GUÍA DE SEGURIDAD DE LAS TIC (CCN-STIC-462)

SEGURIDAD EN JOOMLA

AGOSTO 2016

Edita:

GOBIERNO DE ESPAÑA
MINISTERIO DE LA PRESIDENCIA

CCN Centro Criptológico Nacional

CENTRO CRIPTOLOGICO NACIONAL
2.5.4.13=Qualified Certificate: AAPP-SEP-M-SW-KPSC,
ou=sello electrónico, serialNumber=S2800155J,
o=CENTRO CRIPTOLOGICO NACIONAL, cn=CENTRO
CRIPTOLOGICO NACIONAL, c=ES
2016.08.08 17:31:59 +02'00'

NIPO: 002-16-018-9

Fecha de Edición: agosto 2016

PRÓLOGO

Entre los elementos más característicos del actual escenario nacional e internacional figura el desarrollo alcanzado por las Tecnologías de la Información y las Comunicaciones (TIC), así como los riesgos emergentes asociados a su utilización. La Administración no es ajena a este escenario, y el desarrollo, adquisición, conservación y utilización segura de las TIC por parte de la Administración es necesario para garantizar su funcionamiento eficaz al servicio del ciudadano y de los intereses nacionales.

Partiendo del conocimiento y la experiencia del Centro sobre amenazas y vulnerabilidades en materia de riesgos emergentes, la Ley 11/2002, de 6 de mayo, reguladora del Centro Nacional de Inteligencia, encomienda al Centro Nacional de Inteligencia el ejercicio de las funciones relativas a la seguridad de las tecnologías de la información en su artículo 4.e), y de protección de la información clasificada en su artículo 4.f), a la vez que confiere a su Secretario de Estado Director la responsabilidad de dirigir el Centro Criptológico Nacional en su artículo 9.2.f).

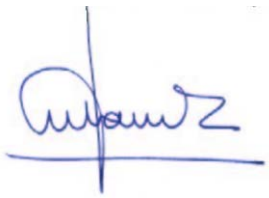
Una de las funciones más destacables que, asigna al mismo, el Real Decreto 421/2004, de 12 de marzo, por el que se regula el Centro Criptológico Nacional es la de elaborar y difundir normas, instrucciones, guías y recomendaciones para garantizar la seguridad de los sistemas de las tecnologías de la información y las comunicaciones de la Administración.

La ley 11/2007, de 22 de junio, de acceso electrónico de los ciudadanos a los servicios públicos, en su artículo 42.2 crea del Esquema Nacional de Seguridad (ENS), que establece las condiciones necesarias de confianza en el uso de los medios electrónicos, a través de medidas para garantizar la seguridad de los sistemas, los datos, las comunicaciones, y los servicios electrónicos.

El Real Decreto 3/2010, de 8 de enero, desarrolla el Esquema Nacional de Seguridad y fija los principios básicos y requisitos mínimos así como las medidas de protección a implantar en los sistemas de la Administración. En su artículo 29 se autoriza que a través de la series CCN-STIC el CCN desarrolle lo establecido en el mismo.

La serie de documentos CCN-STIC se ha elaborado para dar cumplimiento a esta función y a lo reflejado en el ENS, conscientes de la importancia que tiene el establecimiento de un marco de referencia en esta materia que sirva de apoyo para que el personal de la Administración lleve a cabo su difícil, y en ocasiones, ingrata tarea de proporcionar seguridad a los sistemas de las TIC bajo su responsabilidad.

Agosto 2016



Félix Sanz Roldán
Secretario de Estado
Director del Centro Criptológico Nacional

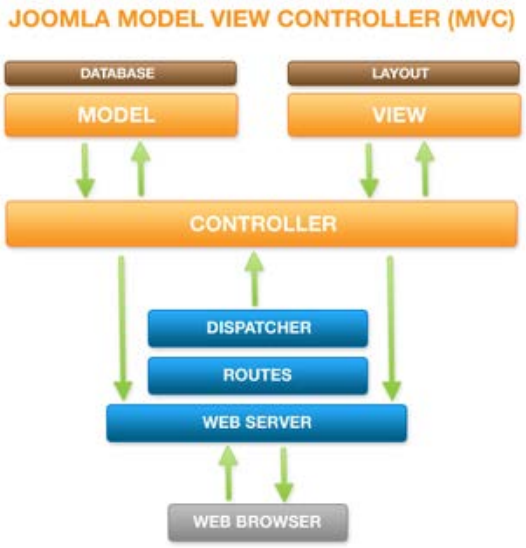
ÍNDICE

1. INTRODUCCIÓN

1. Joomla! [1] es un sistema de gestión de contenidos (o CMS, por las siglas en inglés, Content Management System) que permite desarrollar sitios web dinámicos e interactivos. Permite crear, modificar o eliminar contenido de un sitio web de manera sencilla a través de un panel de administración. Es un software de código abierto, programado o desarrollado en PHP y liberado bajo licencia pública general GNU (GPL).
2. Su nombre es una pronunciación fonética *jumla*, que significa "todos juntos" o "como un todo". Se escogió como una reflexión del compromiso del grupo de desarrolladores y la comunidad del proyecto.

1.1. ARQUITECTURA

3. Está desarrollado en una arquitectura Modelo Vista Controlador (MVC), lo que permite:
    - Interactuar directamente con la parte de vista de Joomla!, ofreciendo la posibilidad de sobrescribir desde el *template* de Joomla! la parte de vista de un componente, módulo o plugin, permitiendo un gran nivel de personalización en el desarrollo.
    - Un desarrollo de componentes módulos y plugins basados en la arquitectura base del CMS.
    - Actualizaciones rápidas, actualizando los elementos requeridos para los cambios de versión con la plataforma funcionando.



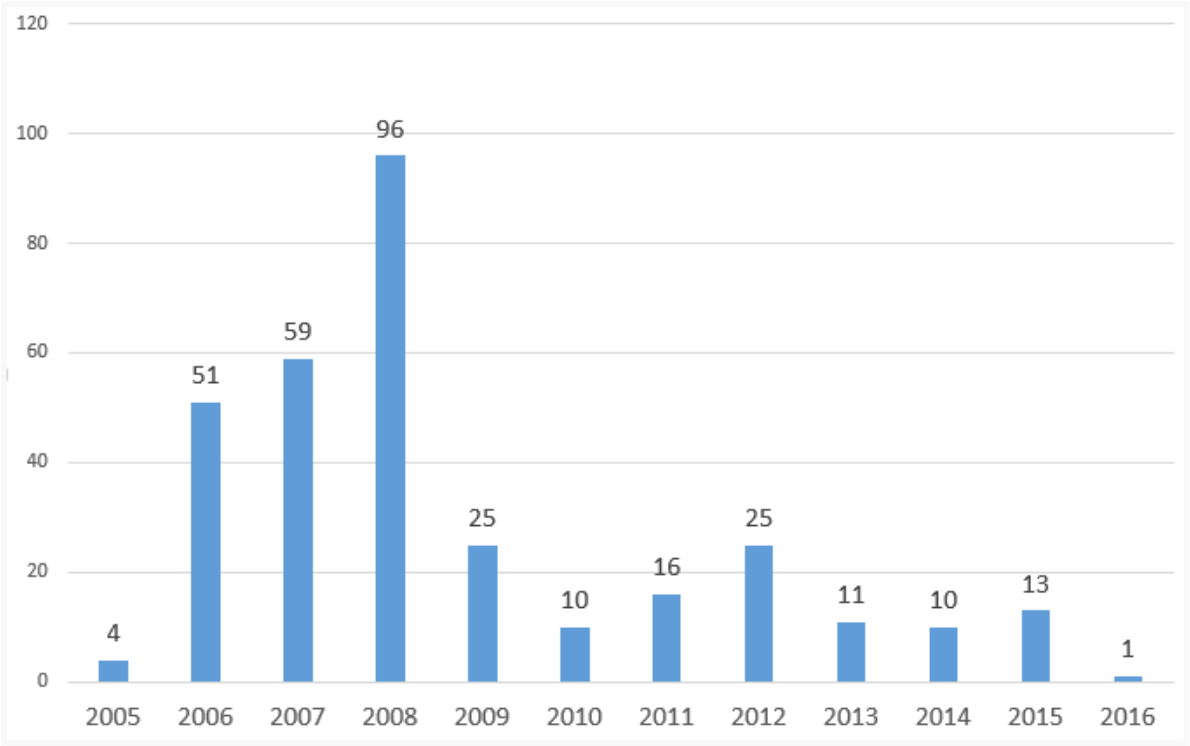
[1] **JOOMLA:** https://es.wikipedia.org/wiki/Joomla

4. Una de las mayores potencialidades que tiene este CMS es que su funcionalidad base puede ser extendida por medio de extensiones, que quedarían organizadas en las siguientes categorías que se enumeran a continuación:

   - **Componentes**: Son extensiones que cargan como elemento funcional principal en Joomla!. Un componente generalmente agrega una funcionalidad de impresión de contenido a Joomla, como podría ser el caso de *com_content*, que es el componente principal de administración de contenido estándar de Joomla. Se caracterizan porque Joomla, de forma nativa, solo soporta el manejo de un solo componente al mismo tiempo.
   - **Módulos**: Son extensiones que permiten cargar en una posición de una plantilla una salida de datos. Generalmente, un módulo nos permitirá colocar la salida de un componente asignado.
   - **Plantillas**: Son extensiones que permiten cambiar la parte de vista del CMS. El archivo "index.php" de la plantilla de Joomla! determina donde cargarán los componentes y los módulos.
   - **Plugins**: Los plugins son extensiones que corren bajo disparadores seleccionados, realizando acciones seleccionadas mientras el CMS realiza la carga.
   - **Lenguajes**: Los lenguajes son archivos de traducción que permiten colocar el CMS en cualquier idioma (Joomla es multi-idioma nativo).

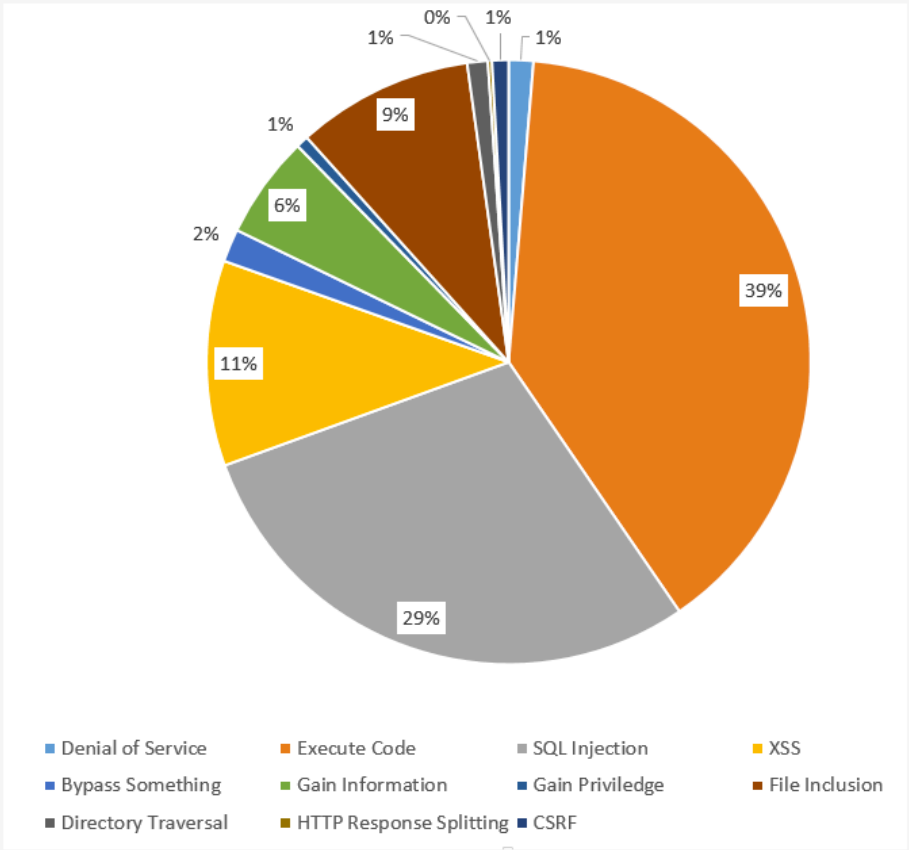
1.2. HISTÓRICO DE VULNERABILIDADES[2]

5. De acuerdo a la página CVE Details, se han reportado 321 vulnerabilidades desde el año 2005 hasta el momento:



[2] **COMPLETE GUIDE ON JOOMLA SECURITY:** https://www.keycdn.com/blog/joomla-security/

6. Cabe destacar que el 39% de las vulnerabilidades públicas conocidas corresponden a RCE (Remote Code Execution):



7. Por este motivo, utilizaremos las mejores prácticas que se muestran a lo largo de esta guía para mantener el CMS lo más seguro posible y mitigar posibles ataques de fuentes externas.

1.3. VERSIONES MÁS IMPORTANTES[3]

8. A continuación mostramos las versiones más relevantes que han sido publicadas, que han sido extraídas de la documentación oficial de Joomla.

9. Si se está ejecutando una versión anterior a la última liberada, deberá ser actualizada. Cada nueva versión soluciona una serie errores y/o problemas de seguridad. Siempre es una buena práctica actualizar a la última versión como primer paso ante un determinado problema de ejecución, así como cualquier problema de seguridad detectado.

Versión del CMS	Disponible	Soporte		Fin de vida útil (EOL)	Tipo de actualización	Notas	Última versión
		Bugs	Seguridad				
1.5	✗	✗	✗	Sept 2012	Migración a 2.5	Plantéate migrar a la 2.5 hoy mismo	**EOL** en la 1.5.26
1.6	✗	✗	✗	Ago 2011	Con un solo clic a 2.5	Actualiza a la 2.5 hoy mismo	1.6.6

[3] **VERSIONES JOOMLA:** https://docs.joomla.org/Joomla!_CMS_versions

1.7	✗	✗	✗	Feb 2012	Con un solo clic a 2.5	Actualiza a la 2.5 hoy mismo	1.7.5
2.5	✗	✗	✗	Dic 2014	Con un clic a la 3.x	Actualiza a 3.5.1 ahora mismo	2.5.28
3.0	✗	✗	✗	May 2013	Con un solo clic a 3.1	Deberías usar la actualización en un clic	3.0.4
3.1	✗	✗	✗	Dic 2013	Con un solo clic a 3.2	Deberías usar la actualización en un clic	3.1.6
3.2	✗	✗	✗	Oct 2014	Con un solo clic a 3.3	Deberías actualizar el PHP de tu servidor a la versión 5.3.10 o superior y actualizar a la 3.3	3.2.7
3.3	✗	✗	✗	Versión 3.4	Con un clic	Deberías usar la actualización en un clic	**3.3.6**
3.4	✗	✗	✗	Versión 3.5	Con un clic	Recomendada para todas las nuevas instalaciones	**3.4.8**
3.5	✓	✓	✓	Versión 3.6	Con un clic	Recomendada para todas las nuevas instalaciones	**3.5.1**
3.6	por definir	-	-	Versión 3.7	Con un clic		
3.7	por definir	-	-	Versión 3.8	Con un clic		
...	...	...	...	...	...	...	...
4.0	2016	-	-				

2. INSTALACIÓN Y ACTUALIZACIÓN

2.1. ANTES DE COMENZAR: REQUERIMIENTOS

10. Antes de comenzar, existen una serie de requerimientos[4] mínimos a cumplir para poder realizar la instalación de forma exitosa:

Software	Recomendado	Mínimo	Más información
PHP (Magic Quotes GPC off)	5.6 + o 7 +[1]	5.3.10 +	www.php.net
Motores de base de datos soportados:			
MySQL[2] (Se requiere soporte InnoDB)	5.5.3 +	5.1 +	www.mysql.com
SQL Server	10.50.1600.1 +	10.50.1600.1 +	www.microsoft.com/sql
PostgreSQL	9.1 +	8.3.18 +	www.postgresql.org
Servidores web soportados:			
Apache(mod_mysql, mod_xml, y mod_zlib)	2.4 +	2.x +	www.apache.org
Nginx	1.8 +	1.0 +	wiki.nginx.org
Microsoft IIS	7	7	www.iis.net

[4] **REQUISITOS PARA SOFTWARE SOPORTADO:** https://docs.joomla.org/Technical_requirements/es

NOTAS ADICIONALES:

- Sólo las versiones de Joomla! 3.5 y posteriores son compatibles con PHP 7
- Joomla! aún no es compatible con MySQL 6.x
- Para poder usar URLs amigables con los motores de búsqueda, se deberá instalar la extensión mod_rewrite de Apache

11. Además, será necesario instalar un servidor Web que gestione las peticiones realizadas por parte del cliente. Durante el desarrollo de esta guía utilizaremos como ejemplo Apache (que deberá ser asegurada siguiendo la guía **CCN-STIC 671 "Configuración segura de servidores web Apache"**), aunque puede utilizarse cualquier otro que soporte PHP y MySQL, como podría ser Nginx.

COMPROBAR LA VERSIÓN DE PHP INSTALADA EN EL SISTEMA

En el caso de que tener acceso por línea de consola al sistema, ejecutaremos el siguiente comando:

```
root@guiaJoomla:~# php -v
PHP 5.5.9-1ubuntu4.11 (cli) (built: Jul  2 2015 15:23:08)
Copyright (c) 1997-2014 The PHP Group
Zend Engine v2.5.0, Copyright (c) 1998-2014 Zend Technologies
    with Zend OPcache v7.0.3, Copyright (c) 1999-2014, by Zend Technologies
root@guiaJoomla:~#
```

En caso contrario, crearemos un fichero, al que llamaremos como ejemplo info_php.php, y lo subiremos a nuestra estructura web. El código que debe contener este fichero será:

```
root@guiaJoomla:/var/www/html# cat info_php.php
<?php phpinfo(); ?>
root@guiaJoomla:/var/www/html#
```

Una vez creado, tan sólo deberemos acceder desde nuestro navegador a http://nuestroservidor/ruta/info_php.php, donde deberemos ver algo como la siguiente captura:

PHP Version 5.5.9-1ubuntu4.11

System	Linux guiaJoomla 3.19.0-25-generic #26~14.04.1-Ubuntu SMP Fri Jul 24 21:16:20 UTC 2015 x86_64
Build Date	Jul 2 2015 14:51:39
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/05-opcache.ini, /etc/php5/apache2/conf.d/10-pdo.ini, /etc/php5/apache2/conf.d/20-json.ini, /etc/php5/apache2/conf.d/20-mysql.ini, /etc/php5/apache2/conf.d/20-mysqli.ini, /etc/php5/apache2/conf.d/20-pdo_mysql.ini, /etc/php5/apache2/conf.d/20-readline.ini

En la parte superior nos mostrará la versión de PHP que está instalada y corriendo en el sistema, en nuestro caso 5.5.9-1. En caso de no estar ejecutando PHP 5, será necesario contactar con los administradores para solicitar su instalación.

Una vez realizada esta prueba, el fichero que hemos creado deberá ser eliminado, ya que puede proporcionar información adicional y necesaria para que un atacante realice una intrusión sobre nuestro sistema.

12. Además de estos requerimientos mínimos, utilizaremos una serie de herramientas de apoyo para realizar la configuración del servicio:

    - Acceso al servidor web (vía consola, SSH o FTP)
    - Un editor de texto
    - Un cliente FTP/SFTP
    - Un navegador Web (en nuestro caso utilizaremos la última versión de Firefox 46.01)

2.2. INSTALACIÓN DE REQUERIMIENTOS

2.2.1. INSTALACIÓN DE MYSQL

13. Antes de comenzar, será necesario descargar la última versión de MySQL disponible desde la web oficial: http://dev.mysql.com/downloads/mysql/. En nuestro caso, descargaremos la última reléase disponible en el momento de la publicación de esta guía: mysql-5.7.12.tar.gz[5].

14. La secuencia de comandos necesaria para la instalación será la siguiente, y será realizado desde la cuenta de "root" o mediante el uso de una cuenta con permisos de administración del sistema:

 # Descargamos MySQL
$ wget https://dev.mysql.com/get/Downloads/MySQL-5.6/mysql-5.7.12.tar.gz
 # Descomprimimos el paquete
$ tar xzvf mysql-5.7.12.tar.gz
$ cd mysql-5.7.12/
 # Comprobación de dependencias
~/mysql-5.7.12$ BUILD/autorun.sh
 # Compilamos MySQL, generando los ejecutables
~/mysql-5.7.12$ CFLAGS="-O3" CXX=gcc CXXFLAGS="-O3 -felide-constructors -fno-exceptions -fno-rtti" ./configure --prefix=/usr/local/mysql --enable-assembler --with-mysqld-ldflags=-all-static

```
root@guiaJoomla:~/mysql-5.7.12# CFLAGS="-O3" CXX=gcc CXXFLAGS="-O3 -felide-constructors -fno-exceptions -fno-rtti"
 ./configure --prefix=/usr/local/mysql --enable-assembler --with-mysqld-ldflags=-all-static
configure.pl : switching CXX compiler from gcc to g++
configure.pl : stripping off -fno-exceptions CXXFLAGS=-O3 -felide-constructors  -fno-rtti
configure.pl : ignoring with-mysqld-ldflags=-all-static
configure.pl : calling cmake /root/mysql-5.7.12  -DCMAKE_INSTALL_PREFIX=/usr/local/mysql -DENABLE_ASSEMBLER=1
-- Running cmake version 2.8.12.2
-- Could NOT find Git (missing:  GIT_EXECUTABLE)
-- Configuring with MAX_INDEXES = 64U
-- The C compiler identification is GNU 4.8.4
-- The CXX compiler identification is GNU 4.8.4
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working CXX compiler: /usr/bin/g++
-- Check for working CXX compiler: /usr/bin/g++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Looking for SHM_HUGETLB
-- Looking for SHM_HUGETLB - found
-- Looking for sys/types.h
-- Looking for sys/types.h - found
-- Looking for stdint.h
-- Looking for stdint.h - found
```

[5] **MD5 (mysql-5.7.12.tar.gz):** af17ba16f1b21538c9de092651529f7c

```
# Instalamos los paquetes generados
~/mysql-5.7.12$ make install
# Añadimos el grupo y usuario ‘mysql’
~/mysql-5.7.12$ groupadd mysql
~/mysql-5.7.12$ useradd -g mysql mysql
~/mysql-5.7.12$ cd /usr/local/mysql
/usr/local/mysql$ chown -R mysql .
/usr/local/mysql$ chgrp -R mysql .
/usr/local/mysql$ scripts/mysql_install_db --user=mysql
/usr/local/mysql$ chown -R root .
/usr/local/mysql$ chgrp -R mysql .
/usr/local/mysql$ chown -R mysql data/
/usr/local/mysql$ chmod -R go-rwx data/
# Generamos un fichero de configuración desde la plantilla
/usr/local/mysql$ cp support-files/my-default.cnf /etc/my.cnf
# Cambiamos la contraseña de acceso del usuario 'root'
/usr/local/mysql$ bin/mysqladmin -u root password 'new-password'
```

15. Una vez que la instalación ha finalizado, arrancaremos el servicio utilizando el siguiente comando:

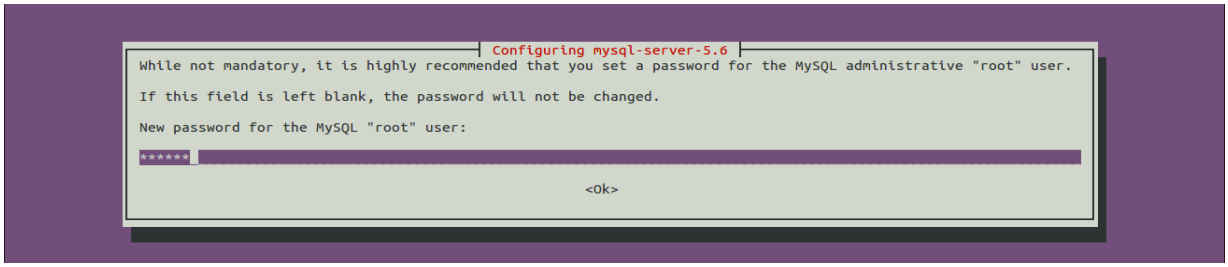
    $ bin/mysqld_safe --user=mysql &

16. Desde este momento dispondremos de MySQL en nuestro sistema, dentro de la ruta ‘/usr/local/mysql’.

17. Como alternativa, en sistemas Debian/Ubuntu, podremos utilizar el gestor de paquetes propio para realizar la instalación, sin compilar las fuentes. Para ello, ejecutaremos los siguientes comandos:

    $ sudo apt-get install mysql-server php5-mysql libapache2-mod-auth-mysql

18. Introduciremos la contraseña de 'root' para poder acceder posteriormente a MySQL:



19. Al finalizar el proceso, tendremos instalada la última version de MySQL así como los módulos necesarios para que poder utilizarlo bajo PHP.

3. INSTALACIÓN DE PHP

3.1. MOD_PHP VS PHP-FPM

20. *mod_php* es el paquete que se instala que por defecto en la mayoría de sistemas, y se encuentra en todos los repositorios de paquetes de cualquier distribución de Linux. Es muy sencillo de instalar y configurar, motivo por el que es el más utilizado.

21. Con este módulo, el intérprete de PHP se ejecuta dentro del proceso de Apache, es decir, no hacen falta llamadas externas a ningún proceso, produciéndose toda la comunicación de forma interna. Sin embargo, cada proceso hijo de Apache debe cargar la extensión, lo que genera una carga mucho mayor de lo necesario cuando se procesan, por ejemplo, ficheros estáticos como imágenes, hojas de estilo CSS o código Javascript. Muchos servidores tienen esta configuración, ya que es por defecto y no se dan cuenta de que también es posible configurar PHP como CGI. Ejecutar PHP como CGI puede ser más seguro en algunos casos, evitando también problemas de permisos de archivos y directorios.

22. *PHP-FPM*, una nueva forma de utilizar PHP con un servidor web, es una implementación alternativa de PHP FastCGI, con algunas características adicionales que pueden ser útiles para los sitios web que se ejecutan en pequeños VPS y servidores. Puede ser utilizado con cualquier servidor web compatible con FastCGI.

23. *PHP-FPM* proporciona a los administradores la capacidad de detener y generar procesos de PHP sin perder las consultas que están siendo procesadas. Puede comenzar con procesos de diferentes uid, gid, y entornos chroot, así como diferentes opciones de configuración de PHP. También puede ejecutar un reinicio de emergencia en todos los procesos en el caso de la destrucción accidental de memoria compartida de caché, si se está utilizando un acelerador.

24. A pesar de que requiere mayor configuración y que el proyecto es relativamente joven (en comparación con *mod_php*), se recomienda su instalación para una ejecución más segura de Joomla!, según la propia documentación del framework. Igualmente, a continuaciónm se expondrá la forma de realizar ambas instalaciones.

3.2. INSTALACIÓN DE MOD_PHP

25. Antes de comenzar, será necesario descargar la última versión de PHP disponible desde la web oficial: http://php.net/downloads.php. En nuestro caso, descargaremos la última release disponible en el momento de la publicación de esta guía: php-5.6.21.tar.gz[6].

26. La secuencia de comandos necesaria para la instalación será la siguiente, y será realizado desde la cuenta de ‘root’ o con permisos de administración del sistema:

```
# Configuramos la instalacion
$ tar zxvf php-5.6.21.tar.gz
$ cd php-5.6.21/
# Especificamos la ruta de instalacion a /usr/local/mysql
$ ./configure --with-mysql=/usr/local/mysql
```

[6] **MD5(php-5.6.21.tar.gz):** 89047bc5219b95dcb47c045774893c4f

```
root@guiaJoomla:~/php-5.6.21# ./configure --with-mysql=/usr/local/mysql
checking for grep that handles long lines and -e... /bin/grep
checking for egrep... /bin/grep -E
checking for a sed that does not truncate output... /bin/sed
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking target system type... x86_64-unknown-linux-gnu
checking for cc... cc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether cc accepts -g... yes
checking for cc option to accept ISO C89... none needed
```

$ make
$ make install
 # Generamos un fichero de configuración desde la plantilla
$ cp php.ini-dist /usr/local/lib/php.ini

NOTA

Existen gran cantidad de opciones de configuración para PHP cuando se compila en entornos Unix-like. La mayoría hacen referencia a ubicaciones concretas o configuraciones que no han sido incluidas en el ejemplo.

Para una lista completa de estas opciones, accederemos a http://php.net/manual/en/configure.about.php y seleccionaremos las más adecuadas a nuestro entorno.

27. Finalmente, nos aseguraremos que desde el archivo de configuración se carga la extensión de MySQL, buscando el siguiente texto (o añadiéndolo a la configuración en caso contrario):

    extension=mysql.so

28. De la misma forma que con MySQL, en sistemas Debian/Ubuntu podremos utilizar el gestor de paquetes propio para la descarga e instalación de PHP, utilizando los siguientes comandos:

    $ apt-get install php5 php5-mysql libapache2-mod-php5

29. Una vez finalizada la instalación, tendremos correctamente configurado PHP para trabajar con Apache, así como MySQL. Podemos comprobar que la instalación se ha realizado correctamente, generando un fichero de información de PHP, llamado "phpinfo.php", en la ruta por defecto de Apache (/var/www/html) con el siguiente contenido:

    <?php
    phpinfo();
    ?>

30. Utilizando el navegador accederemos a la dirección IP de nuestro servidor, de la forma “http://IP/phpinfo.php”. Si obtenemos la pantalla de información de *debugging* de PHP, nuestro servidor estará listo para continuar el proceso de instalación.

3.3. INSTALACIÓN DE PHP-FPM

31. Para la correcta instalación, deberemos instalar “apache2”, “php5”, así como una serie de paquetes adicionales con el comando que se muestra a continuación:

    # sudo apt-get install apache2 libapache2-mod-fcgid apache2-mpm-worker php5 php5-cgi apache2-suexec

32. A continuación, deshabilitamos "mod_php" y habilitamos resto de módulos de Apache necesarios:

    # a2dismod php5
    # a2enmod rewrite suexec include fcgid

33. Editaremos el fichero "/etc/php5/cgi/php.ini" para descomentar las siguientes líneas:

    *; cgi.fix_pathinfo provides *real* PATH_INFO/PATH_TRANSLATED support for* CGI. *PHP's*
    ; previous behaviour was to set PATH_TRANSLATED to SCRIPT_FILENAME, and to not grok
    ; what PATH_INFO is. For more information on PATH_INFO, see the cgi specs. Setting
    ; this to 1 will cause PHP CGI to fix its paths to conform to the spec. A setting
    ; of zero causes PHP to behave as before. Default is 1. You should fix your scripts
    ; to use SCRIPT_FILENAME rather than PATH_TRANSLATED.
    ; http://php.net/cgi.fix-pathinfo
    cgi.fix_pathinfo=1

34. Editaremos el fichero "/etc/apache2/mods-available/fcgid.conf" y lo dejaremos con la siguiente configuración que se muestra:

    <IfModule mod_fcgid.c>
    FcgidConnectTimeout 20
    PHP_Fix_Pathinfo_Enable 1

    <IfModule mod_mime.c>
    AddHandler fcgid-script .fcgi
    </IfModule>
    </IfModule>

```
root@guiaJoomla:/var/www/php-fcgi-scripts/joomla# cat /etc/apache2/mods-available/fcgid.conf
<IfModule mod_fcgid.c>
  FcgidConnectTimeout 20
  PHP_Fix_Pathinfo_Enable 1

  <IfModule mod_mime.c>
    AddHandler fcgid-script .fcgi
  </IfModule>
</IfModule>
```

35. Posteriormente, reiniciaremos Apache para que los cambios realizados surjan efecto:

    /etc/init.d/apache2 restart

36. Ahora comenzaremos la configuración del Virtual Host para nuestra instalación de Joomla! (como ejemplo para esta guía, generaremos un nuevo usuario y grupo para este proceso):

    $ groupadd joomla
    $ useradd -s /bin/false -d /var/www/html/joomla-cms/ -m -g joomla joomla

37. Ejecutaremos PHP utilizando suExec. Obtendremos la información de configuración utilizando el siguiente comando:

    $ /var/lib/apache2/suexec –V

```
root@guiaJoomla:/var/www/html# /usr/lib/apache2/suexec -V
 -D AP_DOC_ROOT="/var/www"
 -D AP_GID_MIN=100
 -D AP_HTTPD_USER="www-data"
 -D AP_LOG_EXEC="/var/log/apache2/suexec.log"
 -D AP_SAFE_PATH="/usr/local/bin:/usr/bin:/bin"
 -D AP_UID_MIN=100
 -D AP_USERDIR_SUFFIX="public_html"
```

38. Como observamos, el directorio raíz está localizado en "/var/www/html". Como nuestro binario de PHP se encuentra en "/usr/lib/cgi-bin/php", y ésta se encuentra fuera de la raíz de suExec, no podremos realizar su ejecución directamente. Debido a que suExec no soporta enlaces simbólicos, la única forma de poder solucionar el problema es crear un Wrapper Script[7] para cada sitio web en el subdirectorio de "/var/www", que se encargará de llamar al binario de PHP que hemos mencionado anteriormente.

