

SIGVI R2

Installation Guide

Index

1. Introduction.....	3
2. Requirements.....	3
2.1. Platform.....	3
2.2. Web Server	3
2.3. Server relational database.....	3
2.4. PHP5 interpreter.....	4
1. Set up.....	4
1.1. System information.....	4
1.2. Setting up the application.....	5
1.3. Database creation.....	5
1.4. Environment changes.....	5
1.1. Use LDAP access.....	6
1.2. Configuring the batch processes.....	6
1.1. First use.....	7
2. Help.....	7

1. Introduction

SIGVI is the acronym in Spanish for Sistema Inteligente de Gestión del Vulnerabilidades Informáticas. It's a tool to discover and manage the vulnerabilities of our systems.

This project is developed and maintained since UPCnet (<http://www.upcnet.es>), ICT service company of the UPC (Universitat Politecnica de Catalunya, <http://www.upc.edu>). It has also been co-funded in 2008 by the Ministry of Industry, Tourism and Trade of Spain (MITYC, <http://www.mityc.es>) to obtain a pre-product.

The SIGVI is a Web application composed by a set of programmed PHP scripts that implements the logic of the application and a relational database where the data are stored. Some scripts are executed as batch processes (usually at night) to perform tasks that do not require human interaction, such as the update of vulnerabilities from the sources, the review of the vulnerabilities in our systems, and so on. The other scripts are the Web application.

This document is trying to clarify how to install a new instance of SIGVI.

2. Requirements

The SIGVI R2 is implemented in PHP5 and you'll need a relational database to store the data.

2.1. Platform

The application has been developed and tested on Linux platforms. The setting up process will be the same on any UNIX platform (with Web directories, which depend most of the Web server).

Because it's an interpreted language it would run on Windows platforms, although there are differences in the configuration of batch processes. We recommend you install the application on a Linux platform.

2.2. Web Server

You need a Web server that allows PHP5 interpreting scripts. This application has been developed and tested in a Web-based **Apache Web Server Version 2** with PHP5 module (optional with the SSL module).

If your Web server is different, you should bear in mind that the application is based on a system directory. There are common directories with classes, images, icons, etc.. And other directories that contain the logic of the application.

Internal references require access on both absolute and relative directories.

Setting up in "apt" based systems:

```
> apt-get install apache2-mpm-prefork libapache2-mod-php5
```

2.3. Server relational database

SIGVI R2 provides a database abstraction layer. This version is provided with access to RDBMS MySQL, Oracle and POSTGRES.

But this is only at a functional level because our initial loading script is MySQL based. In future versions we expect to solve that part providing the necessary code for the rest of supported servers.

We recommend to install MySQL Server version 5.0 or higher. Otherwise, it will be necessary to

modify SQL initial load to adapt it to your system.

Setting up in “apt” based systems:

```
> apt-get install mysql-server-5.0
```

2.4. PHP5 interpreter

We have to configure several PHP5 modules:

- Install library related to your database server (mysql, oracle, POSTGRES)

Using a MySQL database server you will need the PHP module with the necessary libraries (modules php5-mysql, php5-pgsql, ...).

- GD image generation library

The application needs to generate graphical charts and summaries of state. You'll need to install the PHP module to generate images (module php5-gd).

- Shell-line commands

Part of the application are scripts that runs as a batch processes and are interpreted directly by the system. To do this you'll need the PHP module line commands (php5-cli).

1. Set up

We distribute SIGVI in a tar.gz archive that includes:

- Changelog file, which explains the latest developments made in the version of the package
- INSTALL file a brief explaining how to set up the application
- A file to create SQL database application and perform the initial load.
- The directory tree and scripts that make the application itself.

First of all, you must extract the contents of the package:

```
> tar xzvf sigvi-xxxx.tgz
```

1.1. System information

Software:

Operating System: Ubuntu Desktop 7:04

Web Server: Apache Web Server module 2.2.8-1 with PHP5

Shell-line commands PHP5

Database server MySQL 5.0.51a

Environment data:

Root directory of the Web server: /var/www

User on which processes are running Apache: www-data

User of the system that we set up: root

User of MySQL database with permissions to create databases: root

1.2. *Setting up the application*

Within the package there is a directory called "sigvi" that contains the application. We should move this directory within the directory tree of our Web server (in some versions called htdocs):

```
> mv sigvi /var/www/sigvi
```

Then we should modify the permissions of this directory for the Apache running user (www-data in latest versions). If you don't know, you can find out running the command "ps -EFA"

```
> chown -R www-data:www-data /var/www/sigvi
```

```
> chmod -R 750 /var/www/sigvi
```

1.3. *Database creation*

As discussed, SIGVI R2 provides a layer of abstraction database that allow to work with any RDBMS. However the SQL script through which you build the database is based on MySQL.

