|  |  |
| --- | --- |
| Database Firewall Icon  FireWAll SQL  UNAM–CERT | Descripción breve  El proyecto se integrara con la intención de que se integre con el manejador de contenido Drupal 7.  Marcelo Barrera Ismael Hernández  Proyecto Final |

Índice

[Importancia de la seguridad de la información 2](#_Toc413616602)

[Firewall de Base de Datos 4](#_Toc413616603)

[Ataques a bases de datos 5](#_Toc413616604)

[Herramientas utilizadas 7](#_Toc413616605)

[Debian 8](#_Toc413616606)

[Drupal 9](#_Toc413616607)

[Postgresql 10](#_Toc413616608)

[SQLrelay 11](#_Toc413616609)

[Libinjection 12](#_Toc413616610)

[Instalación y configuraciones de postgres 13](#_Toc413616611)

[Requisitos para la instalación 13](#_Toc413616612)

[Instalación 13](#_Toc413616613)

[Instalación e implementación SQL relay 14](#_Toc413616614)

[Plataformas soportadas 14](#_Toc413616615)

[Arquitecturas soportadas 14](#_Toc413616616)

[Biblioteca Rudiments 14](#_Toc413616617)

[Requisitos previos a la instalación 14](#_Toc413616618)

[Script de instalación (sin archivo deb) 14](#_Toc413616619)

[Instalación 14](#_Toc413616620)

[Configuración de SQL relay 17](#_Toc413616621)

[Configuración de Libinjection 19](#_Toc413616622)

[SQL Relay drop-in replacement 20](#_Toc413616623)

[Incorporación de SQL Relay con Drupal y Apache HTTPD 20](#_Toc413616624)

[Exportar las librerías de drop-in replacement de SQL Relay 20](#_Toc413616625)

[Integración de SQL Relay drop-in replacement con Apache HTTPD 21](#_Toc413616626)

[PHP fue instalado con los paquetes libapache2-mod-php5 21](#_Toc413616627)

[PHP es ejecutado usando el módulo mod\_fcgid 21](#_Toc413616628)

[PHP trabaja con el soporte suEXEC de Apache 22](#_Toc413616629)

[Comprobación de la configuración 22](#_Toc413616630)

[Integración con Drupal 23](#_Toc413616631)

[Creación de paquetes deb 24](#_Toc413616632)

[Paquetes deb 24](#_Toc413616633)

[Requisitos 24](#_Toc413616634)

[Creación 24](#_Toc413616635)

[Referencias 26](#_Toc413616636)

# Importancia de la seguridad de la información

La información puede ser definida como un conjunto organizado de datos procesados, que constituyen un mensaje sobre un determinado ente o fenómeno. Los datos se perciben, se integran y generan la información necesaria para producir el conocimiento, que finalmente permite tomar decisiones.

En la actualidad la información tiene valor de acuerdo a los intereses de cada persona, empresa u organización y por esta razón se vuelve de gran importancia protegerla. La información paso a ser uno de los activos más críticos para las empresas y organizaciones. La seguridad de la información es un proceso que debe combinar distintas medidas de seguridad para conseguir su objetivo.

Los objetivos de la seguridad de la información pueden dividirse en tres conceptos básicos:

