GUÍA DE SEGURIDAD DE LAS TIC (CCN-STIC-460)

SEGURIDAD EN WORDPRESS

JULIO 2015

Edita:

NIPO: 002-15-017-9

Fecha de Edición: julio 2015

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.

Julio 2015

Félix Sanz Roldán
Secretario de Estado
Director del Centro Criptológico Nacional

ÍNDICE

ANEXOS

1. INTRODUCCIÓN

1. WordPress es un sistema de gestión de contenido, o CMS por sus siglas en inglés (Content Management System) enfocado, principalmente, a la creación de blogs. Ha sido desarrollado por Automattic bajo licencia GPL, y sobre PHP para entornos que ejecuten MySQL y Apache.
2. WordPress fue creado a partir del desaparecido b2/cafelog y se ha convertido, junto a Movable Type, en el CMS más popular. Las causas de su enorme crecimiento son, entre otras, su licencia, su facilidad de uso y sus características como gestor de contenidos.
3. Otro motivo a considerar sobre su éxito y extensión es la enorme comunidad de desarrolladores y diseñadores, encargados de desarrollarlo en general o crear complementos y temas para la comunidad. En agosto de 2013 era usado por el 18,9 % de todos los sitios existentes en Internet.

1.1. HISTÓRICO DE VERSIONES

4. **Rama 1.x**

   La primera versión final de WordPress se lanzó oficialmente el 3 de enero de 2004, y se le llamó «Miles», por el músico de jazz Miles Davis.

5. **Rama 2.x**

   Desde el lanzamiento de WordPress 2.1, se empezó a usar la versión 4.1 de MySQL, mientras que WordPress 2.0 usa MySQL 3. Con las nuevas versiones 2.x, el equipo de WordPress analizó los servicios de alojamiento web vigentes y concretaron que todavía muchos de estos servicios no admitían MySQL 4. Así, se decidió seguir dando soporte de WordPress 2.0 (actualizaciones de seguridad) hasta 2010, cuando se esperaban que todos los servicios de alojamiento web comenzaran a ofrecer MySQL 4 y 5.

   Además, está la obsolescencia de PHP4 en 2008, lo que provoca que las nuevas versiones de Wordpress se diseñaran basándose en la versión 5, aunque mantenían la compatibilidad inversa (y el soporte de la rama 2.0) en iguales circunstancias que con MySQL.

   - **2.0** (Duke), la versión de Wordpress, denominada «Duke» por el músico Duke Ellington y lanzada el 31 de diciembre de 2005, fue (después de la versión 2.5) realmente el desarrollo más innovador y con más cambios, o al menos de más impacto, hasta esa fecha. Incluía el editor WYSIWIG «TinyMCE», la subida de adjuntos e imágenes, gestión de roles o perfiles de usuarios, caché persistente de contenidos, soporte de diferentes versiones para la base de datos y copia de seguridad de esta, el complemento antispam Akismet (también de Automattic), vista previa de entradas y algunas funciones AJAX, entre otras.15
   - **2.0.12** La rama 2.0.x adquiere soporte oficial, esta versión incluye arreglos de seguridad, mejoras y estética para la rama 2.0.
   - **2.2** (Getz), versión de Wordpress en honor al saxofonista Stan Getz. Esta versión fue lanzada el día 16 de mayo del 2007 e incluye 200 actualizaciones de errores. Su funcionalidad más notable añadida es la integración de *widgets*.
   - **2.3 (Dexter)**, en honor al saxofonista Dexter Gordon, fue lanzada oficialmente el 24 de septiembre de 2007.

- **2.4**, versión cancelada en enero de 2008 para pasar a la versión 2.5. Se canceló por diversos motivos, pero principalmente por los fallos y retraso en el demasiado rápido desarrollo del nuevo panel (dashboard) y que fue reescrito para la versión 2.5.
- **2.5 (Brecker)**, llamada «Brecker» en honor al saxofonista Michael Brecker y que salió oficialmente el 29 de marzo de 2008. Esta versión fue un nuevo punto de inflexión en el desarrollo de Wordpress con especial énfasis en la mejora del panel de administración de Wordpress. Se mejoró el aspecto visual y la usabilidad de la administración, en especial añadiendo funciones AJAX que, en general, mejoraban todo el tablero y en concreto: la gestión de adjuntos y archivos multimedia, de etiquetas, categorías y enlaces, guardado automático de borradores por tiempo, sistema automático de actualización de *plugins*, mejora del sistema de *widgets* para plantillas... También fue importante el añadido de la API para «Shortcode» que permite ejecutar código en las entradas sin el uso directo de incrustación PHP, soporte para el servicio Gravatar en comentarios y generación automática de galerías de imágenes sobre los adjuntos.
    - **2.5.1** versión de mejora en seguridad y corrección de bugs lanzada el 25 de abril de 2008.
- **2.6 (Tyner)**, llamada «Tyner» en honor al pianista de jazz McCoy Tyner y que es lanzada el 15 de julio de 2008.

    La versión Tyner, introduce (según el blog de desarrollo) nuevas mejoras que potencian el uso de Wordpress como CMS. Por ejemplo la gestión de revisiones y versiones de las entradas (diff) al estilo tradicional de las wikis, la posibilidad de sacar del árbol web directorios y archivos sensibles, soporte completo SSL, nuevas versiones de algunas de sus bibliotecas incluidas tales como jQuery y jQuery UI (1.5.1) o TinyMCE, mejora y añadido de *plugins* como «Wordpress Video», pre visualización de plantillas en administración, campos extra para perfiles de usuario (hasta ahora mediante *plugins*) o la posibilidad de establecer varias sub-categorías, entre otras nuevas funciones y, según la noticia oficial, correcciones sobre 194 errores.

    - **2.6.2** Parche dirigido prácticamente en exclusiva a solucionar vulnerabilidades en el registro de usuarios, relacionadas con la generación aleatoria de contraseñas y el posible reseteo de las mismas para otros usuarios. Afecta por lo tanto sólo a sistemas con el registro de usuarios habilitado.
    - **2.6.32** Parche dirigido para solucionar una vulnerabilidad de bajo riesgo en la biblioteca Snoopy, la cual se utiliza para mostrar los *feeds* en el Tablero.