39. Crearemos la ruta necesaria dentro de "/var/www":

    $ mkdir /var/www/php-fcgi-scripts/
    $ mkdir /var/www/php-fcgi-scripts/joomla

40. Crearemos el fichero "php-fcgi-starter" con el siguiente contenido:

    #!/bin/sh
    PHPRC=/etc/php5/cgi/
    export PHPRC
    export PHP_FCGI_MAX_REQUESTS=5000
    export PHP_FCGI_CHILDREN=8
    exec /usr/lib/cgi-bin/php

```
root@guiaJoomla:/var/www/php-fcgi-scripts/joomla# cat /etc/apache2/mods-available/fcgid.conf
<IfModule mod_fcgid.c>
  FcgidConnectTimeout 20
  PHP_Fix_Pathinfo_Enable 1

  <IfModule mod_mime.c>
    AddHandler fcgid-script .fcgi
  </IfModule>
</IfModule>
```

41. Además, estableceremos permisos, propietario y grupo a los anteriormente creados (en nuestro caso “joomla”/”joomla”).

    chmod 755 /var/www/php-fcgi-scripts/joomla/php-fcgi-starter
    chown -R joomla:joomla /var/www/php-fcgi-scripts/joomla

42. Finalmente, modificaremos nuestro fichero responsable de la configuración del sitio en Apache, sea el de por defecto "/etc/apache2/sites-available/000-default.conf", o uno que hayamos creado para configurar un Virtual Host específico, añadiendo las siguientes líneas que aparecen destacadas a continuación:

```
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html/joomla-cms

 <IfModule mod_fcgid.c>
   SuexecUserGroup joomla joomla

   <Directory /var/www/html/joomla-cms/>
    Options +ExecCGI
```

[7] **ADVANCED BASH-SCRIPTING GUIDE - SHELL WRAPPERS:** http://tldp.org/LDP/abs/html/wrapper.html

```
    AllowOverride All
    AddHandler fcgid-script .php
    FCGIWrapper /var/www/php-fcgi-scripts/joomla/php-fcgi-starter .php
    Order allow,deny
    Allow from all
  </Directory>
</IfModule>

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

NOTA
En caso de haber realizado los pasos indicados para activar el soporte de SSL, también será necesario modificar las directivas anteriormente indicadas en el fichero de configuración, por defecto " /etc/apache2/sites-available/default-ssl.conf".
En caso contrario, el soporte de PHP estará desactivado y no se podrán interpretar las páginas correctamente, mostrándolas como texto claro para cualquier visitante.

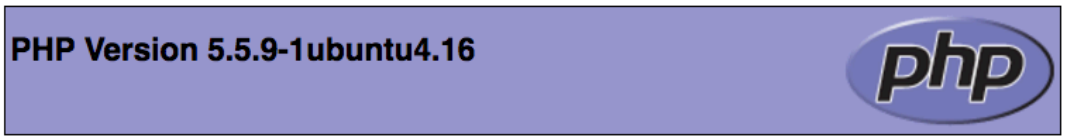
43. Recargamos Apache para que los cambios realizados surjan efecto:

```
/etc/init.d/apache2 reload
```

44. Podemos comprobar que la instalación se ha realizado correctamente, generando un fichero de información de PHP, llamado por ejemplo "info.php", en la ruta por defecto de Apache, con el siguiente contenido:

```
<?php
    phpinfo();
?>
```

45. Utilizando el navegador accederemos a la dirección IP de nuestro servidor, de la forma “http://IP/info.php”, y comprobaremos que la instalación del módulo ha sido satisfactoria:



System	Linux guiaJoomla 3.19.0-25-generic #26~14.04.1-Ubuntu SMP Fri Jul 24 21:16:20 UTC 2015 x86_64
Build Date	Apr 20 2016 14:28:53
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/cgi
Loaded Configuration File	/etc/php5/cgi/php.ini
Scan this dir for additional .ini files	/etc/php5/cgi/conf.d
Additional .ini files parsed	/etc/php5/cgi/conf.d/05-opcache.ini, /etc/php5/cgi/conf.d/10-pdo.ini, /etc/php5/cgi/conf.d/20-json.ini, /etc/php5/cgi/conf.d/20-mysql.ini, /etc/php5/cgi/conf.d/20-mysqli.ini, /etc/php5/cgi/conf.d/20-pdo_mysql.ini, /etc/php5/cgi/conf.d/20-readline.ini
PHP API	20121113
PHP Extension	20121212
Zend Extension	220121212

3.3.1. CONFIGURACIONES DE PHP DIFERENTES POR SITIO WEB

46. Durante el proceso de instalación del módulo, hemos creado diferentes cuentas y su propio script de inicio "php-fcgi-starter", por lo que podremos definir diferentes ficheros de configuración de PHP para cada uno de los sitios que administramos. Para ello, tan sólo debemos copiar el fichero a una ruta visible por “suExec”:

    $ cp /etc/php5/cgi/php.ini /var/www/html/
    $ chown joomla:joomla /var/www/html/php.ini
    $ chmod 440 /var/www/html/php.ini

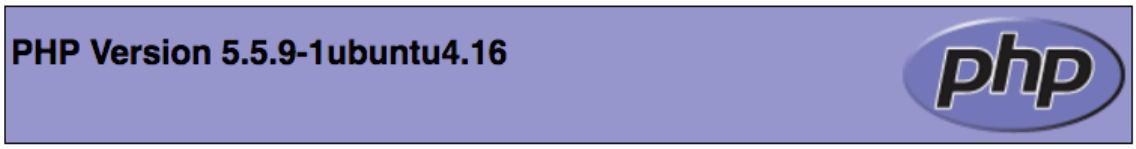
47. Modificamos los parámetros de PHP necesarios, y cambiamos la ruta en el script "/var/www/php-fcgi-scripts/joomla/php-fcgi-starter" para utilizar el nuevo fichero de configuración:

```
root@guiaJoomla:/var/www/php-fcgi-scripts/joomla# cat php-fcgi-starter
#!/bin/sh
PHPRC=/var/www/html/
export PHPRC
export PHP_FCGI_MAX_REQUESTS=5000
export PHP_FCGI_CHILDREN=8
exec /usr/lib/cgi-bin/php
```

48. Finalmente, reiniciaremos el servicio de Apache para cargar la modificación realizada:

    $ /etc/init.d/apache2 restart

49. Accediendo al fichero "info.php" que hemos creado anteriormente, podremos comprobar si los cambios han surgido efecto:



PHP Version 5.5.9-1ubuntu4.16

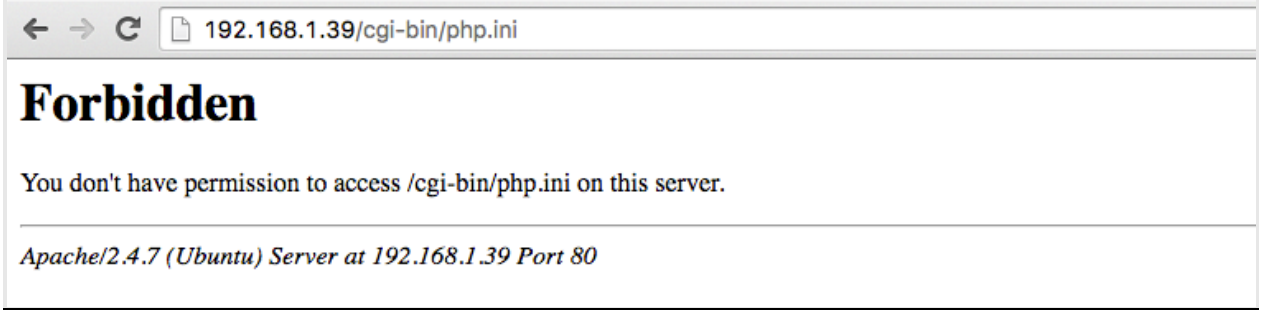
System	Linux guiaJoomla 3.19.0-25-generic #26~14.04.1-Ubuntu SMP Fri Jul 24 21:16:20 UTC 2015 x86_64
Build Date	Apr 20 2016 14:28:53
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/cgi
Loaded Configuration File	/var/www/html/php.ini
Scan this dir for additional .ini files	/etc/php5/cgi/conf.d
Additional .ini files parsed	/etc/php5/cgi/conf.d/05-opcache.ini, /etc/php5/cgi/conf.d/10-pdo.ini, /etc/php5/cgi/conf.d/20-json.ini, /etc/php5/cgi/conf.d/20-mysql.ini, /etc/php5/cgi/conf.d/20-mysqli.ini, /etc/php5/cgi/conf.d/20-pdo_mysql.ini, /etc/php5/cgi/conf.d/20-readline.ini
PHP API	20121113
PHP Extension	20121212

NOTA
Es necesario tomar medidas para que la información de configuración de PHP (contenida en el fichero "php.ini") no sea expuesta a visitantes y posibles atacantes. En nuestro caso, hemos generado el fichero en el directorio "/var/www/html", fuera del directorio raíz del sitio web "/var/www/html/joomla_cms", y con los permisos adecuados.

En caso de que mantenerlo dentro de la ruta del sitio, como por ejemplo el directorio "cgi-bin" como recomiendan algunas documentaciones técnicas, añadiremos las siguientes líneas al fichero ".htaccess":

```
<FilesMatch "^php5?\.(ini|cgi)$">
Order Deny,Allow
Deny from All
Allow from env=REDIRECT_STATUS
</FilesMatch>
```

Estas líneas denegarán el acceso a este tipo de ficheros a cualquier visitante:



50. Otra forma de realizar modificaciones en la configuración de PHP, sin tener que crear un fichero individual por cada Virtual Host de nuestra máquina, será introducir las modificaciones de PHP como parámetro en el script "php-fcgi-starter" de cada sitio. Por ejemplo, si queremos desactivar la opción "allow_url_fopen", dejaremos el siguiente contenido en el fichero:

```
#!/bin/sh
PHPRC=/etc/php5/cgi/
export PHPRC
export PHP_FCGI_MAX_REQUESTS=5000
export PHP_FCGI_CHILDREN=8
exec /usr/lib/cgi-bin/php -d allow_url_fopen=off
```

51. De nuevo, acudiremos a nuestro script de información de PHP, y comprobaremos que los cambios se realizan adecuadamente:

Core

PHP Version	5.5.9-1ubuntu4.16

Directive	Local Value
allow_url_fopen	Off
allow_url_include	Off
always_populate_raw_post_data	Off
arg_separator.input	&
arg_separator.output	&
asp_tags	Off

3.4. ACTIVACIÓN DE MOD_REWRITE

52. Para activar el módulo "mod_rewrite" en Apache, será necesario ejecutar lo siguiente:

```
$ sudo a2enmod rewrite
```

53. Después, revisaremos la configuración por defecto de nuestro Apache (en nuestro caso "etc/apache2/sites-enabled/000-default") y especificaremos el valor "All" para la directiva "AllOverride":

```
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html/joomla-cms

<Directory /var/www/html/joomla-cms>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride All
        Order allow,deny
        allow from all
</Directory>

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

54. Una vez realizado esto, reiniciaremos el servicio:

$ apache2ctl restart

4. INSTALACIÓN DE JOOMLA

4.1. INSTALACIÓN DESDE EL ARCHIVO OFICIAL

55. Si disponemos acceso a la línea de consola de nuestro servidor, podremos descargar la última versión de Joomla! (utilizando wget , Lynx o cualquier navegador de texto), que en el momento de la escritura de esta guía es la **3.5.1**.

56. Ahora deberemos descomprimir el paquete en la ruta definida por nuestro servidor web, utilizando el siguiente comando:

```
root@guiaJoomla:/var/www/html# wget https://github.com/joomla/joomla-cms/releases/download/3.5.1/Joomla_3.5.1-Stab
le-Full_Package.zip
--2016-05-19 16:44:29--  https://github.com/joomla/joomla-cms/releases/download/3.5.1/Joomla_3.5.1-Stable-Full_Pac
kage.zip
Resolviendo github.com (github.com)... 192.30.252.130
Conectando con github.com (github.com)[192.30.252.130]:443... conectado.
Petición HTTP enviada, esperando respuesta... 302 Found
Ubicación: https://github-cloud.s3.amazonaws.com/releases/2464908/c2a9ea34-fb82-11e5-9534-701271ce5d26.zip?X-Amz-A
lgorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAISTNZFOVBIJMK3TQ%2F20160519%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-D
ate=20160519T144733Z&X-Amz-Expires=300&X-Amz-Signature=8c72f755d3468e3c48fe3dba37dfc919ef38dc37d05e708966da6f40a2b
f174b&X-Amz-SignedHeaders=host&actor_id=0&response-content-disposition=attachment%3B%20filename%3DJoomla_3.5.1-Sta
ble-Full_Package.zip&response-content-type=application%2Foctet-stream [siguiente]
--2016-05-19 16:44:30--  https://github-cloud.s3.amazonaws.com/releases/2464908/c2a9ea34-fb82-11e5-9534-701271ce5d
26.zip?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAISTNZFOVBIJMK3TQ%2F20160519%2Fus-east-1%2Fs3%2Faws4_r
equest&X-Amz-Date=20160519T144733Z&X-Amz-Expires=300&X-Amz-Signature=8c72f755d3468e3c48fe3dba37dfc919ef38dc37d05e7
08966da6f40a2bf174b&X-Amz-SignedHeaders=host&actor_id=0&response-content-disposition=attachment%3B%20filename%3DJo
omla_3.5.1-Stable-Full_Package.zip&response-content-type=application%2Foctet-stream
Resolviendo github-cloud.s3.amazonaws.com (github-cloud.s3.amazonaws.com)... 54.231.80.56
Conectando con github-cloud.s3.amazonaws.com (github-cloud.s3.amazonaws.com)[54.231.80.56]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 11806513 (11M) [application/octet-stream]
Grabando a: “Joomla_3.5.1-Stable-Full_Package.zip”

100%[=====================================================================>] 11.806.513  3,28MB/s   en 4,3s

2016-05-19 16:44:35 (2,63 MB/s) - “Joomla_3.5.1-Stable-Full_Package.zip” guardado [11806513/11806513]
```

57. Joomla! se descomprimirá en un directorio "/var/www/html". Finalmente, deberemos eliminar el fichero comprimido de la instalación.

4.2. INSTALACIÓN UTILIZANDO GIT

58. Otra alternativa para la instalación de Joomla! consiste en utilizar la herramienta para desarrolladores GIT, un software de control de versiones que tiene como objetivo el control de los cambios que se puedan producir durante el desarrollo de un determinado software, permitiendo conocer con detalle el estado actual, personas que intervinieron, cambios en determinados archivos del código fuente, restaurar versiones antiguas etc.

59. Lo primero que haremos será configurar el repositorio por primera vez, y descargar la última versión disponible:

```
root@guiaJoomla:/var/www/html# git clone -b master https://github.com/joomla/joomla-cms.git
Clonar en «joomla-cms»...
remote: Counting objects: 496033, done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 496033 (delta 3), reused 0 (delta 0), pack-reused 496027
Receiving objects: 100% (496033/496033), 144.82 MiB | 3.05 MiB/s, done.
Resolving deltas: 100% (342134/342134), done.
Checking connectivity... hecho.
root@guiaJoomla:/var/www/html# ls
joomla-cms
root@guiaJoomla:/var/www/html#
```

60. Con la ejecución de este comando se creará un nuevo directorio llamado "joomla-cms", donde se encontrarán los ficheros de la última versión. Podremos mover este directorio al inicio del directorio o cambiar el nombre para adaptarlo a nuestras necesidades.

DIRECTORIO .GIT EN SERVIDORES PÚBLICOS

El comando anteriormente ejecutado descargará una copia de todos los ficheros, incluido un directorio oculto con el nombre de ".git".

Este directorio puede contener información sensible:

```
root@guiaJoomla:/var/www/html# cd joomla-cms/
root@guiaJoomla:/var/www/html/joomla-cms# ls -al .git
total 772
drwxr-xr-x  8 root root   4096 may 19 16:49 .
drwxr-xr-x 22 root root   4096 may 19 16:49 ..
drwxr-xr-x  2 root root   4096 may 19 16:48 branches
-rw-r--r--  1 root root    265 may 19 16:49 config
-rw-r--r--  1 root root     73 may 19 16:48 description
-rw-r--r--  1 root root     23 may 19 16:49 HEAD
drwxr-xr-x  2 root root   4096 may 19 16:48 hooks
-rw-r--r--  1 root root 729264 may 19 16:49 index
drwxr-xr-x  2 root root   4096 may 19 16:48 info
drwxr-xr-x  3 root root   4096 may 19 16:49 logs
drwxr-xr-x  4 root root   4096 may 19 16:48 objects
-rw-r--r--  1 root root   9737 may 19 16:49 packed-refs
drwxr-xr-x  5 root root   4096 may 19 16:49 refs
root@guiaJoomla:/var/www/html/joomla-cms#
```

Deberemos borrar esta información si se trata de un servicio expuesto al público, o añadir la siguiente regla al fichero ".htaccess":

```
root@guiaJoomla:/var/www/html/joomla-cms# cat .htaccess
RewriteEngine On
RewriteRule ^(.*/)?\.git/ - [F,L]
ErrorDocument 403 "Access Forbidden"
root@guiaJoomla:/var/www/html/joomla-cms#
```

Para comprobar si esta medida adicional está funcionando, accederemos a http://nuestroservidor/ruta/.git/. Si obtenemos un error 403, el directorio habrá quedado protegido.

4.3. INSTALACIÓN EN SISTEMAS CON SELINUX

61. Existen dos alternativas para la instalación de Joomla! en distribuciones basadas en Fedora (RHEL, CentOS etc.) con SELinux activo por defecto:

    - Desde el repositorio oficial de paquetes (generalmente con el comando “yum install joomla”)
    - Descargando el paquete y realizando la instalación por nuestra cuenta, como hemos visto en los puntos anteriores

62. SELinux[8] (Security-Enhanced Linux) es un módulo de seguridad para el kernel Linux que proporciona soporte para políticas de seguridad de control de acceso. Se trata de un conjunto de modificaciones del núcleo y herramientas de usuario que pueden ser agregadas a diversas distribuciones Linux. Su arquitectura se enfoca en separar las decisiones de las aplicaciones de seguridad de las políticas de seguridad mismas y racionalizar la cantidad de software encargado de las aplicaciones de seguridad.

[8] **Security-Enhanced Linux:** https://es.wikipedia.org/wiki/SELinux

63. Cuando la instalación de Joomla! se realiza desde el repositorio oficial de paquetes, se configurará de forma automática los permisos y se activarán las funciones necesarias para que funcione correctamente. En caso de realizar la instalación manual, debemos tener una serie de conceptos en cuenta para el correcto funcionamiento tanto de la instalación, como de la ejecución del framework.

64. Los ficheros de código de Joomla!, dentro del sistema de archivos, son irrelevantes para las funciones asignadas a SELinux. El demonio de Apache, "httpd", posee un grupo asignado dentro de la estructura de políticas de SELinux, por lo que los ficheros y directorios con los que necesite interactuar deben ser incluidos dentro del contexto de "httpd_*" con el siguiente comando:

 # chcon -R -t httpd_sys_content_t /ruta/de/joomla

65. Existen una serie de variables que controlarán la política de Apache, mediante las cuales SELinux controlará ciertos aspectos y funciones, y que deberán ser modificados de forma individual, como:

- El envío de correos electrónicos, para evitar que un posible sistema comprometido pueda actuar como generador/proxy de spam:

 # setsebool -P httpd_can_sendmail=1

- Realizar llamadas a través de *sockets* para conectarse a bases de datos:

 # setsebool -P httpd_can_network_connect_db=1

- Forzar a todos los procesos a tratar todos los contenidos de Apache con las mismas reglas:

 # setsebool -P httpd_unified=1

- Utilizar módulos externos de scripting, como "mod_php", "mod_python", "mod_perl" etc:

 # setsebool –P httpd_builtin_scripting=1

4.4. CONFIGURACIÓN DE LA BASE DE DATOS

66. Para el correcto funcionamiento de Joomla!, será necesario crear una base de datos nueva, con un usuario que tenga permisos para acceder a ella:

```
root@guiaJoomla:/var/www/html/joomla-cms# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 40
Server version: 5.5.44-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE guiajoomla CHARACTER SET utf8 COLLATE utf8_general_ci;
Query OK, 1 row affected (0.00 sec)

mysql> CREATE USER 'usuariojoomla'@'localhost' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER, CREATE TEMPORARY TABLES ON guiajoomla.* T
O 'usuariojoomla'@'localhost' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> EXIT
Bye
root@guiaJoomla:/var/www/html/joomla-cms#
```

67. El ejemplo anterior muestra el acceso como la cuenta de administrador por defecto de MySQL. Este nombre de usuario depende de los valores iniciales con los que la base de datos fuera configurada e instalada al inicio, por lo que en ciertos sistemas puede ser un valor diferente.

68. El resto de valores deberán ser elegidos por parte del usuario, como el nombre de la base de datos o el usuario que se va a utilizar para acceder, así como el campo “hostname”, que será generalmente "localhost".

69. Respecto a las contraseñas de acceso, se recomienda seguir una serie de pautas para la creación y establecimiento de contraseñas seguras:

    - Se deben utilizar al menos 12 caracteres para crear la clave.
    - Se recomienda utilizar en una misma contraseña dígitos, letras y caracteres especiales.
    - Es recomendable que las letras alternen aleatoriamente mayúsculas y minúsculas.
    - Utilizar signos de puntuación, así como diferentes símbolos.
    - En lo posible, utilizar una *passphrase* (una frase de contraseña o secuencia de palabras), debido a su sencillez para recordar y ser más segura, debido a su longitud.

70. Podremos utilizar el siguiente comando de línea de consola para generar una contraseña segura de 12 caracteres:

```
$ tr -dc [:graph:] < /dev/urandom | head -c 12 | xargs -0
X#}C")?q1&Xy
$
```

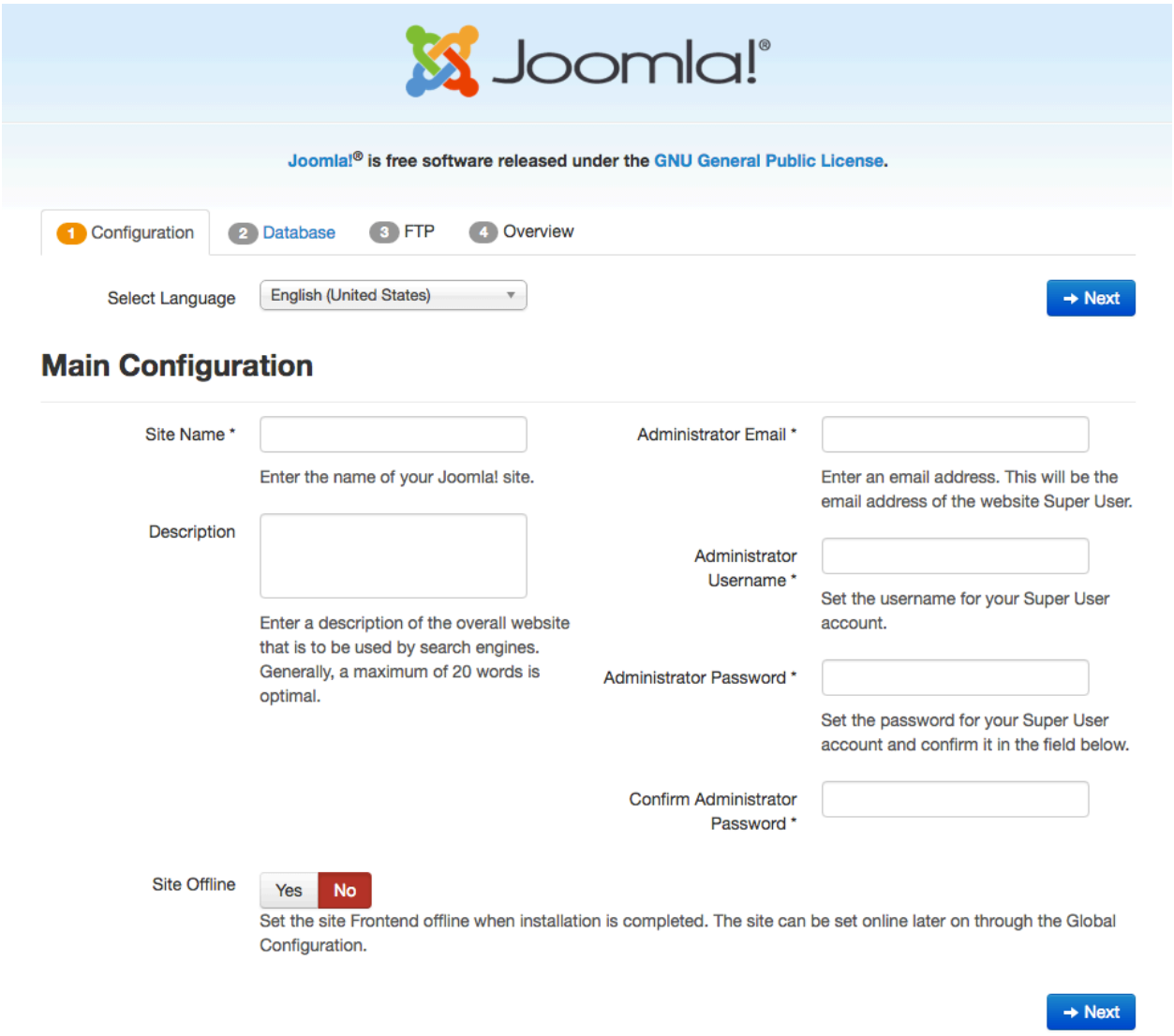
71. Además, en caso de necesidad, podremos modificar el valor de la longitud de la contraseña, incrementando el valor entero después del comando ‘head’:

```
$ tr -dc [:graph:] < /dev/urandom | head -c 20 | xargs -0
Y_W[PQ5Wghi,^J(%)1Re
$
```

4.5. FINALIZANDO LA INSTALACIÓN

72. Ahora que tenemos los ficheros necesarios para realizar la instalación, es necesario configurar Joomla! para indicarle ciertos valores, como la información de la base de datos.

73. Accedemos a la URL “http://direccion_servidor/directorio_joomla/”, y nos encontraremos una pantalla como la siguiente:



74. Deberemos completar el formulario con una serie de datos necesarios:

- **Select Language**: nos permite seleccionar la traducción adecuada en nuestra instalación. Durante el desarrollo de esta guía utilizaremos el idioma Español.
- **Nombre del Sitio**: nombre del nuevo sitio que se generará.
- **Descripción**: corresponde a la etiqueta *META DESCRIPTION* en HTML, que ayuda a indicar cuál es el contenido de nuestra página, de este modo los buscadores de internet recogen esta información y clasifican la página en una temática en función del contenido.
- **Dirección de Email de administrador**: la dirección de correo electrónica correspondiente al administrador del sitio. Será de gran utilidad para poder recuperar la contraseña en el caso de que sea necesario.
- **Nombre del usuario del administrador**: elegiremos un nombre de usuario para el perfil de administrador. A pesar de que Joomla! use "admin" como nombre por defecto, esto es una práctica no recomendable en cuestiones de seguridad, por lo que deberemos seleccionar otro nombre más complicado de averiguar por posibles atacantes externos.

- **Contraseña de administrador**: debe utilizarse una contraseña complicada, ya que el usuario administrador tiene control absoluto del sitio (*frontend* y *backend*). Se recomienda seguir una serie de pautas, ya vistas en puntos anteriores de la guía, para la creación y establecimiento de contraseñas seguras:
  - Se deben utilizar al menos 12 caracteres para crear la clave.
  - Se recomienda utilizar en una misma contraseña dígitos, letras y caracteres especiales.
  - Es recomendable que las letras alternen aleatoriamente mayúsculas y minúsculas.
  - Utilizar signos de puntuación, así como diferentes símbolos.
  - En lo posible, utilizar una *passphrase* (una frase de contraseña o secuencia de palabras), debido a su sencillez para recordar y ser más segura, debido a su longitud.
  - Podremos utilizar el siguiente comando de línea de consola para generar una contraseña segura de 12 caracteres:

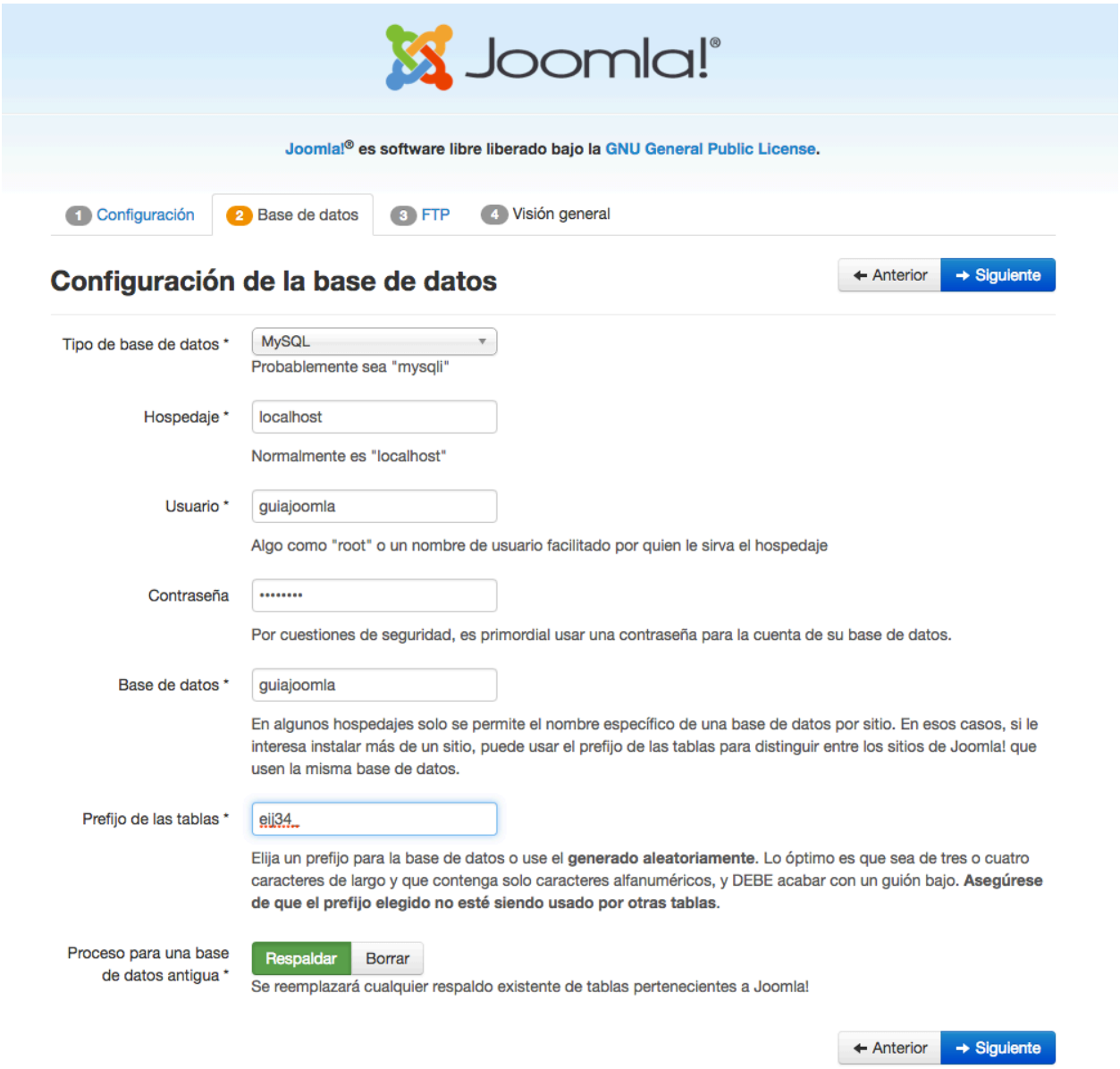
```
$ tr -dc [:graph:] < /dev/urandom | head -c 12 | xargs -0
X#}C")?q1&Xy
$
```

  - Además, en caso de necesidad, podremos modificar el valor de la longitud de la contraseña, incrementando el valor entero después del comando 'head':

```
$ tr -dc [:graph:] < /dev/urandom | head -c 20 | xargs -0
Y_W[PQ5Wghi,^J(%)1Re
$
```

- **Sitio fuera de línea**: indica si el sitio será accesible desde el exterior. En caso de pulsar sobre "Sí", cuando un usuario acceda a Joomla!, recibirá el mensaje de sitio fuera de línea.

75. El siguiente paso nos solicitará los datos para la generación de la base de datos de MySQL que hemos creado anteriormente:



76. Proporcionaremos los datos necesarios para la configuración de Joomla!:

- **Tipo de base de datos**: MySQLi (MySQL Improved) es el *driver* más común utilizado en PHP para proporcionar un interfaz de conexión y trabajo con bases de datos MySQL.
- **Hospedaje**: la máquina donde se encuentra instalada y configurada nuestra base de datos. En nuestro caso, será "localhost", pero se puede introducir la dirección de otra máquina.
- **Usuario**: nombre de usuario creado para la base de datos (en nuestro caso "usuariojoomla").
- **Contraseña**: la contraseña asociada al usuario de la base de datos.
- **Nombre de la base de datos**: el nombre seleccionado (en nuestro caso "guiajoomla").
- **Prefijo de tabla**: este prefijo se genera de forma aleatoria en el momento de la instalación, y permite añadir una seguridad extra a la instalación, ya que un atacante no tendría información del nombre completo de las tablas en caso de encontrar algún fallo de seguridad que lo requiera, como inyecciones de código SQL.

- **Proceso para una base de datos antigua**: nos permite respaldar una instalación anterior de Joomla!, o borrarla e iniciar el proceso desde cero.

77. El siguiente apartado permite la configuración del servicio FTP, que mantendremos desactivado:

78. Finalmente accederemos al paso final, donde podremos tanto instalar datos de ejemplo para usuarios principiantes, como revisar los datos de la configuración introducidos durante los pasos anteriores del proceso de instalación:

79. Una vez que hayamos comprobado que los datos introducidos son correctos, pulsaremos sobre el botón "Instalar". En caso de que sea necesario cambiar alguno, utilizaremos el botón "Anterior" para retroceder a pasos anteriores y modificar estos valores.

80. En el último paso, deberemos pulsar en "Eliminar carpeta de Instalación" para proceder al uso normal del sitio, debido a las comprobaciones de seguridad de Joomla!:

5. BORRADO DE FICHEROS INNECESARIOS

81. Al finalizar la instalación de Joomla!, se borrarán una serie de ficheros que pueden ser eliminados, evitando una posible fuga de información que pueda ayudar a un atacante externo a elaborar un ataque dirigido contra nuestra plataforma.

82. En el caso de haber descargado el fichero desde la web oficial del proyecto, podremos eliminar los siguientes ficheros:

    - *Joomla_3.5.1-Stable-Full_Package.zip (fichero de distribución de la última versión)*
    - *LICENSE.txt (fichero de información de licencia)*
    - *README.txt (fichero de información y ayuda)*
    - *web.config.txt (sólo para versiones de IIS)*
    - *Robots.txt.dist (utilizaremos el indicado en esta guía)*
    - *htaccess.txt (utilizaremos la configuración mostrada a lo largo de la guía)*

83. Si por el contrario hemos realizado la instalación utilizando Git como herramienta de descarga, procederemos a eliminar los siguientes archivos y directorios:

    - *composer.lock y composer.json (fichero de metadatos y dependencias de Composer)*

- *build.xml (fichero de metadatos del proyecto Joomla)*
- *web.config.txt (sólo para versiones de IIS)*
- *LICENSE.txt (fichero de información de licencia)*
- *README.txt y README.md (fichero de información y ayuda)*
- *tests/ (ficheros de pruebas de calidad de Joomla)*
- *phpunit.xml.dist y travisci-phpunit.xml (fichero de configuración pruebas)*
- *Robots.txt.dist (utilizaremos el indicado en esta guía)*
- *htaccess.txt (utilizaremos la configuración mostrada a lo largo de la guía)*

6. SEGURIDAD EN LA BASE DE DATOS

84. MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones, entre las que podemos destacar, además de las realizadas por usuarios o administradores, a empresas como Yahoo!, Alcatel-Lucent, Google, Nokia, Youtube y muchas otras.
85. Como la mayor parte de los servicios, MySQL vendrá pre configurado en nuestro sistema si hemos optado por su instalación como paquete, por lo que deberemos ajustar ciertos parámetros para adaptarnos a las mejores prácticas:

6.1. DESHABILITAR O RESTRINGIR EL ACCESO REMOTO

86. La primera consideración a tener en cuenta es determinar si el servicio de MySQL será accesible desde la red o solamente desde nuestro servidor (el caso más común). En caso de necesitar acceso remoto al servicio, debemos asegurarnos que existe un filtrado de qué direcciones estarán autorizadas, utilizando TCP Wrappers, Iptables o cualquier firewall software/hardware que esté disponible.
87. En caso contrario, evitaremos que MySQL abra un socket de red (sin impedir que las conexiones locales sigan funcionando) que permita esta conexiones, añadiendo el siguiente código a la parte "[mysqld]" del fichero "/etc/my.cnf" o "/etc/msqyl/my.ini":

    Skip-networking

88. Otra opción disponible será forzar a MySQL a escuchar sólo las conexiones realizadas desde "localhost", añadiendo la siguiente línea en el mismo apartado del fichero anterior:

    Bind-address=127.0.0.1

```
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 127.0.0.1
```

6.2. DESHABILITAR EL USO DE ‘LOCAL INFILE’

89. La siguiente tarea a realizar será deshabilitar el uso del comando “LOAD DATA LOCAL INFILE”, que evitará que un posible atacante sea capaz de leer ficheros locales utilizando diferentes tipos de ataques comunes, como un SQL Injection.