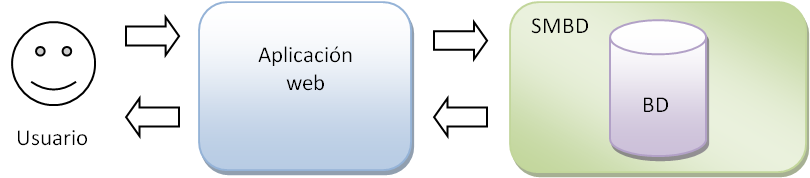
* Confidencialidad: Es posible dividir lo datos en dos tipos datos en movimiento cuando se encuentran en movimiento a través de la red; y datos en reposo cuando permanecen en algún medio de almacenamiento. La confidencialidad significa que solo las personas o sistemas autorizados pueden ver información sensible o clasificada. A su vez esto implica que cualquier persona o sistema sin autorización no deberán tener acceso a los datos. Un método utilizado para proteger los datos en movimiento es el cifrar la información antes de enviarla.
* Integridad: significa que cualquier modificación en los datos solo podrá ser realizada por el sistema o personal autorizado. En caso de que se realizara una modificación a los datos por alguien no autorizado, el sistema debe ser capaz de saberlo y no tomar en consideración los datos corruptos.
* Disponibilidad: La información así como los recursos dentro de una organización deben estar utilizables para el personal autorizado. El que no se pueda acceder genera pérdidas para las empresas.

Una Base de Datos es una estructura computarizada organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite. A su vez la estructura debe ser compartida e integrada. En la base de datos se guardan:

* Datos del usuario final, datos sin elaborar que son de interés para el usuario final.
* Metadatos (datos acerca de los datos), son la forma en la cual los datos del usuario final son integrados y manejados.

La mayoría de las aplicaciones, tanto web como de escritorio, interactúan con usuarios que proporcionan información para su almacenamiento y procesamiento. Generalmente, esta información se resguarda en una base de datos, esta información puede ser de carácter sensible para los clientes o para el desarrollo de procedimientos internos de la organización.

Si las aplicaciones web no cuentan con la seguridad, la información de los usuarios, información de configuraciones, podrían quedar expuestas, o bien, algún usuario malicioso podría intentar extraerla.



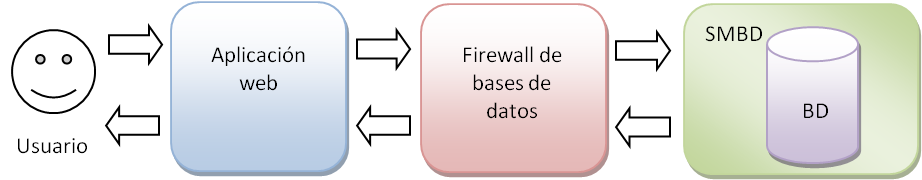
Un tipo de ataque proveniente de usuarios maliciosos hacia un manejador de base de datos es conocido como SQL Injection, que consiste en la “inyección” de consultas SQL para obtener información de la base de datos. Por estas razones las bases de datos resultan deben ser protegidas para limitar el acceso al usuario y de esta forma se conserve la confidencialidad y la integridad.

# Firewall de Base de Datos

El firewall de bases de datos es una aplicación de software que permite filtrar, mediante un conjunto de reglas preestablecidas, las peticiones que llegan al manejador de bases de datos.

Adicionalmente, al instalar una solución orientada para proteger la base de datos, no solo se bloquean las peticiones maliciosas, si no que se puede llevar a cabo el monitoreo de las actividades, generando bitácoras y explotando la información que se almacena en ellas. De esta manera se pueden generar estadísticas que permitan visualizar el posible comportamiento de los atacantes y tomar las medidas de seguridad necesarias.

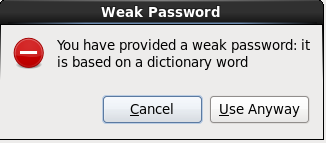
El firewall de bases de datos se coloca entre la aplicación web y el manejador de bases de datos. Cuando algún usuario malicioso intenta obtener acceso a la información almacenada, se cuenta con una capa adicional de protección proporcionada por el firewall de bases de datos que le impedirá poder consultarla. El firewall solo permitirá el paso a los usuarios o a consultas y accesos autorizados con anterioridad.