- **2.7 (Coltrane)** Llamada «Coltrane», en honor al saxofonista John Coltrane, y que es lanzada el 10 de diciembre del 2008.25 Esta versión incorpora, sobre todo, una nueva y muy esperada interfaz gráfica (1.0) con también nuevas mejoras AJAX, corrección de algunos fallos en la actual y, en especial, el cambio, tanto de ubicación como en relación a la usabilidad, del menú general superior a la parte izquierda de la pantalla. El nuevo menú posibilita el acceso a cualquier sección sin importar la profundidad de las sub-secciones gracias a la implementación de AJAX para hacerlo extensible. Además, otro aspecto muy esperado, es la implementación del sistema de actualización automática para el sistema en general y que está basado, como en otras ocasiones ya se ha hecho, en el *plugin* independiente que ya había. Se acaba de implementar el sistema de cookies sobre HTTP en exclusiva, el añadido de adjuntos sin la necesidad de guardarlos como entradas, explorador de archivos para *plugins*, edición en línea (no desde el tablero), la documentación PHPDoc estará disponible y se incluyen funcionalidades para «arrastrar y soltar» elementos en el Tablero, entre otros. Como es habitual se corrige otro gran número de bugs en relación sobre todo a la interfaz y su funcionalidad respecto a navegadores, localizaciones, etc., así como sobre el código del sistema y validaciones varias.
- **2.7.1** Arregla más de 60 errores menores y que es lanzada el 10 de febrero del 2009.
- **2.8 (Baker)** Llamada «Baker» en honor al trompetista Chet Baker. La versión 2.8 incluye características enfocadas a usuarios avanzados, como el nuevo editor de código CodePress que, a diferencia del anterior editor, resalta el código dependiendo del lenguaje que se está usando. Otra nueva característica que se incluye es la instalación de plantillas vía web, similar a la instalación de *plugins* vía web insertada en la versión 2.7
    - **2.8.6**: Se liberó el 12 de noviembre de 2009.
- **2.9 (Carmen)**: Llamada «Carmen» en honor a la cantante y pianista Carmen McRae. Permite hacer ediciones en imágenes (rotar, voltear, recortar), incrustación de multimedia (sin necesidad de código embeded), así como un sistema de papelera para almacenar los comentarios y artículos eliminados, entre otros.
    - **2.9.1**: Esta revisión soluciona los fallos encontrados en la versión 2.9
    - **2.9.2**: Esta revisión soluciona un fallo de seguridad que permite obtener los resultados de la papelera. Además soluciona otros fallos encontrados en la versión 2.9.1

6. Rama 3.x

- **3.0** Fusiona WordPress con «WordPress Mu» para dar soporte multiblogging por defecto. Los menús son editables y las actualizaciones de *plugins* pueden realizarse en masa.
- **3.1** Esta versión, la número 14, fue lanzada el 22 de febrero de 2011 y se llama Django Reinhardt en honor al Jazzista. Las novedades principales de esta versión son:
    - Enlaces internos: con un clic en el nuevo botón para enlaces internos podrás buscar una entrada o revisar el contenido existente para enlazarlo.

- Barra de admin: contiene varios enlaces para acceder a diversas pantallas de administración. Por defecto, la barra de admin se muestra cuando un usuario ha accedido y está visitando el sitio, y no se muestra en las pantallas de administración en las instalaciones simples (sin multisitio activado). Para las instalaciones con multisitio se muestra tanto cuando estás visitando el sitio como en las pantallas de administración.
- Mejoras en la interfaz de escritura: los nuevos usuarios de WordPress encontrarán la pantalla de escritura mucho más limpia que antes, con la mayoría de las opciones ocultas por defecto (puedes hacer clic en Opciones de pantalla de la parte superior para volverlas a mostrar).
- Formatos de entrada: la información de los formatos pueden usarla los temas para personalizar la presentación de una entrada.
- Administrador de la red: se han movido los menús del Super administrador y las páginas relacionadas de la pantalla de admin habitual a la nueva Pantalla de administrador de la red.
- Pantallas de administración en modo de lista: puedes ordenar las columnas de las pantallas con listados (páginas, entradas, comentarios, etc) para mejorar la paginación.
- Mejoras del exportador/importador: hay muchos cambios en la información del autor, mejora en el manejo de taxonomías y términos, soporte correcto de menús de navegación, etc.
- Mejoras en el tipo de contenido personalizado: permite a los desarrolladores crear páginas de archivo y disponer de más controles de las capacidades y mejores menús.
- Consultas avanzadas: permite a los desarrolladores realizar consultas múltiples de taxonomías y campos personalizados.
- Un esquema de color azul para la administración más fresco que centra la atención en tu contenido.

- **3.2** Esta versión, la número 15, fue lanzada el 4 julio de 2011 y se llama Gershwin en honor al compositor y pianista George Gershwin. Las novedades principales de esta versión son:
  - Nueva interfaz de administración: se ha repintado la interfaz de administración, dando un aspecto más fluido, nuevos iconos, y nuevo diseño de la barra lateral.
  - TinyMCE actualizado: el editor de entradas y páginas ha sido revisado y acomodado dando un aspecto más minimalista.
  - Escritura sin distracciones: nuevos botones, interfaz más minimalista. Para que así al escribir no haya más distracciones. Agregada una opción de Pantalla Completa.
  - Mejoras de velocidad: mejoras en la velocidad y lectura del PHP.
  - Mejoras en la API de listado de tablas: más flexibilidad para uso de la API por parte de terceros.
  - PHP 5.2.4 o superior obligatorio.

- MySQL 5 obligatorio.
- No más soporte para Internet Explorer 6: y da una alerta para que actualices a otro navegador.
- Nuevo tema por defecto: hace su aparición «Twenty Eleven (2011)» como tema predeterminado en la instalación de WordPress. Incluye una imagen de cabecera.
- Página de Crédito: créditos de cada desarrollador de WordPress que haya participado en esta versión.

- **3.2.1** Revisión de problemas de incompatibilidad con JSON y ajustes en el Escritorio y el Twenty Eleven.
- **3.3** Esta versión, la número 16, fue lanzada el 12 de diciembre de 2011 y se llama Sonny. Las novedades principales de esta versión son:
  - Carga de archivos más rápida: permite arrastrar y soltar para cargar archivos, y se admiten los formatos .7z y .rar.
  - Barra lateral del escritorio: menús flotantes, diseño adaptable a otras interfaces.
  - Bienvenida al escritorio: se te da la bienvenida al WordPress 3.3 y te muestran las novedades.
  - Muestra de novedades: a medida que se edita algo salen pequeñas burbujas de texto con la información que fue editada en la versión.
  - Co-edición mejorada: ahora solo se muestra si de verdad están editando la entrada que tú quieres editar.
  - Importa desde Tumblr.
  - Mejores *widgets*.
  - Actualizado jQuery a la versión 1.7.1: agregado jQuery UI.
  - Más flexibilidad en enlaces permanentes: más libertad a la hora de elegir la estructura de tus enlaces permanentes.
- **3.3.1** Revisión de seguridad lanzada el 4 de enero de 2012, entre los cambios importantes:
  - Límite máximo de 50 MB en multisitios.
  - Solución a wp_print_styles(), que provocaba que hubiese estilos y scripts que se mezclaran en la zona de administración.
  - Ahora se muestra correctamente $userdata.
  - Los usuarios con la capacidad de listar usuarios podían cambiar a un Administrador a Suscriptor.
- **3.5** 11 de diciembre de 2012. Algunas mejoras de esta versión son:
  - Mejora de la librería multimedia.
  - Creación de un asistente para personalizar los temas.

