Reading local decoder file.
2013/05/23 19:14:42 ossec-testrule: INFO: Started (pid: 16358).
ossec-testrule: Type one log per line.

184.121.127.168 - - [21/May/2013:13:10:08 -0400] "GET /member/404.php HTTP/1.1"
404 11736 "http://myserver.com/contact/alignement.php?id=591" "Mozilla/5.0
(iPhone; CPU iPhone OS 6_0_1 like Mac OS X) AppleWebKit/536.26 (KHTML, like
Gecko) Version/6.0 Mobile/10A523 Safari/8536.25"

**Phase 1: Completed pre-decoding.

 full event: '184.121.127.168 - - [21/May/2013:13:10:08 -0400] "GET
/member/404.php HTTP/1.1" 404 11736 "http://myserver.com/contact/alignement.php?
id=591" "Mozilla/5.0 (iPhone; CPU iPhone OS 6_0_1 like Mac OS X)
AppleWebKit/536.26 (KHTML, like Gecko) Version/6.0 Mobile/10A523
Safari/8536.25"'

 hostname: 'ks388513'
 program_name: '(null)'

 log: '184.121.127.168 - - [21/May/2013:13:10:08 -0400] "GET /member/404.php
HTTP/1.1" 404 11736 "http://myserver.com/contact/alignement.php?id=591"
"Mozilla/5.0 (iPhone; CPU iPhone OS 6_0_1 like Mac OS X) AppleWebKit/536.26
(KHTML, like Gecko) Version/6.0 Mobile/10A523 Safari/8536.25"'
```

```
**Phase 2: Completed decoding.

 decoder: 'web-accesslog'
 srcip: '184.121.127.168'
 url: '/member/404.php'
 id: '404'

**Phase 3: Completed filtering (rules).

 Rule id: '100002'
 Level: '0'
 Description: '(null)'
```

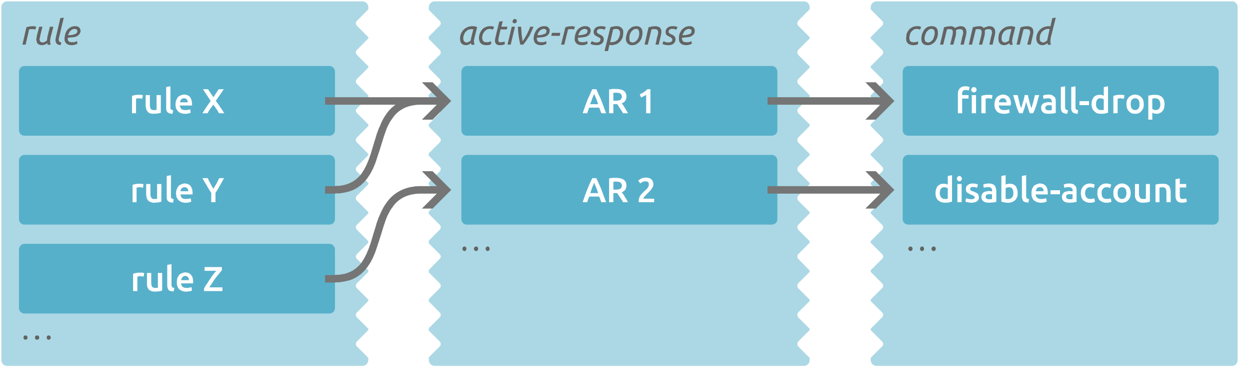
How it works:

- Phase 1 is used to treat the log itself;
- Phase 2 apply decoder (i.e. regex present in `etc/decoder.xml` and `etc/local_decoder.xml`), to grab some field:
  - decoder: the name of the decoder use to get interesting stuff;
  - `srcip`;
  - `url`;
  - `id`;
- Phase 3: rules are played. They use info returned by decoders (for example ip).

In our case, what we have here: we generated rule id 100002, with level 0, and without description (my fault: I forgot to put one). Our rule has been triggered has expected. This test confirm it is working.

5.4 Enable active-response

To enable active-response, you will need to edit `etc/ossec.conf`. It works as follow:



Active-response are links between rules and commands:

command tag: there are several predefined commands in `etc/ossec.conf` (firewall-drop, host-deny, disable-account, route-null, etc) and you can of course create yours. Just put a script or executable in `active-response/bin`

rules: rules are triggers by OSSec, and are defined in `rules/` directory

active-response tag: you define link between rules and command with an active-response XML tag in `etc/ossec.conf`. the link can be rule id, rule level…

The general configuration in `etc/ossec.conf` is:

```
<ossec_config>
   <command>
      <!--
      command options here
      -->
   </command>
   <active-response>
      <!--
      command options here
      -->
   </active-response>
</ossec_config>
```

5.4.1 1st example: block ssh bruteforce attempt

```
<active-response>
   <command>firewall-drop</command>
   <location>agent</location>
   <rules_id>5712</rules_id>
   <timeout>600</timeout>
</active-response>
```

Notes:

- When OSSec detect a ssh brute-force, it will execute firewall-drop command (which is iptables on a Linux) and will block connection to the IP of the offender.
- The offender will be block 600 seconds

5.4.2 2nd example: block longer repetitive offender

```
<active-response>
   <command>firewall-drop</command>
   <location>all</location>
   <rules_id>5712,5720,5551</rules_id>
   <timeout>600</timeout>
   <repeated_offenders>30,60,120</repeated_offenders>
</active-response>
```

Notes:

- This time we detect a ssh brute-force attack via several rules.
- when triggered, it will block offender on all agents (location=all), except the server (all means “all agents”).
- If you want to block also on the server, duplicate this active-response and replace `<location>all</location>` by `<location>server<location>`
- The offender will be block 600 seconds, but if we see it again after, it will be block for longer: 30 minutes, after 60 minutes, and after 120 minutes.

5.4.3 3rd example: block all alerts above level 10

```
<active-response>
   <command>firewall-drop</command>
   <location>all</location>
   <level>10</level>
   <timeout>600</timeout>
   <repeated_offenders>30,60,120</repeated_offenders>
</active-response>
```

Here instead of using `<rules_id>`, we use level. You can also use `<rules_group>`.

5.4.4 4th example: do not receive alerts that are blocked by active-response

SSH brute-force attack are very common. Personally I don’t like to receive by email rules that are blocked. Just uncaught alerts. So I edit `rules/local_rules.xml`:

```
<group name="mygroup">
   ...
   <rule id="100006" level="1">
      <if_sid>5712,5720,5551</if_sid>
      <description>sshd bruteforce attempts</description>
   </rule>
   ...
</group>
```

and in `etc/ossec.conf`:

```
<active-response>
   <command>firewall-drop</command>
   <location>all</location>
   <rules_id>100006</rules_id>
   <timeout>600</timeout>
   <repeated_offenders>30,60,120</repeated_offenders>
</active-response>
```

5.5 Writing your own rules

Now we saw how to write rules, and setup active response, here are some others rules example, to be more comfortable with rules writing.

5.5.1 Blocking brute-force attack (frequency and time-frame)

Also you have a proftpd server, and people try to brute-force it to try to discover login/password.

```
<rule id="100003" level="10" frequency="8" timeframe="43200">
   <if_matched_sid>11203</if_matched_sid>
   <same_source_ip />
   <description>FTP brute force (multiple failed logins).</description>
</rule>
```

The new keywords here are frequency and timeframe.

This rule will be triggered when the same 11203 rule will be seen 8 times during a 43200s window.

You now just have to add the 100003 to active-response. In `etc/ossec.conf`:

```
<active-response>
   <command>firewall-drop</command>
   <location>all</location>
   <rules_id>100003,100006</rules_id>
   <timeout>600</timeout>
   <repeated_offenders>30,60,120</repeated_offenders>
</active-response>
```

You can test it with `bin/ossec-logtest`, by fetching 10 times the same log. For example with:

```
May 23 14:23:15 ns15 proftpd[614]: 184.121.127.168
(101.64.16.226[101.64.16.226]) - USER theadmin: no such user found
from::ffff:101.64.16.226 [101.64.16.226] to 209.172.63.238:21
```

5.5.2 Blocking DoS on a web server (change rule level + frequency and timeframe)

I bet you will better use iptables for that, but in one case I couldn't. So what I have done:

```
<rule id="100100" level="1">
   <if_sid>31108</if_sid>
   <description>A web page</description>
</rule>

<rule id="100101" level="9" timeframe="15" frequency="250">
   <if_matched_sid>100100</if_matched_sid>
   <same_source_ip/>
   <description>Multiple access in a short time from same ip</description>
   <group>attack,recon,</group>
</rule>
```

Unfortunately there is a `<different_url>` but no `<same_url>` tag

So I created a new alert each time a web-page is accessed (31108), and when an IP asks too much web-pages (250) in a relative short period of time (15s), I block it.

I had to create the 100100 rule because 31108 is level 0, and rule with level 0 are discarded directly, and not keep by OSSec to see if there are « timeframe » rule that can apply to it.

5.5.3 Blocking on specific hosts (hostname option)

Sometimes you have rules that works great on some servers, but produce lots of false positive on other.

You can write rule that just apply to a set of agents:

```
<rule id="100005" level="10">
   <if_sid>40111</if_sid>
      <hostname>prod_server_1</hostname>
      <hostname>prod_server_2</hostname>
   <hostname>prod_server_3</hostname>

   <description>Mutiple authentication failures on prod server</description>
   <description>from same source ip.</description>
   <group>authentication_failures,</group>
</rule>
```

5.5.4 Blocking web attack on specific keywords (regex)

A) The problematic: web scan

OSSec comes by default with word-press detection, and generic multiple 4xx error (rule 31151 in `rules/web_rules`) but I want to kick off for very long time people scanning for specific web application. For example I got these logs:

```
137.116.136.162 - - [31/May/2013:08:10:44 -0400] "GET
/cpanelsql/scripts/setup.php HTTP/1.1" 404 304
137.116.136.162 - - [31/May/2013:08:11:39 -0400] "GET /pMA/scripts/setup.php
HTTP/1.1" 404 298
137.116.136.162 - - [31/May/2013:08:11:39 -0400] "GET /pma/scripts/setup.php
HTTP/1.1" 404 298
137.116.136.162 - - [31/May/2013:08:11:42 -0400] "GET /scripts/setup.php
HTTP/1.1" 404 294
137.116.136.162 - - [31/May/2013:08:11:42 -0400] "GET /Scripts/setup.php
HTTP/1.1" 404 294
137.116.136.162 - - [31/May/2013:08:11:44 -0400] "GET
/sqlmanager/scripts/setup.php HTTP/1.1" 404 305
```

```
137.116.136.162 - - [31/May/2013:08:11:44 -0400] "GET /sql/scripts/setup.php
HTTP/1.1" 404 298
137.116.136.162 - - [31/May/2013:08:11:47 -0400] "GET
/typo3/phpmyadmin/scripts/setup.php HTTP/1.1" 404 311
137.116.136.162 - - [31/May/2013:08:11:49 -0400] "GET
/web/phpmyadmin1/scripts/setup.php HTTP/1.1" 404 309
137.116.136.162 - - [31/May/2013:08:11:49 -0400] "GET
/web/phpmyadmin2/scripts/setup.php HTTP/1.1" 404 309
137.116.136.162 - - [31/May/2013:08:11:47 -0400] "GET
/typo3/phpmyadmin/scripts/setup.php HTTP/1.1" 404 311
137.116.136.162 - - [31/May/2013:08:11:52 -0400] "GET
/xampp/phpmyadmin/scripts/setup.php HTTP/1.1" 404 311
```

The common criteria of all there logs is:

- a 404 error;
- coming from the same IP;
- with some keywords: `/[Ss]cripts/setup.php` on several directory (phpmyadmin, sqlmanager, pma …);
- in a relative short timeframe.

Indeed someone is using a tool like dirb (http://dirb.sourceforge.net/), to scan the website. and we want to stop him.

So we need to write 2 rules:

- a first rule that match each of this line;
- another one that see our first rule several time in a short period of time.

B) Building the rule

We must first create a rule matching one of the line we want. Before writing anything, if I use bin/ossec-log, I see:

```
# bin/ossec-log
137.116.136.162 - - [31/May/2013:08:10:44 -0400] "GET
/cpanelsql/scripts/setup.php HTTP/1.1" 404 304
…
**Phase 3: Completed filtering (rules).
 Rule id: '31101'
 Level: '5'
…
```

So the rule is already tagged as 31101. Indeed when ossec apply it:

- decode it (see that it is a webpage, and fill the url and id fields);
- tagged the log as 31100 (in `rules/web_rules.xml`);
- after it has been retagged 31101 (in rules_web_rules.xml) because it is a 4xx http return code;
- and then arrives at `rules/local_rules.xml`.

I will override that in `rules/local_rules.xml`:

```
<rule id="100106" level="5">
   <if_sid>31101</if_sid>
   <url>/cpanelsql/scripts/setup.php</url>
   <description>known sensitive web url.</description>
   <group>attack,</group>
</rule>
```

Now it catch the first line I got in my logs.

C) OSSec OR pattern

The tricky part: create a rule that match several keyword. You have 2 ways to do it:

- use a `<match>` field, where you can only use special keyword “`^`”, “`|`” and “`$`”. If the alert is a web page, you can use the `<url>` field;
- use a `<regex>` field, where you can use more regexp syntax.

First without regexp, we can write:

```
<rule id="100106" level="5">
   <if_sid>31101</if_sid>
   <url>/scripts/setup.php|/Scripts/setup.php</url>
   <description>known sensitive web url.</description>
   <group>attack,</group>
</rule>
```

With this expression, I will catch all URL containing “`[Ss]scripts/setup.php`”

D) OSSec regexp

If I want to be more restrictive, I can use regexp. OSSec has regex support, but it is not Perl compliant. As stated in http://www.ossec.net/doc/syntax/regex.html#os-match:

Expressions

- \w → A-Z, a-z, 0-9 charactersasd
- \d → 0-9 characters
- \s → For spaces " "
- \t → For tabs.
- \p → ()*+,-.:;<=>?[] (punctuation characters)
- \W → For anything not \w
- \D → For anything not \d
- \S → For anything not \s
- \. → For anything

Modifiers

- + → To match one or more times (eg \w+ or \d+)
- * → To match zero or more times (eg \w* or \p*)

Special Characters

- ^ → To specify the beginning of the text.
- $ → To specify the end of the text.
- | → To create an "OR" between multiple patterns.

Characters Escaping

To utilize the following characters they must be escaped:

- \$ → $
- \(→ (
- \) →)
- \\ → \
- \| → |

So with a regexp my rule can be:

```
<rule id="100106" level="5">
   <if_sid>31101</if_sid>
   <url>/scripts/setup.php|/Scripts/setup.php</url>
   <regex>/pMA/scripts/setup.php|/pma/scripts/setup.php|</regex>
   <regex>/scripts/setup.php|/Scripts/setup.php|</regex>
   <regex>phpmyadmin\.*/scripts/setup.php</regex>
   <description>known sensitive web url.</description>
   <group>attack,</group>
</rule>
```

Explanations:

- the rule is valid if it comes from a web log with HTTP code 4xx (`<if_sid>31101</if_sid>`);
- the rule is valid if the URL contains `/[Ss]cripts/setup.php` (`<url>/scripts/setup.php|/Scripts/setup.php</url>`);
- AND to be more restrictive, the rule need also to match the multiline regular expression (`<regex>/pMA/scripts/setup.php|/pma/scripts/setup.php|/scripts/setup.php|/Scripts/setup.php|phpmyadmin\.*/scripts/setup.php</regex>`). The regex can be on one line, or as mentionned above on serveral lines. OSSec concanates them.

I admit: it is overkill

E) The scan rule detection

Last we must now create a “timeframe” rule:

```
<rule id="100107" level="10" timeframe="8" frequency="120">
   <if_matched_sid>100106</if_matched_sid>
   <same_source_ip/>
   <description>Sensitive web scan from same IP</description>
   <group>web,appsec,attack</group>
</rule>
```

and in `etc/ossec.conf` I will add another active-response:

```
<active_response>
   <command>firewall-drop<command>
   <location>agent</location>
   <rules_id>100107</rules_id>
   <timeout>3600</timeout>
</active-response>
```

5.5.5 Summary: avoid the false positive 31151 rule pitfall

As discussed above, if rule 31151 catch too much things, here is what I write in `rules/local_rules.xml`:

```
<group name="web,accesslog,">
   <rule id="100103" level="5">
      <if_sid>31151</if_sid>
      <description>maybe a false positif webattack attempt</description>
   </rule>
   <!-- if we have a 4xx on a sensitive url -->
   <rule id="100106" level="5">
      <if_sid>31101</if_sid>
      <url>/scripts/setup.php|/Scripts/setup.php|editor/filemanager/connectors/uploadtest.html|/wp-login.php|/w00tw00t.at.ISC.SANS.DFind|/phppath/php|/wp-content/uploads|/admin.php|/administrator/index.php|/phpmyadmin|/phpMyAdmin|/websql|/php-my-admin|/member.php|/login.php|/reg.asp|/Class/Post.asp|/user/register|/tiki-register.php|/administrator|/wp-content/cache|/wp-comments-post.php</url>
      <description>known sensitive web url.</description>
      <group>attack,</group>
   </rule>
   <rule id="100107" level="10" timeframe="120" frequency="8">
      <if_matched_sid>100106</if_matched_sid>
      <same_source_ip/>
      <description>Sensitive web scan from same IP</description>
      <group>web,appsec,attack</group>
   </rule>
   <!-- avoid logging favicon -->
   <rule id="100110" level="0">
      <if_sid>31101</if_sid>
      <url>/favicon</url>
      <description>Ignore /favicon* uri</description>
      <group>attack,</group>
   </rule>
</group>
```

5.5.6 Receiving active-response actions

On each agents, change etc/ossec.conf and add:

```
<localfile>
   <log_format>syslog</log_format>
   <location>/var/ossec/logs/active-responses.log</location>
</localfile>
```

So now OSSec can see when active-response are triggers, and rules 601,602 (in `rules/ossec_rules.xml`) are active, but they have level 3, so usually below “email_alert_level” (which is generally 6+). The best way to receive them: write new rules in `rules/local_rules.xml`:

```
<group name="ossec, active_response_notification">
   <rule id="10601" level="6">
      <if_sid>601</if_sid>
      <description>Host Blocked by firewall-drop.sh Active Response</description>
      <group>active_response,</group>
   </rule>

   <rule id="10602" level="6">
      <if_sid>602</if_sid>
      <description>Host Unblocked by firewall-drop.sh Active Response</description>
      <group>active_response,</group>
   </rule>
</group>
```

Now, if your email_alert_level is 6, you will receive active-response actions by email.

If you have set `<location>all</location>` on your active-response. you will receive as many active-response alerts as agents.

TO GO FURTHER

CHAPTER 6

6 TO GO FURTHER

6.1 Geoip support

If you want geoip support, you will need to recompile the server (if you took my deb, it is already compiled with):

Step A) Get and install Maxmind GeoIP API