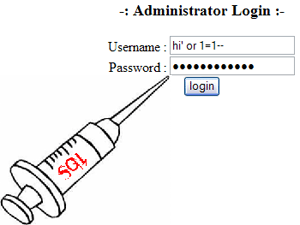
# Ataques a bases de datos

Se expondrán algunos de los principales ataques realizados a bases de datos:

* Obtención de usuarios y contraseñas: El hacer uso de contraseñas débiles o contraseñas por default crea un hueco de seguridad que puede ser utilizado por los atacantes. Al dejar los usuarios y contraseñas por default cualquier persona puede conocer la contraseña con realizar una simple investigación. Al usar contraseñas débiles, estas pueden ser obtenidas mediante herramientas que realizan ataques por diccionario o por fuerza bruta para conocer las mismas.



* Escalamiento de privilegios: Consiste en obtener más privilegios de los que el usuario debería tener. En ocasiones los usuarios tienen más privilegios de los que deberían. Por ejemplo un usuario que solo requiere conocer información de la base de datos mas no modificarla, debe tener exclusivamente privilegios de lectura en la parte correcta de la base de datos (no la lectura de todas las tablas). En muchos casos esto puede mitigarse colocando las configuraciones adecuadas para los usuarios y hasta donde tiene acceso cada uno.
* Explotación de servicios y características no utilizadas: Al realizar la instalación de los manejadores de base de datos en varias ocasiones los administradores optan por realizar una instalación completa (con todas las características). Esto aunque sea más cómodo genera servicios que no son utilizados. Al tener este tipo de servicios instalados, también se contara con las vulnerabilidades de estos. Si no se utiliza algún servicio o característica se recomienda que este permanezca desinstalado o deshabilitado.
* Explotación de vulnerabilidades corregidas con parches: aplicar los parches y actualizaciones a los manejadores de base de datos disminuye las vulnerabilidades de este. El problema es que muchas veces las actualizaciones y parches son tomados a la ligera lo que permite a los atacantes explotar vulnerabilidades que en realidad ya fueron corregidas por el equipo encargado del mantenimiento de la base de datos.
* Inyección SQL: Cuando los campos de la aplicación web son vulnerables, permiten introducir sentencias SQL, se puede afectar a la base de datos desde leer información a la que no se debería tener acceso, hasta la modificación o destrucción de la misma. Este ataque se aprovecha de que al mandar información mediante un campo vulnerable no se necesita de un usuario y contraseña, solo se envía la sentencia SQL. Este ataque puede ser mitigado utilizando un Firewall de base de datos, específicamente un Firewall para filtrar sentencias SQL maliciosas.



* Robo de respaldos sin cifrar: En ocasiones los atacantes se encuentran dentro de la misma empresa u organización. Por esta razón los respaldos que se realicen deben contar con seguridad lógica y física. En ocasiones por mantener segura la base de datos los administradores olvidan que los respaldos de base de datos contienen la misma información (con diferencias por el tiempo en el que se realizó el respaldo)y por lo tanto es igual de sensible y debe ser protegida.

# Herramientas utilizadas

## Debian



Debian es una distribución de sistema operativo donde el kernel puede ser Linux o kFreeBSD (para este proyecto se utiliza el kernel de Linux). Esta distribución es libre y gratuita, principalmente bajo la licencia GNU. El proyecto debían comenzó en 1993 y su primer lanzamiento fue en 1996. La versión utilizada para este proyecto es la versión 7 que tiene por nombre “Wheezy”. La interfaz por default es GNOME, aunque es posible utilizar otro tipo de interfaz. El navegador por defecto es Iceweasel. Debían es utilizado en distintos campos como el de la medicina, educación, química etc.

## Drupal



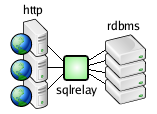
Drupal es un sistema de gestión de contenidos que se utiliza para crear sitios web dinámicos con varias funcionalidades. Es software libre, escrito en PHP, que cuenta con una amplia y activa comunidad de usuarios y desarrolladores. Se trata de un sistema modular con una arquitectura muy consistente, que permite que los módulos creados por cualquier desarrollador puedan interactuar con el núcleo del sistema. Para este proyecto se utiliza Drupal en su versión 7.