- **3.6** De nombre clave «Oscar» fue lanzada el día 1 de agosto de 2013, algunas de las mejoras que presenta son:
  - Nuevo tema «Twenty Thirteen».
  - Mejoras en la sección de administración.
  - Cambios en la interfaz de modelos de entradas.
  - Interfaz de menús actualizada.
  - Nuevo visor de revisiones y guardado automático de entradas.
  - Bloqueo de entradas.
  - Visualizador de vídeos HTML5.
- **3.6.1** 11 de septiembre de 2013 (actualizaciones de seguridad)
- **3.7** De nombre clave «Basie», fue lanzada el 24 de octubre de 2013; se destacan las siguientes características:
  - Actualizaciones automáticas en segundo plano.
  - Nuevo sistema de comprobación de contraseñas.
  - Mejoras en el buscador.
- **3.8** De nombre clave «Parker», fue lanzada el 12 de diciembre de 2013; se destacan las siguientes características:
  - Lanzamiento de un nuevo tema «Twenty Fourteen».
  - 8 nuevos colores para la administración de WordPress.
  - Interfaz de administración, adaptada a todos los dispositivos.
- **3.9** De nombre clave «Smith», fue lanzada el 16 de abril de 2014; se destacan las siguientes características:
  - Mejora en la edición de imágenes.
  - Mejora en el buscador de temas.
  - Añadidas listas de reproducción de audios y vídeos.

2. INSTALACIÓN Y ACTUALIZACIÓN

2.1. ANTES DE COMENZAR: REQUERIMIENTOS

7. Antes de comenzar, existen una serie de requerimientos mínimos a cumplir para poder finalizar la instalación de forma exitosa:
   1. PHP Versión 5.2.4 o superior con el límite de memoria de al menos 64Mb
   2. MySQL versión 5.0 o superior
   3. (Opcional) El módulo Apache mod_rewrite (para el uso de Permalinks)

8. 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:

```
administrador@GuiaWordpress:~$ php -v
PHP 5.5.9-1ubuntu4.5 (cli) (built: Oct 29 2014 11:59:10)
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
```

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:

```
administrador@GuiaWordpress:/var/www/html$ cat info_php.php
<?php phpinfo(); ?>
administrador@GuiaWordpress:/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:

System	Linux GuiaWordpress 3.13.0-43-generic #72-Ubuntu SMP Mon Dec 8 19:35:06 UTC 2014 x86_64
Build Date	Oct 29 2014 11:56:57
Server API	Apache 2.0 Handler

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.

9. 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 SFTP)
    - Un editor de texto
    - Un cliente FTP/SFTP
    - Un navegador Web (en nuestro caso utilizaremos la última versión de Firefox 34.0.5)

2.2. INSTALACIÓN DE REQUERIMIENTOS

2.2.1. INSTALACIÓN DE MYSQL

10. 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 "release" disponible en el momento de la publicación de esta guía: mysql-5.6.25.tar.gz.

11. La secuencia de comandos necesaria para la instalación será la siguiente, y será realizada desde la cuenta de 'root' o con permisos de administración del sistema:

$ wget https://dev.mysql.com/get/Downloads/MySQL-5.6/mysql-5.6.25.tar.gz
 # Descomprimimos el paquete
$ tar xzvf mysql-5.6.25.tar.gz
$ cd mysql-5.6.25/
 # Comprobación de dependencias
~/mysql-5.6.25$ BUILD/autorun.sh
 # Compilamos MySQL, generando los ejecutables
~/mysql-5.6.25$ CFLAGS="-O3" CXX=gcc CXXFLAGS="-O3 -felide-constructors -fno-exceptions -fno-rtti" ./configure --prefix=/usr/local/mysql --enable-assembler --with-mysqld-ldflags=-all-static
 # Instalamos los paquetes generados
~/mysql-5.6.25$ make install

```
Scanning dependencies of target abi_check
[  0%] Built target abi_check
Scanning dependencies of target zlib
[  0%] Building C object zlib/CMakeFiles/zlib.dir/adler32.c.o
[  0%] Building C object zlib/CMakeFiles/zlib.dir/compress.c.o
[  0%] Building C object zlib/CMakeFiles/zlib.dir/crc32.c.o
[  0%] Building C object zlib/CMakeFiles/zlib.dir/deflate.c.o
[  0%] Building C object zlib/CMakeFiles/zlib.dir/gzio.c.o
[  0%] Building C object zlib/CMakeFiles/zlib.dir/infback.c.o
[  0%] Building C object zlib/CMakeFiles/zlib.dir/inffast.c.o
[  0%] Building C object zlib/CMakeFiles/zlib.dir/inflate.c.o
[  0%] Building C object zlib/CMakeFiles/zlib.dir/inftrees.c.o
[  0%] Building C object zlib/CMakeFiles/zlib.dir/trees.c.o
[  0%] Building C object zlib/CMakeFiles/zlib.dir/uncompr.c.o
[  1%] Building C object zlib/CMakeFiles/zlib.dir/zutil.c.o
Linking C static library libzlib.a
[  1%] Built target zlib
```

NOTA

En caso de fallar el comando anterior (*make install*) se procederá a instalar los paquetes faltantes:
- cmake
- g++
- libncurses

Si estamos en un sistema Debian/Ubuntu bastará con ejecutar el siguiente comando para instalar las dependencias anteriores: *apt-get install cmake g++ libncurses5_dev*

 # Añadimos el grupo y usuario 'mysql'
~/mysql-5.6.25$ groupadd mysql
~/mysql-5.6.25$ useradd -g mysql mysql
~/mysql-5.6.25$ 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'

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

    $ bin/mysqld_safe --user=mysql &

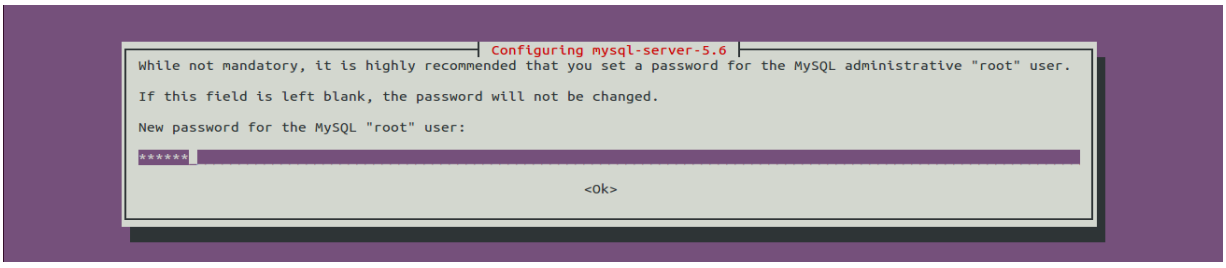
13. Desde este momento dispondremos de MySQL en nuestro sistema, dentro de la ruta '/usr/local/mysql'.

14. 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-5.6 php5-mysql libapache2-mod-auth-mysql

```
$ sudo apt-get install mysql-server-5.6 php5-mysql libapache2-mod-auth-mysql
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  php5-common php5-json
Suggested packages:
  mailx tinyca php5-user-cache
The following NEW packages will be installed:
  libapache2-mod-auth-mysql mysql-server-5.6 php5-common php5-json php5-mysql
0 upgraded, 5 newly installed, 0 to remove and 100 not upgraded.
Need to get 559 kB/6019 kB of archives.
After this operation, 50.9 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

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



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

2.2.2. INSTALACIÓN DE PHP

17. 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-5.6.5.tar.gz.