90. Además, en ciertas ocasiones, el comando “LOCAL INFILE” será utilizado para acceder a ficheros del sistema operativo, como “/etc/passwd”, intentando ejecutar una sentencia como la siguiente:

    mysql> LOAD DATA LOCAL INFILE '/etc/passwd' INTO TABLE table1

    O incluso de forma más sencilla:

    mysql> SELECT load_file("/etc/passwd")

91. Para deshabilitar el uso de este comando, añadiremos el siguiente código a la sección ‘[mysqld]’ de nuestro fichero de configuración:

    set-variable=local-infile=0

6.3. MODIFICAR EL NOMBRE DE USUARIO DEL ADMINISTRADOR

92. El nombre de usuario por defecto de la instalación de MySQL es "root", por lo que los atacantes intentarán acceder de diferentes formas a esta cuenta para lograr control completo de nuestro sistema.

93. Para endurecer esta tarea, modificaremos el nombre de la cuenta a otra más complicada de adivinar de forma remota, añadiendo una contraseña compleja. Para renombrar la cuenta de administrador del sistema, introduciremos el siguiente comando en la consola de MySQL:

    mysql> RENAME USER root TO nuevo_user;

94. El comando “RENAME USER” se introdujo por primera vez en la version 5.0.2, por lo que si nos encontramos con una versión anterior, será necesario realizar el proceso de forma diferente con los siguientes comandos:

    mysql> use mysql;
    mysql> update user set user="nuevo_usuario" where user="root";
    mysql> flush privileges;

6.4. ELIMINAR LA BASE DE DATOS DE PRUEBAS

95. MySQL, por norma general, almacena una base de datos, llamada ‘test’, que puede ser accedida por usuarios anónimos, por lo que podría recibir gran número de ataques. Para eliminar esta base de datos, utilizaremos el comando DROP de la siguiente forma desde la línea de comandos:

    mysql> drop database test;

96. También podemos realizar esta tarea desde la Shell utilizando ‘mysqladmin’:

    $ mysqladmin -u username -p drop test

6.5. ELIMINAR LA CUENTA ANÓNIMA Y OTRAS CUENTAS OBSOLETAS

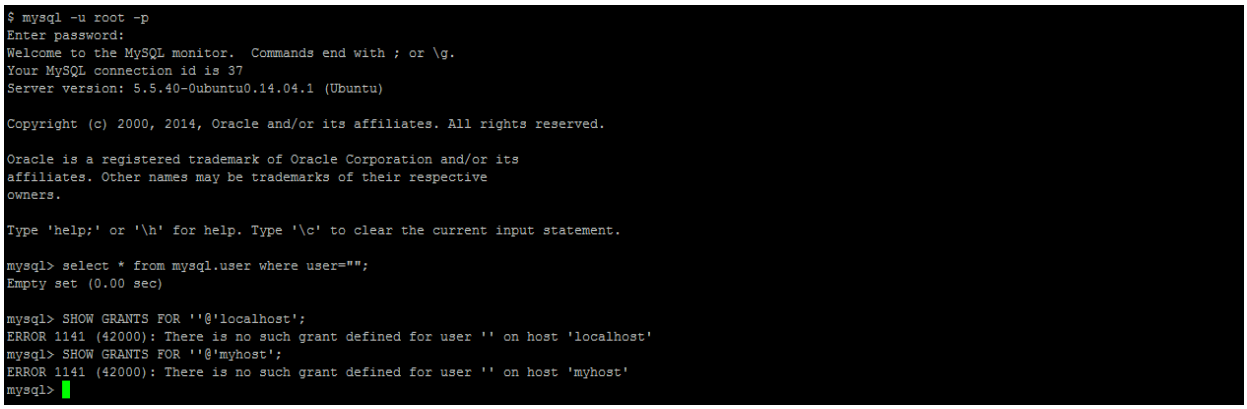
97. Por defecto, MySQL activará ciertas cuentas anónimas, que no necesitan contraseña, por lo que cualquier atacante podría conseguir acceso al sistema de forma sencilla. Para comprobar si están activadas en nuestro servidor, realizaremos la siguiente consulta:

    *mysql> select * from mysql.user where user="";*

98. Otra forma de realizar la consulta, sería de la siguiente forma:

    mysql> SHOW GRANTS FOR ''@'localhost';
    mysql> SHOW GRANTS FOR ''@'myhost';

99. En sistemas seguros, no deberíamos obtener resultado alguno:



100. Se deberán eliminar estas cuentas, ejecutaremos el siguiente comando:

    mysql> DROP USER "";

101. El comando DROP USER fue introducido en la versión 5.0. Si nos encontramos con una versión anterior de MySQL, podremos eliminar estas cuentas con las siguientes consultas:

    mysql> use mysql;
    mysql> DELETE FROM user WHERE user="";
    mysql> flush privileges;

6.6. REDUCIR LOS PRIVILEGIOS DEL SISTEMA

102. Se deberán reducir los privilegios con los que se ejecuta MySQL para evitar la exposición de nuestro sistema a gran cantidad de ataques.

103. En el caso de que estemos trabajando con versiones superiores a la 5.0, los permisos estarán ajustados y no será necesario realizar ninguna variación.

104. En otro caso, existen algunas premisas que deberemos cumplir para asegurar el servicio. Primero, comprobaremos que el directorio del servicio está asignado al grupo y usuario ‘mysql’:

    $ ls -l /var/lib/mysql

105. Comprobaremos, además, que el usuario "mysql" y "root" tienen acceso al directorio "/var/lib/mysql".

106. Finalmente, los binarios que se encuentran en el directorio "/usr/bin/", deberán tener como propietario al usuario ‘root’ o el correspondiente a la ejecución de "mysql". Ningún otro usuario debe tener permisos de escritura a estos ficheros:

    *$ ls -l /usr/bin/my**

107. La siguiente captura muestra un ejemplo de un sistema con los permisos correctos:

```
$ ls -l /usr/bin/my*
-rwxr-xr-x 1 root root 3301584 oct 11 03:13 /usr/bin/myisamchk
-rwxr-xr-x 1 root root 3190416 oct 11 03:13 /usr/bin/myisam_ftdump
-rwxr-xr-x 1 root root 3177104 oct 11 03:13 /usr/bin/myisamlog
-rwxr-xr-x 1 root root 3211440 oct 11 03:13 /usr/bin/myisampack
-rwxr-xr-x 1 root root 2903024 oct 11 03:13 /usr/bin/my_print_defaults
-rwxr-xr-x 1 root root 3469920 oct 11 03:13 /usr/bin/mysql
-rwxr-xr-x 1 root root  111539 oct 11 00:02 /usr/bin/mysqlaccess
-rwxr-xr-x 1 root root 3338016 oct 11 03:13 /usr/bin/mysqladmin
lrwxrwxrwx 1 root root      10 oct 11 03:12 /usr/bin/mysqlanalyze -> mysqlcheck
-rwxr-xr-x 1 root root 3456608 oct 11 03:13 /usr/bin/mysqlbinlog
-rwxr-xr-x 1 root root   11075 oct 11 00:02 /usr/bin/mysqlbug
-rwxr-xr-x 1 root root 3338976 oct 11 03:13 /usr/bin/mysqlcheck
-rwxr-xr-x 1 root root 3745984 oct 11 03:13 /usr/bin/mysql_client_test
-rwxr-xr-x 1 root root    4245 oct 11 00:02 /usr/bin/mysql_convert_table_format
-rwxr-xr-x 1 root root   23756 oct 11 00:02 /usr/bin/mysqld_multi
-rwxr-xr-x 1 root root   24885 oct 11 00:02 /usr/bin/mysqld_safe
-rwxr-xr-x 1 root root 3404960 oct 11 03:13 /usr/bin/mysqldump
-rwxr-xr-x 1 root root    7402 oct 11 00:02 /usr/bin/mysqldumpslow
-rwxr-xr-x 1 root root    3315 oct 11 00:02 /usr/bin/mysql_find_rows
-rwxr-xr-x 1 root root    1261 oct 11 00:02 /usr/bin/mysql_fix_extensions
-rwxr-xr-x 1 root root   34852 oct 11 00:02 /usr/bin/mysqlhotcopy
-rwxr-xr-x 1 root root 3334336 oct 11 03:13 /usr/bin/mysqlimport
-rwxr-xr-x 1 root root   14785 oct 11 00:02 /usr/bin/mysql_install_db
lrwxrwxrwx 1 root root      10 oct 11 03:12 /usr/bin/mysqloptimize -> mysqlcheck
-rwxr-xr-x 1 root root 2911352 oct 11 03:13 /usr/bin/mysql_plugin
lrwxrwxrwx 1 root root      10 oct 11 03:12 /usr/bin/mysqlrepair -> mysqlcheck
-rwxr-xr-x 1 root root   39016 oct 11 03:11 /usr/bin/mysqlreport
-rwxr-xr-x 1 root root    8198 oct 11 00:02 /usr/bin/mysql_secure_installation
-rwxr-xr-x 1 root root   17473 oct 11 00:02 /usr/bin/mysql_setpermission
-rwxr-xr-x 1 root root 3332992 oct 11 03:13 /usr/bin/mysqlshow
-rwxr-xr-x 1 root root 3352064 oct 11 03:13 /usr/bin/mysqlslap
-rwxr-xr-x 1 root root 3589872 oct 11 03:13 /usr/bin/mysqltest
-rwxr-xr-x 1 root root 2872944 oct 11 03:13 /usr/bin/mysql_tzinfo_to_sql
-rwxr-xr-x 1 root root 2979096 oct 11 03:13 /usr/bin/mysql_upgrade
-rwxr-xr-x 1 root root 2898224 oct 11 03:13 /usr/bin/mysql_waitpid
-rwxr-xr-x 1 root root    3888 oct 11 00:02 /usr/bin/mysql_zap
```

6.7. ACTIVAR LOS LOGS DEL SISTEMA

108. Se deberá activar la generación de logs añadiendo las siguientes líneas a la sección "[mysqld]" de nuestro fichero de configuración:

log =/var/log/mylogfile

```
[mysqld]
#
# * Basic Settings
#
user            = mysql
pid-file        = /var/run/mysqld/mysqld.pid
socket          = /var/run/mysqld/mysqld.sock
port            = 3306
basedir         = /usr
datadir         = /var/lib/mysql
tmpdir          = /tmp
lc-messages-dir = /usr/share/mysql
skip-external-locking
#
# Log
#
log=/var/log/fichero_de_logs_mysql
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 127.0.0.1
```

109. Además, comprobaremos que sólo el usuario "root" o "mysql" tiene acceso a estos ficheros (al menos permisos de escritura).

6.8. ELIMINAR HISTORIAL

110. Durante el proceso de instalación y generación de bases de datos, tablas etc. es probable que se utilice información sensible que debe de ser eliminada de tal forma que cualquier atacante que consiga acceso a nuestro sistema no tenga acceso a ella. Esta información puede ser almacenada dentro del historial, por lo que debe ser eliminada, así como todas las peticiones que almacena.

111. Esta tarea puede realizarse incluyendo el siguiente comando en nuestro fichero de inicio de sesión de la Shell (en el caso de utilizar Bash sería .bashrc):

cat /dev/null > ~/.mysql_history

6.9. LIMITAR LOS PERMISOS DE USUARIO

112. Para la operativa normal de Joomla!, como generar nuevos páginas, crear usuarios o instalar extensiones, el usuario utilizado para acceder a nuestra base de datos sólo debe tener permisos de lectura y escritura; es decir, debe poder ejecutar sentencias SELECT, INSERT, UPDATE y DELETE, por lo que podremos eliminar el resto de privilegios.

113. Para ello, introduciremos la siguiente consulta en nuestra consola MySQL:

mysql> REVOKE ALL PRIVILEGES ON guiajoomla. FROM usuariojoomla@'localhost';*

114. Después, asignaremos nuevos privilegios a nuestro usuario:

mysql> GRANT SELECT, INSERT, UPDATE, DELETE ON guiajoomla. TO usuariojoomla@'localhost';*

NOTA:
Algunos *plugins*, temas o actualizaciones de Joomla! pueden requerir realizar cambios estructurales en la base de datos, como añadir nuevas tablas etc. En ese caso, antes de realizar la instalación será necesario otorgar privilegios completos a nuestro usuario de Joomla! en MySQL. En este caso el procedimiento será el contrario:

mysql> REVOKE ALL PRIVILEGES ON guiajoomla. FROM usuariojoomla@'localhost';*

mysql> GRANT ALL ON usuariojoomla. TO usuariojoomla@'localhost' IDENTIFIED BY 'password';*

Se recomienda generar un *backup* del sistema antes de realizar estos cambios mayores.

7. CONFIGURACIÓN SEGURA DE PHP

7.1. INSTALACIÓN DE SUHOSIN

115. A pesar de que existen gran cantidad de recursos para mejorar la seguridad de PHP, muchos administradores no los implantan en sus sistemas. En este caso vamos a utilizar una herramienta de securización para PHP, llamada Suhosin, que generará una configuración más robusta y prevendrá gran cantidad de ataques, ya que está diseñada para proteger tanto a servidores como usuarios de vulnerabilidades, tanto conocidas como desconocidas, en las aplicaciones y Core de PHP.

Algunas de las características principales que proporciona Suhosin son:

- Protección contra buffer overflows (Canary y SafeUnlink Protection).
- Protección del core y extensiones de PHP contra vulnerabilidades format string.

- Protección contra el secuestro de sesiones, usando cifrado de forma transparente para la aplicación.
- Soporte para sha256(), sha256_file() y CRYPT_BLOWFISH en crypt().
- Cifrado de cookies.
- Baneo automático en la carga de ficheros binarios o ejecutables ELF.
- Configuración de respuestas frente a violaciones de política de seguridad: bloqueo, redirección a nivel de cliente, envío de código HTTP o ejecución de script PHP.
- etc.

116. Para la instalación en sistemas Ubuntu y Debian, ejecutaremos los siguientes comandos:

```
$ wget https://sektioneins.de/files/repository.asc
$ sudo apt-key add repository.asc
$ echo 'deb http://repo.suhosin.org/ ubuntu-trusty main' >> /etc/apt/sources.list
$ apt-get update
$ apt-get install php5-suhosin-extension
$ php5enmod suhosin
$ apache2ctl restart
```

117. En el caso de que deseemos una instalación manual, deberemos descargar la última versión estable (en el momento de la escritura de esta guía es la 0.9.38), y ejecutar los siguientes comandos:

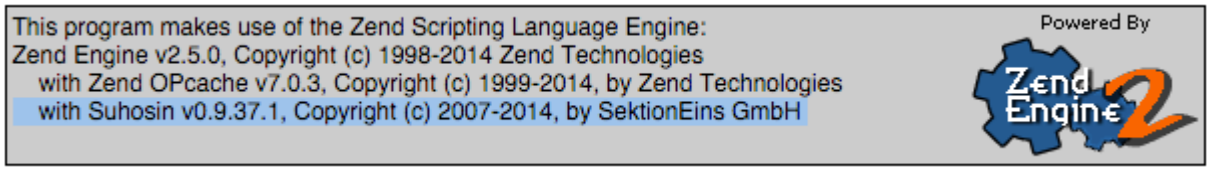
```
$ cd /tmp
$ wget https://download.suhosin.org/suhosin-0.9.38.tar.gz
$ tar xzvf suhosin-0.9.38.tar.gz
$ cd suhosin-0.9.38
$ phpize
$ make
$ make test
$ make install
```

118. Esto debería instalar la extensión 'suhosin.so' en el directorio correcto. Ahora sólo tendremos que añadir la siguiente directiva al fichero php.ini:

```
extension=suhosin.so
```

119. Se puede obtener más información detallada para realizar la instalación desde la página oficial del proyecto: https://suhosin.org/stories/install.html#installing-on-debian-and-ubuntus.

120. Finalmente, crearemos un fichero de prueba con la función "phpinfo()", tal como hicimos en el inicio de esta guía, para comprobar que la instalación ha sido satisfactoria, pudiendo visualizar:



121. Debemos recordar nuevamente que este fichero de prueba debe ser borrado inmediatamente después de haber realizado las comprobaciones necesarias.

7.2. CONFIGURACIÓN DE PHP

122. Existen gran cantidad de variables que pueden ser asignadas vía Apache, así como en el fichero de configuración de PHP. Lo más recomendable es mantener la configuración de seguridad de PHP centralizada en el fichero "php.ini".

123. Por ejemplo, implementar el *flag* "httpOnly" cuando se genera una cookie puede reducir el riesgo de un ataque XSS (Cross-Site Scripting), ya que quedará protegida de lectura y escritura por parte de scripts del lado del usuario (solo el servidor y el navegador tendrán acceso a la información guardada en los mismos). Además, implementaremos el *flag* "httpSecure", que impedirá que un posible atacante acceda o intercepte los datos de la sesión, ya que estos sólo viajarán únicamente por canales HTTPS seguros.

124. También añadiremos el dominio o dirección de a máquina, para inhabilitar cualquier sesión que no provenga de ésta (utilizando la cabecera *Referer*), y securizando otros valores como la entropía de la generación de ésta. Por último, seleccionaremos el algoritmo hash utilizado para generar el identificador ID de la sesión, utilizando en este caso el llamado "whirlpool".

125. Para realizar estas tareas, agregaremos el siguiente código a nuestro fichero "php.ini":

```
session.cookie_httponly = 1
session.cookie_secure = 1
session.use_only_cookies = 1
session.referer_check ="nuestro_dominio.es"
session.entropy_file = "/dev/urandom"
session.entropy_length = 16
session.hash_function = whirlpool
```

126. A continuación, procederemos a controlar y limitar ciertos recursos valiosos del sistema, como la memoria máxima utilizable, el tamaño máximo de fichero que podrá cargar un usuario o el tiempo de ejecución. En nuestro mismo fichero incluiremos las siguientes líneas:

```
memory_limit = 128M
max_input_time = 60
max_execution_time = 30
upload_max_filesize = 2M
```

127. Como medida adicional, también vamos a ocultar la versión de PHP que está siendo utilizada en el sistema, para reducir información que pudiera utilizar un atacante, añadiendo la siguiente línea:

```
expose_php = off
```

128. También deshabilitaremos funciones de PHP que permiten a los scripts referenciar a otras URLs:

```
allow_url_include = off
allow_url_fopen = off
```

129. Debemos ser conscientes de que existen una serie de funciones de PHP que se consideran peligrosas. Será necesario comprobar si nuestra instalación y módulos asociados a Joomla! no utilizan alguna de esta funciones (esto podemos realizarlo, por ejemplo, utilizando el comando "grep" desde la línea de comandos) ya que dependiendo de nuestra configuración, si las deshabilitamos, puede dejar de funcionar nuestro Joomla! :

disable_functions=exec,passthru,system,shell_exec,popen,proc_open,pcntl_exec,eval,assert,preg_replace,create_function,dl,phpinfo,posix_mkfifo,posix_getlogin,posix_ttyname,getenv,get_current_user,proc_get_status,get_cfg_var,disk_free_space,disk_total_space,diskfreespace,getcwd,getlastmod,getmygid,getmyinode,getmypid,getmyuid,extension_loaded,curl_exec,curl_multi_exec,fsockopen,parse_ini_file,show_source,symlink

130. Nos aseguraremos que la ruta donde se guardan las sesiones se encuentra fuera del directorio web y no pueden ser leídas por otros usuarios del sistema. También indicaremos a PHP que el directorio donde se realizarán las cargas del fichero se encontrará fuera del directorio web, con las siguientes líneas:

session.save_path = "/var/lib/php"
upload_tmp_dir = "/tmp"

131. En el caso en el que no necesitemos o hagamos uso de la subida de ficheros a nuestro sitio, será necesario deshabilitar esta opción:

file_uploads= off;

8. ADMINISTRACIÓN Y NAVEGACIÓN SOBRE SSL

132. Como hemos visto anteriormente, reforzar nuestro Joomla! es una tarea vital, ya que existen muchos factores externos que pueden amenazar su seguridad, a pesar de las diferentes medidas que vayamos incorporando.

133. Por este motivo, vamos a activar el protocolo HTTPS.

NOTA:
Para poder completar este paso, SSL deberá estar correctamente configurado en nuestro servidor. En caso contrario, seguiremos los pasos mostrados a continuación:

```
# Activamos SSL sobre Apache
$ a2enmod ssl
# Reiniciamos el servicio
$ service apache2 restart
# Creamos un nuevo directorio donde almacenar la clave y certificado del servidor
$ mkdir /etc/apache2/ssl
# Generamos un nuevo certificado con 365 días de validez, así como la clave asociada
$ openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/apache.key -out /etc/apache2/ssl/apache.crt
```

Ahora, utilizaremos el editor de texto que prefiramos para modificar el fichero de configuración de Apache SSL "/etc/apache2/sites-available/default-ssl.conf" y para adaptarlo a nuestras necesidades, incluyendo el nuevo certificado que hemos generado. Modificaremos los campos apropiados, como "ServerName", de la misma forma que las conexiones HTTP, incluyendo las siguientes líneas, si no se encontraran activas:

```
SSLEngine on
SSLCertificateFile /etc/apache2/ssl/apache.crt
SSLCertificateKeyFile /etc/apache2/ssl/apache.key
```

Finalmente, activaremos el nuevo sitio y recargaremos el servicio:

```
$ sudo a2ensite default-ssl
$ service apache2 reload
```

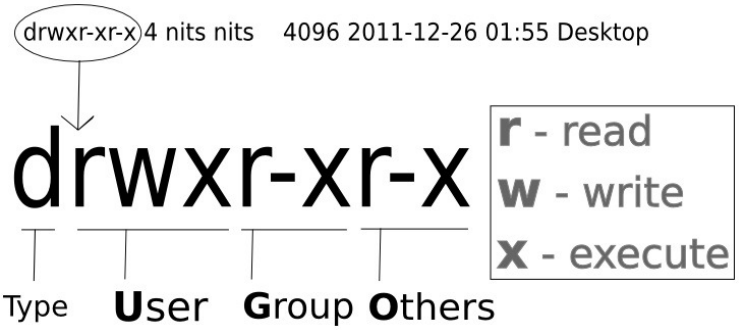
9. CONFIGURACIÓN DE PERMISOS

134. Los distintos ficheros y directorios poseen permisos que especifican quien y que puede leer, escribir, modificar y accederlos. Esto es importante, puesto que Joomla! puede necesitar acceso de escritura a ficheros en su directorio para activar ciertas funcionalidades.

135. El modo de permisos se realiza sumando los valores para el usuario, el grupo y para el resto. Este diagrama se muestra como:

- **R**ead (4): permitido leer ficheros/leer directorio
- **W**rite (2): permitido escribir/modificar ficheros/directorios
- e**X**ecute (1): permitido ejecutar el archivo o acceder al directorio

136. Estos permisos pueden ser representados en notación octal, que no es más que un valor en base 8. Siendo así, si el primer elemento de izquierda a derecha está activo (lectura) su valor es 4. Si el segundo elemento está activo (escritura) su valor es dos y si el tercer elemento está activo (ejecución), su valor es 1. Por ejemplo, si quisiéramos representar en notación octal permisos de lectura y ejecución, el valor sería 5. Para permisos de lectura y escritura el valor es 6 y para permisos de lectura, escritura y ejecución el valor es 7.[9]



137. Además, existen una serie de categorías de permisos:

- **U**suario o propietario (user, abreviada como u)
- **G**rupo (group, abreviada como g)
- **O**tros (others, abreviada como o)

138. Cada categoría de permisos se representa con tres caracteres. El primer conjunto de caracteres representa la categoría de usuario, el segundo conjunto representa la categoría de grupo y el tercer conjunto representa a la categoría otros. Cada uno de los tres caracteres representa los permisos de lectura, escritura y ejecución respectivamente.

[9] **FUENTE:** http://hipertextual.com/archivo/2014/05/permisos-linux/

139. Visto de esta forma, algunos ejemplos con su simbología completa serian:

Modo	Cadena perms.	Explicación
0477	-r--rwxrwx	el propietario solo puede leer (4), los otros y el grupo tienen acceso total rwx (7)
0677	-rw-rwxrwx	el propietario tiene sólamente lectura y escritura rw (6), los otros y el grupo tienen acceso total rwx (7)
0444	-r--r--r--	todos pueden sólamente leer (4)
0666	-rw-rw-rw-	todos pueden sólamente leer y escribir rw (6)
0400	-r--------	el propietario tiene solo lectura(4), los otros y el grupo no tienen permiso(0)
0600	-rw-------	el propietario tiene sólamente lectura y escritura rw, los otros y el grupo no tienen permiso(0)
0470	-r--rwx---	el propietario tiene sólamente lectura, el grupo tiene acceso total rwx, los otros no tienen permiso
0407	-r-----rwx	el propietario tiene sólamente lectura, los otros tienen acceso total rwx, el grupo no tiene permiso
0670	-rw-rwx---	el propietario tiene sólamente lectura y escritura rw, el grupo tiene acceso total rwx, los otros no tienen permiso
0607	-rw----rwx	el propietario tiene sólamente lectura y escritura rw, el grupo no tienen permiso y los otros tienen acceso total rwx

140. Los permisos de Joomla! serán diferentes de un host a otros host, así que esta guía solamente detalla principios genéricos. No puede cubrir todos los casos.

141. Típicamente, todos los ficheros deberían pertenecer a la cuenta de usuario de tu servidor web, y deberían ser escribibles por esa cuenta. En hosting compartido, los ficheros nunca deben pertenecer al proceso mismo del servicio web (generalmente será *www*, o *apache*, o *nobody*).

142. Cualquier fichero que necesite acceso de escritura desde Joomla! debería pertenecer al grupo de la cuenta de usuario utilizada por Joomla! (la cual puede que sea diferente de la cuenta del servidor). Por ejemplo, puede que se disponga de una cuenta específica de FTP que se utilice para cargar/descargar ficheros en el servidor, pero que ésta se encuentre utilizando un usuario separado, en un grupo de usuario aparte, como *apache*, *www-data* o *nobody*. Si Joomla! se ejecuta como la cuenta de FTP, esta cuenta necesita tener permisos de escritura, ser el propietario de los ficheros, o pertenecer a un grupo que tenga permiso de escritura. En este último caso, implicaría que los permisos están establecidos más permisivamente que por defecto:

- 775 en vez de 755 para directorios.
- 644 para ficheros.
- 750 para el directorio raíz (en nuestro caso /var/www/html/).
- En ningún momento se otorgarán permisos 777 bajo ningún concepto. Comprobaremos que no existe ningún fichero con estos permisos ejecutando el siguiente comando en la ruta raíz del servidor web:

```
root@guiaJoomla:/var/www/html/joomla-cms# find . -type f -perm 0777
root@guiaJoomla:/var/www/html/joomla-cms#
```

143. Son muchas las ocasiones en que pueden surgir graves problemas de seguridad por unos permisos inadecuados de estos ficheros y directorios, por lo que especificaremos unas reglas básicas:

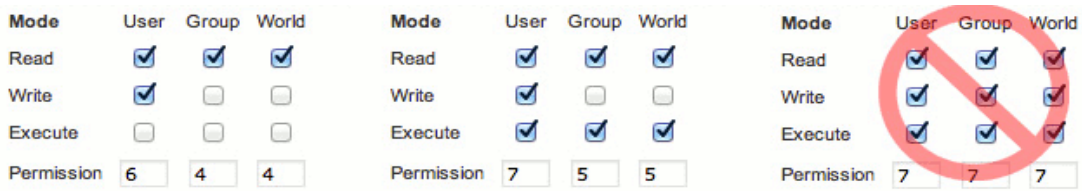
1. Los ficheros del núcleo de Joomla! deberían poseer permisos de escritura solamente por la cuenta de usuario utilizada.

2. Para garantizar la seguridad de los archivos, utilizaremos el siguiente comando:

   $ find /var/www/html/directorio_joomla -type d -exec chmod u=rwx,g=rx,o= '{}' \;

3. Para garantizar la seguridad de los directorios, es necesario mantenerlos en 755.

   $ find /var/www/html/directorio_joomla -type f -exec chmod u=rw,g=r,o= '{}' \;



4. Si por algún motivo debe modificarse un fichero o dar permisos de escritura a un directorio, deben ser 666 y comprobar si funciona. En caso de error, procederemos a asignarle los permisos 777 y una vez finalizado, le asignaremos el valor estándar de nuevo: 644 para ficheros y 755 para directorios.

144. Existen ciertos ficheros dentro de la instalación de Joomla! que deberían ser especialmente protegidos, como "configuration.php", que contiene la información relativa al sitio, base de datos, usuario y contraseña utilizada para conectarse al servidor MySQL etc, añadiendo la siguiente línea a nuestro fichero ".htaccess":

```
<FilesMatch "configuration.php">
Order allow,deny
Deny from all
</FilesMatch>
```

145. Además, podremos bloquear el acceso a otro tipo de ficheros, como logs, ficheros de configuración etc. con el siguiente bloque:

```
<FilesMatch "\.(htaccess|htpasswd|ini|phps|log|sh|conf)$">
Order allow,deny
Deny from all
</FilesMatch>
```

10. CREACIÓN DE UN FICHERO ‘ROBOTS.TXT’

146. Todos los buscadores, como Google, Bing, etc., para analizar los millones de webs existentes, utilizan programas que van continuamente escaneando todo tipo de información y añadiéndola a sus bases de datos (indexación). A estos programas se les denomina de diferentes formas, como robots, arañas, crawlers, etc. En el caso de Google, su robot principal se llama Googlebot. El fichero "robots.txt" te permite bloquear o permitir el acceso de los robots a las partes de tu web que a ti te interesen. Cuando el robot llegue a tu sitio web, comprobará si existe el fichero robots.txt, y si es ese el caso, seguirá las instrucciones que ahí se le indiquen para analizar la información.

147. Hay que tener en cuenta que aunque este es el funcionamiento general de la mayoría de motores de búsqueda (buscadores), esto no quiere decir que todos sigan estas reglas. Gente que se dedica a fines poco éticos como SPAM, etc., utiliza sus propios motores para captar información y darle un mal uso, saltándose todas estas reglas. Si tienes información privada en tu web, protégela con contraseñas o con cualquier otro sistema que evite que cualquiera pueda verla.

148. Por defecto, el fichero robots.txt de las instalaciones de Joomla! contiene:

```
# If the Joomla site is installed within a folder such as at
# e.g. www.example.com/joomla/ the robots.txt file MUST be
# moved to the site root at e.g. www.example.com/robots.txt
# AND the joomla folder name MUST be prefixed to the disallowed
# path, e.g. the Disallow rule for the /administrator/ folder
# MUST be changed to read Disallow: /joomla/administrator/
#
# For more information about the robots.txt standard, see:
# http://www.robotstxt.org/orig.html
#
# For syntax checking, see:
# http://tool.motoricerca.info/robots-checker.phtml

User-agent: *
Disallow: /administrator/
Disallow: /bin/
Disallow: /cache/
Disallow: /cli/
Disallow: /components/
Disallow: /includes/
Disallow: /installation/
Disallow: /language/
Disallow: /layouts/
Disallow: /libraries/
Disallow: /logs/
Disallow: /modules/
Disallow: /plugins/
Disallow: /tmp/
```

149. Cualquier directorio que deseemos excluir de este indexado de buscadores, deberá introducir con la opción Disallow, indicando la ruta completa al directorio deseado.

NOTA:
Existe una serie de datos adicionales que necesitaremos saber:
1. Se recomienda no incluir en un fichero robots.txt más de 200 líneas de Disallow.
2. Una vez que incluyamos el Disallow de una carpeta, si nos equivocamos y queremos volver a indexarla, pueden pasar hasta 3 meses desde que se agregue hasta que los buscadores vuelvan a mostrarla.
3. Se utilizan los Disallow cuando se quiera bloquear carpetas completas. Si es sólo una página suelta, las etiquetas meta son más eficaces.
4. Para ver qué ha indexado Google de nuestra web y compararlo con lo que le hemos dicho que no indexe en meta o robots.txt, introduciremos en la búsqueda “site:www.direccionweb.com“.

