Institut Eurécom
Corporate Communications Department
2229, route des Crêtes
B.P. 193
06904 Sophia Antipolis
FRANCE

Research Report RR-04-103

White Paper:

OWL: Installation testing and validation[1]

October 31, 2004

Fabien Pouget

Institut Eurécom

Email: {pouget@eurecom.fr}

[1] This research is supported by a research contract with France Telecom R&D, contract N. 425-17044

OWL : Installation testing and validation

Fabien Pouget
Email: {pouget}@eurecom.fr
Eurecom
2229, Route des Crêtes ; BP 193
06904 Sophia Antipolis Cedex ; France

Abstract:

In this paper, we report on the installation of the alert correlation console OWL implemented by France Telecom R&D. The version number is 1.1beta2 and has been released in 2004. OWL has been developed as a central place to retrieve and archive events/alerts coming from multiple sources. It also provides some correlation capabilities based on statistical analysis of alert densities and cross-information between sources.
We have installed the console on a Linux Red Hat 9.0 machine without any major problem. We report in this paper the installation processes as well as some general remarks on the console installation.

Keywords:

Alert Correlation, OWL, installation.

Corresponding Author:
Fabien Pouget (pouget@eurecom.fr)
Tél: +33 (4) 93 00 29 26
Fax: +33 (4) 93 00 26 27

TABLE OF CONTENTS

1 Introduction

Alert correlation tools can be basically classified into three main categories, respectively named “Log Analysis Tools”, “Management Consoles” and “Experimental Tools”. We report the interested reader to [PoDa03] for more detailed information on this topic. A review of the state of the art is presented as well as common tools which aim at correlating alerts produced by diverse security elements. In this document we present OWL (Organized Window on Logs), a Management Console developed by France Telecom R&D. This tool has been installed and briefly tested at the Institut Eurecom, where it will be used and modified in a near future, in order to incorporate honeypot information into its correlation engine. This paper aims at describing the installation phases from the packages we received. A deeper analysis and usage of this tool will be described in another document. It is organized as follows: Section 2 presents the tool and its capabilities. Section 3 deals with the installation procedures. Section 4 presents some remarks and Section 5 concludes this experimental report.

2 OWL Presentation

2.1 Brief Overview

OWL is management console that retrieve and archive events coming from multiple sources. These sources can be as different as:

- Intrusion Detection Systems (Snort)
- Vulnerability scanners (Nessus)
- Honeypots (honeyd)
- Firewalls (CheckPoint, Cisco_IOS, Juniper Netscreen)
- Host-based logs (Syslog, authentication logs)

Put in other words, OWL is a well-defined data-centric architecture associated to a correlation engine and a web-based access. A short description can be found in [Owl03]. It processes such heterogeneous logs by means of parsers. It also tries to store uniform data and enhance incoming information thanks to lookups (DNS, Whois). Alert information is correlated to ease the administrator task. Finally, an interface has been developed to display an analyze output.

As a result, this console is more promising than the existing like ACID (analysis Console for Intrusion Databases) as it can handle many various sources. A complete database management is provided so that the user does not need to know the whole postgreSQL architecture. It allows defining groups of users. Roles are granted to each group. In addition, the database follows the IDMEF standard. Thus, we can imagine that other tools using this standard will be easily integrated in a near future.

2.2 Packages Contents

We received three packages, respectively:

- owlsrc_v1.1b2_20040818.tgz, containing the source files of OWL.
- snortsrc_v1_1b2_20040818.tgz containing the source code of the snort-based sensor with IDMEF plugin.
- rules_v1_1b2_20040818.tgz containing the rule sets for the previous sensor.

```
$> tar -zxvf XXXX.tgz
```

This command creates many directories, which are:

Directory	***Description***	***Do we need to look inside for the install process?***
adodb	The database abstraction library for PHP	No
docs	Docs: the documents repository	Yes
etc	Scripts directory (shell and sql)	Yes
jpgraph	The PHP graph plotting library	No
logs	Parsers and correlation modules	No
owl	The site pages	Yes

The */docs* directory contains 12 files in html or pdf format. This would have been more convenient to have all files into a global OWL document. Indeed, there are some redundancies (and even contradictions) between documents like *Installation_OWL.html* and *Architecture_and_Installation.pdf*. Moreover, some documents are written in English and others in French but they do not sometimes provide similar information. As for an illustration, requirements are quite different. In one document, only four packages are listed. We have followed the *Architecture_and_Installation.pdf* instructions and the different install steps are presented in the following section.