18. 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-VERSION.tar.gz
$ cd php-VERSION
# Especificamos la ruta de instalacion a /usr/local/mysql
$ ./configure --with-mysql=/usr/local/mysql
$ 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 configuración extensiones 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.

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

```
extension=mysql.so
```

20. 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
```

```
$ apt-get install php5 php5-mysql libapache2-mod-php5
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  php5-cli php5-common php5-json php5-readline
Suggested packages:
  php-pear php5-user-cache
The following NEW packages will be installed:
  libapache2-mod-php5 php5 php5-cli php5-common php5-json php5-mysql
  php5-readline
0 upgraded, 7 newly installed, 0 to remove and 100 not upgraded.
Need to get 4854 kB of archives.
After this operation, 19.9 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

21. Un 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();
?>
```

22. Utilizando el navegador accederemos 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.

2.2.3. ACTIVACIÓN DE MOD_REWRITE

23. Para activar el módulo 'mod_rewrite' en Apache, será necesario ejecutar los siguientes comandos como 'root' o con permisos de administrador del sistema:

    $ sudo a2enmod rewrite

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

        <Directory /var/www/html/wordpress/>
                AllowOverride All
                # Uncomment this directive is you want to see apache2's
                # default start page (in /apache2-default) when you go to /
                #RedirectMatch ^/$ /apache2-default/
        </Directory>
```

25. Una vez realizado esto, reiniciaremos el servicio:

    $ apache2ctl restart

2.3. INSTALACIÓN DE WORDPRESS

2.3.1. INSTALACIÓN DESDE EL ARCHIVO OFICIAL

26. Si disponemos acceso a la línea de consola de nuestro servidor, podremos descargar la última versión de Wordpress utilizando wget (o Lynx o cualquier navegador de texto):

```
root@GuiaWordpress:/var/www/html# wget http://wordpress.org/latest.tar.gz
--2014-12-30 15:07:12--  http://wordpress.org/latest.tar.gz
Resolviendo wordpress.org (wordpress.org)... 66.155.40.249, 66.155.40.250
Conectando con wordpress.org (wordpress.org)[66.155.40.249]:80... conectado.
Petición HTTP enviada, esperando respuesta... 302 Moved Temporarily
Ubicación: https://wordpress.org/latest.tar.gz [siguiente]
--2014-12-30 15:07:12--  https://wordpress.org/latest.tar.gz
Conectando con wordpress.org (wordpress.org)[66.155.40.249]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 6183711 (5,9M) [application/octet-stream]
Grabando a: "latest.tar.gz"

100%[======================================>] 6.183.711   3,28MB/s   en 1,8s

2014-12-30 15:07:15 (3,28 MB/s) - "latest.tar.gz" guardado [6183711/6183711]
```

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

```
root@GuiaWordpress:/var/www/html# tar -zxvf latest.tar.gz
wordpress/
wordpress/wp-settings.php
wordpress/wp-cron.php
wordpress/wp-comments-post.php
wordpress/wp-activate.php
```

28. Wordpress se descomprimirá en un directorio nuevo, llamado wordpress, dentro del mismo directorio raíz donde hemos descargado el fichero latest.tar.gz

2.3.2. INSTALACIÓN UTILIZANDO SVN

29. Otra alternativa para la instalación de Wordpress consiste en utilizar la herramienta para desarrolladores Subversion (SVN), una herramienta de control de versiones open source basada en un repositorio, cuyo funcionamiento se asemeja enormemente al de un sistema de ficheros.

30. Utiliza el concepto de revisión para guardar los cambios producidos en el repositorio. Entre dos revisiones sólo guarda el conjunto de modificaciones, optimizando así al máximo el uso de espacio en disco. SVN permite al usuario crear, copiar y borrar carpetas con la misma flexibilidad con la que lo haría si estuviese en su disco duro local.

31. El repositorio SVN de Wordpress contiene varias ramas (branches) o secciones. Cuando se realiza la instalación, hay que seleccionar cual deseamos utilizar. Las principales opciones serán:

    - **trunk**: contiene la última versión que está en desarrollo. Esta rama puede contener múltiples errores, pero puede ser de utilidad en el caso de querer comprobar la funcionalidad de nuestros *plugins* o temas con la siguiente "release" de Wordpress.
    - **Última *release***: es la rama recomendada si queremos utilizar la última versión funcional y estable de esta herramienta.

32. Seleccionaremos la última versión estable que haya sido lanzada de Wordpress accediendo a la web oficial (https://wordpress.org/download/). En el proceso a continuación, veremos cómo en un directorio "wordpress" descargaremos los ficheros necesarios de esta nueva instalación:

```
root@GuiaWordpress:/var/www/html/wordpress# svn co http://core.svn.wordpress.org/tags/4.1 .
A    wp-content
A    wp-content/index.php
A    wp-content/plugins
A    wp-content/plugins/hello.php
A    wp-content/plugins/index.php
A    wp-content/themes
A    wp-content/themes/index.php
A    wp-content/themes/twentythirteen
A    wp-content/themes/twentythirteen/genericons
A    wp-content/themes/twentythirteen/genericons/Genericons-Regular.otf
A    wp-content/themes/twentythirteen/genericons/genericons.css
A    wp-content/themes/twentythirteen/genericons/COPYING.txt
```

33. El punto (.) al final de comando es importante, ya que descargará los ficheros necesarios dentro del directorio donde nos encontremos. En caso contrario, acabaremos creando un directorio con el mismo nombre de la versión instalada, en este caso "4.1".

DIRECTORIO .SVN EN SERVIDORES PÚBLICOS

El comando 'svn co' (*checkout*) descargará una copia de todos los ficheros, incluido un directorio oculto con el nombre de '.svn'.

Este directorio puede contener información sensible, como podrían ser contraseñas:

```
root@GuiaWordpress:/var/www/html/wordpress# ls -al .svn/
total 972
drwxr-xr-x   4 root root   4096 dic 30 15:20 .
drwxr-xr-x   6 root root   4096 dic 30 15:20 ..
-rw-r--r--   1 root root      3 dic 30 15:17 entries
-rw-r--r--   1 root root      3 dic 30 15:17 format
drwxr-xr-x 258 root root   4096 dic 30 15:20 pristine
drwxr-xr-x   2 root root   4096 dic 30 15:20 tmp
-rw-r--r--   1 root root 970752 dic 30 15:20 wc.db
```

Debemos incluir una regla para proteger el acceso público a estos directorios, creando un fichero .htaccess:

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

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

Forbidden

You don't have permission to access /.svn/entries on this server.

2.3.3. CONFIGURACIÓN DE LA BASE DE DATOS

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

```
$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 44
Server version: 5.5.40-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2014, 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 guiawordpress;
Query OK, 1 row affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON guiawordpress.* TO "usuariowp"@"hostname" 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
```

35. 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.

36. 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'.

37. 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.