150. Existen también una serie de robots que pueden resultar perjudiciales para nuestro sitio web, por lo que limitaremos el acceso de éstos añadiendo las siguientes líneas al fichero:

```
User-agent: 123peoplebot
Disallow: /
User-agent: AACrawler
Disallow: /
User-agent: AhrefsBot
disallow: /
User-agent: Alexibot
Disallow: /
User-agent: Aqua_Products
User-agent: MJ12bot
Disallow: /
User-agent: MJ12bot/v1.4.3
Disallow: /
User-agent: MLBot
Disallow: /
User-agent: moget
Disallow: /
User-agent: moget/2.1
```

Disallow: /
User-agent: asterias
Disallow: /
User-agent: Atomic_Email_Hunter
Disallow: /
User-agent: b2w/0.1
Disallow: /
User-agent: BackDoorBot/1.0
Disallow: /
User-agent: BacklinkCrawler
Disallow: /
User-agent: Baiduspider
Disallow: /
User-agent: Baiduspider+
Disallow: /
User-agent: Baiduspider-cpro
Disallow: /
User-agent: Baiduspider-favo
Disallow: /
User-agent: Baiduspider-image
Disallow: /
User-agent: Baiduspider-mobile
Disallow: /
User-agent: Baiduspider-news
Disallow: /
User-agent: Baiduspider-sfkr
Disallow: /
User-agent: Baiduspider-video
Disallow: /
User-agent: BlowFish/1.0
Disallow: /
User-agent: Bookmark search tool
Disallow: /
User-agent: BotALot
Disallow: /
User-agent: BotRightHere
Disallow: /
User-agent: BuiltBotTough
Disallow: /
User-agent: Bullseye/1.0
Disallow: /

Disallow: /
User-agent: NetAnts
Disallow: /
User-agent: netEstate NE Crawler
Disallow: /
User-agent: NICErsPRO
Disallow: /
User-agent: Offline Explorer
Disallow: /
User-agent: OnetSzukaj
Disallow: /
User-agent: Openbot
Disallow: /
User-agent: Openfind
Disallow: /
User-agent: Openfind data gatherer
Disallow: /
User-agent: Oracle Ultra Search
Disallow: /
User-agent: PerMan
Disallow: /
User-agent: ProPowerBot/2.14
Disallow: /
User-agent: ProWebWalker
Disallow: /
User-agent: proximic
Disallow: /
User-agent: psbot
Disallow: /
User-agent: Python-urllib
Disallow: /
User-agent: QueryN Metasearch
Disallow: /
User-agent: Radiation Retriever 1.1
Disallow: /
User-agent: RepoMonkey
Disallow: /
User-agent: RepoMonkey Bait & Tackle/v1.01
Disallow: /
User-agent: RMA
Disallow: /

User-agent: BunnySlippers
Disallow: /
User-agent: Check Url Bot
Disallow: /
User-agent: CheeseBot
Disallow: /
User-agent: CherryPicker
Disallow: /
User-agent: CherryPickerElite/1.0
Disallow: /
User-agent: CherryPickerSE/1.0
Disallow: /
User-agent: Cityreview
Disallow: /
User-agent: Copernic
Disallow: /
User-agent: CopyRightCheck
Disallow: /
User-agent: cosmos
Disallow: /
User-agent: Crescent
Disallow: /
User-agent: CyberPatrol SiteCat Webbot
Disallow: /
User-agent: DittoSpyder
Disallow: /
User-agent: DomainWatcher
Disallow: /
User-agent: EmailCollector
Disallow: /
User-agent: EmailSiphon
Disallow: /
User-agent: EmailWolf
Disallow: /
User-agent: EroCrawler
Disallow: /
User-agent: Exabot
Disallow: /
User-agent: exooba
Disallow: /
User-agent: ExtractorPro

User-agent: ScoutJet
Disallow: /
User-agent: SeznamBot/3.0
Disallow: /
User-agent: searchpreview
Disallow: /
User-agent: spbot/3.1
Disallow: /
User-agent: SiteSnagger
Disallow: /
User-agent: SpankBot
Disallow: /
User-agent: spanner
Disallow: /
User-agent: Sosospider/2.0
Disallow: /
User-agent: suzuran
Disallow: /
User-agent: Szukacz/1.4
Disallow: /
User-agent: Teleport
Disallow: /
User-agent: TeleportPro
Disallow: /
User-agent: Telesoft
Disallow: /
User-agent: The Intraformant
Disallow: /
User-agent: TheNomad
Disallow: /
User-agent: TightTwatBot
Disallow: /
User-agent: toCrawl/UrlDispatcher
Disallow: /
User-agent: Toplistbot
Disallow: /
User-agent: True_Robot
Disallow: /
User-agent: True_Robot/1.0
Disallow: /
User-agent: turingos

Disallow: /
User-agent: Ezooms/1.0
Disallow: /
User-agent: FairAd Client
Disallow: /
User-agent: Fasterfox
Disallow: /
User-agent: Flaming AttackBot
Disallow: /
User-agent: Foobot
Disallow: /
User-agent: Gaisbot
Disallow: /
User-agent: GetRight/4.2
Disallow: /
User-agent: gigabot
Disallow: /
User-agent: GrapeshotCrawler/2.0
Disallow: /
User-agent: grapeFX/0.9
Disallow: /
User-agent: Harvest/1.5
Disallow: /
User-agent: heritrix
Disallow: /
User-agent: hloader
Disallow: /
User-agent: httplib
Disallow: /
User-agent: HTTrack
Disallow: /
User-agent: HTTrack 3.0
Disallow: /
User-agent: humanlinks
Disallow: /
User-agent: ia_archiver
Disallow: /
User-agent: ia_archiver/1.6
Disallow: /
User-agent: ICCrawler - ICjobs
Disallow: /
Disallow: /
User-agent: TurnitinBot
Disallow: /
User-agent: TurnitinBot/1.5
Disallow: /
User-agent: twiceler
Disallow: /
User-agent: twiceler.
Disallow: /
User-agent: URL Control
Disallow: /
User-agent: URLSpion
Disallow: /
User-agent: URL_Spider_Pro
Disallow: /
User-agent: URLy Warning
Disallow: /
User-agent: VCI
Disallow: /
User-agent: VCI WebViewer VCI WebViewer Win32
Disallow: /
User-agent: Web Image Collector
Disallow: /
User-agent: WebAlta Crawler
Disallow: /
User-agent: WebAuto
Disallow: /
User-agent: WebBandit
Disallow: /
User-agent: WebBandit/3.50
Disallow: /
User-agent: webbericht.com ?
Disallow: /
User-agent: WebCapture 2.0
Disallow: /
User-agent: WebCopier
Disallow: /
User-agent: WebCopier v.2.2
Disallow: /
User-agent: WebCopier v3.2a
Disallow: /

User-agent: Infohelfer/1.3.0
Disallow: /
User-agent: InfoNaviRobot
Disallow: /
User-agent: Iron33/1.0.2
Disallow: /
User-agent: JennyBot
Disallow: /
User-agent: jobs.de-Robot
Disallow: /
User-agent: Kenjin Spider
Disallow: /
User-agent: Keyword Density/0.9
Disallow: /
User-agent: larbin
Disallow: /
User-agent: LexiBot
Disallow: /
User-agent: libWeb/clsHTTP
Disallow: /
User-agent: LinkextractorPro
Disallow: /
User-agent: LinkScan/8.1a Unix
Disallow: /
User-agent: LinkWalker
Disallow: /
User-agent: LNSpiderguy
Disallow: /
User-agent: lwp-trivial
Disallow: /
User-agent: lwp-trivial/1.34
Disallow: /
User-agent: Mail.Ru
Disallow: /
User-agent: Mail.RU_Bot/2.0
Disallow: /
User-agent: magpie-crawler/1.1
Disallow: /
User-agent: Mata Hari
Disallow: /
User-agent: MIIxpc
User-agent: WebDataCentreBot
Disallow: /
User-agent: WebEnhancer
Disallow: /
User-agent: WebSauger
Disallow: /
User-agent: Website Quester
Disallow: /
User-agent: Webster Pro
Disallow: /
User-agent: WebStripper
Disallow: /
User-Agent: Webwiki
Disallow: /
User-agent: WebZip
Disallow: /
User-agent: WebZip/4.0
Disallow: /
User-agent: WebZIP/4.21
Disallow: /
User-agent: WebZIP/5.0
Disallow: /
User-agent: Wget
Disallow: /
User-agent: wget
Disallow: /
User-agent: Wget/1.5.3
Disallow: /
User-agent: Wget/1.6
Disallow: /
User-agent: WWW-Collector-E
Disallow: /
User-agent: Xenu´s
Disallow: /
User-agent: Xenu´s Link Sleuth 1.1c
Disallow: /
User-agent: Yandex
Disallow: /
User-agent: YandexBot/3.0
Disallow: /
User-agent: Yeti

Disallow: /

User-agent: MIIxpc/4.2

Disallow: /

User-agent: Mister PiX

Disallow: /

Disallow: /

User-agent: Zeus

Disallow: /

User-agent: Zeus 32297 Webster Pro V2.9 Win32

Disallow: /

11. RESTRICCIONES DE ACCESO AL BACKEND

151. Un módulo de Apache necesario para securizar los accesos al backend será "mod_authz_host", ya que es capaz de restringir el acceso a páginas como "/index.php/component/users" o "/administrator", y apenas necesita configuración.

152. Por ejemplo, para evitar que cualquier usuario pueda editar un nodo, salvo las direcciones IP 192.168.1.1 y 192.168.1.2, incorporamos el siguiente código en el fichero de configuración de Apache:

```
<Location - "/administrator/">
    Order Deny, Allow
    Deny from all
    Allow from 192.168.1.1 192.168.1.2
</Location>
```

153. Este tipo de restricción de acceso también puede realizarse utilizando "mod_rewrite", escribiendo el siguiente código dentro del fichero ".htaccess" o el fichero de configuración de Apache. En este caso sólo permitiremos el acceso a "administrator/" a las direcciones IP anteriormente indicadas:

```
<IfModule mod_rewrite.c>
    RewriteEngine on
    # Allow only internal access to admin
    RewriteCond %{REMOTE_ADDR} !^(192\.168\.1\.1|192\.168\.1\.2)$
    RewriteRule ^administrator/.* - [F]
</IfModule>
```

154. Finalmente, podremos incluir una autenticación adicional en el directorio. El primer paso que debemos realizar será crear un fichero ".htpasswd", conteniendo los usuarios/contraseñas que tienen permiso para acceder a nuestra área protegida. Este fichero puede ser generado de varias formas:

- **Local**: utilizaremos la herramienta htpasswd, incluído en la mayoría de distribuciones que tienen Apache instalado. Para crear el fichero, utilizaremos el siguiente comando:

  $ htpasswd –B –C 15 –c .htpasswd Nombre_de_Usuario

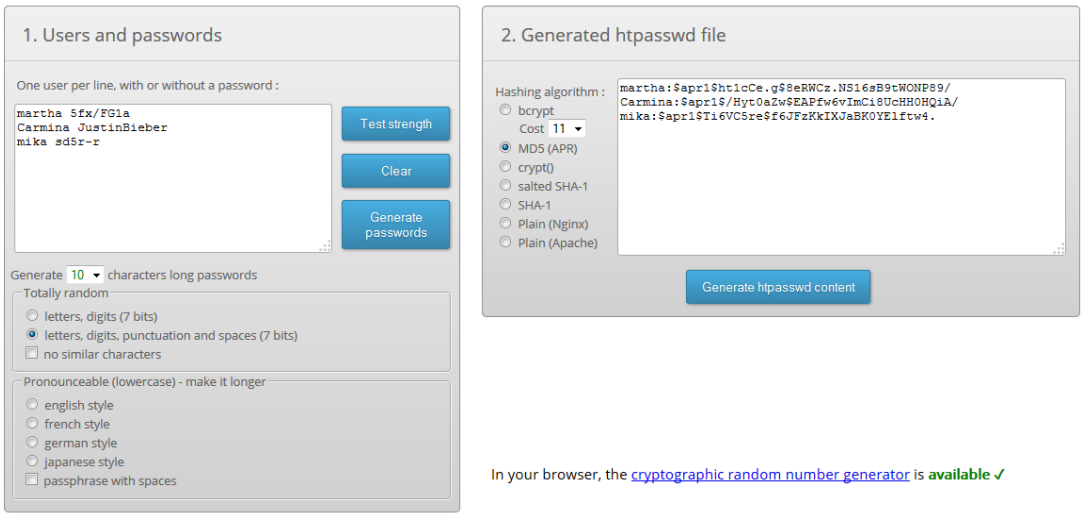
  Si deseamos crear un nuevo fichero, autorizando al usuario "adminhttpauth", introduciremos el siguiente comando y su contraseña:

```
$ htpasswd -B -C 15 -c .htpasswd adminhttpauth
New password:
Re-type new password:
Adding password for user adminhttpauth
$
```

El parámetro –B forzará a htpasswd a utilizar un cifrado más seguro para la contraseña (bcrypt), en lugar del utilizado por defecto (md5). Además, incrementaremos el tiempo de computación utilizado para generar la nueva contraseña (por defecto es 4, pero puede alcanzar valores de hasta 31).

En caso de que no queramos crear el fichero, tan solamente añadir un nuevo usuario, prescindiremos del parámetro "-c".

- **Online**: también es posible generar este fichero utilizando herramientas online. Existen gran cantidad de páginas que ofrecen este servicio, aunque nos centraremos en http://aspirine.org/htpasswd_en.html, debido a la gran cantidad de opciones que permite. Tan solo deberemos introducir el nombre de usuario y la contraseña, separada por un espacio, en el cuadro de la izquierda, y el resultado saldrá tras pulsar "Generate htpasswd content" en el cuadro de la derecha.



NOTA:
No es aconsejable realizar esta serie de tareas en páginas desconocidas, ya que podríamos estar exponiendo las credenciales de acceso de nuestro servicio a posibles atacantes.

Esta opción sólo debe utilizarse en caso de no disponer en ningún acceso a la herramienta htpasswd dentro de nuestro servidor local o cualquier otra máquina de la que se disponga acceso.

155. Una vez tenemos el fichero ".htpasswd" generado, lo subiremos al servidor, pero fuera de la ruta del directorio que utiliza nuestro servicio web. Si, por ejemplo, Apache está configurado para utilizar el directorio por defecto "/var/www/html/", podremos utilizar el directorio "/var/www" para almacenar este fichero, o cualquiera dentro de nuestro sistema que no pueda ser accedido de forma pública o por otro usuario.

NOTA:
El fichero .htpasswd nunca deberá estar alojado en un directorio expuesto de forma pública por el servidor Web.

156. Agregaremos la siguiente información a nuestro fichero ".htaccess", o crearemos uno nuevo para este propósito:

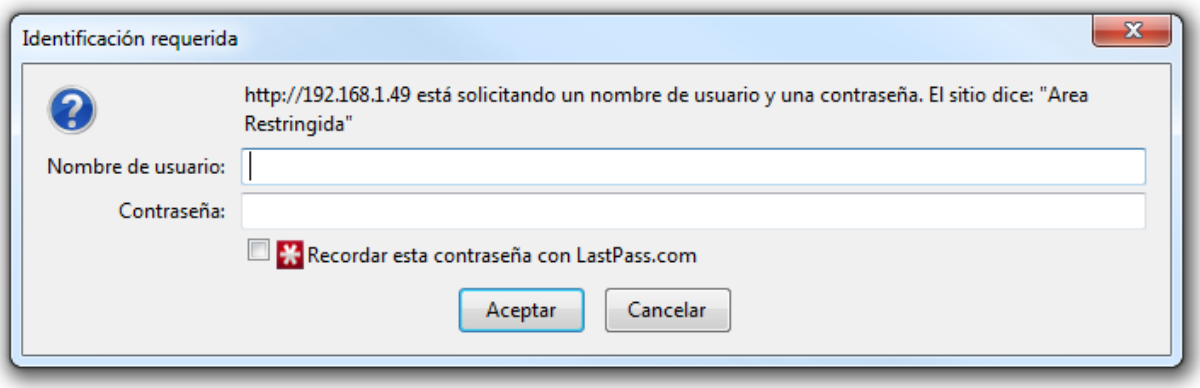
```
# enable basic authentication
AuthType Basic
# this text is displayed in the login dialog
AuthName “Area Restringida”
# The absolute path of the Apache htpasswd file. You should edit this
AuthUserFile /ruta/de/.htpasswd
```

```
# Allows any user in the .htpasswd file to access the directory
require valid-user
```

157. Una vez hayamos creado nuestros ficheros ".htpasswd" y ".htaccess", debemos configurar los permisos correctos para su acceso:

```
# Permisos rw-r--r-- para htaccess
# chmod 644 /ruta/a/.htaccess
# Permisos rw-r----- para .htpasswd
# chmod 640 /ruta/a/.htpasswd
```

158. Ahora trataremos de nuevo de acceder al panel de administración de nuestro Joomla!. A diferencia de accesos anteriores, encontraremos un cuadro de autenticación solicitando unas credenciales antes de poder acceder a la zona de login:



NOTA:
La autenticación HTTP es sencilla de implementar, pero las contraseñas se enviarán por la red, codificadas en Base64 y en texto claro.

Por este motivo, es siempre aconsejable utilizar HTTPS.

159. Un método más seguro para acceder al backend, es la utilización de un certificado digital de acceso.

12. INCREMENTANDO LA SEGURIDAD CON HTACCESS

160. El propio paquete de instalación de Joomla! contiene un fichero "htaccess.txt" de ejemplo para mejorar la seguridad de nuestro sitio, con las siguientes directrices:

```
## No directory listings
IndexIgnore *

## Can be commented out if causes errors, see notes above.
Options +FollowSymlinks
Options -Indexes

## Mod_rewrite in use.

RewriteEngine On

## Begin - Rewrite rules to block out some common exploits.
# If you experience problems on your site block out the operations listed below
# This attempts to block the most common type of exploit `attempts` to Joomla!
#
# Block out any script trying to base64_encode data within the URL.
```

```
RewriteCond %{QUERY_STRING} base64_encode[^()]*\([^)]*\) [OR]
# Block out any script that includes a <script> tag in URL.
RewriteCond %{QUERY_STRING} (<|%3C)([^s]*s)+cript.*(>|%3E) [NC,OR]
# Block out any script trying to set a PHP GLOBALS variable via URL.
RewriteCond %{QUERY_STRING} GLOBALS(=|\[|\%[0-9A-Z]{0,2}) [OR]
# Block out any script trying to modify a _REQUEST variable via URL.
RewriteCond %{QUERY_STRING} _REQUEST(=|\[|\%[0-9A-Z]{0,2})
# Return 403 Forbidden header and show the content of the root homepage
RewriteRule .* index.php [F]
#
## End - Rewrite rules to block out some common exploits.

## Begin - Custom redirects
#
# If you need to redirect some pages, or set a canonical non-www to
# www redirect (or vice versa), place that code here. Ensure those
# redirects use the correct RewriteRule syntax and the [R=301,L] flags.
#
## End - Custom redirects

##
# Uncomment following line if your webserver's URL
# is not directly related to physical file paths.
# Update Your Joomla! Directory (just / for root).
##

# RewriteBase /

## Begin - Joomla! core SEF Section.
#
RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}]
#
# If the requested path and file is not /index.php and the request
# has not already been internally rewritten to the index.php script
RewriteCond %{REQUEST_URI} !^/index\.php
# and the requested path and file doesn't directly match a physical file
RewriteCond %{REQUEST_FILENAME} !-f
# and the requested path and file doesn't directly match a physical folder
RewriteCond %{REQUEST_FILENAME} !-d
# internally rewrite the request to the index.php script
RewriteRule .* index.php [L]
#
## End - Joomla! core SEF Section.
```

161. La primera directiva que añadiremos, será ocultar la información del servidor web y versión que actualmente se está ejecutando sobre nuestra máquina. Si lanzamos una petición a un fichero no existente, obtendremos lo siguiente:

```
<address>Apache/2.4.7 (Ubuntu) Server at 127.0.0.1 Port 80</address>
```

162. Para ocultar esta información, es necesario añadir al fichero:

```
ServerSignature Off
```

163. También bloquearemos peticiones HTTP inseguras, intentos de ataque en la cabecera Referer o en las Cookie, bloquearemos escaneos automatizados de herramientas conocidas y algunas variantes de ataques de Inyección SQL:

```
RewriteCond %{REQUEST_METHOD} ^(HEAD|TRACE|DELETE|TRACK) [NC,OR]
RewriteCond %{THE_REQUEST} (\\r|\\n|%0A|%0D) [NC,OR]

RewriteCond %{HTTP_REFERER} (<|>|'|%0A|%0D|%27|%3C|%3E|%00) [NC,OR]
RewriteCond %{HTTP_COOKIE} (<|>|'|%0A|%0D|%27|%3C|%3E|%00) [NC,OR]
RewriteCond %{REQUEST_URI} ^/(,|;|:|<|>|">|"<|/|\\\.\.\\).{0,9999} [NC,OR]

RewriteCond %{HTTP_USER_AGENT} ^$ [OR]
RewriteCond %{HTTP_USER_AGENT} ^(java|curl|wget) [NC,OR]
RewriteCond %{HTTP_USER_AGENT}
(winhttp|HTTrack|clshttp|archiver|loader|email|harvest|extract|grab|miner) [NC,OR]
RewriteCond %{HTTP_USER_AGENT} (libwww-
perl|curl|wget|python|nikto|sqlmap|nessus|fimap|webshag|dirbuster|^w3af.sourceforget.net|Openvas|j
brofuzz|scan) [NC,OR]
RewriteCond %{HTTP:Acunetix-Product} ^WVS
RewriteCond %{HTTP_USER_AGENT} (<|>|'|%0A|%0D|%27|%3C|%3E|%00) [NC,OR]

RewriteCond %{QUERY_STRING}
(;|<|>|'|"|\)|%0A|%0D|%22|%27|%3C|%3E|%00).*(/\*|union|select|insert|cast|set|declare|drop|upd
ate|md5|benchmark) [NC,OR]
RewriteCond %{QUERY_STRING} \.\./\.\. [OR]
RewriteCond %{QUERY_STRING} (localhost|loopback|127\.0\.0\.1) [NC,OR]
RewriteCond %{QUERY_STRING} \.[a-z0-9] [NC,OR]
RewriteCond %{QUERY_STRING} (<|>|'|%0A|%0D|%27|%3C|%3E|%00) [NC]
RewriteRule .* index.php [F]
```

164. Bloquearemos peticiones sospechosas de ataques ya conocidos en Joomla![10]:

```
# If the request query string contains /proc/self/environ (by SigSiu.net)
RewriteCond %{QUERY_STRING} proc/self/environ [OR]
# Legacy variable injection (these attacks wouldn't work w/out Joomla! 1.5's Legacy Mode plugin)
RewriteCond %{QUERY_STRING} mosConfig_[a-zA-Z_]{1,21}(=|\%3D) [OR]
# Block out any script trying to base64_encode/base64_decode data to send via URL
RewriteCond %{QUERY_STRING} base64_(en|de)code\(.*\) [OR]
## IMPORTANT: If the above line throws an HTTP 500 error, replace it with these 2 lines:
# RewriteCond %{QUERY_STRING} base64_encode\(.*\) [OR]
# RewriteCond %{QUERY_STRING} base64_decode\(.*\) [OR]
# Block out any script that includes a <script> tag in URL
RewriteCond %{QUERY_STRING} (<|%3C).*script.*(>|%3E) [NC,OR]
# Block out any script trying to set a PHP GLOBALS variable via URL
RewriteCond %{QUERY_STRING} GLOBALS(=|\[|\%[0-9A-Z]{0,2}) [OR]
# Block out any script trying to modify a _REQUEST variable via URL
RewriteCond %{QUERY_STRING} _REQUEST(=|\[|\%[0-9A-Z]{0,2})
# Return a 403 Forbidden header and show the content of the root homepage
RewriteRule .* index.php [F]
```

165. Las siguientes líneas protegerán contra posibles ataques de Remote File Inclusion[11] (RFI):

```
########## Begin - File injection protection, by SigSiu.net
RewriteCond %{REQUEST_METHOD} GET
RewriteCond %{QUERY_STRING} [a-zA-Z0-9_]=http:// [OR]
RewriteCond %{QUERY_STRING} [a-zA-Z0-9_]=(\.\.//?)+ [OR]
RewriteCond %{QUERY_STRING} [a-zA-Z0-9_]=/([a-z0-9_.]//?)+ [NC]
RewriteRule .* - [F]
```

[10] **MASTER HTACCESS:** https://github.com/nikosdion/master-htaccess/blob/master/htaccess.txt

[11] **REMOTE FILE INCLUSION:** https://es.wikipedia.org/wiki/Remote_File_Inclusion

########## End - File injection protection

166. Otra opción que deberemos añadir es impedir a posibles atacantes que localicen el gestor de contenidos que estamos ejecutando, bloqueando ciertas peticiones a componentes concretos que pueden facilitar estar tarea:

Disallow visual fingerprinting of Joomla! sites (module position dump)
Initial idea by Brian Teeman and Ken Crowder, see:
http://www.slideshare.net/brianteeman/hidden-joomla-secrets
Improved by @nikosdion to work more efficiently and handle template
and tmpl query parameters
RewriteCond %{QUERY_STRING} (^|&)tmpl=(component|system) [NC]
RewriteRule . - [L]*
RewriteCond %{QUERY_STRING} (^|&)t(p|emplate)= [NC]
RewriteRule . - [F]*

167. PHP, igual que otra gran cantidad de programas, contiene los denominados Huevos de Pascua (Easter Eggs), que son pequeñas bromas incluidas dentro del código de la distribución. A pesar de no tener un impacto directo en la seguridad del sistema, estos Easter Eggs pueden ayudar a un atacante a conocer nuestra versión de PHP y buscar vulnerabilidades que puedan afectarlo, por lo que bloquearemos estas peticiones:

Disallow PHP Easter Eggs (can be used in fingerprinting attacks to determine
your PHP version). See http://www.0php.com/php_easter_egg.php and
http://osvdb.org/12184 for more information
RewriteCond %{QUERY_STRING} \=PHP[a-f0-9]{8}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{12} [NC]
RewriteRule . - [F]*

13. ELIMINANDO LA EXPOSICIÓN DEL CMS Y VERSIÓN

168. Dentro de la estructura HTML generada por Joomla!, existen una serie de etiquetas *META* que ayudan a los buscadores a obtener información de la descripción de la página, palabras claves con la que relacionarla, o el lenguaje. Entre ellas se encuentra la etiqueta *Generator*, que indicar el gestor de contenidos con el que se ha creado la página. Dentro de cada página, si accedemos al código, podremos encontrarla:

```
<meta name="generator" content="Joomla! - Open Source Content Management" />
<title>Home</title>
```

169. También figura cuando solicitas visualizar el *feed* RSS:

```
<?xml version="1.0" encoding="utf-8"?>
<!-- generator="Joomla! - Open Source Content Management" -->
<rss version="2.0" xmlns:atom="http://www.w3.org/2005/Atom">
        <channel>
                <title>Guia Joomla!</title>
                <description><![CDATA[]]></description>
                <link>http://192.168.1.39/index.php?id=14</link>
                <lastBuildDate>Tue, 24 May 2016 15:26:07 +0000</lastBuildDate>
                <generator>Joomla! - Open Source Content Management</generator>
                <atom:link rel="self" type="application/rss+xml" href="http://192.168.1.39/index.php/component/content/?
id=14&amp;Itemid=473&amp;format=feed&amp;type=rss"/>
                <language>es-es</language>
        </channel>
</rss>
```

170. Joomla!, dentro de la configuración global, permite desactivar que se muestre la versión ejecutada en el servidor, pero no permite eliminar la etiqueta completamente. Esta información permite a un posible atacante conocer y poder realizar ataques más específicos y mayor probabilidad de éxito contra la plataforma:

171. Por este motivo, tendremos que modificar el código de la plataforma para ocultar el mensaje. Realizaremos el siguiente cambio dentro del fichero "libraries/cms/application/site.php", comentando la línea original, sustituyéndola por la que nosotros queramos o dejándola en blanco:

```
// Add version number or not based on global configuration
if ($this->get('MetaVersion', 0))
{
        $document->setGenerator('Joomla! - Open Source Content Management - Version ' . JVERSION);
}
else
{
        $document->setGenerator(null);
        //$document->setGenerator('Joomla! - Open Source Content Management');
}
```

172. Con esta modificación, la etiqueta meta “Generator” no figurará en el código HTML, y tampoco del *feed RSS*:

```
<?xml version="1.0" encoding="utf-8"?>
<!-- generator="" -->
<rss version="2.0" xmlns:atom="http://www.w3.org/2005/Atom">
        <channel>
                <title>Guia Joomla!</title>
                <description><![CDATA[]]></description>
                <link>http://192.168.1.39/index.php?id=14</link>
                <lastBuildDate>Tue, 24 May 2016 15:27:12 +0000</lastBuildDate>
                <generator></generator>
                <atom:link rel="self" type="application/rss+xml" href="http://192.168.1.39/index.php/component/content/?
id=14&amp;Itemid=473&amp;format=feed&amp;type=rss"/>
                <language>es-es</language>
        </channel>
</rss>
```

173. En cuanto al *feed ATOM*, podemos comprobar que la información continua figurando:

```
<?xml version="1.0" encoding="utf-8"?>
<!-- generator="" -->
<feed xmlns="http://www.w3.org/2005/Atom"  xml:lang="es-es">
        <title type="text">Guia Joomla!</title>
        <subtitle type="text"></subtitle>
        <link rel="alternate" type="text/html" href="http://192.168.1.39"/>
        <id>http://192.168.1.39/index.php</id>
        <updated>2016-05-24T15:23:25+00:00</updated>
        <author>
                <name>Guia Joomla!</name>
        </author>
        <generator uri="https://www.joomla.org"></generator>
        <link rel="self" type="application/atom+xml" href="http://192.168.1.39/index.php/component/content/?id=14&amp;Itemid=473&amp;format=feed&amp;type=atom"/>
</feed>
```

174. Será necesario editar el fichero "libraries/joomla/document/renderer/feed/atom.php" y comentar la línea asociada, o sustituirla por la información deseada:

```
//$feed .= "    <generator uri=\"https://www.joomla.org\"" . $versionHtmlEscaped . ">" . $data->getGenerator() . "</generator>\n";
$feed .= '    <link rel="self" type="application/atom+xml" href="' . str_replace(' ', '%20', $url . $syndicationURL) . "\"/>\n";
```

Este cambio modificará la respuesta mostrada anteriormente, ocultando la información relevante:

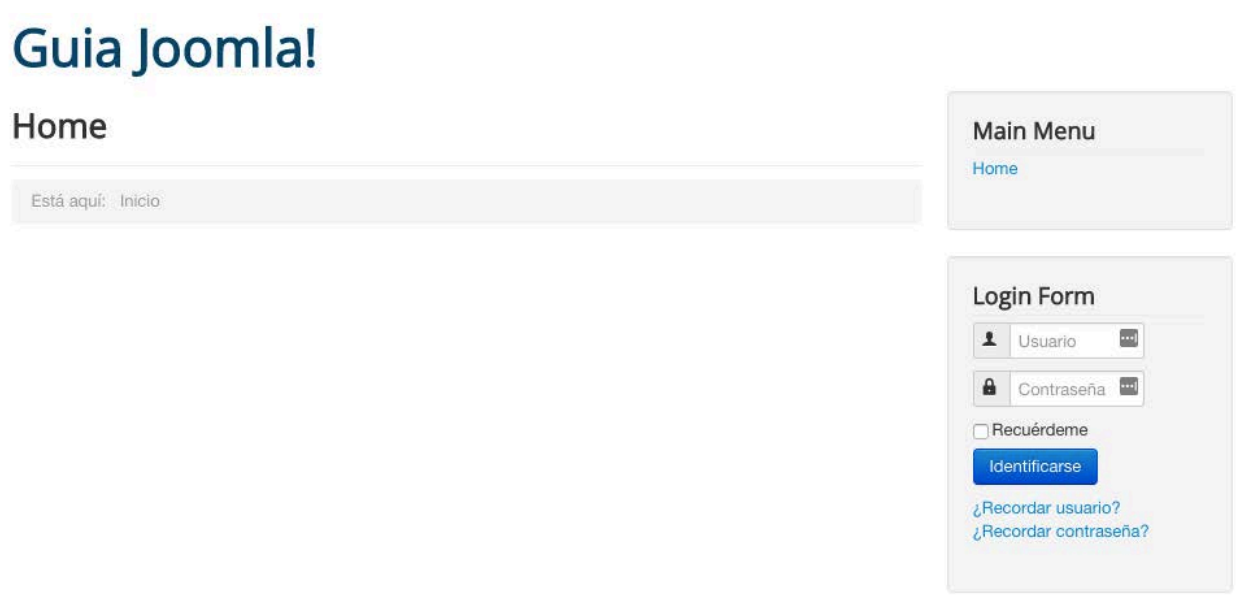
```
<?xml version="1.0" encoding="utf-8"?>
<!-- generator="" -->
<feed xmlns="http://www.w3.org/2005/Atom"  xml:lang="es-es">
        <title type="text">Guia Joomla!</title>
        <subtitle type="text"></subtitle>
        <link rel="alternate" type="text/html" href="http://192.168.1.39"/>
        <id>http://192.168.1.39/index.php</id>
        <updated>2016-05-24T15:24:58+00:00</updated>
        <author>
                <name>Guia Joomla!</name>
        </author>
        <link rel="self" type="application/atom+xml" href="http://192.168.1.39/index.php/component/content/?id=14&amp;Itemid=473&amp;format=feed&amp;type=atom"/>
</feed>
```

14. GESTION DE USUARIOS Y ACCESOS

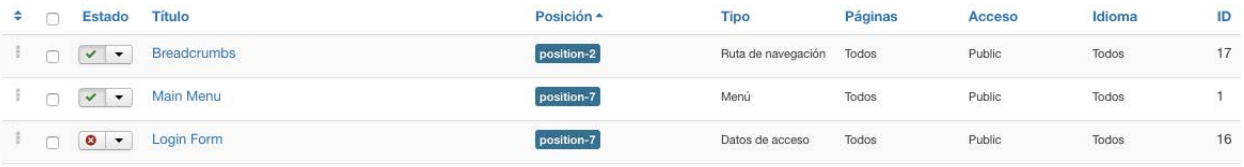
175. Existen dos categorías generales para los usuarios de un sitio Joomla!:

- **Invitados**: son visitantes que han accedido directamente a nuestro sitio web, a través de cualquier enlace o un buscador. Generalmente tendrán acceso a la mayor parte del contenido y podrán navegar libremente por el sitio
- **Usuarios registrados**: poseen credenciales de acceso, es decir, un nombre de usuario y una contraseña que les permite acceder a un área restringida al resto de los visitantes, recibiendo algún privilegio asociado.