## Postgresql



PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Tiene soporte completo para claves foráneas, uniones, vistas, disparadores y procedimientos almacenados (en varios lenguajes). Incluye la mayoría de los tipos de datos del SQL 2008, incluyendo INTEGER, numérico, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, y TIMESTAMP. También soporta almacenamiento de objetos binarios grandes, como imágenes, sonidos o vídeo. Cuenta con interfaces nativas de programación para C / C + +, Java,. Net, Perl, Python, Ruby, Tcl, ODBC, entre otros.

## SQLrelay



SQLrelay es un manejador de conexión a base de datos. Permite acelerar y mejorar la escalabilidad de una base de datos, principalmente de una que realice conexiones web, balanceo de carga para la base de datos, acceso a la base de datos desde plataformas que no son compatibles, migrar de una base datos a otra etc. Algunas de sus ventajas son:

* Conexión de base de datos persistente
* Funciones proxy
* Balanceo de carga
* Filtrado y enrutamiento de Query

SQL Relay proporciona la API nativa con funciones avanzadas como variables se unen, de varias filas obtiene, resultado del lado del cliente establece el almacenamiento en caché y las operaciones suspendidas por muchos lenguajes de programación diferentes.

* C
* C++
* Perl
* Python
* PHP
* Ruby
* Java
* TCL

ErlangSQL Relay también ofrece controladores para muchas capas abstracción de bases de datos existentes.

* Perl DBD
* Python DB
* Ruby DBD
* PHP Pear DB
* Zope
* ODBC

## Libinjection

Es una pequeña biblioteca en C utilizada para detectar ataques SQLi en campos donde el usuario realiza alguna consulta o inserción a la base de datos.

* Es código abierto
* Baja cantidad de falsos-positivos
* Buena detección e SQLi
* Diseñado para ambientes de alto rendimiento
* Fácil de probar
* Fácil de integrar
* Licencia BSD.
* Se basa en el análisis léxico de SQL y no utiliza expresiones regulares.
* Código C estándar, se puede compilar en Linux / Unix / BSD, Mac y Windows, se compila como C + +.
* Solo consume 500Kb por segundo en hardware.
* En memoria 500 bytes.
* Alrededor de 1000 líneas de código en sólo tres archivos.
* No hay dependencias de bibliotecas externas.
* Calidad, exactitud, precisión.
* Rapidez mil inspecciones por segundo.
* Construido y probado utilizando gcc, clang, y g+ +.
* Análisis estático utilizando Cppcheck y analizador estático de clang.
* Probado contra 85000 muestras de sql injection.
* Probado contra los anteriores y actuales falsos positivos.

# Instalación y configuraciones de postgres

## Requisitos para la instalación

Aparte de la instalación de postgreSQL se necesitan instalar:

* postgresql-server-dev-all: bibliotecas para desarrollo en postgreSQL
* g++ :compilador

## Instalación

El siguiente script debe ser corrido con permisos de Root para su correcta ejecución. El script se encargara de añadir la biblioteca de desarrollo de postgreSQL y el compilador.

#!/bin/bash

# verificar tipo de sistema

UNIX\_KIND**=`uname -s`**

**if** **[** "`echo Linux | grep \"$UNIX\_KIND\"`" **]** #si es linux

**then**

**echo** "Tu sistema es: '$UNIX\_KIND'"

**else**

**echo** "El sistema: '$UNIX\_KIND' no es soportado por este script"

**exit** 1

**fi**

# Verificar root

**if** **[** **-z** "`id | grep \"uid=0(root)\"`" **]** #si no hay string que concuerde con el uid root...

**then**

**echo** "Necesitas ser root para instalar!!"

**echo** " "

**echo** "Abortando instalacion..."

**exit** 1

**fi**

# ----------------------

# Instalación de servicios y bibliotecas

# ----------------------

# Instalación de Postgres

apt-get **-**y install postgresql

# Instalación de bibliotecas para desarrollo en postgres

apt-get **-**y install postgresql-server-dev-all

# Instalación del compilador

apt-get **-**y install g++

**echo** “**----------------------**“

**echo** “Instalación Finalizada”

**echo** “**----------------------**“

# Instalación e implementación SQL relay

## Plataformas soportadas

* Versiones modernas de Linux o Unix
* Solaris
* FreeBSD
* NetBSD
* OpenBSD
* Mac OS X
* Haiku
* Cygwin (algunos detalles menores)