To create the database we use a user with MySQL privileges, in our case we use "root", and the file provided in the package "sigvi-version.sql"

```
# mysql -u root -p < sigvi-1.3.0.sql
```

If no errors have occurred, we have the SIGVI database up and running and all the tables initialized.

1.4. *Environment changes*

To configure the web access we have to edit the file app.conf.php in the directory "conf" provided in the package and change the parameters HOME, ADM_EMAIL, SERVER_URL, and the database configuration.

1.4.1. *HOME & SERVER_URL*

Take special care with the constant HOME, it is the internal reference to the application itself through the browser.

How can I define this variables?

•Case 1: If we want to access the application through a URL like this:

<http://server.localdomain.domain/sigvi> then the HOME value will be: `define("HOME", "/sigvi")`
without the slash at the end, and the SERVER_URL value will be:
`define("SERVER_URL", "server.localdomain.domain")`

•Case 2: If we want to access the application through a URL like this:

http://server.localdomain.domain/my_applications/sigvi_r2 then the HOME value will be:
`define("HOME", "/my_applications/sigvi_r2")` and the SERVER_URL value will be:
`define("SERVER_URL", "server.localdomain.domain")`

•Case 3: If we installed SIGVI in a Virtual Host and we want to access the application through a URL like this: <http://server.localdomain.domain:81/> then the HOME value will be: `define("HOME", "")`

without any slash, and the SERVER_URL value will be:
`define("SERVER_URL", "server.localdomain.domain:81")`

1.1. Use LDAP access.

SIGVI allows authenticate users against external services. In this version we provide a plugin for authentication via LDAP.

You only have to edit the `app.conf.php` inside the `conf` directory and modify at the end of the file according to your environment.

Users who have the value “extern” to true, will be authenticated using this method.

1.2. Configuring the batch processes

The batch processes are responsible to update your database. We use them to download and update the table of vulnerabilities, to see which products installed on your servers are affected by a vulnerability, alert administrators, etc..

Although the application provides mechanisms through the Web interface to manually run the most important processes, we recommend the configuration of batch execution to automate tasks and minimize the human failure.

On UNIX systems configuration is done via `crontab`, which can be done in two ways: using the system file `/etc/crontab` or the user's `crontab` (`crontab -l` command to view or `crontab -e` to edit it).

First you have to edit the file `cron.sh` that is within the directory “cron”. Review the first three variables:

- PHP: point to the php interpreter, usually “`/usr/bin/php5`”
- DIR: the directory where the SIGVI was installed, usually “`/var/www/sigvi`”
- OUTPUT: the file for saving the standard output and the processes errors.

Edit the `crontab` file. Remember that the user who run the `crontab` will need access to the SIGVI directory, and remember to put your SIGVI installation directory.

- Case 1: edit `/etc/crontab`, and add a line like this:

```
0 5 * * * www-data /var/www/sigvi/cron/cron.sh > /dev/null 2>&1
```

In this case, the user `www-data` has access to the SIGVI directory.

- Case 2: execute `crontab -e` as user `root` or user `crontab`, and add this line:

```
0 5 * * * /var/www/sigvi/cron/cron.sh > /dev/null 2>&1
```

Finally, we recommend to execute manually the shell script to prove that everything is running right.

If you are using Windows, you'll need to create a programmed task to execute the `.php` scripts, like it is done in the script “`cron.sh`”

- `plugins/sources/load_vulnerabilities.php`
- `plugins/sources/check_server_vulnerabilities.php`
- `plugins/discoverer/cron/check_repository_updates.php`

1.1. First use.

Open a web browser and go to: <http://server.localdomain.domain/sigvi>. In the login page put the initial user: “admin”, with password: “admin”.

Remember to change this password as soon as possible.

When authenticated you'll see the main menu and a information icon telling you the next steps you have to do after the installation of a instance of SIGVI. For more information please read the user manual, you'll find details on this first tasks you must do.

2. Help

If you need some help, you can ask to sebastian.gomez@upcnet.es

Thanks in advance for your interest, and we hope you'll find SIGVI useful.