3 Installation Processes

3.1 Installation Requirements

We show in table 1 the packages that are suggested in the install document compared to the ones we effectively used.

Requirements under Debian Operating System		Packages installed on the RedHat OS
perl, perl-base, perl-doc, perl-modules	5.8.3-3	perl-5.8.0-88 (rpm)
libperl5.8	5.8.3-3	
Libdbi-perl	1.41-1	
libdbd-pg-perl	1.32-1	DBD-Pg-1.32 (.tar.gz)
libxml-libxml-common-perl	0.13-4	
libxml-libxml-perl	1.56-6	
libxml-namespacesupport-perl	1.08-3	
libxml-sax-perl	0.12-4	libxml-sax-perl_0.12 (.tar.gz)
libdatetime-perl	0.1901-1	
php4 , php3-gd	4.3.4-4	php-4.3.4-3 (rpm) php-gd-4.3.4-3 (rpm)
php4-pgsql	4.3.3-2	php-pgsql-4.3.3-6 (rpm)
libpcap0.7	0.7.2-6	libpcap-0.7.2-1
libxml2	2.6.10-3	Libxml2-2.6.10-1 (rpm)

postgreSQL	7.1.3-> 7.3.x	PostgreSQL-7.3.2-3
		Other installed libraries: NetAddr-IP-3.20 (.tar.gz) Params-Validate-0.74 (.tar.gz) Date-Time-0.1901 (.tar.gz) Date-Time-Locale-0.09 (.tar.gz) Date-Time-TimeZone-0.28 (.tar.gz) Module-Build-0.25 (.tar.gz)

3.2 Some RedHat variants

Some packages might be hard to find. That is why we have tried other versions. This leads to the following remark: it would be better to include such packages in the OWL distribution. We can imagine an additional directory, where it would be possible to get all required packages for Debian or Red Hat distributions. This would not be a hard work, but this would make the install a lot faster.

While running the *install_owl.sh* file, we have realized that some Perl libraries were missing: The NetAddr-IP.pm library is required by the update_hosts.pl script. DateTime.pm library is required by archivage.pl and change_detect.pl scripts.

This leads to the following modifications in the scripts:

In *change_detect.pl*: 'use lib /PATH_TO/DateTime-0.1901/lib'

In *archivage.pl:* 'use lib /PATH_TO/DateTime-0.1901/lib'

The other libraries are required during the install process (traditional './configure', 'make', 'make install') but do not imply modifying the existing scripts.

There are some warnings that occur during the installation ('Malformed UTF-8 character') but they do not seem to interfere with the console running correctly.

3.3 Platform up and running: test

We have installed a Snort sensor source. The install was very easy. A weird thing is the number of *snort.conf* files that were created. A 'locate snort.conf' command gives the following results:

1. /Supervision/rules/snort.conf
2. /Supervision/snort21/etc/snort.conf
3. /Supervision/snort/etc/snort-eth0/snort.conf
4. /Supervision/snort/etc/snort-eth0.1/snort.conf
5. /Supervision/snort/etc/snort-eth0.2/snort.conf
6. /Supervision/snort/etc/snort-eth0.3/snort.conf

In order to determine which one is used, we have to look at the /etc/init.d/snortrc file. The file number 3 is actually the one used by snortrc. According to this same init file, there should be as many config files as interfaces (SNORTETCDIR=$SNORTDIR/etc/*snort-${IFACE}*). The fact is that a new /etc/snort-eth0.X is created at each *install_snort.sh* run. We can imagine during the pre-action "Cleaning snort installation" that it is asked whether we want to keep the /etc files or not. I guess this corresponds to the original Snort implementation. However, we do not really need to archive as many /etc/snort-eth0 folders as install attempts. Do we?

```
/bin/cp -fR etc/* ${CfgDir}
```

The default DBI with RedHat 9.0 distribution is version 1.32. However, the DBI version 1.35 is at least required by the install_driver(pg). Thus, we have installed the last version (DBI-1.45). The installation is quite straightforward.

4 General Remarks

4.1 Installation Documentation

Installation documents remain quite light. We would have preferred a step-by-step presentation.

Some documents need to be updated or dropped. For instance, the *Description_des_programmes_perl.html* presents -u dbuser and -p dbpass options. These options are obsolete with the last postgreSQL versions where they are replaced by -U dbuser and -W dbuser. In addtion, the example is not accurate. When we launch the given command, it is asked to provide the mandatory database name.