176. En las últimas versiones de Joomla! está deshabilitado por defecto el registro de nuevos usuarios. Esta característica debe ser mantenida por seguridad, y adicionalmente eliminaremos el formulario de acceso que se encuentra accesible desde el sitio web:

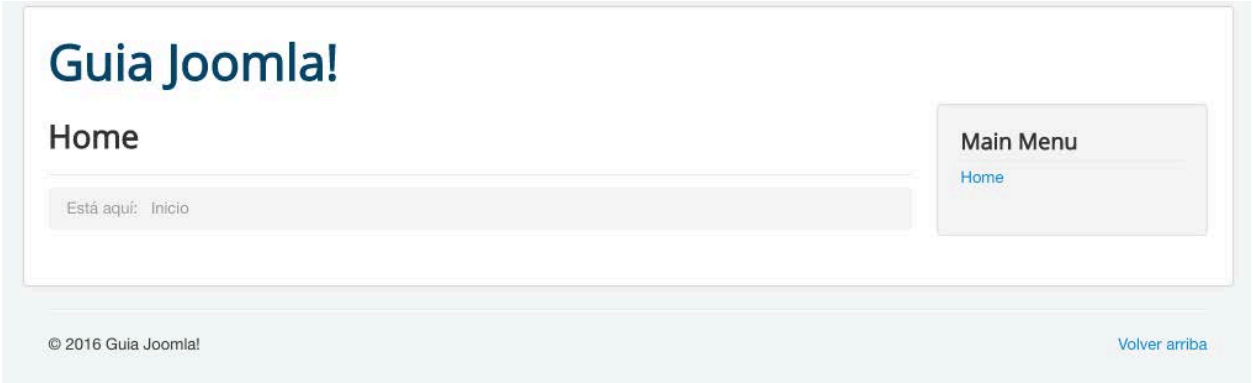


177. Para ello, accederemos al menú "Extensiones"->"Módulos", y desactivaremos el módulo "Login Form":

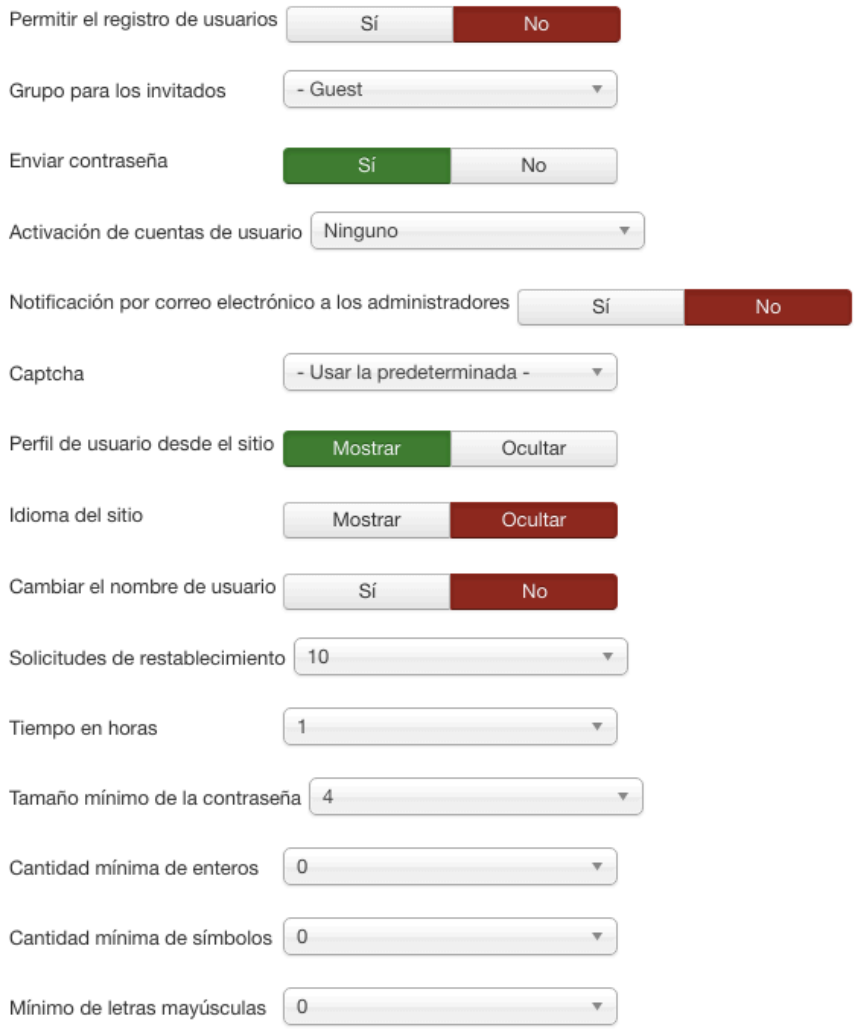


Estado	Título	Posición	Tipo	Páginas	Acceso	Idioma	ID
✓	Breadcrumbs	position-2	Ruta de navegación	Todos	Public	Todos	17
✓	Main Menu	position-7	Menú	Todos	Public	Todos	1
✗	Login Form	position-7	Datos de acceso	Todos	Public	Todos	16

178. Comprobaremos, accediendo a la página principal, que el formulario de acceso a usuarios, así como las opciones "Recordar usuario" o "Recordar contraseña", han sido eliminadas correctamente:



179. En ese caso de que la opción de registro de nuevos usuarios sea completamente necesaria para las funcionalidades de nuestro sitio web, deberemos realizar una serie de modificaciones en la configuración para incrementar la seguridad por defecto. Para ello accederemos a "Configuración Global"->"Usuarios" y activaremos la opción de "Permitir el registro de usuarios" que por defecto se encuentra desactivada:



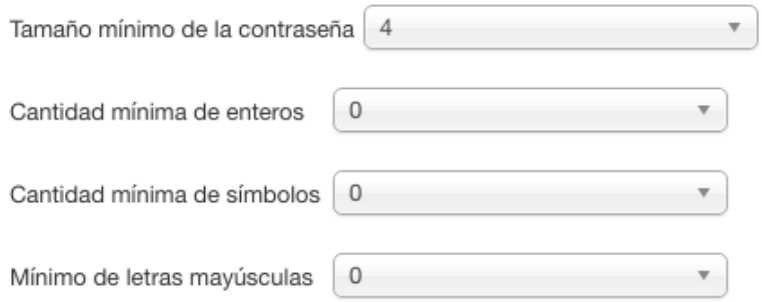
180. A continuación, modificaremos el proceso de "Activación de cuentas de usuario", pudiendo seleccionar las siguientes opciones:

- **Ninguno**: no es necesario realizar activación, por lo que el usuario puede acceder directamente en cuanto finaliza el proceso. Esta opción no debe ser activada por motivos de seguridad.
- **Por sí mismo**: el usuario indicará una dirección de email donde, tras finalizar el proceso, recibirá un enlace con un token para activar la cuenta
- **Administrado**: una vez finalizado el proceso de alta realizado en "Por sí mismo", los administradores recibirán un correo electrónico con un enlace y un token para activar cada una de las cuentas de forma individual. Para ello, necesitaremos activar la opción siguiente "Notificación por correo electrónico a los administradores". Mientras esta acción no se realice y los administradores comprueben los datos del solicitante de forma manual, la cuenta no se encontrará activa. Esta opción es la más recomendable desde el punto de vista de seguridad

181. A continuación, modificaremos los valores de la opción "Solicitudes de restablecimiento", que permite controlar el número de peticiones para restablecer la contraseña en un tiempo determinado. Por norma general, este valor puede ser bajo, considerándose seguro la solicitud de 3 restablecimientos en un periodo de 1 hora.

15. FORTALEZA Y POLÍTICA DE CONTRASEÑAS

182. Dentro del panel de configuración anterior, en "Configuración Global"->"Usuarios", finalmente podremos generar una política para la generación de contraseñas del sitio. Por defecto, Joomla! permitirá al usuario seleccionar una contraseña de al menos cuatro caracteres, sin necesidad de añadir códigos numéricos, símbolos o mayúsculas:



183. En función de la criticidad de los datos y la información manejada por el sitio, esta política podrá ser más o menos estricta. Por norma general, se considerará segura para un sitio sin acceso a información confidencial, la creación de una política que incluya las siguientes características:

- Longitud de al menos 8 caracteres.
- Al menos 1 carácter numérico.
- Al menos 1 símbolo.
- Al menos 1 carácter en mayúsculas.

16. INSTALACIÓN DE EXTENSIONES

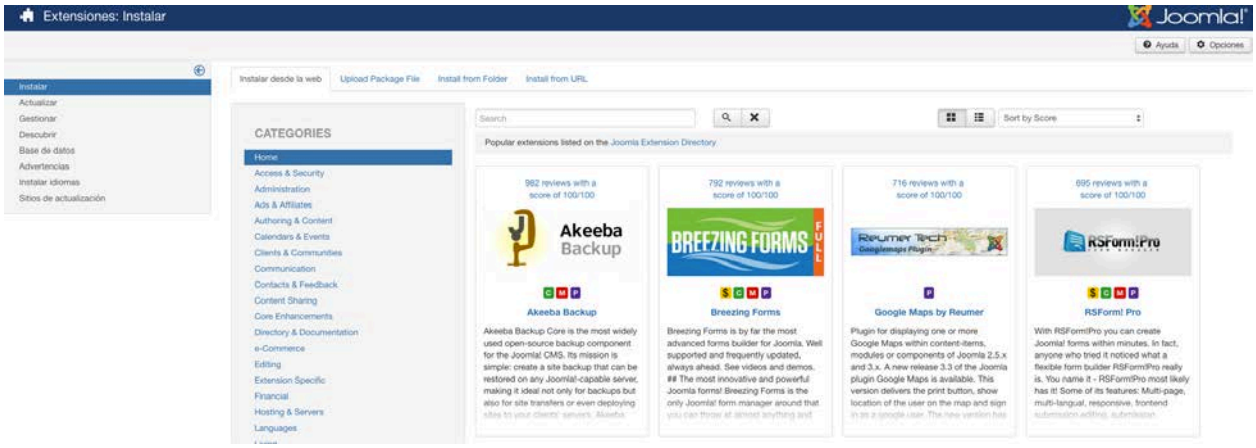
184. Joomla! es un gestor de contenidos modular, que permite añadir funcionalidades a nuestro sitio de una forma sencilla, tanto escritas por los propios desarrolladores del proyecto, como por terceros.

185. Debemos diferenciar las extensiones que existen:

- **Componentes**: se trata de un código que añade funcionalidades complejas a nuestro sitio web, y generalmente estarán compuestos de la parte de administración (BackEnd) y la parte pública (FrontEnd). Alguno de los componentes por defecto que se instalarán con Joomla! serán el sistema de gestión de contenido, los formularios de contacto y los enlaces web.

- **Módulos**: sirven para facilitar el acceso a un determinado sitio de nuestra web, facilitando la navegación a los usuarios, y que pueden ser insertados en diferentes posiciones de nuestra plantilla de forma flexible. Algunos de ellos, por ejemplo, pueden mostrar un código HTML seleccionado por el administrador, crear un atajo para utilizar el buscador, o vinculados a un componente, como por el ejemplo el gestor de contenido.
- **Plugins**: añaden funcionalidad a Joomla! y resto de extensiones, pero solamente cuando se produce un evento predefinido. Cuando este se produce, las funciones del plugin se ejecutarán en secuencia.

186. La instalación de estas extensiones se realiza de forma sencilla desde el BackEnd de administración, y para realizarlo, lo haremos desde el menú "Extensiones"->"Gestionar":

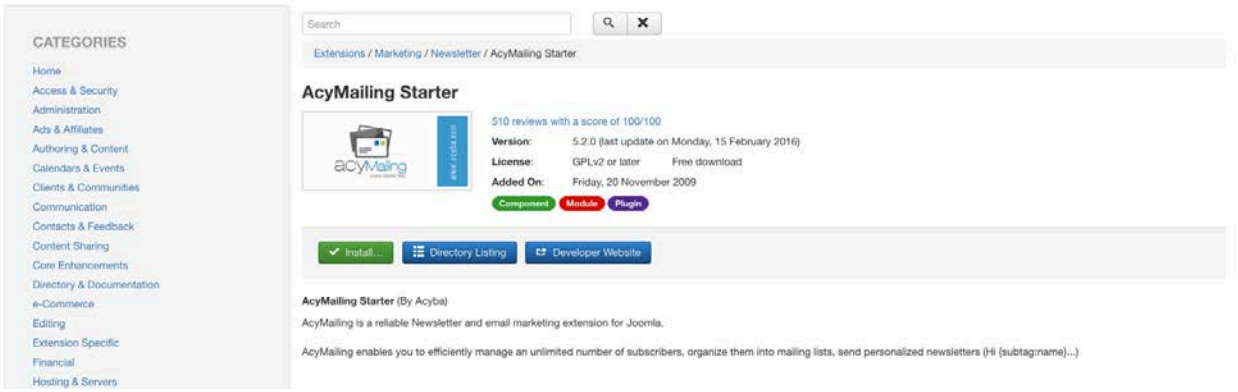


NOTA
Es recomendable realizar una copia de seguridad de nuestro sitio antes de la instalación de una extensión, ya que puede provocarse alguna incompatibilidad, error de carga o producir inestabilidad en el gestor de contenidos que no permita continuar su ejecución de forma correcta.

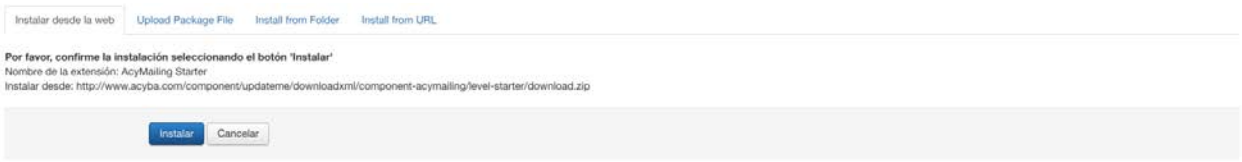
187. Una vez en esta sección de configuración, tendremos una serie de opciones para realizar la instalación:

188. **Instalar desde Web**

Permite la instalación de una extensión sin la necesidad de descargarla previamente en nuestro equipo, pudiendo realizar la búsqueda desde el propio panel. Será la forma recomendable de instalación, aunque existen algunos casos en los que no será útil, como en extensiones de pago, que requiera el registro y compra previa. El proceso será sencillo, seleccionaremos la extensión deseada y pulsaremos sobre el botón "Install":



Confirmamos nuestra selección de nuevo:

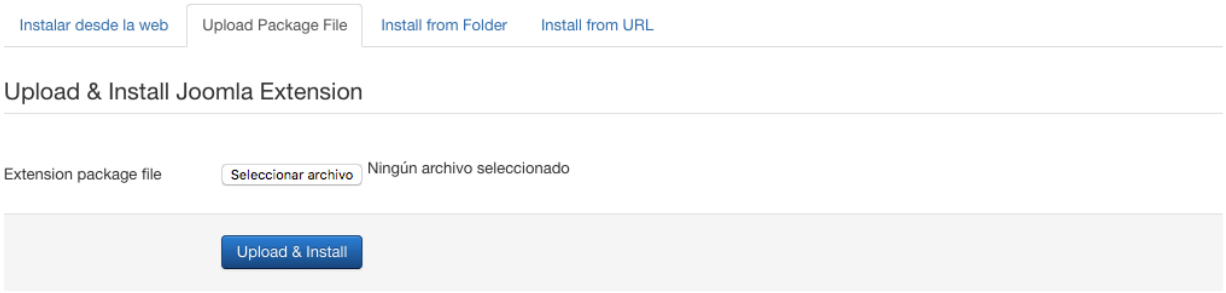


Y finalmente obtendremos el resultado satisfactorio de la instalación:

El resto de opciones generan un proceso similar, ofreciendo diferentes modalidades para la subida de archivos:

189. Subir archivo e instalar

Si hemos descargado previamente la extensión a nuestro equipo o no es posible acceder al servidor de descarga desde nuestro sitio Web, podemos utilizar el formulario para seleccionar el fichero comprimido y automáticamente será instalado en nuestro sitio.

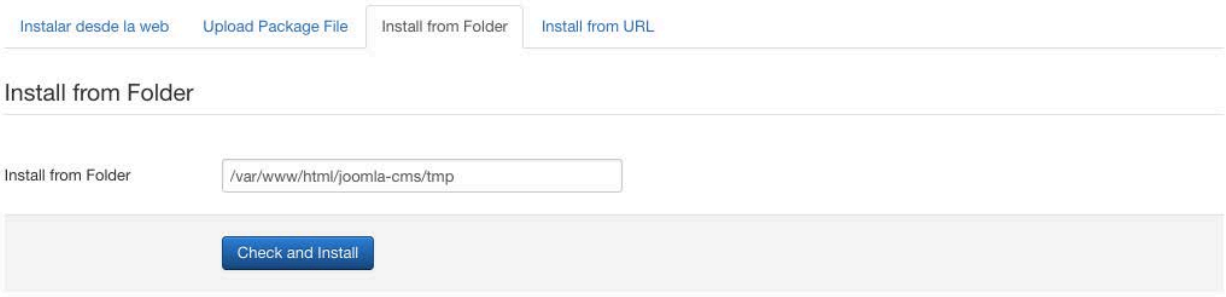


Dependiendo de la configuración de PHP, es posible que tengamos restringido un tamaño máximo para la carga de ficheros en nuestro servidor. En estos casos, podemos modificar la configuración del fichero "php.ini" para incrementar estos valores, modificando las siguientes variables:

upload_max_filesize = 10M
post_max_size = 10M

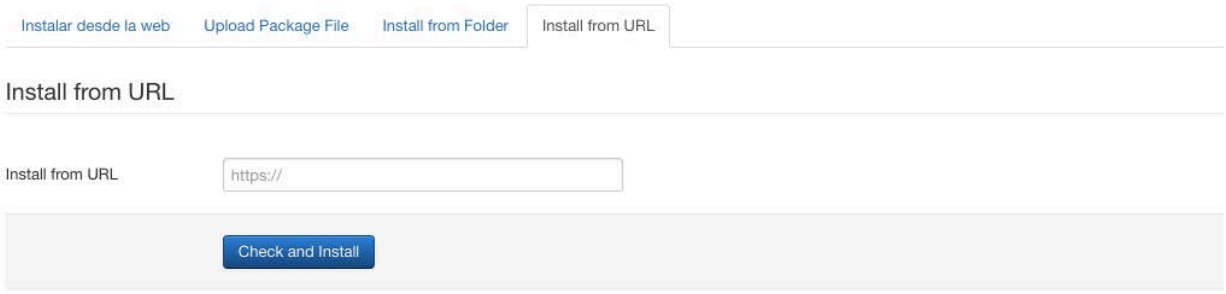
190. **Instalar desde directorio**

En el caso de que no podamos realizar la carga desde un equipo o modificar los valores anteriormente descritos, podremos instalar la extensión utilizando este método. Para ello deberemos indicar la ruta al directorio donde descomprimiremos la extensión que previamente hemos subido al servidor utilizando cualquier método disponible, SFTP, SCP etc. Por defecto, la ruta será el directorio temporal "tmp" incluido en el directorio raíz de la instalación de Joomla!.



191. **Instalar desde URL**

Nos permite especificar la URL donde se encuentra el fichero comprimido de la extensión a instalar, y Joomla! se encargará de descargarlo, extraer la información necesaria y realizar las tareas necesarias para la instalación.



17. FORZADO DE CONEXIÓN SOBRE SSLZ

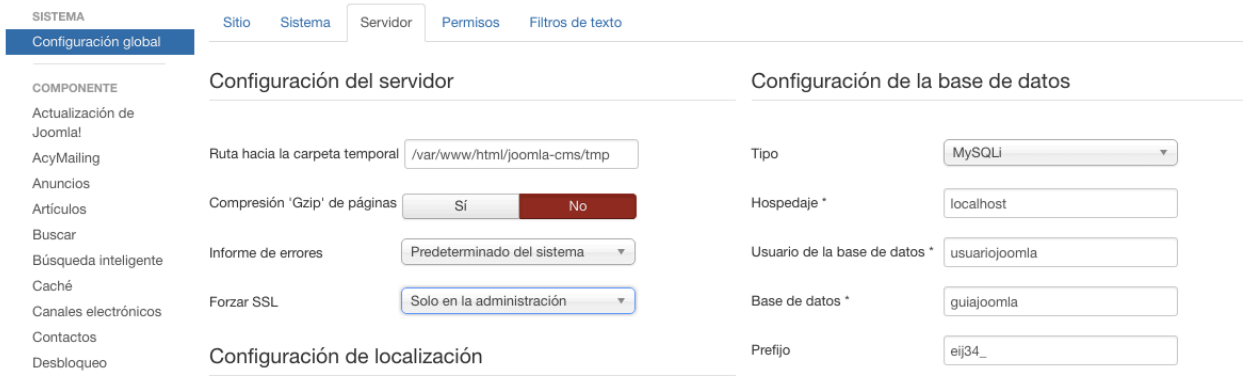
17.1. UTILIZANDO HTACCESS

192. Si deseamos forzar al cliente a utilizar HTTPS para la visualización completa del contenido de nuestro Joomla!, tanto de FrontEnd como BackEnd, incluiremos la siguiente directiva dentro del fichero ".htaccess" del directorio raíz de la aplicación:

```
RewriteEngine On
# Enable the Rewrite capabilities
RewriteCond %{HTTPS} !=on
# Check to make sure the connection is not already HTTPS
RewriteRule ^/?(.*) https://%{SERVER_NAME}/$1 [R,L]
# This rule will redirect users from their original location, to the same location but using HTTPS.
# The leading slash is made optional so that this will work either in httpd.conf or .htaccess context
```

17.2. DESDE EL BACKEND DE ADMINISTRACIÓN

193. También es posible diferenciar si lo que necesitamos es activar la conexión segura sobre todo el sitio, como en el método anterior, o solamente en el BackEnd de administración. Esta tarea se puede configurar desde el menú "Configuración Global"->"Servidor:



194. Como observaremos en las opciones de configuración, disponemos de tres opciones:

- **Ninguna**: no se fuerza la navegación segura, por lo que el visitante podrá acceder sobre HTTP/HTTPS según su elección o el enlace utilizado para visitar la página.
- **Solo administración**: la navegación por la parte pública de la página puede realizarse por HTTP/HTTPS, pero el acceso al BackEnd de administración se fuerza a realizarse bajo conexiones seguras HTTPS. Esta opción es la más recomendable
- **Todo el sitio**: en casos específicos donde se requiera que toda la navegación, tanto por la parte pública como privada del sitio, sea segura, la activaremos

195. Esta configuración también puede realizarse desde el terminal, editando el fichero de configuración "configuration.php":

```
public $proxy_user = '';
public $proxy_pass = '';
public $massmailoff = '0';
public $MetaRights = '';
public $sitename_pagetitles = '0';
public $force_ssl = '2';
```

196. La variable "$force_ssl" es la encargada de gestionar este proceso, y podemos asignarle los siguientes valores que corresponden con las opciones anteriormente mostradas del interfaz de gestión de Joomla!:

- **0** -> Ninguna
- **1** -> Sólo administración
- **2** -> Todo el sitio

18. USO DE DOBLE FACTOR

197. Durante los últimos dos años, muchos servicios online han comenzado a ofrecer autenticación de doble factor. Se trata de una medida de seguridad extra que, frecuentemente, requiere de un código obtenido a partir de una aplicación, o un mensaje SMS, además de una contraseña para acceder al servicio.

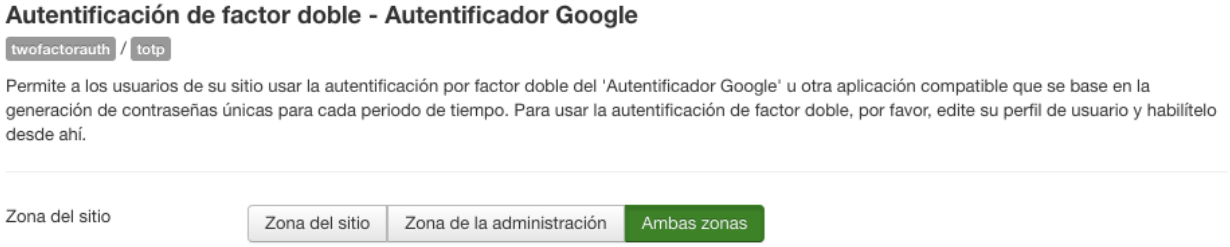
198. Los sistemas de doble factor de autenticación son mucho más seguros que las contraseñas. Muchos ataques de notoriedad pública, como los perpetrados contra cuentas de empresas de medios en Twitter el año pasado, no hubieran ocurrido si hubiera habido un sistema de doble factor implementado. Incluso si un atacante logra infectar un equipo y roba una contraseña, el acceso no podrá ser logrado ya que no cuentan con el código de acceso.

199. El inicio de una sesión con una contraseña, es la autenticación de un solo paso. Se basa únicamente en algo que una persona sabe. La autenticación de dos pasos, por definición, es un sistema en el que utiliza dos de los tres factores posibles para probar la identidad, en lugar de sólo uno:

- **Algo que eres**: sistemas como el reconocimiento facial, las huellas dactilares, el patrón de vena en tu mano, patrón de retina etc.
- **Algo que conoces**: por ejemplo, una contraseña, un PIN o un patrón de autenticación, como la basada en patrones que ofrecen los dispositivos Windows 8 y Android
- **Algo que tienes**: generalmente un token, un dispositivo en tu posesión. Existen dos tipo primarios de dispositivos de generación de token:
  - Tokens utilizados en línea (basados en reto/respuesta).
  - Tokens que pueden ser usados offline (tiempo, secuencia o basados en OTP).

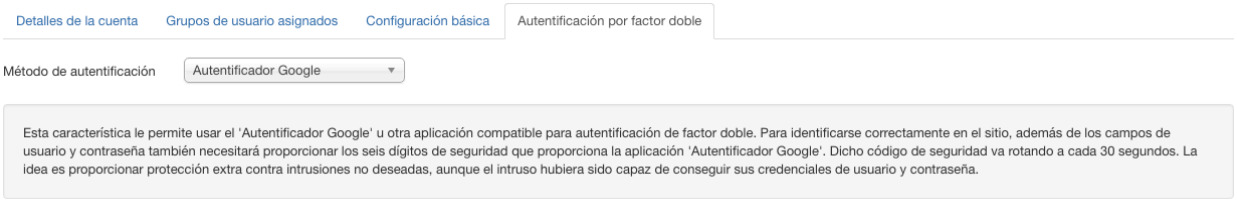
200. La instalación de Joomla! se distribuye con un plugin que permite utilizar esta autenticación de doble factor, pero primero deberemos habilitar éste desde el menú "Extensiones"->"Plugins":

201. Utilizaremos el sistema de factor doble de Google. Podremos seleccionar si deseamos este sistema para el acceso para el FrontEnd, BackEnd o para ambos (la opción recomendable), pulsando sobre el título del plugin:



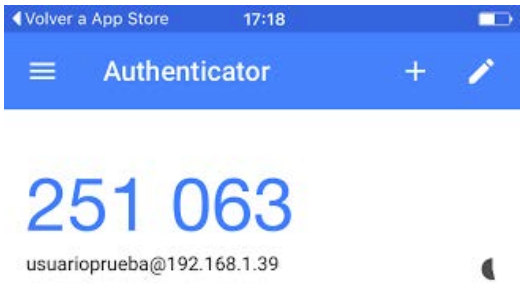
202. Ahora será necesario configurar de forma individual el servicio para cada uno de los usuarios del sistema, accediendo a "Usuarios"->"Gestionar":

203. Habilitaremos este método, al menos, para los Super User de nuestro sitio pulsando sobre cada uno, y dentro de los detalles de la pestaña "Autenticación por doble factor" seleccionando "Autenticador Google" como Método de autenticación:



204. Una vez activado, con los datos de nuestra cuenta abriremos la aplicación en nuestro dispositivo móvil e introduciremos los valores de forma manual o escanearemos el código QR desde la aplicación de Google Authenticator:

205. Automáticamente aparecerán los códigos cada 30 segundos en la aplicación de nuestro dispositivo:



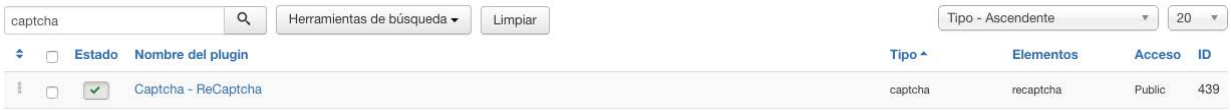
206. Introducimos el código en el campo "Código de seguridad" que encontraremos más abajo y salvaremos la configuración del usuario:

207. Comprobaremos que el acceso es correcto, accediendo a la zona de FrontEnd o BackEnd, e introduciendo el código asociado en el nuevo campo "Clave secreta":

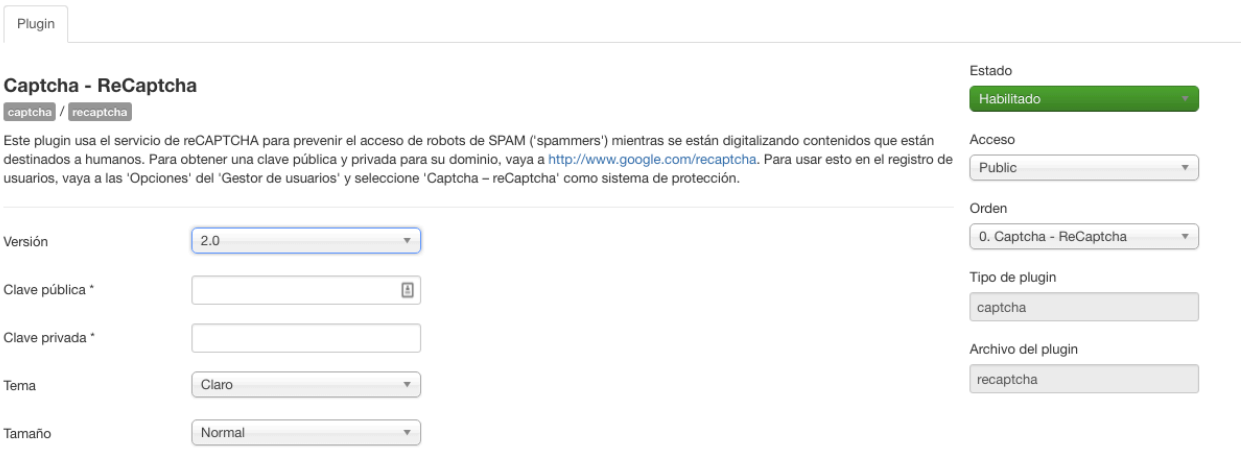
19. HABILITACIÓN DE RECAPTCHA

208. Un CAPTCHA es una prueba desafío-respuesta, que a menudo se encuentra insertado dentro de los formularios web para determinar si un usuario es humano. El propósito de estos captcha es bloquear el envío de spam, a través de estos formularios, por bots automatizados, que colocan el contenido no deseado en todas las partes del sitio que pueden. Para sistemas Joomla! utilizaremos el sistema reCHAPTCHA, que utiliza el sistema mejorado de Google.