38. 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
$
```

39. 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
$
```

2.3.4. FINALIZANDO LA INSTALACIÓN

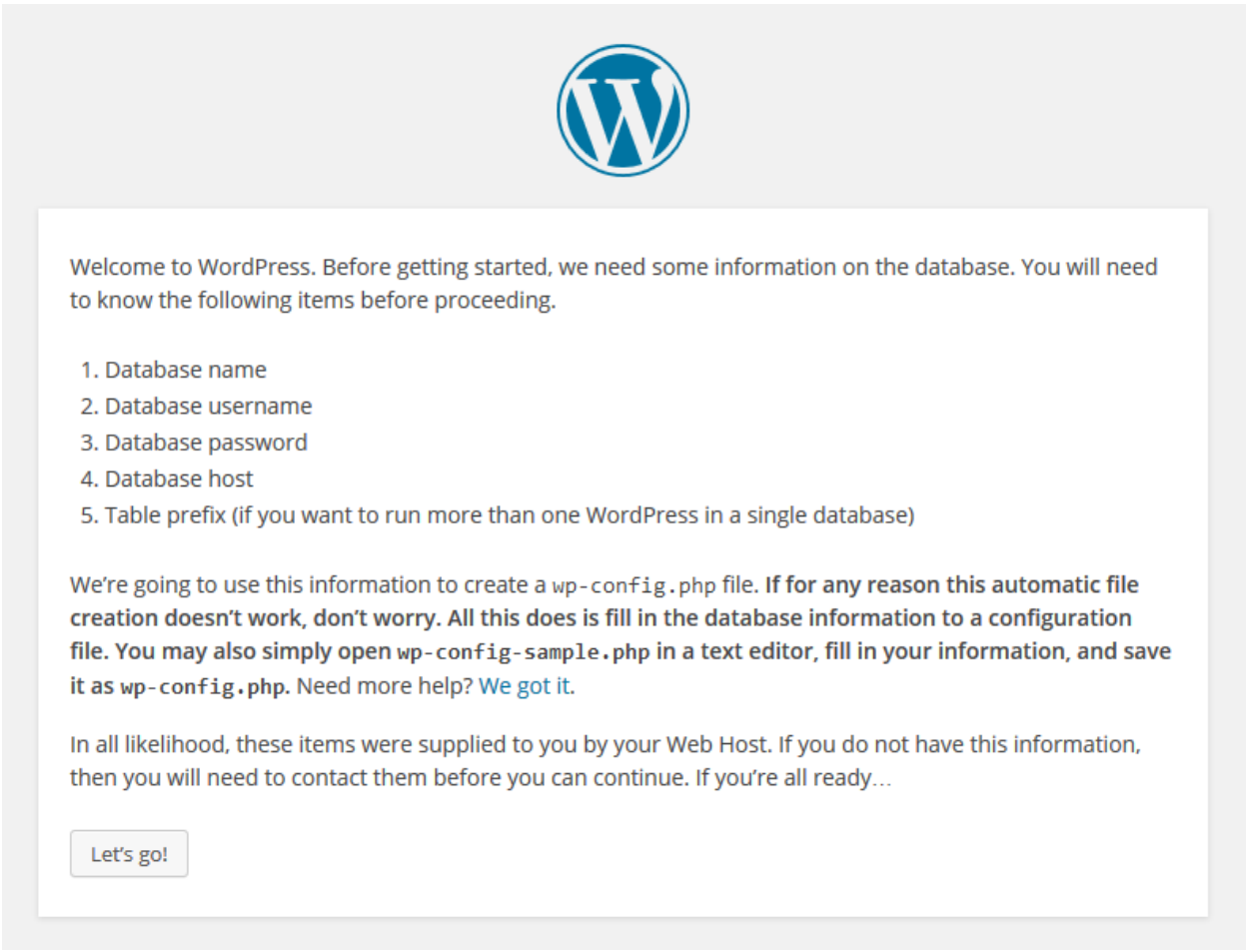
40. Ahora que tenemos los ficheros necesarios para realizar la instalación, es necesario configurar Wordpress para indicarle ciertos valores, como la información de la base de datos. Esta tarea puede realizarse de dos formas:

    1. **MANUAL**: abrimos el fichero 'wp-config-sample.php' con nuestro editor de texto, e introducimos los valores necesarios para las variables:
        - DB_NAME: el nombre de la base de datos que utilizará Wordpress
        - DB_USER: el usuario creado para acceder a la base de datos
        - DB_PASSWORD: la contraseña correspondiente
        - DB_HOST: el *hostname* de la máquina donde se realizará la instalación

```php
<?php
/**
 * The base configurations of the WordPress.
 *
 * This file has the following configurations: MySQL settings, Table Prefix,
 * Secret Keys, and ABSPATH. You can find more information by visiting
 * {@link http://codex.wordpress.org/Editing_wp-config.php Editing wp-config.php}
 * Codex page. You can get the MySQL settings from your web host.
 *
 * This file is used by the wp-config.php creation script during the
 * installation. You don't have to use the web site, you can just copy this file
 * to "wp-config.php" and fill in the values.
 *
 * @package WordPress
 */

define('FORCE_SSL_ADMIN', true);

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'nombre_basedatos');

/** MySQL database username */
define('DB_USER', 'usuario_mysql');

/** MySQL database password */
define('DB_PASSWORD', 'passwd_mysql');

/** MySQL hostname */
define('DB_HOST', 'localhost');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8');
```

Y a continuación salvamos el fichero como wp-config.php, dentro del directorio raíz donde Wordpress será instalado (en nuestro caso /var/www/html/wordpress).