In the *Architecture_and_Installation.pdf* file, it is mentioned in several places a directory called 'supervision'. However this folder is never created.

In the same document, we find some sections in English, and others in French (3.8).

The cron job at the end is surprising. Why should we clean every day the postgres database? This should depend on the traffic collected. A certain threshold could limit the number of entries. Another solution would be to send a warning when the number of collected entries reach a certain threshold and then to ask the administrator if he wants to clean the oldest ones.

4.2 Scripts

Some scripts have French comments. Some others have strange characters due to the French accents that are not recognized with some Operating System configurations. This leads to a bunch of warnings and errors like the following:

"Malformed UTF-8 character (unexpected non-continuation byte XXXX, immediately after start byte YYYY) at <FILE> line ZZ."

A list of such errors is given in Annex A. They should be replaced in next versions by English comments.

It is not really easy to understand which script should be modified in order to perform the install. For instance, where can we change the RootDir variable value? It is not clear which scripts should be launched/modified to customize which variables. By default, we directly edited the *supervision_owl* file (which is not mentioned in the document) in order to customize these values.

Generally speaking, we wish that more details on apache/openssl/php/postgreSQL had been provided. Their installation is not obvious, all the more if we want for instance to configure apache with openssl, php and postgreSQL. Missing parameters can make the install very annoying. Some configuration examples or tricks can be presented in order to help the user building a default platform. Some people may want to customize such services, but a default configuration will make the install very fast for people who do not have this need.

I had to modify the apache_owl.conf file: the SSLCertificate files did not point to the good root folder (as I tried another path than /supervision in a second step). However, this file should be updated automatically depending on the OWL configuration as described in supervision_owl. Furthermore, the ssl.dir folder did not exist. This explains some errors at the install time when the certificate creation is asked.

How to launch the OWL service? This information has not been found in the user document. A glance at *install_corr_owl.sh* reveals that there is an automatic startup called 'owlrc', which is confirmed by the command 'ls /etc/init.d | grep owl'. However it seems this information is vital for end-users. It should definitely appear in the installation guide.

What is the interest of keeping old files in the install process? (see *install_owl.sh* 'mv ${FILE_X} ${FILE_X}.old'). Could it be an option during the install, like: *It seems another version of OWL is already installed. Do you want to archive old files?*

4.3 Database

A major remark concerns PostGreSQL as described in Section 3.3.1. People are not supposed to be postGreSQL experts. The installation requires few basic commands. They can be included either in a script or in the install document. Some of them are listed in Annex B.

Something surprising: the database architecture appears nowhere.

4.4 The web interface

The *guest* user has no real utility at the moment.

Where are the help pages?

It is written on page 8 that the installation process also creates an empty Owl alert database. However, when we first log as *operator* and click on the *Alert menu,* we get the following message: "La base de données n' est pas disponible" (or an error message like: "Fatal error: Call to a member function on a non-object in /var/www/html/alerts/overview.php on line 223."

The correlation modules like change_detect.pl, change_nessus.pl or cache_host_dns.pl should be incorporated in the database. Otherwise, the database seems to be a "database management console" without correlation capabilities. Some parameters could be directly modified from the interface to tune these scripts.

5 Conclusion

We would conclude that some changes should be made in order to make the install easier. At this time writing, it is necessary to have a look at each script in order to understand the good commands to launch. Thus, we would suggest two points:

- Make the install documents unified. A step-by-step install procedure is required, with some clear help on how to install and configure apache/php/openssl/postgresql quickly. People who do not have experience with these tools will be lost. It can be links/references to relevant web pages or concrete examples of default install.

- Gather all install scripts into a general one, with multiple options, depending on the kind of installation.
- Provide the install packages that are not easy to find and enrich the list with non Debian-specific versions.

6 ANNEXE A

Files	***Remarks***
etc/configure_owl.sh	Accents with ECHO commands lines 16, 31, 39, 40 (French comments)
etc/configure_snort.sh	Accents with ECHO commands lines 16, 28, 32, 37, 38, 44, 49, 50 (French comments)
logs/Sensor.pm	Accents with PRINT command Line 341. (French Comments)

7 ANNEXE B

```
su -l postgres
initdb –pgdata=/var/lib/pgsql/data
/usr/bin/pg_dl -D /var/lib/pgsql/data -l logfile start
service postgresql start
createdb toto
psql toto
\c toto
\q
```