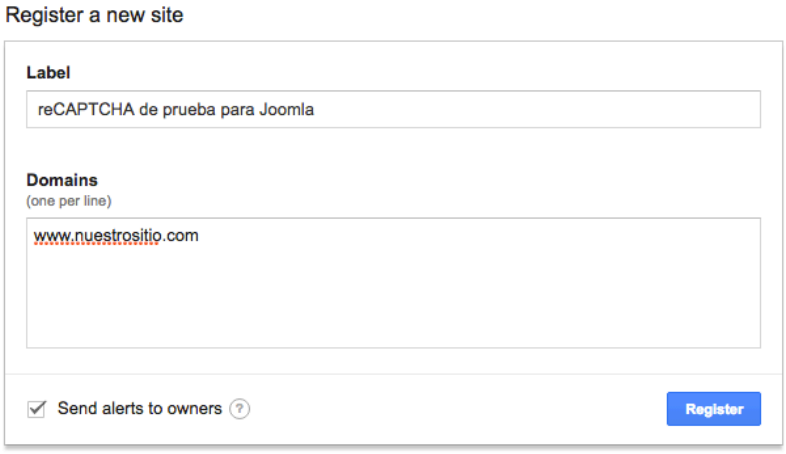
209. Esta funcionalidad se encuentra en el paquete de instalación original, por lo que no deberemos realizar la instalación de ningún paquete adicional. Primero, accederemos al menú "Extensiones"->"Plugins" y realizaremos su activación:



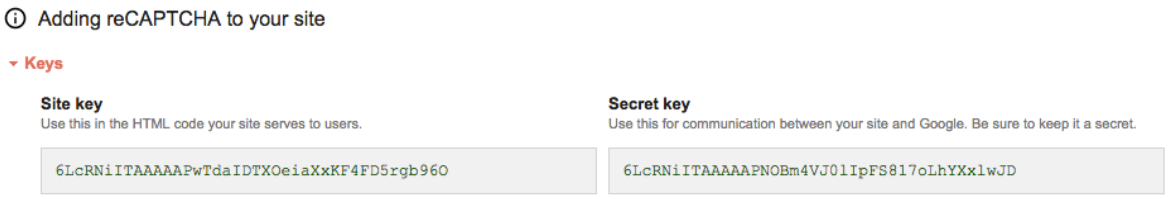
210. Después, pulsaremos sobre el plugin para acceder a la configuración:



211. Seleccionaremos la versión "2.0", y deberemos obtener los datos correspondientes a "Site key" y "Secret key", obligatorios para el correcto funcionamiento. Para obtenerlos, visitaremos la URL "http://www.google.com/recaptcha/admin", e introduciremos en la pantalla mostrada la descripción de nuestro sitio así como los dominios donde el sistema funcionará:

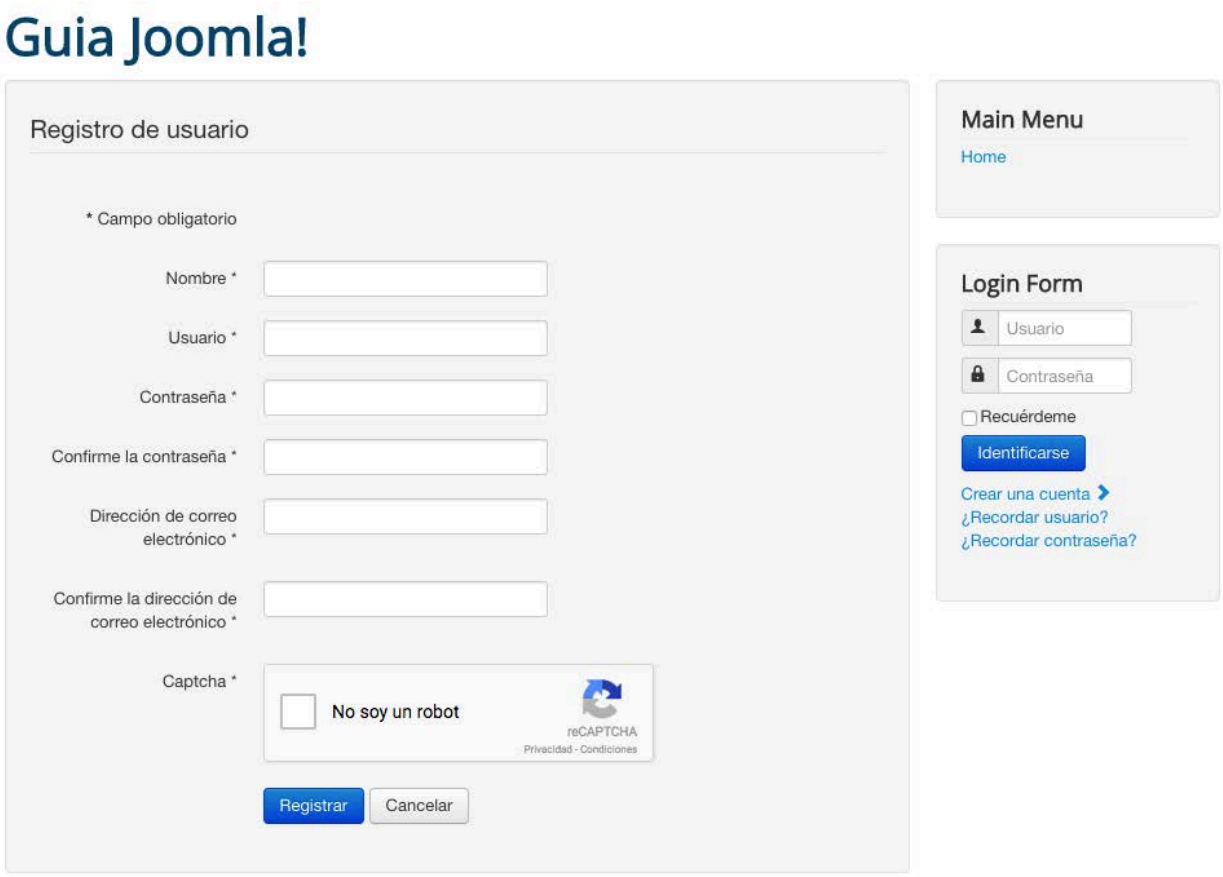


212. Una vez hayamos pulsado sobre "Register", accederemos a la página de configuración del sitio donde obtendremos los valores que necesitamos insertar en la configuración del sistema en Joomla!:



213. Después, accederemos a "Sistema"->"Configuración Global" y lo seleccionaremos como "Captcha predeterminado":

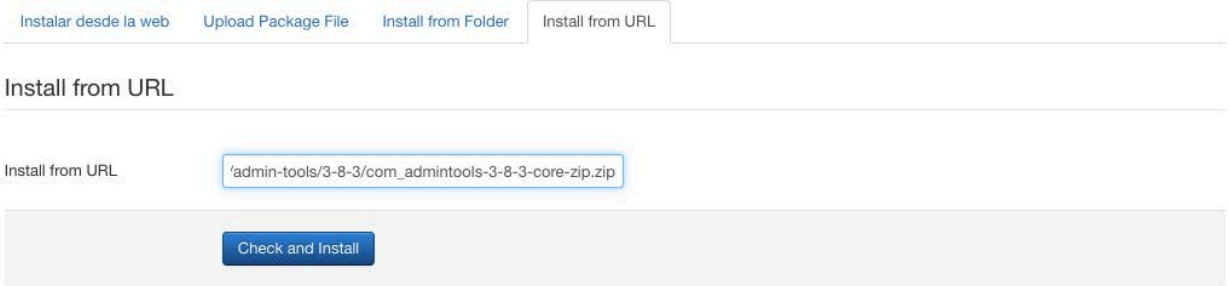
214. Si hemos realizado la configuración correctamente, deberíamos tener habilitado el sistema en nuestro sitio, por ejemplo en el formulario de alta de usuarios, si necesariamente se encontrara activado:



20. MÓDULOS A INSTALAR

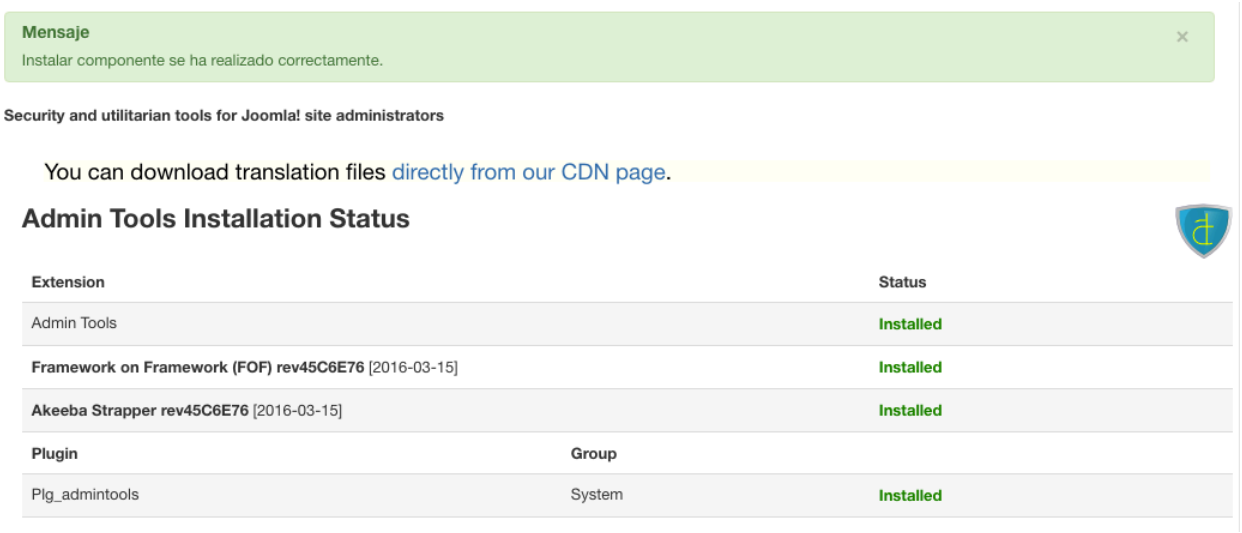
20.1. AKEEBA ADMIN TOOLS[12]

215. La suite de herramientas "Akeeba Admin Tools" nos permite realizar algunas tareas de seguridad de forma sencilla desde el backend de Joomla!. La instalación la realizaremos utilizando la URL[13] de descarga de la página principal de la extensión. En el momento de la escritura de esta guía, la versión estable más reciente es la 3.8.3:

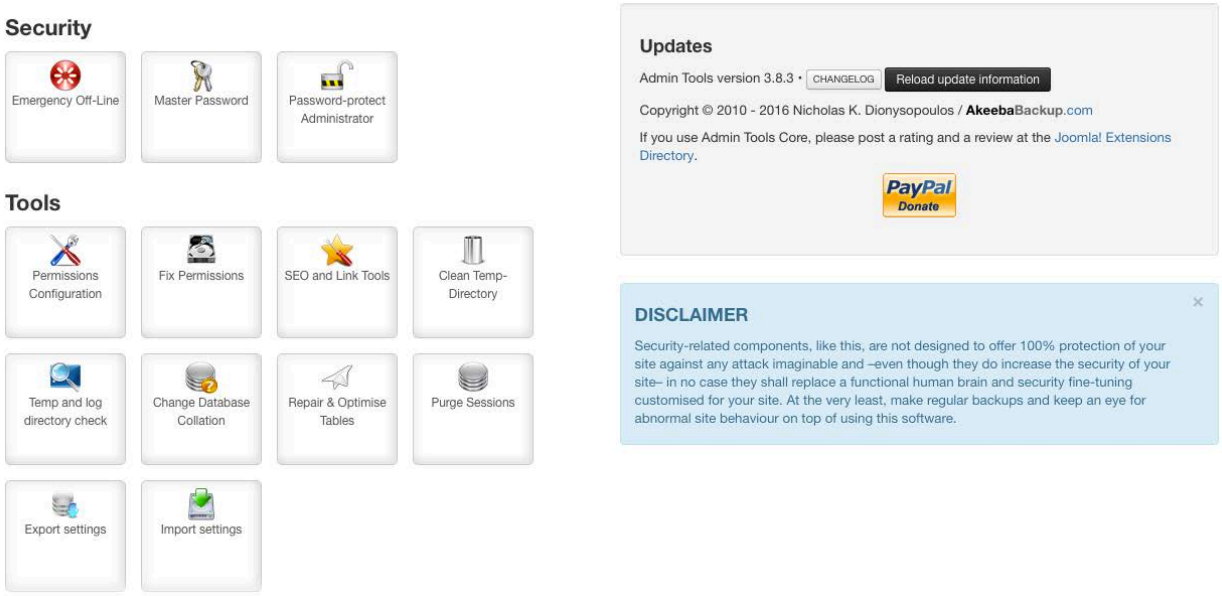


[12] http://extensions.joomla.org/extension/admin-tools
[13] https://www.akeebabackup.com/download/admin-tools/

216. Una vez pulsemos, deberíamos ver un mensaje indicando la correcta instalación de la extensión, así como de los plugins asociados:



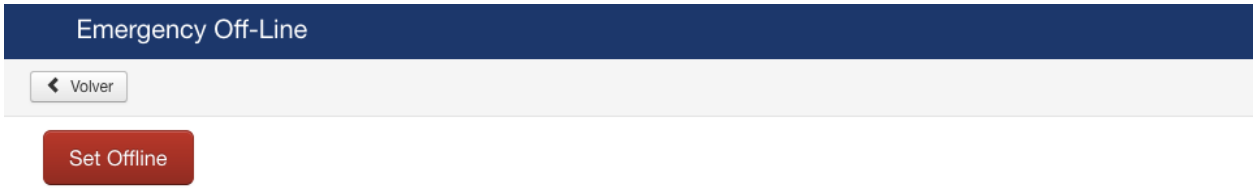
217. Podremos acceder a la configuración de la extensión desde el ménu "Componentes"->"Admin Tools":



218. Desde esta pantalla, tendremos acceso a las funcionalidades de la extensión, cuyos módulos más relevantes, desde el punto de vista de seguridad, describiremos a continuación:

20.1.1. EMERGENCY OFF-LINE

219. Al pulsar sobre este botón, accederemos a una nueva pantalla que nos permitirá dejar el sitio "fuera de línea", bloqueando el acceso a cualquier visitante, a excepción de la dirección IP utilizada por el administrador en ese momento.



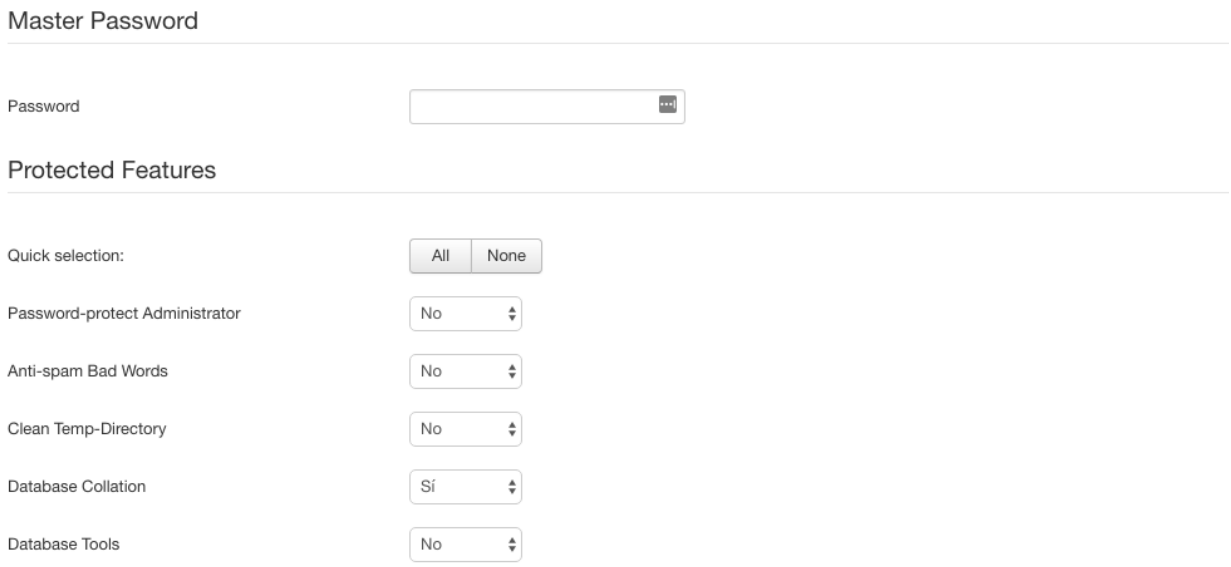
220. Se creará el siguiente fichero ".htaccess", que será mantenido hasta deshabilitar esta opción desde el mismo panel desde el que fue activado:

```
RewriteEngine On
RewriteCond %{REMOTE_HOST}      !192\.168\.1\.36
RewriteCond %{REQUEST_URI}      !offline\.html
RewriteCond %{REQUEST_URI}      !(\.png|\.jpg|\.gif|\.jpeg|\.bmp|\.swf|\.css|\.js)$
RewriteRule (.*)            offline.html    [R=307,L]
```

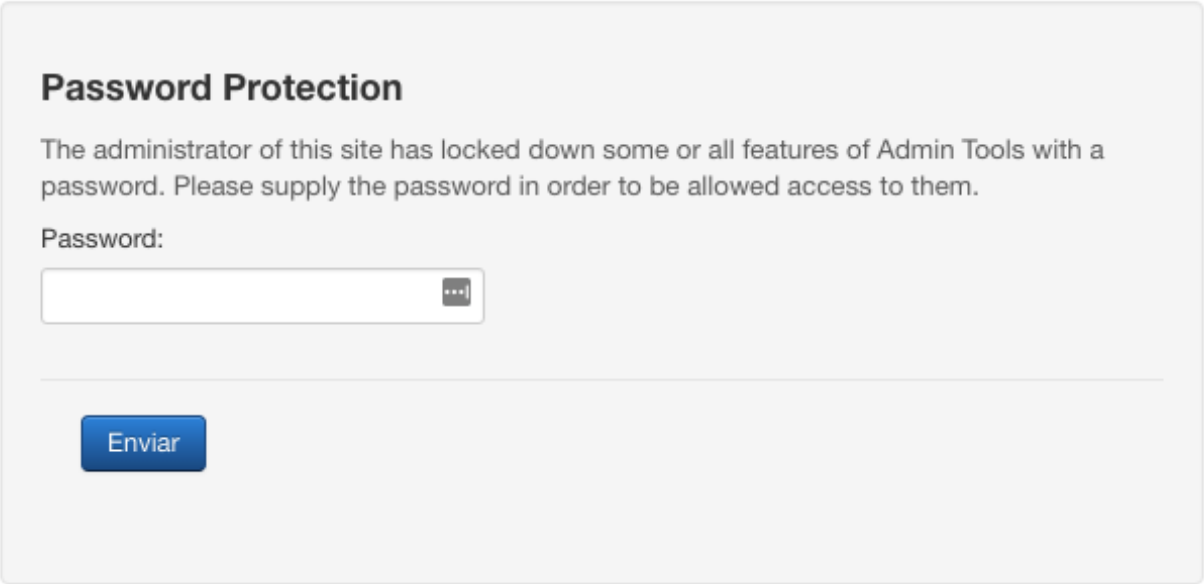
NOTAS ADICIONALES:
En caso de perder la conexión desde la dirección IP utilizada por el administrador, el acceso al BackEnd de Joomla! quedará bloqueado, por lo que será necesario realizar una conexión segura al servidor y eliminar el fichero ".htaccess" para habilitar de nuevo el acceso a todos los visitantes.

20.1.2. MASTER PASSWORD

221. Dotará a la extensión de una capa adicional de seguridad, al permitir incluir una contraseña para acceder a algunos de los módulos que gestiona:



222. Cuando esta contraseña sea activada, al acceder al BackEnd de configuración de la extensión, encontraremos la siguiente pantalla solicitándola para poder acceder a los módulos protegidos:



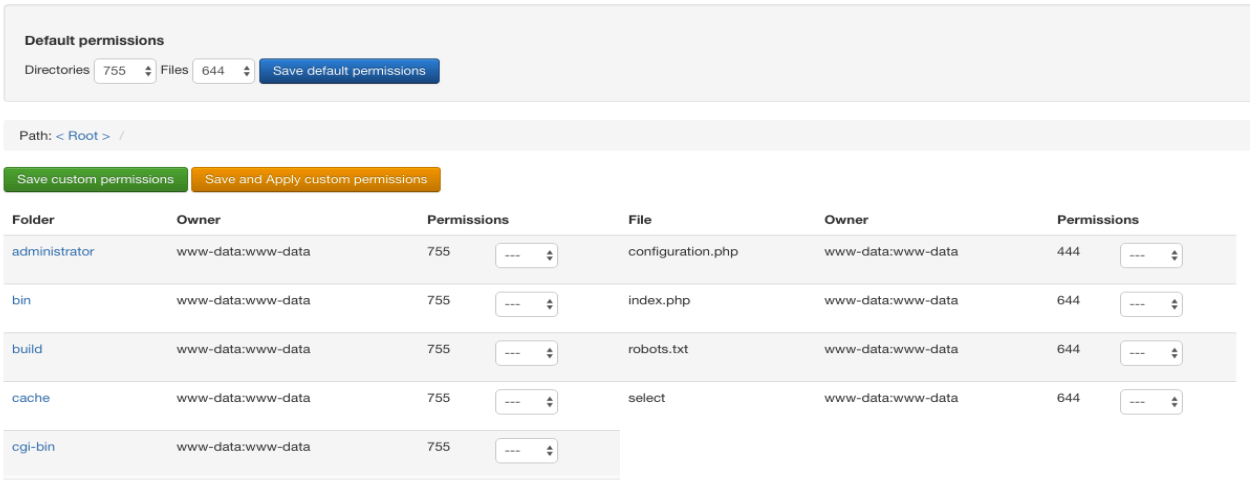
20.1.3. PASSWORD-PROTECT ADMINISTRATOR

223. Nos permite añadir una capa de autenticación al acceso a la zona de administrador ("/administrator") de Joomla!, de la misma forma que hemos visto en la sección "Restricciones de acceso al backend" de esta misma guía.

20.1.4. PERMISSIONS CONFIGURATION

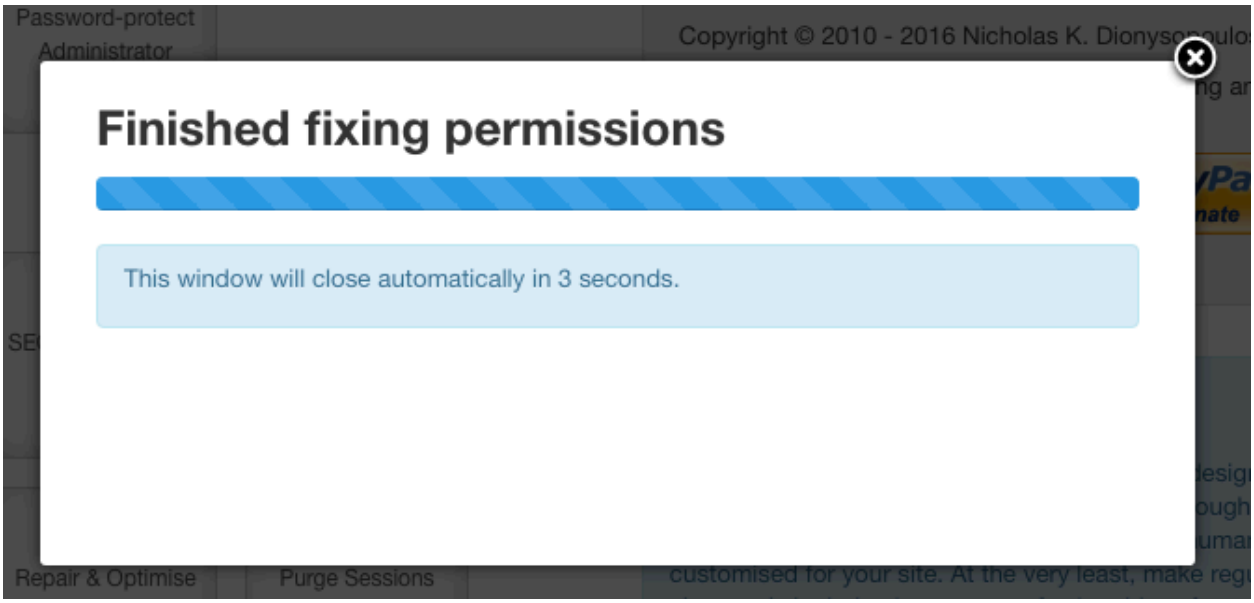
224. Este módulo nos permite comprobar los permisos configurados de los ficheros y directorios de la instalación de Joomla!. Como ya hemos visto anteriormente, será necesario asignar, por seguridad, los permisos 755 a directorios y 644 a ficheros.

225. Podremos visualizar estos permisos y modificar la configuración de estos, si hemos realizado alguna modificación necesaria, por ejemplo, para una extensión/módulo/plugin que haga uso de permisos especiales:



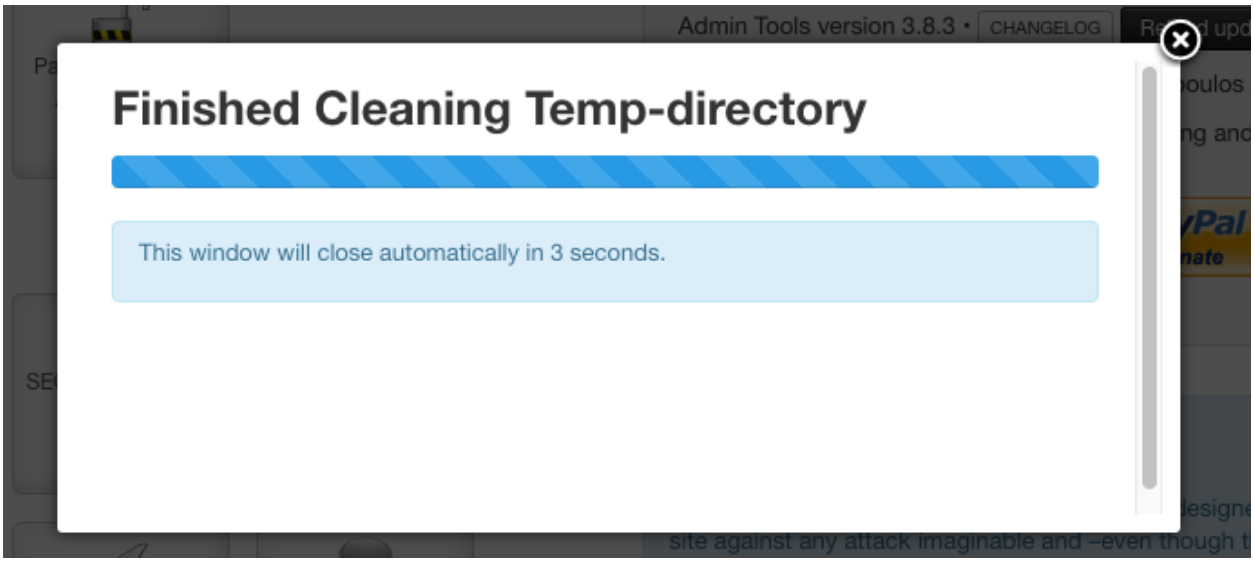
20.1.5. FIX PERMISSIONS

226. Dada la configuración mostrada anteriormente, al pulsar sobre este icono aplicaremos la política de permisos que haya sido establecida de forma rápida y sencilla:



20.1.6. CLEAN TEMP-DIRECTORY

227. Nos permite realizar una limpieza del directorio temporal de Joomla! sin necesidad de acceder al servidor por línea de comandos o utilizando algún protocolo de transferencia de ficheros:



20.1.7. PURGE SESSIONS

228. Elimina toda la información asociada a sesiones de usuarios dentro de Joomla!, reiniciando además las tablas asociadas en la base de datos, incluida la del administrador que nos encontremos utilizando.

20.2. JSECURE LITE[14]

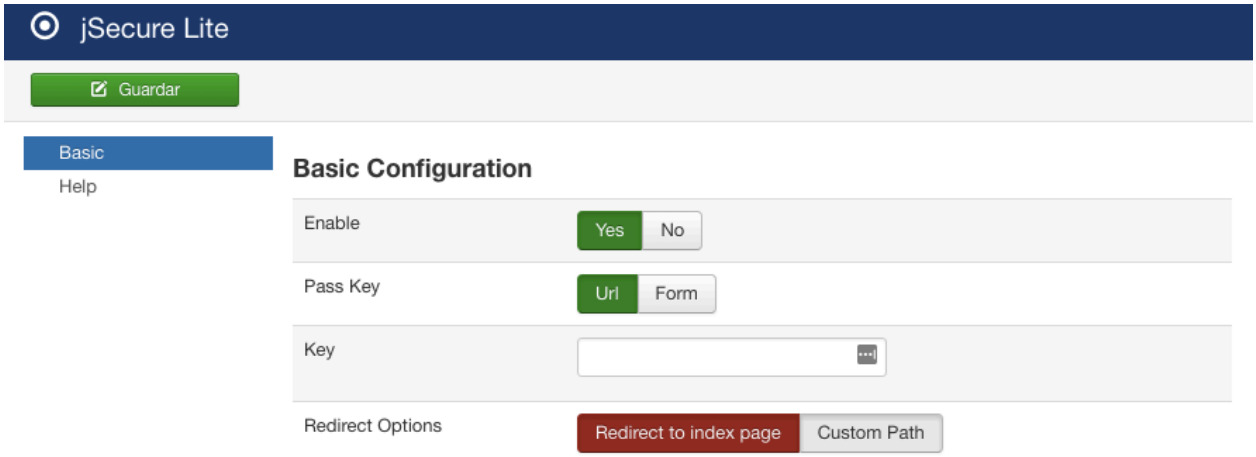
229. Este sencillo componente nos permite añadir una capa adicional de seguridad al acceso a nuestro backend de Joomla!, modificando la ruta de acceso a la sección de administrador añadiendo una palabra de seguridad.

230. Generalmente el acceso a esta zona de administración se realiza a través de la URL "http://misitiojoomla/administrator/". Gracias a este módulo, este acceso sólo será posible utilizando una palabra de seguridad que se añade a la ruta, quedando "http://misitiojoomla/administrator/?palabraclaveseguridad".

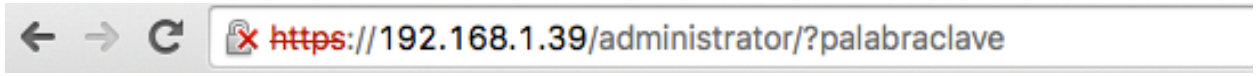
231. El primer paso para la correcta instalación será su descarga desde la página del creador. Una vez realizado un registro gratuito, descargaremos el fichero de instalación "jsecurelite1.0.unzipfirst.zip", que descomprimiremos en nuestro equipo.

232. Después, seleccionaremos la versión de la extensión correspondiente a nuestro Joomla! (en este caso "jsecurelite_joomla3.0.x"), y realizaremos la instalación desde "Extensiones"->"Gestionar"->"Upload Package File".

233. Una vez finalizada, configuraremos nuestra nueva palabra de paso desde "Componentes"->"jSecure Lite":



234. Una vez activado, si intentamos el acceso a través del directorio "/administrator/", de forma predeterminada nos redirigirá a la página de inicio, o a la URL que hayamos indicado si activamos la opción "Custom Path" de "Redirect Options". En caso de utilizar la palabra de paso adecuada, mostrará el acceso al backend:

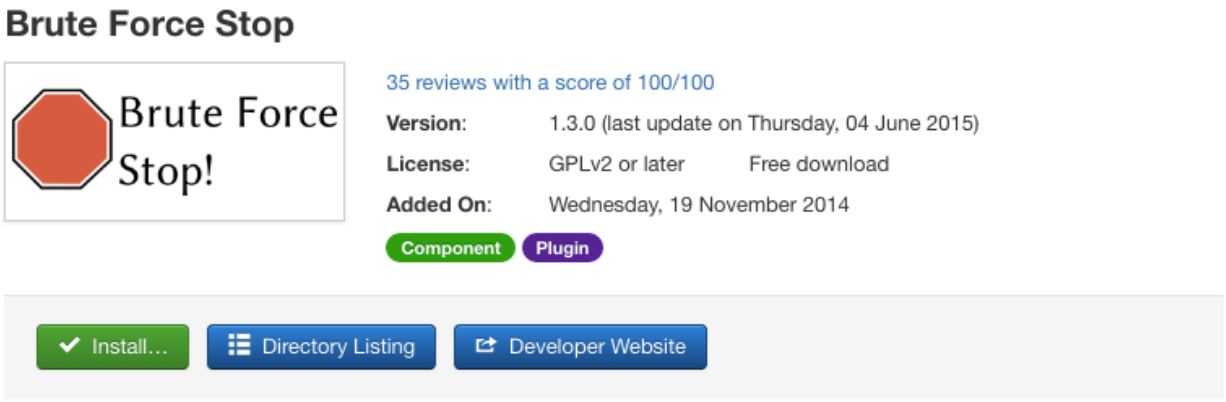


20.3. BRUTE FORCE STOP

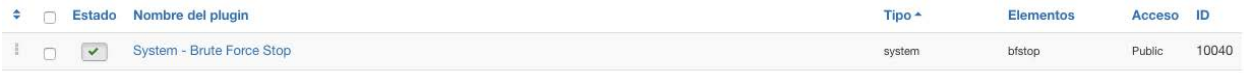
235. Esta extensión permite evitar los ataques de fuerza bruta en los formularios de acceso de nuestro sitio web, almacenando información acerca de los intentos fallidos de autenticación y bloqueando el acceso a la IP involucrada cuando se alcanza un número predefinido de errores, y permitiendo el envío de notificaciones a los administradores del sistema.

[14] http://www.joomlaserviceprovider.com/extensions/joomla/non-commercial/jsecure-lite.html

236. Realizaremos la instalación desde el propio gestor de extensiones de Joomla!:



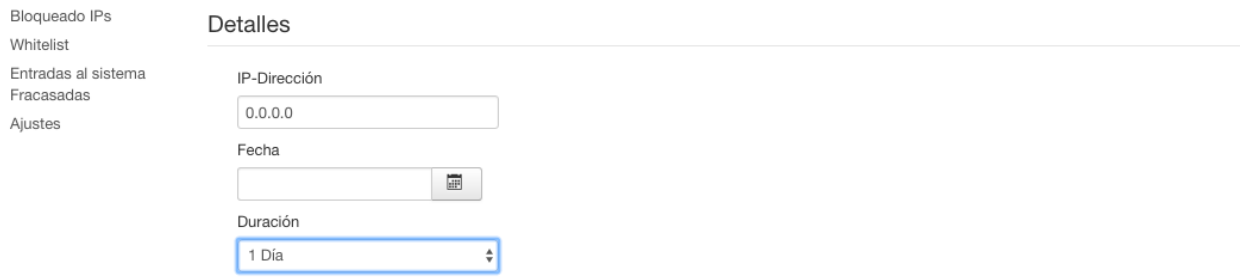
237. Una vez la extensión ha sido instalada, será necesario habilitar el plugin correspondiente para activar su funcionamiento. Para ello accederemos al menú "Extensiones"->"Plugin" y procederemos a buscarlo por el nombre "Brute Force Stop" para posteriormente activarlo.



238. Ahora procederemos a la configuración, desde el menú "Extensiones"->"Brute Force Stop Administration":

20.3.1. BLOQUEADO IPS

239. Mostrará la lista de direcciones IP bloqueadas, así como la duración, la fecha del bloqueo y su estado actual. Podremos crear un nuevo bloqueo pulsando en el botón "Nuevo":

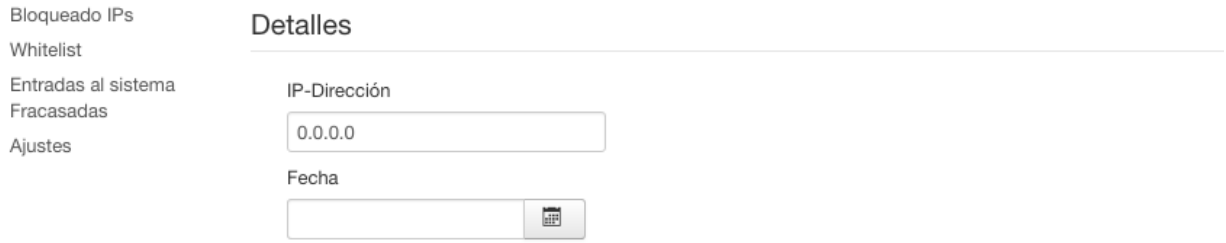


240. Deberemos rellenar los siguientes datos:

- IP-Dirección: dirección IP que deseemos incluir en la lista negra.
- Fecha: indica cuándo comenzará el bloqueo especificado.
- Duración: podremos especificar el intervalo de duración de este bloqueo, o mantenerlo de forma permanente.

20.3.2. WHITELIST

241. Mostrará la lista de direcciones IP en la lista blanca de acceso, así como la fecha de alta. Podremos incorporar una nueva dirección a esta lista blanca pulsando sobre "Nuevo":

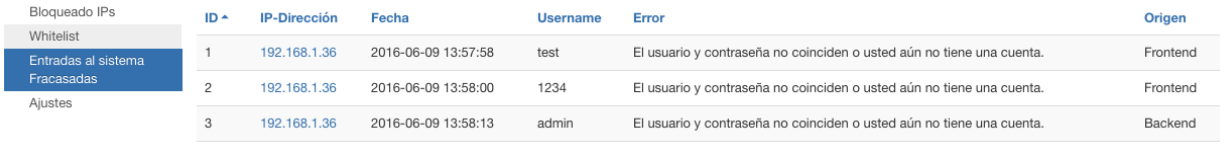


242. Deberemos rellenar los siguientes datos:

- IP-Dirección: dirección ip que deseemos incluir en la lista blanca.
- Fecha: indica cuando tendrá validez esta regla.