## Arquitecturas soportadas

* X86
* X86\_64
* ARM
* Sparc
* Mips
* PPC
* Otros

## Biblioteca Rudiments

Para que SQL Relay pueda funcionar se debe contar con la biblioteca Rudiments. Rudiments es una biblioteca de código abierto de C++ que provee de clases base para distintas cosas como:

* Demonios
* Clientes y servidores
* Envolturas (wrappers)

## Requisitos previos a la instalación

* Biblioteca Rudiments

## Script de instalación (sin archivo deb)

### Instalación

Por defecto la instalación se realizara en /usr/local/firstwork. Este directorio debe ser creado antes de la instalación o teniendo privilegios de Root. En el siguiente script se utilizan los privilegios de Root para la creación de los mismos.

#!/bin/bash

# verificar tipo de sistema

UNIX\_KIND**=`uname -s`**

**if** **[** "`echo Linux | grep \"$UNIX\_KIND\"`" **]** #si es linux

**then**

**echo** "Tu sistema es: '$UNIX\_KIND'"

**else**

**echo** "El sistema: '$UNIX\_KIND' no es soportado por este script"

**exit** 1

**fi**

# Verificar root

**if** **[** **-z** "`id | grep \"uid=0(root)\"`" **]** #si no hay string que concuerde con el uid root...

**then**

**echo** "Necesitas ser root para instalar!!"

**echo** " "

**echo** "Abortando instalacion..."

**exit** 1

**fi**

# ----------------------

#Instalación de PHP

# ----------------------

apt-get install php5 php5-pgsql php5-dev

# ----------------------

#Instalación de rudiments

# ----------------------

wget "http://downloads.sourceforge.net/project/rudiments/rudiments/0.48/rudiments-0.48.tar.gz"

**tar** **-**xzvf rudiments-0.48.tar.gz

**cd** rudiments-0.48

**./**configure

**make**

**make** install

**cd** **..**

# ----------------------

#Instalación de sqlrelay

# ----------------------

wget "http://downloads.sourceforge.net/project/sqlrelay/sqlrelay/0.57/sqlrelay-0.57.tar.gz"

**tar** **-**xzvf sqlrelay-0.57.tar.gz

**cd** sqlrelay-0.57

**./**configure

**make**

**make** install

**cd** **..**

# ----------------------

# Iniciar servicio

# ----------------------

**export** PATH**=$PATH:/**usr**/**local**/**firstworks**/**bin

sqlr-start –id instanciapsql

sqlr-start –id router

# Configuración de SQL relay

Las configuraciones de SQL Realy, se hacen dentro del archivo con extensión .conf comúnmente sqlrelay.conf aunque puede tener cualquier nombre, este archivo suele situarse en el directorio ../firstworks/etc/

tags

Cada instancia se define mediante tags de SQL Relay.

El tag router define una instancia para rutear y filtrar los querys hacia otra instancia de SQL Relay, Si este tag está presente, la instancia no debe contener tags de conexiones.

El tag route designa una instancia de SQL Relay para rutear los querys. Debe contener un conjunto de tags de querys que definen qué consultas se van a rutear a la instancia designada. Esta etiqueta sólo es válida dentro de un tag de router.

El tag filter define un conjunto de consultas que de ser filtrados (es decir, si un cliente envía una de estas consultas, el servidor de retransmisión de SQL no se ejecutará en la base de datos y se devolverá un error). Esta etiqueta sólo es válida dentro de un tag de router.