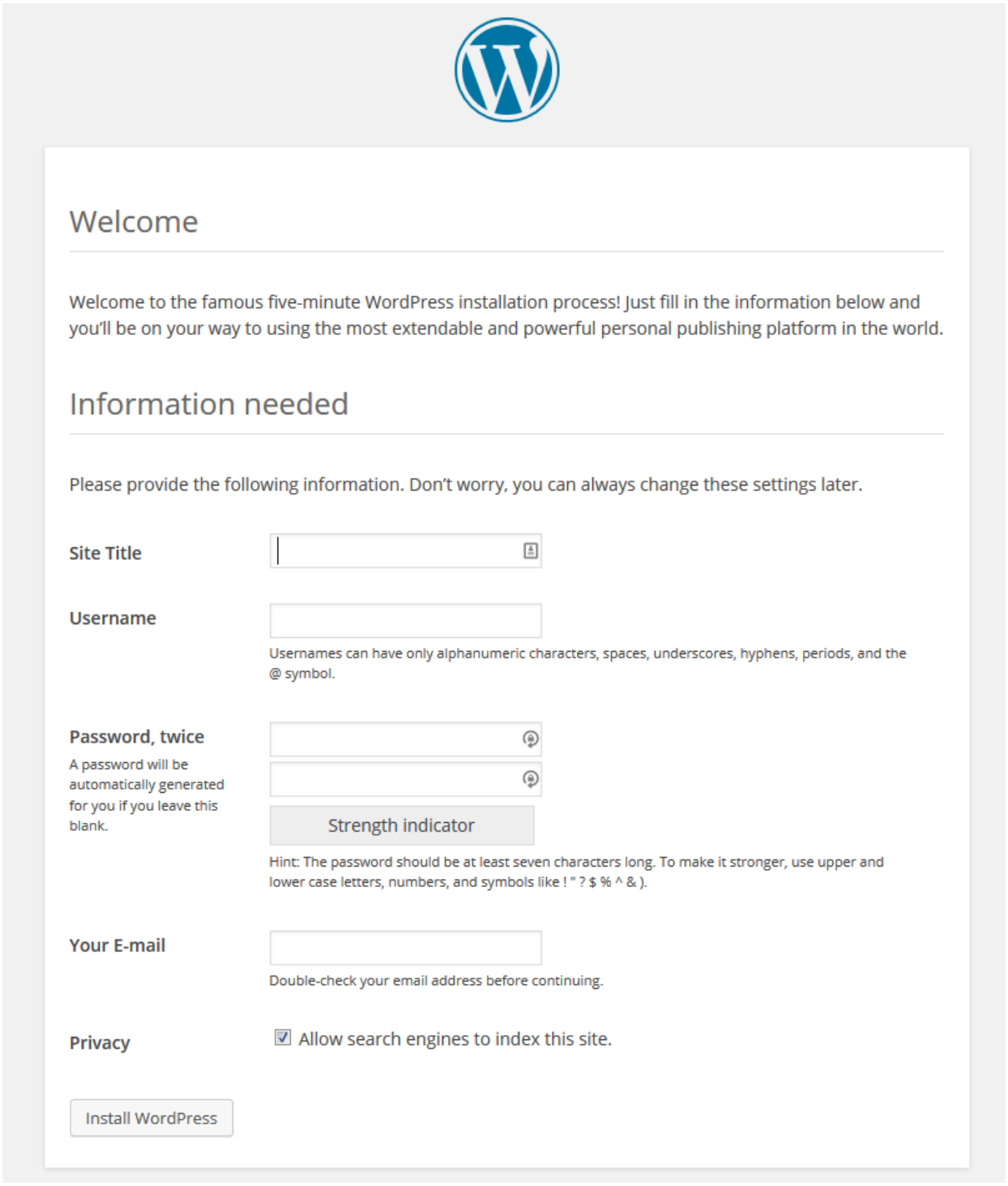
2. **AUTOMÁTICA**: Si no existe el fichero wp-config.php, al navegar a nuestro nuevo sitio, tendremos que introducir los valores en los formularios que Wordpress nos presentará para ello:



NOTA:
Se deberá modificar el valor de 'Table prefix' dentro del fichero 'wp-config.php' a otro aleatorio, para añadir una nueva capa de seguridad y evitar posibles ataques a *plugins* o componentes por defecto.

41. Una vez introducidos estos datos de configuración interna, llegaremos a la pantalla de la configuración propia de Wordpress, donde deberemos introducir una serie de valores adicionales:

- **Site title**: el título que se mostrará en la generación de páginas dinámicas
- **Username**: el nombre de usuario del administrador del sitio (no se recomienda emplear nombres de usuario comunes como "admin", "root", etc.)
- **Password**: la contraseña correspondiente al administrador (se recomienda seguir las pautas indicadas anteriormente)
- **Your E-mail**: la dirección de correo asociada al administrador
- **Privacy**: activaremos esta opción si queremos que nuestro sitio sea visible a todo el mundo, incluido los diferentes motores de búsqueda en Internet. En caso de que no queramos que nuestra web quede indexada por estos motores de búsqueda, deberemos desactivar esta opción.



42. Una vez finalicemos de introducir los valores que hemos seleccionado, pulsamos en "Install Wordpress" y la instalación habrá finalizado.

3. SEGURIDAD EN LA BASE DE DATOS

43. MySQL es un sistema de gestión de bases de datos relacional, multi-hilo 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.

44. 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.

45. En los siguientes apartados se describen los pasos necesarios para mejorar la seguridad de la base de datos, sin embargo se recomienda antes emplear el script de autoconfiguración que trae MySQL "mysql_secure_installation":

```
root@ubuntu:~# mysql_secure_installation

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MySQL
root user without the proper authorisation.

You already have a root password set, so you can safely answer 'n'.

Change the root password? [Y/n]
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
 ... Success!

Remove anonymous users? [Y/n]
 ... Success!

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n]
 ... Success!

By default, MySQL comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n]
 - Dropping test database...
ERROR 1008 (HY000) at line 1: Can't drop database 'test'; database doesn't
 ... Failed!  Not critical, keep moving...
 - Removing privileges on test database...
 ... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n]
 ... Success!

Cleaning up...

All done!  If you've completed all of the above steps, your MySQL
installation should now be secure.
```

46. Este script permitirá cambiar la contraseña por defecto del usuario 'root', eliminar los usuarios anónimos, deshabilitar el acceso remoto del usuario 'root' y eliminar las bases de datos de prueba.

3.1. DESHABILITAR O RESTRINGIR EL ACCESO REMOTO

47. 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.

48. En caso contrario, evitaremos que MySQL abra un socket de red (sin impedir que las conexiones locales sigan funcionando) que permita estas conexiones, añadiendo el siguiente código a la parte '[mysqld]' del fichero '/etc/my.cnf' o '/etc/msqyl/my.ini':

    Skip-networking

49. 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
```

3.2. DESHABILITAR EL USO DE 'LOCAL INFILE'

50. 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.

51. 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

52. O incluso de forma más sencilla:

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

53. 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

3.3. MODIFICAR EL NOMBRE DE USUARIO DEL ADMINISTRADOR

54. 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.

55. Para endurecer esta tarea, modificaremos el nombre de la cuenta a otra más complicada de adivinar de forma remota, añadiendo además 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;

56. 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;

3.4. ELIMINAR LA BASE DE DATOS DE PRUEBAS

57. 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;

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

$ mysqladmin -u username -p drop test

3.5. ELIMINAR LA CUENTA ANÓNIMA Y OTRAS CUENTAS OBSOLETAS

59. 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="";*

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

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

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

```
$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 37
Server version: 5.5.40-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2014, 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> select * from mysql.user where user="";
Empty set (0.00 sec)

mysql> SHOW GRANTS FOR ''@'localhost';
ERROR 1141 (42000): There is no such grant defined for user '' on host 'localhost'
mysql> SHOW GRANTS FOR ''@'myhost';
ERROR 1141 (42000): There is no such grant defined for user '' on host 'myhost'
mysql>
```

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

mysql> DROP USER "";

63. 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;

3.6. REDUCIR LOS PRIVILEGIOS DEL SISTEMA

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

65. 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.

66. 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

```
total 36880
-rw-r--r-- 1 root  root         0 dic 30 14:40 debian-5.5.flag
drwx------ 2 mysql mysql     4096 ene 10 22:37 guiawordpress
-rw-rw---- 1 mysql mysql 27262976 ene 11 01:11 ibdata1
-rw-rw---- 1 mysql mysql  5242880 feb  9 18:13 ib_logfile0
-rw-rw---- 1 mysql mysql  5242880 feb  9 18:13 ib_logfile1
drwx------ 2 mysql root     4096 dic 30 14:41 mysql
-rw-rw---- 1 root  root         6 dic 30 14:41 mysql_upgrade_info
drwx------ 2 mysql mysql     4096 dic 30 14:41 performance_schema
```

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

68. Finalmente, los binarios que se encuentran en el directorio '/usr/bin/', deberán tener como propietario al usuario 'root' o al que ejecute el servicio 'mysqld'. Ningún otro usuario debe tener permisos de escritura sobre estos ficheros:

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

69. 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
```