20.3.3. REGISTRO DE INTENTOS DE LOGIN FALLIDO

243. Al acceder a esta pestaña, se mostrará por pantalla un listado de los últimos accesos no válidos a nuestro sitio web, detallando la dirección IP utilizada, la fecha y hora del acceso, el nombre de usuario utilizado, el tipo de error y el origen de la alerta, que puede ser realizada desde el FrontEnd o el BackEnd de nuestro Joomla!:



20.3.4. AJUSTES

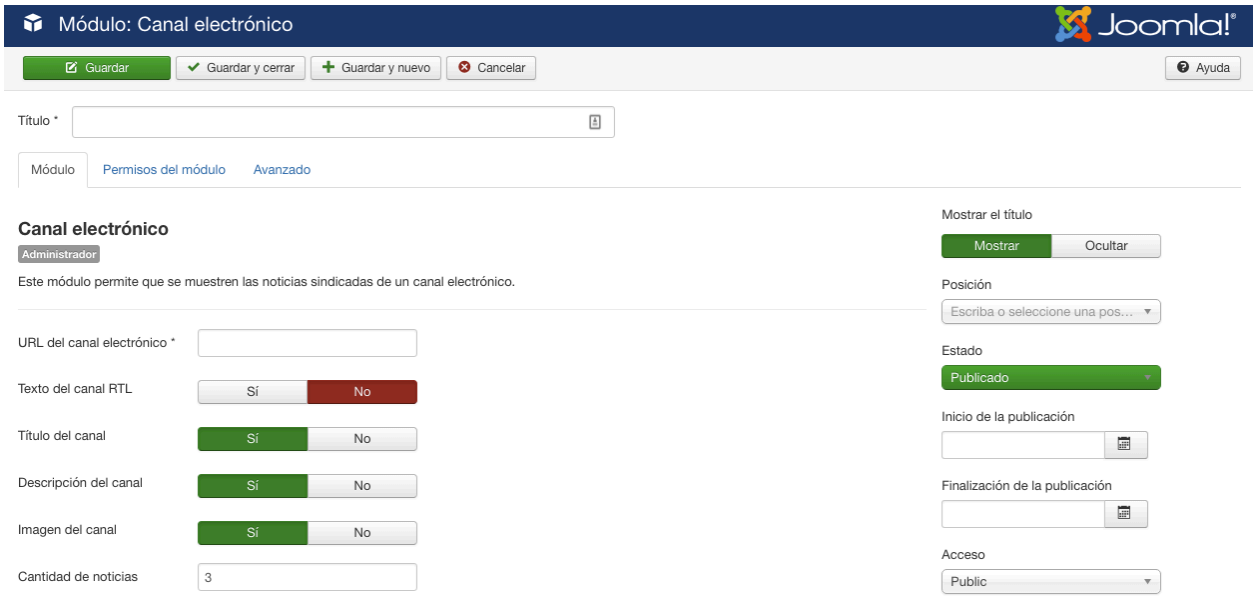
244. Permite comprobar si la configuración de la notificación del plugin está funcionando correctamente, pulsando sobre el botón "Notificación de Prueba", que enviará un correo electrónico de muestra al administrador.

21. INTEGRACIÓN DE FEEDS DE SEGURIDAD EN EL BACKEND

245. Integrar todas las extensiones y medidas de seguridad descritas a lo largo de este documento ayudarán a mejorar la seguridad de nuestro Joomla!, aunque frecuentemente se publicarán diferentes vulnerabilidades, tanto en el core como en las extensiones, que pueden afectar a nuestros sistemas y provocar una intrusión de un atacante.

246. Por ello, se recomienda integrar el Feed de seguridad oficial de Joomla! en el BackEnd de administración, donde se publicarán este tipo de vulnerabilidades y todos los datos relacionados, como criticidad, versiones afectadas, posibles soluciones etc.

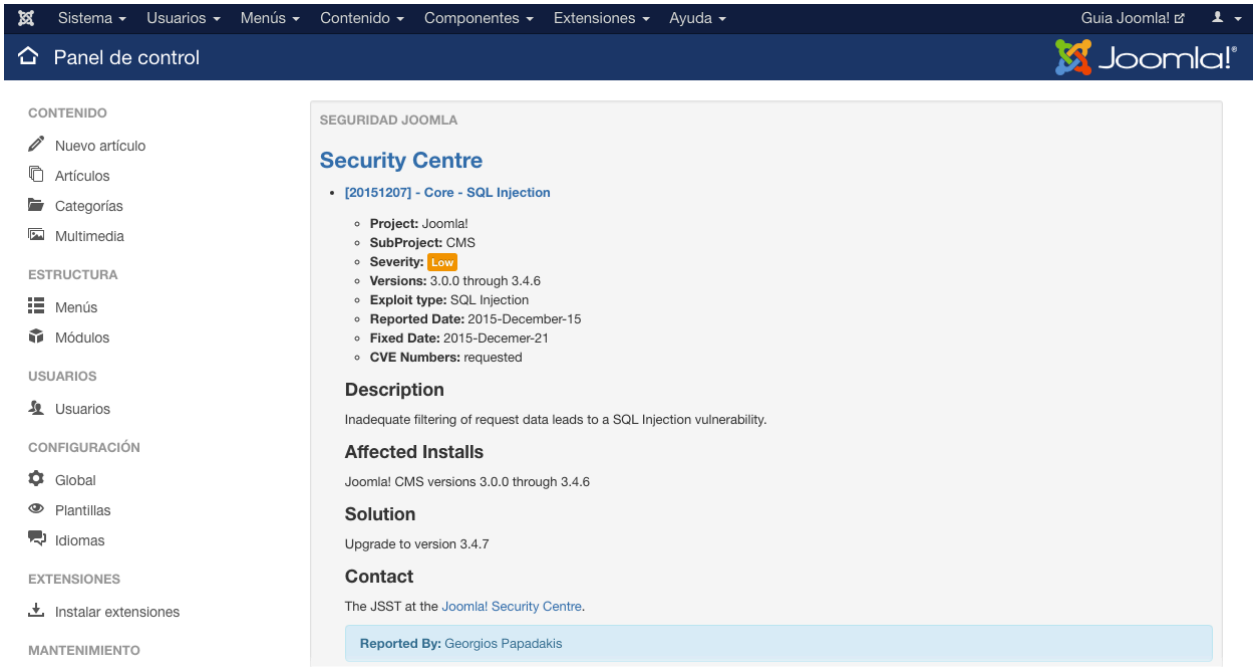
247. Accedemos al menú "Extensiones"->"Módulos" y en el desplegable de la barra de herramientas, seleccionamos "Administrador", y después pulsamos sobre el botón "Nuevo". Elegiremos el tipo de módulo "Canal Electrónico", y procederemos a su configuración:



248. Solamente deberemos introducir:

- Título con el que será mostrado en nuestro panel de administración.
- La URL del canal electrónico, que en este caso corresponde a *http://feeds.joomla.org/JoomlaSecurityNews?format=xml*
- Finalmente, como posición seleccionaremos el valor "cpanel".

249. Una vez introducidos estos valores, pulsaremos sobre "Guardar" y al pulsar sobre el botón de inicio de nuestro BackEnd, deberíamos poder visualizarlo:



250. También será una acción obligatoria, en función de los módulos, plugins o extensiones que hayamos instalado en nuestro sitio, añadir el feed oficial de seguridad de las extensiones de Joomla!, repitiendo el proceso anterior, pero incorporando la URL

http://feeds.joomla.org/JoomlaSecurityVulnerableExtensions?format=xml

22. CREACIÓN Y RECUPERACIÓN DE BACKUPS

251. Es necesario realizar un correcto respaldo de los sitios en producción, ya que, a pesar de que los discos donde se almacena la información suelen ser bastantes seguros, están expuestos a fallos lógicos, ataques de terceros, errores humanos, cortes de luz etc. que ponen en peligro la información almacenada. Recuperar la información sin un adecuado respaldo es posible, pero puede llegar a ser un proceso lento y excesivamente costoso, por lo que muchos usuarios desisten de recuperarla.

252. Esta guía no es objeto de discusión sobre los diferentes métodos de almacenamiento de backup, que van desde un soporte físico como una memoria USB o DVD, hasta el servicio externo de una empresa.

253. Se recomienda tener las siguientes consideraciones generales en cuenta a la hora de realizar una copia de seguridad:

- **Identificar los datos críticos:** No toda la información almacenada en un servidor es relevante. Es importante, durante la fase de planificación, que se tenga claro que información es la más importante, para no gastar tiempo, esfuerzo y espacio en copiar datos que no son relevantes.
- **Determinar la frecuencia adecuada:** hay que determinar la periodicidad de los backup, en función de cada cuánto tiempo cambia nuestra información. Por ejemplo, si nuestros archivos cambian una vez en semana no tiene sentido hacer copias de seguridad diarias, ya que estaremos almacenando los mismos datos una y otra vez.
- **Sistemas de copia continua:** lo ideal es que el backup esté permanentemente actualizado, es decir, que en caso de pérdida de la información podamos recuperar todo, incluso lo que haya sucedido sólo unos segundos antes del incidente.
- **Rápida recuperación:** Prácticamente tan importante es la correcta gestión de una réplica de nuestra información como el uso de un mecanismo que nos permita su restauración en el menor espacio de tiempo posible.
- **Verificar los backups:** Es aconsejable que cada cierto tiempo se compruebe que los datos que se almacenan en el backup son los correctos, y que se puede acceder a ellos sin ningún tipo de problema.
- **Recuperación granular:** Puede que sea necesario recuperar tan sólo un archivo concreto, no toda su estructura, por lo que deberemos contar con esta posibilidad en la fase de planificación.
- **Externalizar una copia cifrada:** Cuando estemos tratando con información de gran importancia o confidencialidad, se deberá considerar la posibilidad de realizar un segundo backup en un depósito seguro, fuera de las instalaciones donde se encuentra el servicio. De esta manera, las posibilidades de pérdida de información se verán reducidas al mínimo.

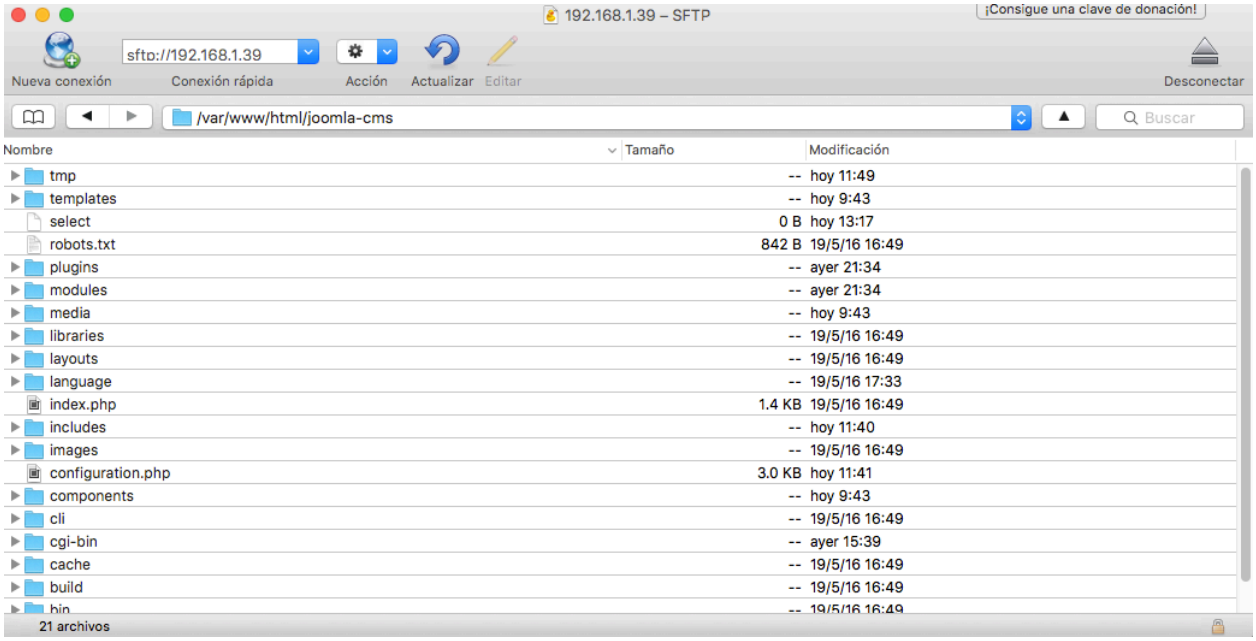
- **Destrucción de los backups:** a la hora de destruir estas copias de seguridad es fundamental asegurarnos de que el proceso cumple con las leyes de protección de datos, así como asegurar que ningún tipo de información pueda caer en manos de un tercero o un posible atacante.

22.1 BACKUP VIA FTP Y PHPMYADMIN

254. Las fases necesarias para realizar esta tarea serán las siguientes:

255. **Copia de seguridad de los ficheros de Joomla!**

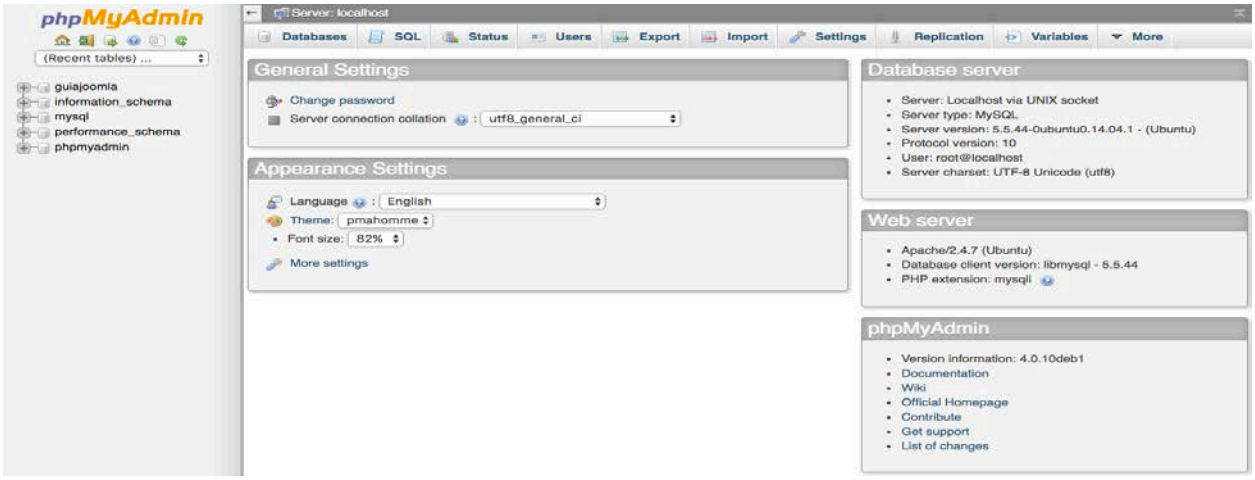
Este proceso se realizará utilizando cualquier forma de copia de archivos, sea a un soporte físico externo, o a otra máquina utilizando FTP, SCP etc.



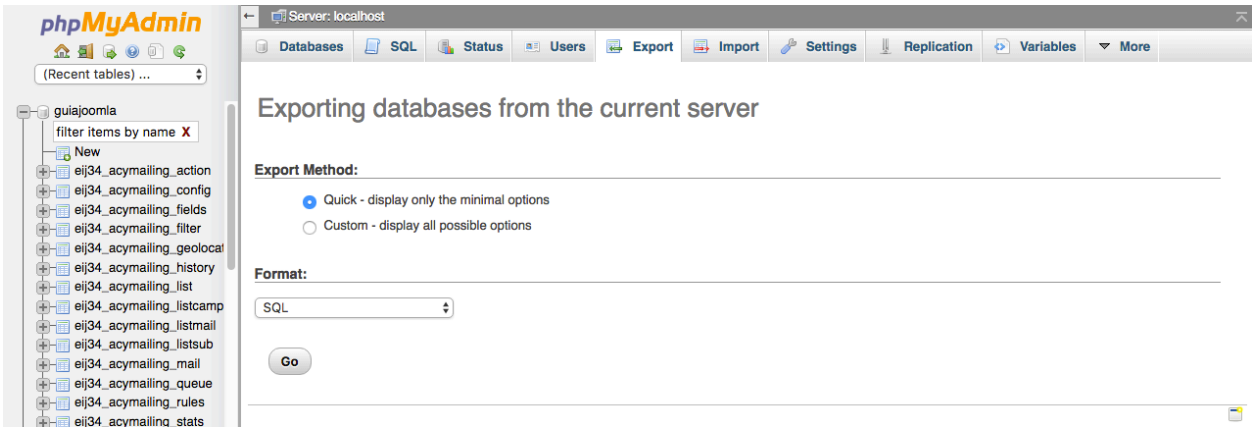
CYBERDUCK es uno de los clientes que permite FTP/SFTP/WebDav etc.

256. **Copia de seguridad de la base de datos con phpMyAdmin:**

Accederemos a nuestra instancia personal de phpMyAdmin, y seleccionaremos de la lista que se encuentra a la izquierda la base de datos que deseamos exportar, en nuestro caso "guiajoomla"

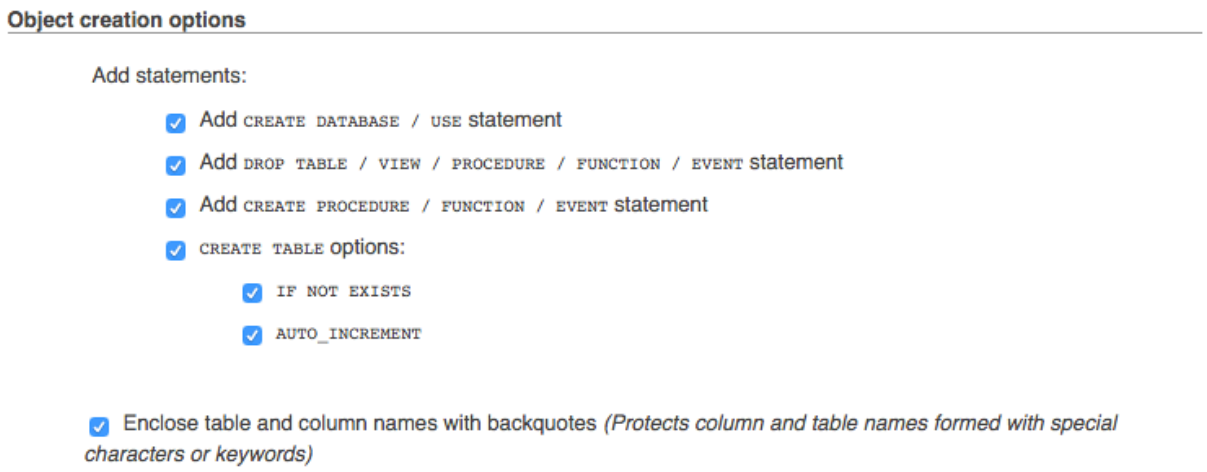


A continuación pulsamos sobre la pestaña "Export":



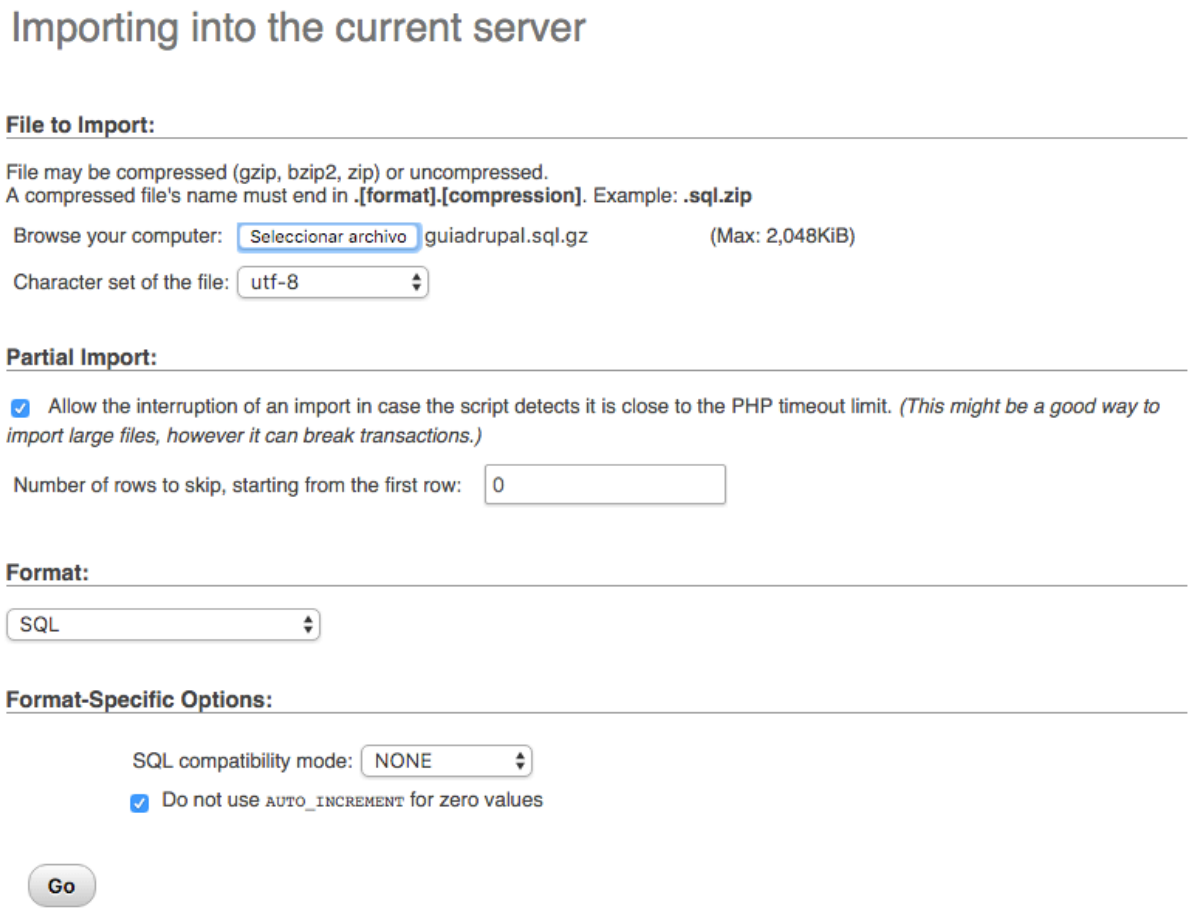
Una vez en la nueva pantalla, seleccionaremos el método "Custom" y pulsaremos sobre las siguientes opciones:

- *Tables*: seleccionamos todas.
- *Output/Compression*: gzipped.
- *Format*: SQL.
- *Objetct creation options*: seleccionamos todas.

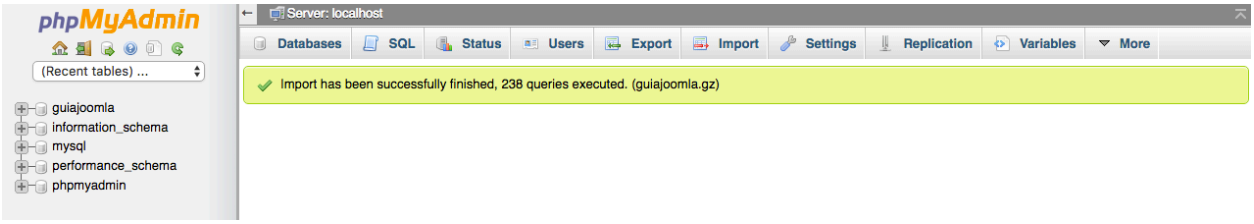


Pulsamos sobre el botón "Go", y a través del navegador, nos descargará el fichero generado.

Para restaurar el backup, sólo tendremos que copiar los ficheros del directorio de Joomla! de nuevo en la ruta especificada, utilizando el mismo sistema (FTP, SCP etc.), y accederemos a phpMyAdmin y pulsaremos sobre la pestaña "Import":



Seleccionamos el fichero que contiene la copia de la base de datos, y pulsamos sobre el botón "Go". Una vez realizado, el sistema nos mostrará si la restauración de la base de datos ha sido o no satisfactoria:



22.2 BACKUP VIA LÍNEA DE COMANDOS

257. Copia de los ficheros del sistema

De la misma forma que en el método anterior, podemos copiar los ficheros a un equipo personal u otro servidor utilizando protocolos como FTP, SCP etc. Otra opción es copiarlos a una ruta diferente o generar un archivo comprimido que pondremos a salvo:

$ cp -rp /ruta/de/joomla/ruta/del/backup
o
$ tar cvf joomlabackup.tgz /ruta/de/joomla

258. **Copia de seguridad de la base de datos**

Utilizaremos la herramienta "mysqldump", que nos permitirá generar una copia de la base de datos con el siguiente comando:

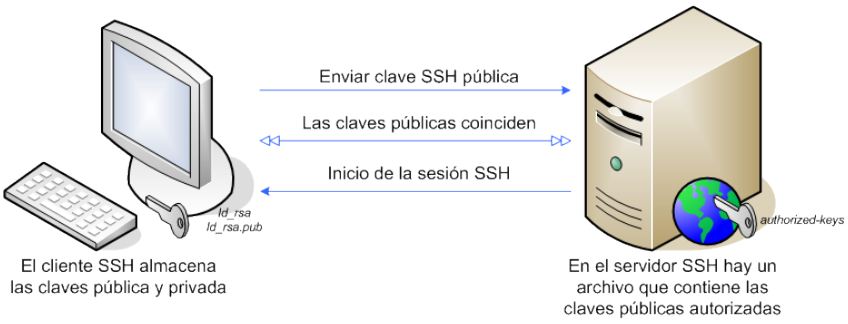
$ mysqldump -u NOMBRE_USUARIO -p'PASSWORD' nombre_bbdd > /ruta/del/backup/backup_ddbb.sql

La restauración del backup se realizará a través del comando:

$ mysql -u NOMBRE_USUARIO -p'PASSWORD' nombre_bbdd < /ruta/del/backup/backup_ddbb.sql

22.3 SCRIPT DE AUTOMATIZACIÓN DE COPIA DE SEGURIDAD

259. Para realizar este tipo de copia de seguridad, utilizaremos una conexión segura SSH a nuestro servidor remoto, sin utilizar contraseña y en su lugar utilizando una clave pública generada por el servidor.



260. Para ello, primero tendremos que realizar una serie de pasos sencillos:

1. Asegurarnos que el servidor remoto de respaldo tiene SSH instalado, y la cuenta de usuario que vamos a utilizar dispone del directorio ".ssh" en su interior.
2. Crearemos el par de clave privada y pública en nuestro servidor con el comando "ssh-keygen":

```
root@guiaJoomla:~# ssh-keygen -b 4096 -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
ab:e2:7c:15:f3:c7:96:7b:6f:a7:17:11:01:f0:f0:62 root@guiaJoomla
The key's randomart image is:
+--[ RSA 4096]----+
|           o.....|
|            +  . |
|           E o  .|
|        o . .  . |
|        S+ . .  .|
|        ... =  . |
|       ..  o .  .|
|   .. ..    . . +|
|   .oo.      ..=o|
+-----------------+
root@guiaJoomla:~#
```

3. Copiamos la clave pública a nuestro equipo de respaldo con el comando "ssh-copy-id", que introducirá nuestra clave en el fichero "authorized_keys" remoto:

```
root@guiaJoomla:~# ssh-copy-id -i .ssh/id_rsa.pub joomlabackup@          0.69
The authenticity of host '139.162.200.69 (139.162.200.69)' can't be established.
ECDSA key fingerprint is e3:8b:bc:22:d4:ec:2d:0b:8c:7f:8c:6f:fd:41:b2:74.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
joomlabackup@139.162.200.69's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'joomlabackup@139.162.200.69'"
and check to make sure that only the key(s) you wanted were added.
```

4. Comprobamos que podemos realizar la conexión sin contraseña:

```
root@guiaJoomla:~# ssh joomlabackup@          0.69
Linux (none) 4.4.0-x86_64-linode63 #2 SMP Tue Jan 19 12:43:53 EST 2016 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
joomlabackup@(none):~$
```

I. REALIZANDO LA COPIA DE SEGURIDAD

261. Una vez que tengamos esta configuración realizada, procederemos a instalar el script para la realización de copias de seguridad en nuestro servidor. Necesitaremos modificar una serie de variables iniciales para adaptarlo a nuestra configuración, como:

- **BLOG_DIR**: directorio donde Joomla! se encuentra instalado.
- **BACKUP_DIR**: directorio local donde se almacenará el backup.
- **REMOTE_USER**: usuario remoto que se usará en la conexión SSH.
- **REMOTE_HOST**: dirección de la máquina a la que nos conectaremos por SSH.
- **REMOTE_BACKUP_DIR**: directorio remoto donde almacenaremos el backup.

262. El script que crearemos en nuestra máquina contendrá el siguiente código:

```
#!/bin/bash
#
# Script para la realizacion de copias de seguridad en Joomla! por SSH
#
# Ruta de la instalacion de Joomla!
BLOG_DIR=/var/www/html/joomla-cms/
# Directorio donde almacenar las copias en local
BACKUP_DIR=/var/backups/joomla
# Datos de conexion remota por SSH/SCP
REMOTE_USER=joomlabackup
REMOTE_HOST=XXX.XXX.XX.XX
# Directorio remoto donde almacenar las copias de seguridad
REMOTE_BACKUP_DIR=/backups/joomla

# Variables de configuracion de acceso a la base de datos de Joomla!
DB_NAME=guiajoomla
DB_USER=usuariojoomla
DB_PASS=password
DB_HOST=127.0.0.1

echo "Realizando volcado de la base de datos... "
DUMP_NAME=${DB_NAME}-$(date +%Y%m%d).sql.bz2
mysqldump --user=${DB_USER} --password=${DB_PASS} --host=${DB_HOST} --databases ${DB_NAME} |
bzip2 -c > ${BACKUP_DIR}/${DUMP_NAME}
if [ "$?" -ne "0" ]; then
    echo "error!"
    exit 1
fi

echo "Realizando copia de seguridad de los ficheros de Joomla... "
TAR_NAME=${BLOG_DIR##*/}-$(date +%Y%m%d).tar.bz2
tar -cjf ${BACKUP_DIR}/${BLOG_DIR##*/}-$(date +%Y%m%d).tar.bz2 --exclude cache ${BLOG_DIR}
if [ "$?" -ne "0" ]; then
    echo "error!"
    exit 2
fi


echo "Enviando copia de seguridad de BBDD/ficheros a host remoto... "
scp "${BACKUP_DIR}/${DUMP_NAME}" "$REMOTE_USER@$REMOTE_HOST:$REMOTE_BACKUP_DIR"
scp "${BACKUP_DIR}/${TAR_NAME}" "$REMOTE_USER@$REMOTE_HOST:$REMOTE_BACKUP_DIR"

if [ "$?" -ne "0" ]; then
    echo "Se ha producido un error!"
    exit 3
fi
echo "Copia de seguridad finalizada!"
```

263. Una vez creado en nuestro sistema, su ejecución es muy sencilla. Tan sólo deberemos otorgarle permisos de ejecución y lanzarlo desde la línea de consola:

```
root@guiaJoomla:~# ./backup.sh
Realizando volcado de la base de datos...
Realizando copia de seguridad de los ficheros de Joomla!...
tar: Eliminando la `/' inicial de los nombres
Enviando copia de seguridad de BBDD/ficheros a host remoto...
guiajoomla-20160525.sql.bz2                                   100%   31KB  30.7KB/s   00:00
joomla-20160525.tar.bz2                                       100%  160MB   1.2MB/s   02:14
Copia de seguridad finalizada!
root@guiaJoomla:~#
```

264. Ahora que tenemos la certeza de que el script está funcionando correctamente, podremos programarlo para realizar copias de seguridad diarias de forma automática y sin necesidad de la interacción del administrador del sistema. Para ello sólo tendremos que añadir una nueva entrada en el "crontab" del sistema, especificando la hora a la que debe realizarse la copia de seguridad. En nuestro ejemplo configuraremos la tarea para que se realice de forma automática todos los días a las 2.30 AM:

```
$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
30 2 * * * /path_del_script/backup.sh > /dev/null 2>&1
$
```

II. RESTAURANDO LA COPIA DE SEGURIDAD

265. El proceso para restaurar la copia de seguridad será el siguiente:

1. Descargar los ficheros correspondientes a la copia de seguridad de la base de datos y de los ficheros de Joomla! a nuestro servidor.
2. Importamos el backup en formato *.sql* a MySQL y comprobamos que se restaura correctamente.

```
root@guiaJoomla:/var/backups/joomla# ls
guiajoomla-20160525.sql.bz2  joomla-20160525.tar.bz2
root@guiaJoomla:/var/backups/joomla# bunzip2 guiajoomla-20160525.sql.bz2; mysql -u usuariojoomla -p < guiajoomla-20160525.sql
Enter password:
root@guiaJoomla:/var/backups/joomla# mysql -u usuariojoomla -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 247
Server version: 5.5.44-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| guiajoomla         |
+--------------------+
2 rows in set (0.00 sec)

mysql>
```

3. Descomprimimos el archivo correspondiente a la copia de seguridad de los ficheros. Es importante saber que se expandirá la ruta completa donde el anterior Joomla! fue instalado, por lo que será importante mantenerla. En caso que los directorios correspondientes ya existan, procederemos a mover el directorio de la instalación a la ruta completa.

```
root@guiaDrupal:/var/backups/drupal# mv drupal-20160518.tar.bz2 /var/www/html/drupal/
root@guiaDrupal:/var/backups/drupal# cd /var/www/html/drupal/
root@guiaDrupal:/var/www/html/drupal# tar -xjf drupal-20160518.tar.bz2
root@guiaDrupal:/var/www/html/drupal# ls
autoload.php    composer.lock  dbcontent.sql           example.gitignore  modules    README.txt  sites    update.php  vendor
composer.json   core           drupal-20160518.tar.bz2 index.php          profiles   robots.txt  themes   var         web.config
```

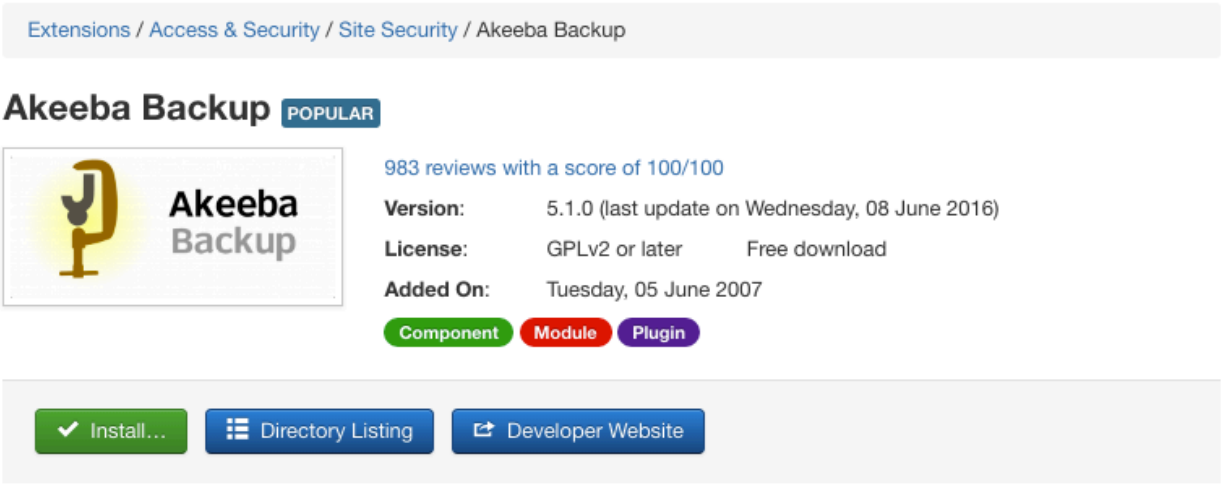
4. Comprobamos que los ficheros tienen los permisos correspondientes. La información detallada de este proceso se encuentra en la sección “Configuración de permisos”.