Atributos

Instancia

* id – El ID de la instancia
* port – El número de puerto en el que escucharasocket - The unix socket (filename) the listener should listen on (note, if multiple instances are configured with the same port or socket, they cannot be run at the same time)
* dbase – El tipo de base de datos a la que se conectara. Puede ser: oracle8, mysql, postgresql, sqlite, freetds, sybase, odbc, db2, firebird o mdbtools
* connections – El número de demonios sqlr-connection que iniciran cuando se utilie sqlr-start. Debe ser mayor o igual a 0.
* maxconnections – El numero máximos de demonios de sqlr-connection

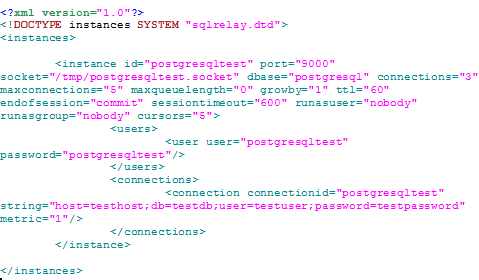
Usuario

* user – El nombre de usuario para especificar el sqlr-listener
* password – El password que utilizara el listener para conectarse, normalmente en texto plano, pero puede ir cifrado utilizando el módulo de cifrado.

Conexión

* connectionid – El ID de la conexión.
* string – La cadena que utilizara el demonio para realizar la conexión.
* Para bases de datos de postgresql, La cadena para conectar es: "user=USER;password=PASSWORD;db=DB;host=HOST;port=PORT"
* user: El nombre de usuario de SQLRelay que se utilizara para entrar en la base de datos.
* password: El password de SQLRelay que se utilizara para entrar en la base de datos.
* db: El nombre de la base de datos que se utilizará.
* host: El host al que se conectará.
* port: El puerto remoto, opcional, por default 3306.
* socket: El unix socket que se utilizará, opcional.

Ejemplo del archivo .conf para postgresql.



Ejecutar SQL relay

El sqlr-start se proporciona programa para simplificar el proceso de inicio.

sqlr-start-id ID [-config CONFIGFILE] [-localstatedir localstatedir]

CONFIGFILE es el nombre del archivo de configuración para utilizar. El archivo de configuración por defecto es / usr / local / firstworks / etc / sqlrelay.conf

# Configuración de Libinjection

Existen bastantes archivos dentro del directorio C, pero sólo se tendrá que copiar los siguientes 3 archivos en el árbol de código fuente:

* c / libinjection.h
* c / libinjection\_sqli.c
* c / libinjection\_sqli\_data.h

# SQL Relay drop-in replacement

Una de las características más importantes de SQL Relay es que provee de un drop-in replacement library para PostgreSQL.

Drop-in replacement library, es una librería de objetos compartidos (Archivos \*.so), que en conjunto de la directiva ***LD\_PRELOAD***, se pueden ejecutar ciertos comandos y llamadas al manejador de bases de datos Postgres con las bibliotecas compiladas de SQL Relay. Esto logra una mejor interesacción de SQL Relay con el cliente nativo de Postgres (psql).

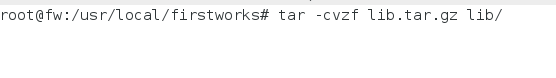
## Incorporación de SQL Relay con Drupal y Apache HTTPD

Para poder unir el firewall SQL generado con SQL Relay con el servidor Web Apache HTTPD se deben realizar los siguientes pasos

## Exportar las librerías de drop-in replacement de SQL Relay

**Del lado del servidor del firewall SQL**

* Exportar el directorio *lib* que usa SQL Relay, en una instalación normal se encuentra en el directorio */usr/local/firstworks/lib,* para esto se crea un archivo \*.tar.gz para exportar el directorio antes mencionado.