```
wget http://www.maxmind.com/download/geoip/api/c/GeoIP-1.4.8.tar.gz
tar xzvf GeoIP-1.4.8.tar.gz
cd GeoIP-1.4.8
./configure
make
su
make install
```

Note:

- on a redhat 64bits server, I had to make a symlink:
  `ln -s /usr/local/lib/libGeoIP.so.1 /lib64`
- on an Ubuntu 12.04, I didn't have to install it by hand. You can install libgeoip-dev and libgeoip1

Step B) Get Maxmind GeoIP DB

```
wget http://geolite.maxmind.com/download/geoip/database/GeoLiteCity.dat.gz
gunzip GeoLiteCity.dat.gz
wget http://geolite.maxmind.com/download/geoip/database/GeoLiteCityv6-beta/GeoLiteCityv6.dat.gz
gunzip GeoLiteCityv6.dat.gz
su
cp GeoLiteCity*.dat /var/ossec/etc/
```

Step C) Compile OSSec with GeoIP enabled

```
get ossec-hids-2.7.tar.gz
tar xzvf ossec-hids-2.7.tar.gz
cd ossec-hids-2.7
cd src
make setgeoip
cd ..
su
./install.sh
```

Note:

- on Ubuntu 12.04, before compiling I had to change `src/analysis/Makefile` to add "-lGeoIP" to loga_OBJS:

```
…
loga_OBJS = ${LOCAL} ${PLUGINS} ${DBS} ${ALERTS} ${OS_XML} ${OS_REGEX} ${OS_NET}
${OS_SHARED} ${OS_ZLIB} ${CPRELUDE} -lGeoIP
…
```

Step D) Modify etc/ossec.conf file

```
<ossec_config>
   <global>
      <!-- to specify GeoIP database file location -->
      <geoip_db_path>/etc/GeoLiteCity.dat</geoip_db_path>
      <geoip6_db_path>/etc/GeoLiteCityv6.dat</geoip6_db_path>
   </global>

   <alerts>
      <!-- to add GeoIP info in alerts -->
      <use_geoip>yes</use_geoip>
   </alerts>
</ossec_config>
```

Step E) Restart ossec

```
/var/ossec/bin/ossec-control restart
```

Note:

- it is not yet possible to block based on countries, this will be just for information in alerts.

6.2 PCI-DSS coverage

If you plan to be PCI-DSS compliant, some section can be covered by OSSec (or Ossim):

- the file integrity checking of OSSec covers PCI-DSS sections 11.5 and 10.5.5
- the log monitoring capability of OSSec also covers PCI-DSS section 10 in a whole.

6.3 ossim

- http://sourceforge.net/projects/os-sim/

If you need a full SIEM (Security Information and Event Management) system, you can give a try with Alienvault Ossim: http://communities.alienvault.com/.

Alienvault provides an ISO to install from scratch a SIEM central server (based on Ubuntu). This SIEM server acts as an OSSec server but contains also:

- openVAS;
- ntop;
- nagios;
- snort;
- arpwatch;
- …

It is intended to use it more as appliance you deploy, rather than server and clients you install on an already working machines

6.4 OSSec and LXC

This is out of scope of this book, but OSSec is particularly effective when you use Containers virtualisation technology (LXC or openVZ on Linux, jails on Freebsd): you can install an OSSec server on the host, monitor the logs of all your containers, and if a container is compromised, the attacker will never this the OSSec agent running, because it is not in the container process view.

6.5 Snoopy

If you want to add a total traceability of application execution, you can have a look to snoopy http://sourceforge.net/projects/snoopylogger/

This is a LD_PRELOAD_LIBRARY that log each call to exec() system call. I manage successfully to track a breach of security on a client. But beware when you install it, I got some trouble on certain version of RHEL.