266. Una vez finalizado este paso, nuestra copia de seguridad habrá sido restaurada correctamente y podremos acceder de nuevo a nuestro Joomla! de la forma habitual.

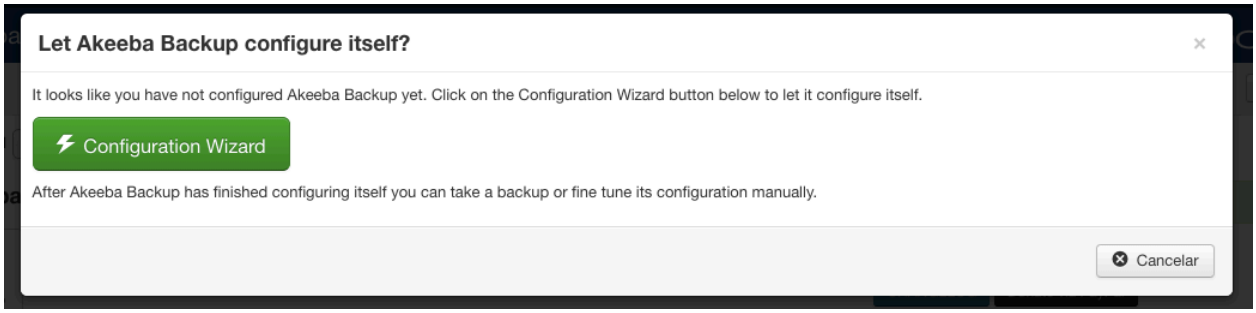
22.4 AKEEBA BACKUP

267. Esta extensión permite generar copias de seguridad de nuestro sitio a través del BackEnd. Además, generará un fichero de instalación similar al original de Joomla! que permitirá recuperar o transferir nuestro sitio de forma sencilla.

268. Procederemos a la instalación de esta extensión desde el propio gestor:



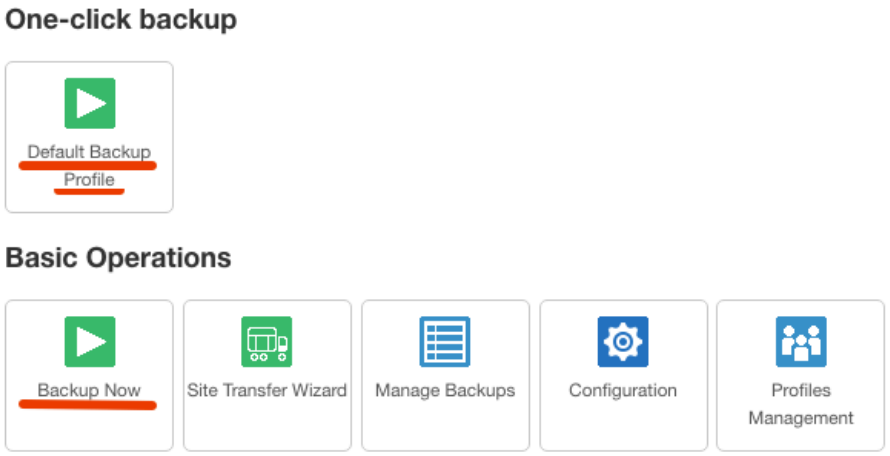
269. Accederemos a la configuración desde el menú "Extensiones"->"Akeeba Backup", en primer lugar encontraremos un asistente de configuración que procederemos a configurar pulsando sobre el botón "Configuration Wizard":



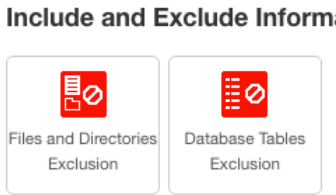
270. Una vez se hayan realizado las diferentes pruebas de rendimiento para adaptar a la realización de las copias de seguridad a los recursos de nuestro servidor, podremos elegir entre diferentes opciones del menú principal:

22.4.1. GENERACIÓN DE COPIAS DE SEGURIDAD

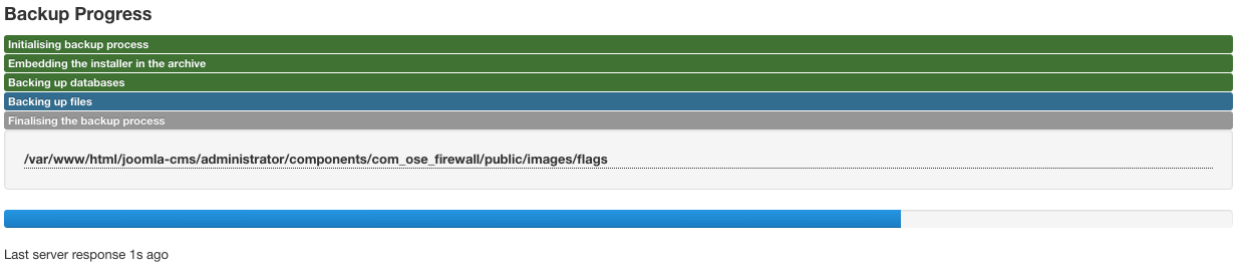
271. Podremos realizar backups desde dos accesos del menú principal de la extensión: "Default Backup Profile" y "Backup Now":



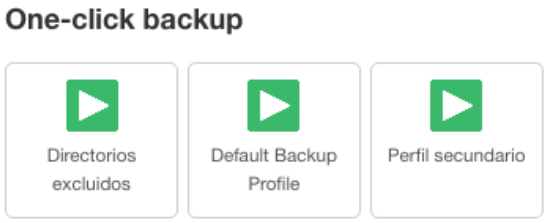
272. La principal diferencia entre ambos es el uso de perfiles. Cuando se utiliza el asistente de configuración, se creará un perfil por defecto (Default Backup Profile) que contendrá los valores óptimos en base a nuestra instalación y recursos. Pueden generarse diferentes perfiles, incluyendo en cada uno los ficheros/directorios o tablas de la base de datos que deseamos excluir de los backups, utilizando los botones inferiores:



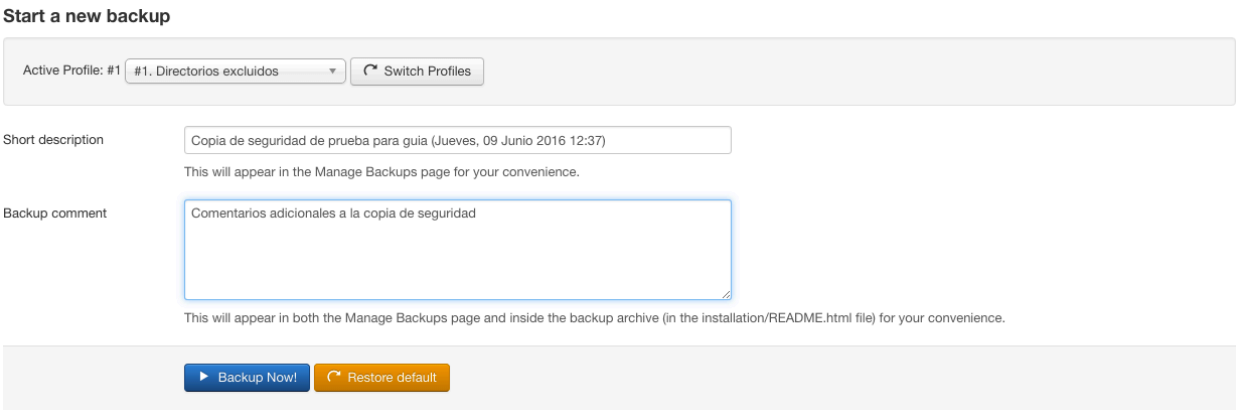
273. Una vez dispongamos de estos perfiles, si pulsamos sobre el botón "Default Backup Profile" automáticamente se creará un backup con el perfil, nombre y ubicación por defecto:



274. Según generemos perfiles nuevos de backup, aparecerán en la pantalla de inicio para realizar las copias con tan sólo una pulsación sobre el icono correspondiente:



275. Si deseamos realizar una copia de seguridad más personalizada, pulsaremos sobre "Backup Now", y accederemos al siguiente menú:



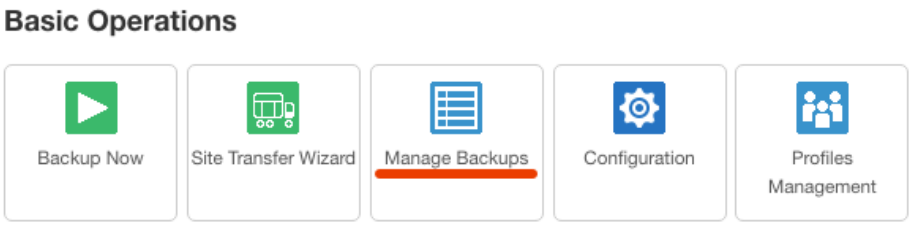
276. Desde la nueva pantalla deberemos especificar:

- Active Profile: el perfil y configuración que se utilizará en la copia de seguridad.
- Short description: nombre corto asociado a la copia.
- Backup comment: podremos introducir comentarios adicionales o cualquier información relevante que será asociada al fichero generado.

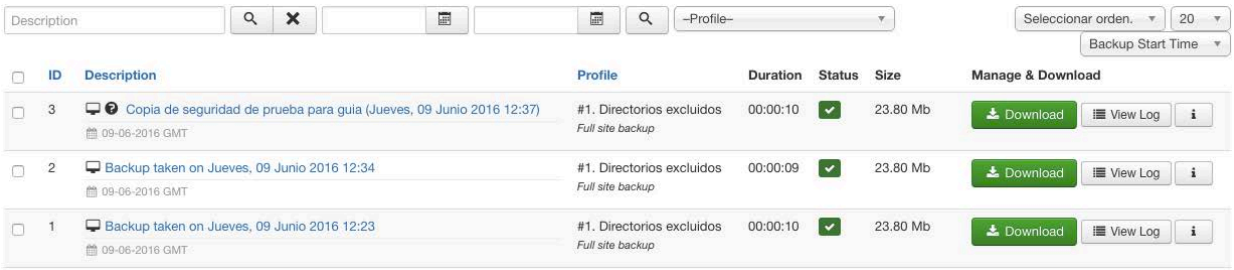
277. Una vez finalizado, pulsaremos sobre el botón "Backup Now!".

22.4.2. GESTIÓN DE COPIAS DE SEGURIDAD

278. La gestión de copias de seguridad puede realizarse pulsando sobre el botón "Manage Backups":



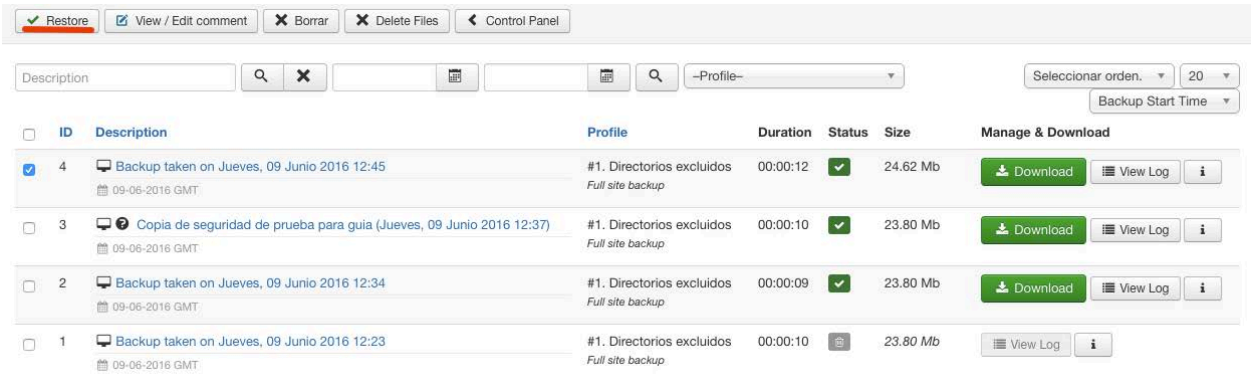
279. Se mostrará un listado completo de las copias de seguridad realizadas por la extensión, pudiendo realizar búsquedas por la descripción de esta, fecha de generación, perfil utilizado etc:



280. Para descargar una copia de seguridad concreta, pulsaremos sobre el botón "Download" asociado, pulsaremos el botón de información "i", que nos proporcionará el nombre de fichero y la ruta donde se encuentra ubicado, para poder acceder a él desde la línea de comandos o cualquier programa de transmisión de ficheros:

22.4.3. RECUPERACIÓN DE COPIAS DE SEGURIDAD

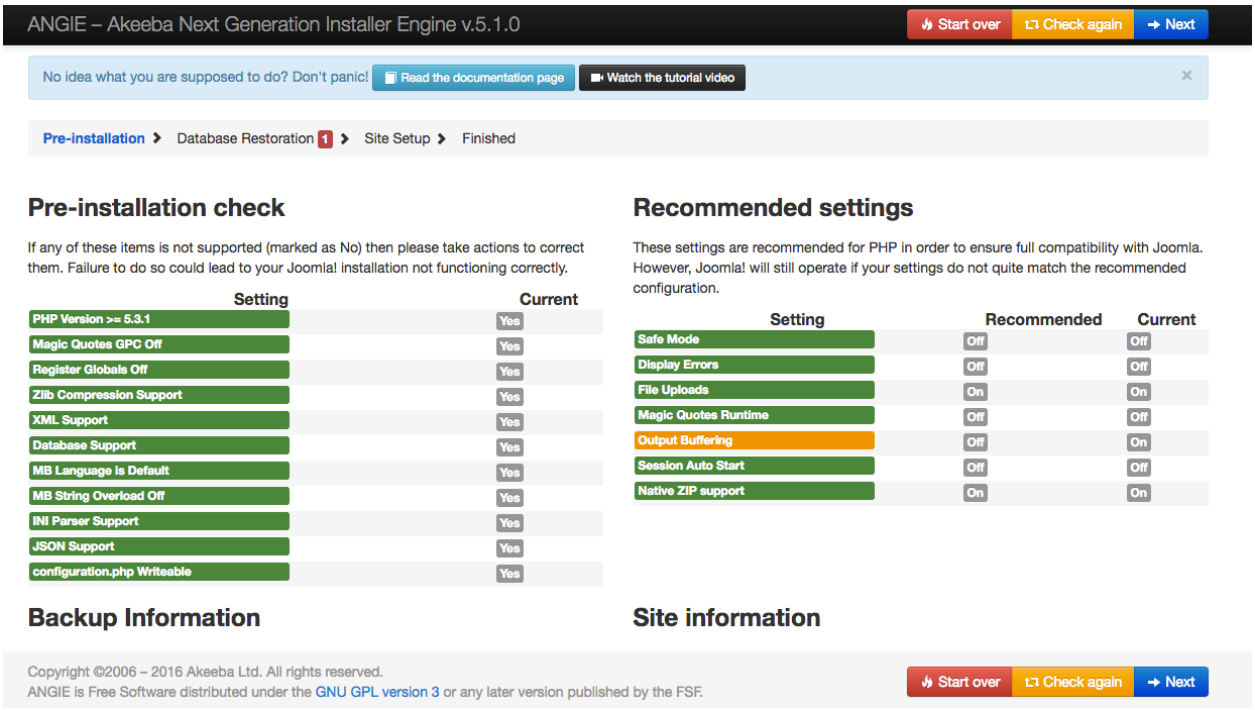
281. La recuperación de una copia de seguridad se realiza de forma sencilla desde el propio gestor de la extensión. Seleccionaremos la copia que deseamos restaurar, y pulsaremos sobre el botón "Restore":



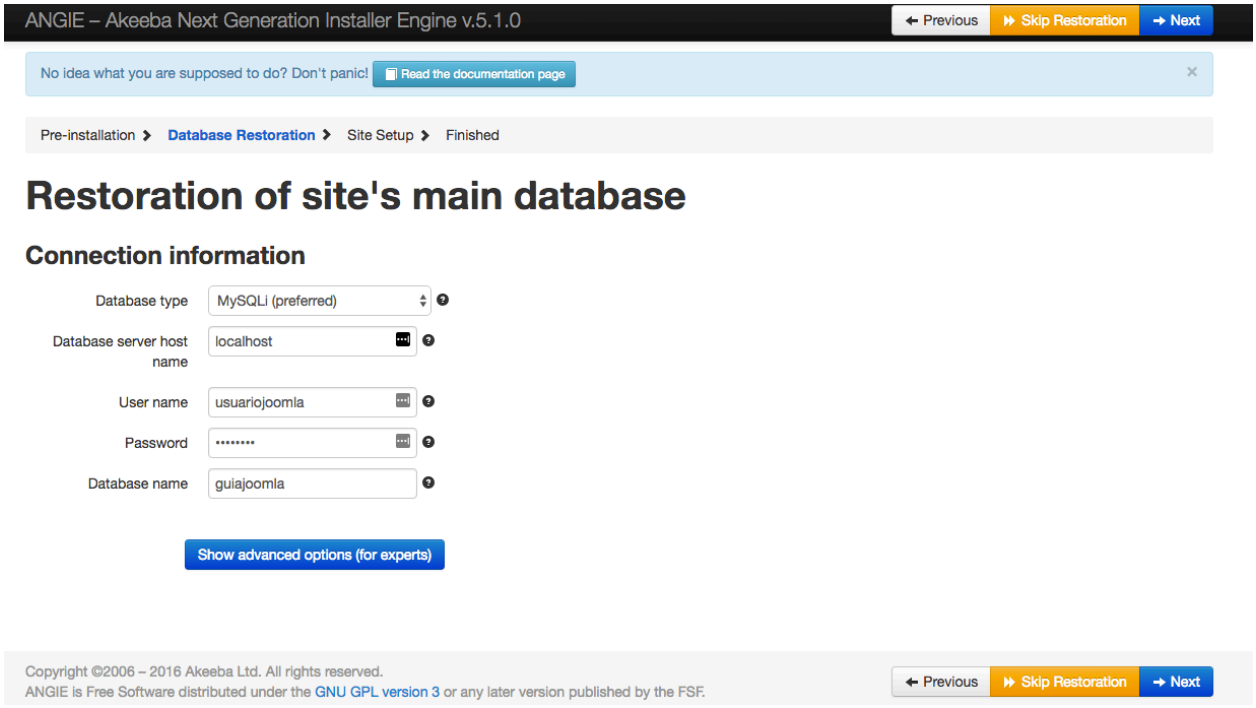
282. En el siguiente menú sólo deberemos especificar el método de restauración. Esta restauración puede ser tanto local como en una máquina remota que queremos replicar nuestra instalación para, por ejemplo, tener un entorno de preproducción donde realizar pruebas.

283. Si deseamos restaurar la copia en nuestra máquina local, pulsaremos sobre "Direct". En caso de querer restaurar una copia de seguridad de una máquina remota, podremos utilizar la opción de FTP configurando todos los parámetros de acceso. Accederemos a una pantalla que nos dará acceso al script de restauración, tanto local como remoto si hemos utilizado otra máquina:

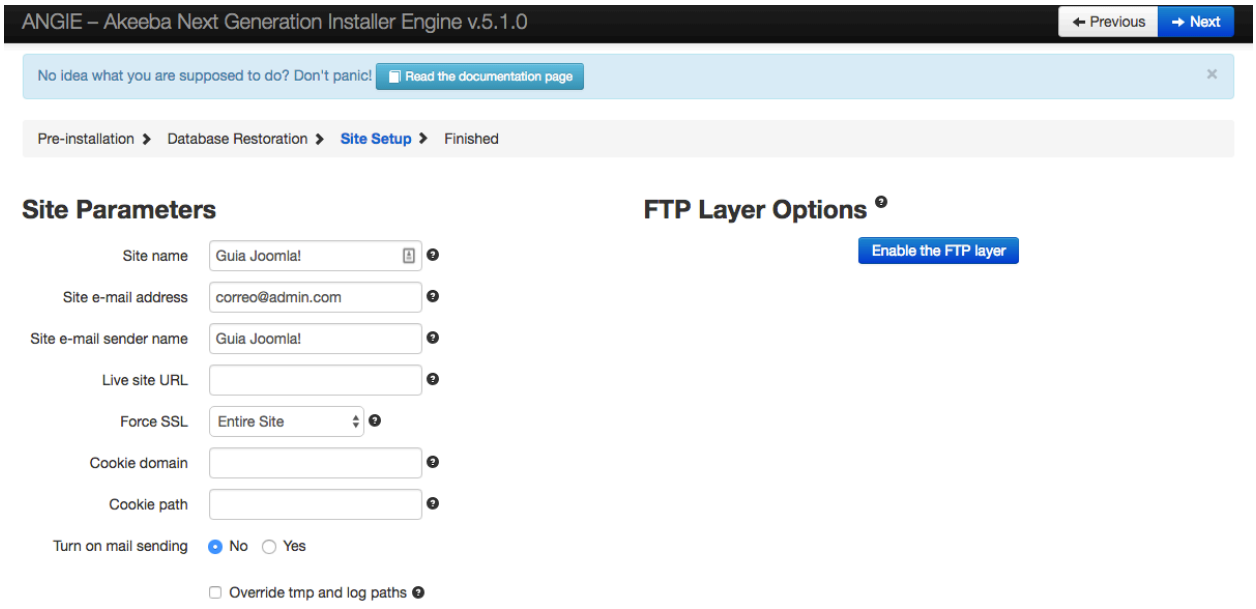
284. Se abrirá una nueva pantalla, que nos mostrará si la máquina donde será restaurada la copia de seguridad cumple los requerimientos:



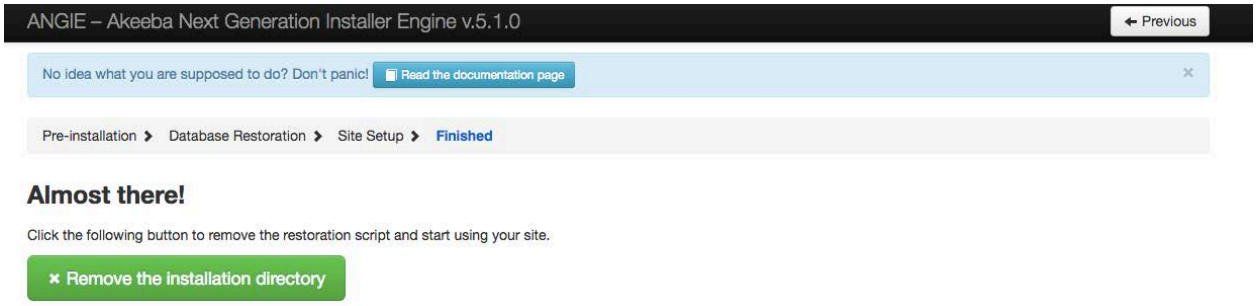
285. Pulsaremos en "Next", si las dependencias son correctas, para acceder a la configuración de la base de datos:



286. Si deseamos realizar una restauración de la base de datos pulsaremos sobre "Next", o en caso contrario pulsaremos sobre "Skip Restoration". Finalmente encontraremos los parámetros de configuración del sitio de Joomla!, como si del instalador original se tratara:



287. Una vez en el último paso, la restauración del sitio se habrá realizado con éxito, por lo que pulsaremos sobre el botón "Remove the installation directory" para eliminar todos los datos asociados con el script de recuperación de la copia de seguridad:



288. En el caso que deseemos restaurar una copia de seguridad en una máquina remota donde no dispongamos de acceso FTP remoto, utilizaremos el script de Akeeba Kickstart[15] para realizar la carga de forma local dentro de la máquina:

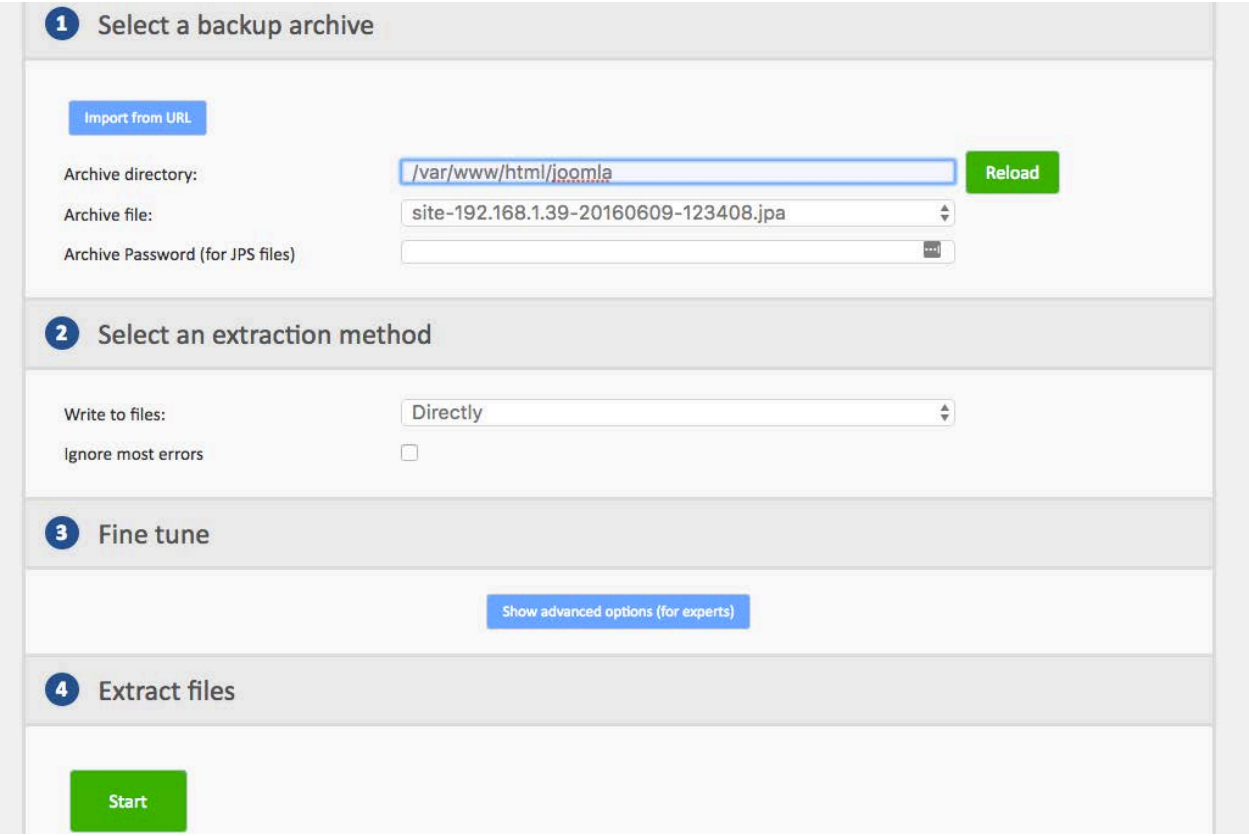
```
root@guiadrupal:/var/www/html/joomla# wget https://www.akeebabackup.com/download/akeeba-kickstart/4-2-1/kickstart-core-4-2-1-zip.zip
--2016-06-09 15:10:39--  https://www.akeebabackup.com/download/akeeba-kickstart/4-2-1/kickstart-core-4-2-1-zip.zip
Resolviendo www.akeebabackup.com (www.akeebabackup.com)... 107.6.162.122
Conectando con www.akeebabackup.com (www.akeebabackup.com)[107.6.162.122]:443... conectado.
Petición HTTP enviada, esperando respuesta... 303 See other
Ubicación: http://cdn.akeebabackup.com/downloads/kickstart/4.2.1/kickstart-core-4.2.1.zip [siguiente]
--2016-06-09 15:10:40--  http://cdn.akeebabackup.com/downloads/kickstart/4.2.1/kickstart-core-4.2.1.zip
Resolviendo cdn.akeebabackup.com (cdn.akeebabackup.com)... 54.230.62.229, 54.230.62.149, 54.230.62.51, ...
Conectando con cdn.akeebabackup.com (cdn.akeebabackup.com)[54.230.62.229]:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 278494 (272K) [application/zip]
Grabando a: "kickstart-core-4-2-1-zip.zip"

100%[=======================================================================================>] 278.494     1,06MB/s   en 0,2s

2016-06-09 15:10:40 (1,06 MB/s) - "kickstart-core-4-2-1-zip.zip" guardado [278494/278494]
```

289. Una vez descomprimido el fichero dentro de la ruta adecuada de nuestro servidor web, cargaremos por SCP o cualquier otro método el fichero con la copia de seguridad realizada, dentro del mismo directorio donde hemos descargado el script. Ahora accederemos a la URL de nuestro sitio http://nuestrositio/kickstart.php:

[15] https://www.akeebabackup.com/products/akeeba-kickstart.html



290. Comprobaremos que la ruta indicada es la correcta y que se ha detectado el fichero de backup que deseamos restaurar, e iniciamos el proceso. Una vez finalizado, podremos acceder al mismo instalador que hemos visto anteriormente para finalizar la instalación de la copia de seguridad, pulsando sobre "Run the Installer":

23. RECUPERACIÓN ANTE UN COMPROMISO DE SEGURIDAD

291. A pesar de las medidas anteriormente descritas, es posible que en algún momento suframos un compromiso en nuestro sistema, porque un atacante descubra una vulnerabilidad, o más habitualmente porque algún *bot*/*gusano* acabe tomando el control de forma automatizada de nuestro sitio.

292. En este apartado veremos algunos de los pasos que serán necesarios para mitigar el impacto de este compromiso y restaurar el servicio:

1. **Realizar una copia de seguridad con fines forenses**: a pesar de que nuestro sitio haya sido infectado, es importante realizar una copia de seguridad de los datos. Esta copia debe ser completa, incluyendo toda la información de la base de datos y ficheros en el sistema. Si fuera posible, realizaremos un snapshot de los servidores que hayan sido comprometidos.
2. **Restaurar el servicio**: a la hora de restaurar el servicio, lo más recomendable es realizar una instalación limpia del sistema, así como la distribución desde el sitio oficial. Si conocemos con exactitud la fecha del compromiso, procederemos a restaurar, si existe, una copia de seguridad de la base de datos y los ficheros necesarios. Solamente en casos concretos, seremos capaces de asegurar que ficheros han sido comprometidos, restaurarlos, asegurar el sitio y mantener los datos almacenados.
3. **Modificar la contraseña de acceso a la base de datos**: debido a que en el momento inicial no conoceremos el alcance del ataque, será necesario generar un nuevo usuario y contraseña para gestionar la base de datos de MySQL. En el caso de que hayamos restaurado una copia de seguridad anterior, entraremos en la consola de administración de la base de datos y procederemos al cambio de contraseña:

   mysql> SET PASSWORD FOR 'usuario'@'' = PASSWORD('nueva_contraseña');*

   Deberemos modificar esta contraseña también en el fichero de configuración de Joomla! "/ruta/web/joomla/configuration.php".

   public $password = 'nueva_contraseña'
4. **Revisión de los usuarios**: deberemos comprobar los roles, así como los usuarios que se encuentran activos en la plataforma. Es posible que durante el proceso del ataque, se haya creado un usuario nuevo con permisos avanzados para mantener el acceso a la máquina comprometida. Revisaremos la información almacenada en la tabla correspondiente (el prefijo será diferente en cada instalación al ser generado de forma aleatoria en el momento de instalación):

```
mysql> select username,email,name from eij34_users;
+---------------+------------------------+------------+
| username      | email                  | name       |
+---------------+------------------------+------------+
| adminguia     | correo@admin.com       | Super User |
| usuarioprueba | usuarioprueba@localhost | Prueba     |
+---------------+------------------------+------------+
2 rows in set (0.00 sec)
```

5. **Reiniciar las contraseñas de todos los usuarios**: En el caso de utilizar una instalación nueva, las contraseñas de los usuarios deben ser diferentes a las anteriores, debido a que han podido ser obtenidas durante el ataque. En el caso de estar utilizando una copia de seguridad de respaldo, procederemos a utilizar la siguiente *query* desde la consola de administración de MySQL para modificarlas, sobre todo las referentes a super usuarios del sistema:

   mysql> UPDATE `prefijotabla_users` SET `password` = MD5('new_password') WHERE `prefijotabla _users`.`username` = "usuario_afectado";
6. **Revisión y reinstalación de módulos**: ahora procederemos a localizar los módulos que estaban anteriormente instalados y los reinstalaremos del repositorio oficial de Joomla! de nuevo.

7. **Google Webmaster Tools**: las herramientas de Webmaster de Google nos permitirán comprobar si nuestro sitio ha sido marcado como infectado, además de obtener información valiosa de los ficheros sospechosos. Si hemos sido incluidos en la lista de sitios infectados, una vez finalizada la limpieza, deberemos solicitar la revisión de nuestro sitio.
8. **Asegurar el sitio de nuevo**: una vez finalizado el proceso anterior, procederemos de nuevo a aplicar las medidas de seguridad explicadas a lo largo de esta guía.

24. CHECKLIST DE REVISIÓN DE SEGURIDAD

TAREAS A REALIZAR	REALIZADA	NO REALIZADA	NO APLICA
MySQL			
Restricción de acceso remoto	☐	☐	☐
Modificación del usuario administrador	☐	☐	☐
Eliminado de cuenta de invitados y obsoletas	☐	☐	☐
Reducción de privilegios del sistema	☐	☐	☐
Activación de logs	☐	☐	☐
Eliminado de historial	☐	☐	☐
Limitación de los permisos del usuario de Joomla	☐	☐	☐
PHP			
Instalación mod_php	☐	☐	☐
Instalación php-fpm	☐	☐	☐
Instalación de Suhosin	☐	☐	☐
Configuración segura de variables	☐	☐	☐
GENERAL			
Eliminación de ficheros innecesarios de instalación	☐	☐	☐
Configuración segura de permisos	☐	☐	☐
Revisión del fichero 'robots.txt'	☐	☐	☐
Restricción de acceso al BackEnd	☐	☐	☐
Incremento de seguridad mediante .htaccess	☐	☐	☐
Eliminación de información expuesta del CMS	☐	☐	☐
Eliminación de información expuesta de versión	☐	☐	☐
Gestión segura de usuarios y accesos	☐	☐	☐
Fortaleza y creación de política de contraseñas	☐	☐	☐
Activación de la navegación sobre SSL	☐	☐	☐
Uso de autenticación de doble factor	☐	☐	☐
Habilitación de reCAPTCHA	☐	☐	☐
Integración de feeds de seguridad al BackEnd	☐	☐	☐

MODULOS			
Akeeba Admin Tools	☐	☐	☐
Jsecure Lite	☐	☐	☐
Brute Force Stop	☐	☐	☐
BACKUPS			
Generación manual de copias de seguridad	☐	☐	☐
Adecuación y uso del script de automatización	☐	☐	☐
Instalación y configuración de Akeeba Backup	☐	☐	☐