**Lado del servidor web**

* Crear los directorios necesarios para asimilar la misma configuración del servidor del firewall SQL.



* Descomprimir el archivo \*.tar.gz



## Integración de SQL Relay drop-in replacement con Apache HTTPD

Existen tres maneras de lograr la incorporación de SQL Relay drop-in replacement con Apache HTTPD, dependiendo de la manera de instalación y configuración de Apache y PHP

1. PHP fue instalado con los paquetes libapache2-mod-php5, que es la opción más utilizada en sistemas Linux Debian y Ubuntu
2. PHP es ejecutado usando el módulo mod\_fcgid
3. PHP trabaja con el soporte suEXEC de Apache

Para efectos prácticos se detallará la instalación de la primera opción ya que es la cual se emplea con mayor frecuencia, y se mencionará como se realizará en las otras opciones, pero con menos énfasis.

### PHP fue instalado con los paquetes libapache2-mod-php5

1. Se debe localizar el archivo de variables de entorno de apache, el cual es *ennvars.*



1. Posteriormente agregar al final dela rchivo la siguiente sentencia

export LD\_PRELOAD=/usr/local/firstworks/lib/libpqsqlrelay.so

1. Reiniciar el servidor con el siguiente comando.

# service apache2 restart

### PHP es ejecutado usando el módulo mod\_fcgid

1. Localizar el archivo que manda a llamar a PHP y modificarlo de la siguiente manera

#!/bin/sh

export PHPRC = /etc/php5/cgi

export LD\_PRELOAD = /usr/local/firstworks/lib/libpqsqlrelay.so

exec /usr/lib/cgi-bin/php

1. Reiniciar el servidor Apache

# service apache2 restart

### PHP trabaja con el soporte suEXEC de Apache

1. Agregar en la configuración del VirtualHost las siguientes directivas de Apache.

ScriptAlias /php\_amon/ "/usr/local/bin/"  
AddHandler php-script .php  
Action php-script /php5-cgi

1. Crear el archive ***/usr/local/bin/php5-cgi*** con el siguiente contenido:

#!/bin/sh  
export LD\_PRELOAD=/usr/local/firstworks/lib/libpqsqlrelay.so  
exec /path/of/the/real/php5 "$@"

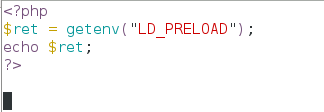
1. Reiniciar el servidor Apache

# service apache2 restart

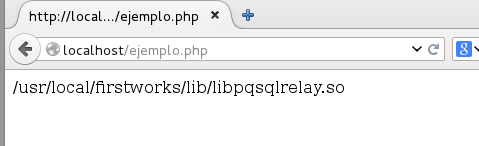
### Comprobación de la configuración

1. Comprobar la configuración creando el archivo *ejemplo.php*, con el siguiente contenido

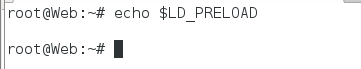
$ vim /var/www/ejemplo.php



1. Acceder por medio de un navegador web al recurso creado y verificar que la variable de entorno en conjunto con la directiva de LD\_PRELOAD fue modificada con éxito.



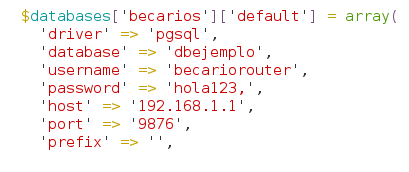
1. Realizar el siguiente comando para comprobar que la configuración realizada solo afecta al servidor de aplicaciones Apache y no al resto del sistema operativo



## Integración con Drupal

Después dela configuración, la integración con Drupal es transparente para el CMS, por lo cual la configuración de la conexión a la base de datos se hace de la manera definida por el sitio oficial de Drupal.

Ejemplo de configuración de conexión de Drupal con una base de datos en Postgres.



# Creación de paquetes deb

## Paquetes deb

Los paquetes son la agrupación instalable de los diferentes archivos necesarios para que un programa funcione. En GNULinux los paquetes no son ejecutables, sino que son gestionados por terceras aplicaciones. Además, los paquetes de Linux suelen ser muchísimo más compactos y reducidos ya que no traen consigo las librerías compartidas (dependencias) y, llegado el caso de necesitarse, serán instaladas o se nos informará de la necesidad de hacerlo, según el gestor de paquetes que estemos utilizando.

Los paquetes DEB (.DEB), suelen formarse por un nombre que alude al paquete seguido de los números de versión del paquete y puede contener también un indicativo de la plataforma para la que ha sido creado y compilado.

## Requisitos

Para la generación de paquetes deb se requiere contar con las siguientes bibliotecas

* build-essential
* autoconf
* automake
* autotools-dev
* dh-make
* debhelper
* devscripts
* fakeroot
* xutils
* lintian
* pbuilder

## Creación

Para la creación de los paquetes deb de rudiments y de SQL relay se puede utilizar el siguiente script

#!/bin/bash

# ----------------------

#Instalación de requisitos

# ----------------------

sudo apt-get install build-essential autoconf automake autotools-dev dh-make debhelper devscripts fakeroot xutils lintian pbuilder

sudo apt-get install php5 php5-pgsql php5-dev

sudo apt-get install checkinstall

# ----------------------

#rudiments deb

# ----------------------

wget "http://downloads.sourceforge.net/project/rudiments/rudiments/0.48/rudiments-0.48.tar.gz"

**tar** **-**xzvf rudiments-0.48.tar.gz

**cd** rudiments-0.48

**./**configure

**make**

sudo checkinstall **--**fstrans**=**no

**cd** **..**

# ----------------------

#sqlrelay deb

# ----------------------

wget "http://downloads.sourceforge.net/project/sqlrelay/sqlrelay/0.57/sqlrelay-0.57.tar.gz"

**tar** **-**xzvf sqlrelay-0.57.tar.gz

**cd** sqlrelay-0.57

**./**configure

**make**

sudo checkinstall **--**fstrans**=**no

**cd** **..**

Este script creara los archivo en la ubicación de las carpetas de SQL relay y de Rudiments.

# Referencias

<https://media.blackhat.com/bh-us-12/Turbo/Galbreath/BH_US_12_Galbreath_Libinjection_Slides.pdf>

<http://www.imperva.com/Products/DatabaseFirewall>

<https://www.debian.org/doc/manuals/packaging-tutorial/packaging-tutorial.en.pdf>

<http://www.linux.com/learn/tutorials/307110-easy-package-creation-with-checkinstall>

<http://www.linux.com/learn/tutorials/307110-easy-package-creation-with-checkinstall>

<http://revista.seguridad.unam.mx/numero-18/firewall-de-bases-de-datos>

<http://rudiments.sourceforge.net/about.html>

<http://www.darkreading.com/risk/hackers-choice-top-six-database-attacks/d/d-id/1129481>

<http://www.redseguridad.com/opinion/articulos/la-importancia-y-la-necesidad-de-proteger-la-informacion-sensible>

<http://www.webupd8.org/2010/01/how-to-create-deb-package-ubuntu-debian.html>

<http://www.webupd8.org/2010/01/how-to-create-your-own-gpg-key.html>

<https://wiki.debian.org/HowToPackageForDebian>

<http://www.linuxuser.co.uk/tutorials/make-your-own-deb-and-rpm-packages>

<http://www.modsecurity.org/>

<http://askubuntu.com/questions/140998/compiling-source-into-a-deb-package>

<http://santi-bassett.blogspot.mx/2014/07/how-to-create-debian-package.html>

<https://github.com/tonejo/firewall-sql>

<http://www.cyberciti.biz/faq/ubuntu-linux-how-do-i-install-deb-packages/>

<http://www.linuxjournal.com/content/using-checkinstall-build-packages-source>

<http://www.postgresql.org.es/>