3.7. ACTIVAR LOS LOGS DEL SISTEMA

70. 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
```

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

3.8. ELIMINAR HISTORIAL

72. Durante el proceso de instalación y generación de bases de datos, tablas etc. es probable que se utilice información sensible que debe ser eliminada de 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.

73. 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 .basrhc):

    cat /dev/null > ~/.mysql_history

3.9. LIMITAR LOS PERMISOS DE USUARIO

74. Para la operativa normal de Wordpress, como generar nuevos posts, comentarios, crear usuarios o instalar plugins, 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.

75. Para ello, introduciremos la siguiente consulta en nuestra consola MySQL (en nuestro ejemplo, la base de datos que hemos utilizado para Wordpress es ‘guiawordpress’ y el usuario con los permisos de acceso ‘usuariowp’):

    mysql> REVOKE ALL PRIVILEGES ON guiawordpress. FROM 'usuariowp'@'localhost';*

76. Después, asignaremos nuevos privilegios a nuestro usuario:

    mysql> GRANT SELECT, INSERT, UPDATE, DELETE ON guiawordpress. TO 'usuariowp'@'*':*

NOTA:
Algunos *plugins*, temas o actualizaciones de Wordpress 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 Wordpress en MySQL.

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

3.10. MODIFICAR EL PREFIJO DE LAS TABLAS DE WORDPRESS

77. Wordpress aplica un prefijo a todas las tablas de la base de datos que le asignemos, por ejemplo wp_posts, wp_terms, etc. Este prefijo se puede modificar durante la instalación, e incluso más tarde, y nos ayudará a añadir una nueva capa de seguridad para impedir a posibles atacantes usar técnicas de SQL Injection para tomar el control de nuestro servidor.

78. Esta acción no deberá ser realizada si este prefijo ha sido cambiado previamente.

NOTA:
Se recomienda realizar un *backup* de la base de datos antes de realizar este tipo de cambios estructurales.

79. El prefijo usado actualmente se encuentra dentro del fichero 'wp-config.php':

```
$ grep table_prefix wp-config.php
$table_prefix  = 'wp_';
$
```

80. Para modificar el valor, editaremos nuestro fichero de configuración y elegiremos uno nuevo que sea más complicado de adivinar, como por ejemplo:

$table_prefix = wp_VzQCxSJv7uL_';

81. Realizar el cambio en el fichero de configuración no modificará de forma automática todas las tablas que hayan sido creadas con anterioridad en Wordpress, por lo que será necesario realizar una actualización de forma manual de la base de datos. Primero, renombraremos las 11 tablas que crea Wordpress en la instalación por defecto:

```
$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 69
Server version: 5.5.40-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2014, 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> use guiawordpress;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> RENAME table `wp_commentmeta` TO `wp_VzQCxSJv7uL_commentmeta`;
Query OK, 0 rows affected (0.03 sec)

mysql> RENAME table `wp_comments` TO `wp_VzQCxSJv7uL_comments`;
Query OK, 0 rows affected (0.02 sec)

mysql> RENAME table `wp_links` TO `wp_VzQCxSJv7uL_links`;
Query OK, 0 rows affected (0.01 sec)

mysql> RENAME table `wp_options` TO `wp_VzQCxSJv7uL_options`;
Query OK, 0 rows affected (0.02 sec)

mysql> RENAME table `wp_postmeta` TO `wp_VzQCxSJv7uL_postmeta`;
Query OK, 0 rows affected (0.02 sec)

mysql> RENAME table `wp_posts` TO `wp_VzQCxSJv7uL_posts`;
Query OK, 0 rows affected (0.01 sec)

mysql> RENAME table `wp_terms` TO `wp_VzQCxSJv7uL_terms`;
Query OK, 0 rows affected (0.01 sec)

mysql> RENAME table `wp_term_relationships` TO `wp_VzQCxSJv7uL_term_relationships`;
Query OK, 0 rows affected (0.02 sec)

mysql> RENAME table `wp_term_taxonomy` TO `wp_VzQCxSJv7uL_term_taxonomy`;
Query OK, 0 rows affected (0.02 sec)

mysql> RENAME table `wp_usermeta` TO `wp_VzQCxSJv7uL_usermeta`;
Query OK, 0 rows affected (0.02 sec)

mysql> RENAME table `wp_users` TO `wp_VzQCxSJv7uL_users`;
Query OK, 0 rows affected (0.01 sec)

mysql>
```

82. Para ello será necesario lanzar las siguientes consultas, como hemos visto, desde la línea de comandos de MySQL:

mysql> RENAME table `wp_commentmeta` TO `wp_VzQCxSJv7uL_commentmeta`;

```
mysql> RENAME table `wp_comments` TO `wp_VzQCxSJv7uL_comments`;
mysql> RENAME table `wp_links` TO `wp_VzQCxSJv7uL_links`;
mysql> RENAME table `wp_options` TO `wp_VzQCxSJv7uL_options`;
mysql> RENAME table `wp_postmeta` TO `wp_VzQCxSJv7uL_postmeta`;
mysql> RENAME table `wp_posts` TO `wp_VzQCxSJv7uL_posts`;
mysql> RENAME table `wp_terms` TO `wp_VzQCxSJv7uL_terms`;
mysql> RENAME table `wp_term_relationships` TO `wp_VzQCxSJv7uL_term_relationships`;
mysql> RENAME table `wp_term_taxonomy` TO `wp_VzQCxSJv7uL_term_taxonomy`;
mysql> RENAME table `wp_usermeta` TO `wp_VzQCxSJv7uL_usermeta`;
mysql> RENAME table `wp_users` TO `wp_VzQCxSJv7uL_users`;
```

NOTA:
En caso de observar que existe alguna tabla adicional, perteneciente a un *plugin* o instalación posterior, será necesario modificar el prefijo también.

El objetivo es modificar TODAS las tablas para que contengan el nuevo prefijo sin que la instalación de Wordpress se vea afectada.

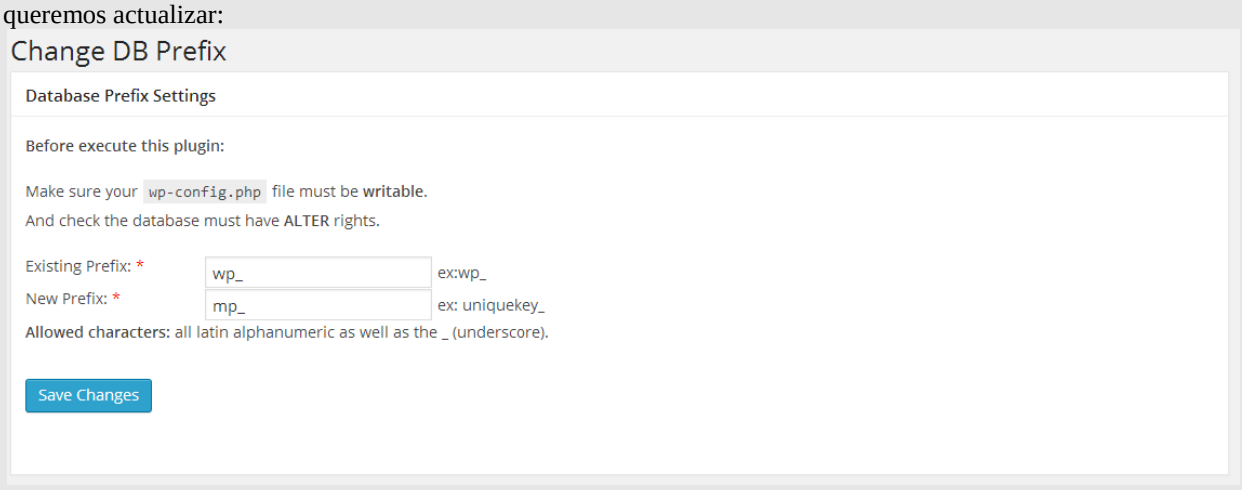
83. Ahora, deberemos buscar en la tabla 'options' cualquier ocurrencia con el anterior prefijo y modificarlo de forma manual. Para ello, utilizaremos la siguiente sentencia:

```
mysql> SELECT * FROM `wp_VzQCxSJv7uL_options` WHERE `option_name` LIKE '%wp_%';
```

84. Finalmente, dentro de la tabla 'usermeta' buscaremos también cualquier coincidencia con el anterior prefijo 'wp_' y lo modificaremos de forma manual:

```
mysql> SELECT * FROM `wp_VzQCxSJv7uL_usermeta` WHERE `meta_key` LIKE '%wp_%';
```

NOTA:
Este cambio puede realizarse también de forma automática utilizando diferentes *plugins* del repositorio oficial de Wordpress. El más sencillo hasta la fecha es 'Change DB Prefix', y una vez instalado con el procedimiento habitual, nos mostrará una pantalla donde deberemos especificar el valor actual del prefijo de las tablas, así como el nuevo queremos actualizar:



4. CREACIÓN DE UN FICHERO 'ROBOTS.TXT'

85. 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.

86. 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.

87. Por defecto, el fichero robots.txt de las instalaciones de Wordpress contiene:

    *User-agent: **
    Disallow: /wp-admin/
    Disallow: /wp-includes/

88. Modificaremos este fichero para asegurarnos de que los buscadores no indexarán información sensible acerca de nuestra instalación.

    *User-agent: **
    (Indicamos que estas reglas son aplicables a todos los buscadores)
    Disallow: /cgi-bin/
    (Indicamos que no indexen programas que pudiese haber en cgi-bin)
    Disallow: /?*
    (Indicamos que no indexe búsquedas, que incluyen el símbolo ?)

89. Cualquier directorio que deseemos excluir de este indexado de buscadores, lo introduciremos con la opción "Disallow", indicando la ruta completa al directorio deseado. Se deberá evitar introducir directorios propios de Wordpress, como 'wp-admin' o 'wp-includes', ya que podría facilitar información a un atacante sobre el tipo de CMS utilizado en el servidor.

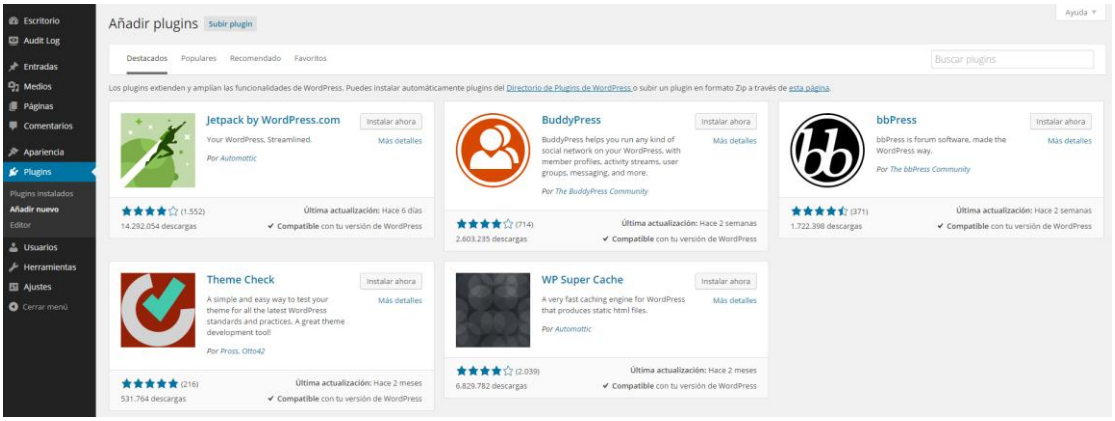
NOTA:
Existen una serie de datos adicionales que necesitaremos saber:
1. Se recomienda no incluir en un fichero robots.txt más de 200 líneas de Disallow.
2. Una vez que incluyamos el Disallow de una carpeta, si nos equivocamos y queremos volver a indexarla, pueden pasar hasta 3 meses desde que se agregue 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 tener una idea de lo que 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".

5. PROCEDIMIENTO PARA LA INSTALACIÓN DE PLUGINS

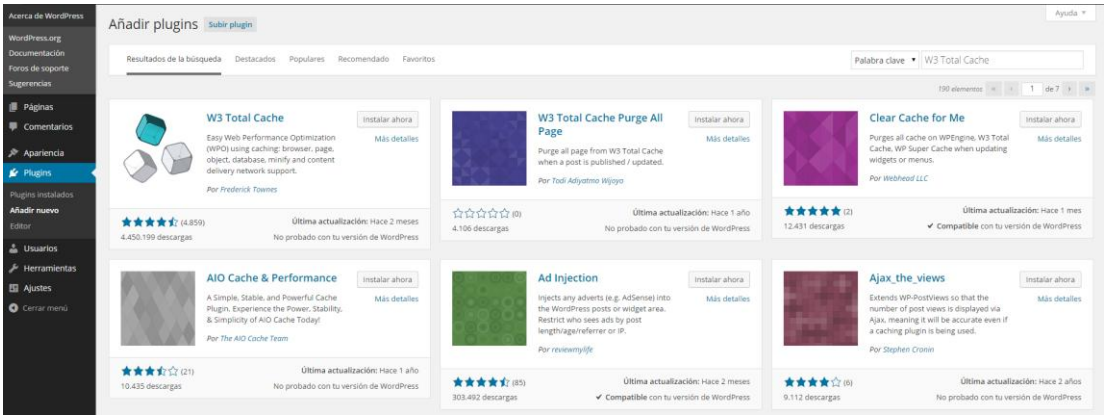
5.1. AUTOMÁTICO

90. Para la instalación de plugins de forma automática, seguiremos los siguientes pasos:

- Ir a *Plugins* -> Añadir Nuevo.



- En el campo "Buscar" escribir el nombre del *plugin*. Ej: W3 Total Cache
- Hacer clic en "Instalar Ahora" del *plugin* deseado.



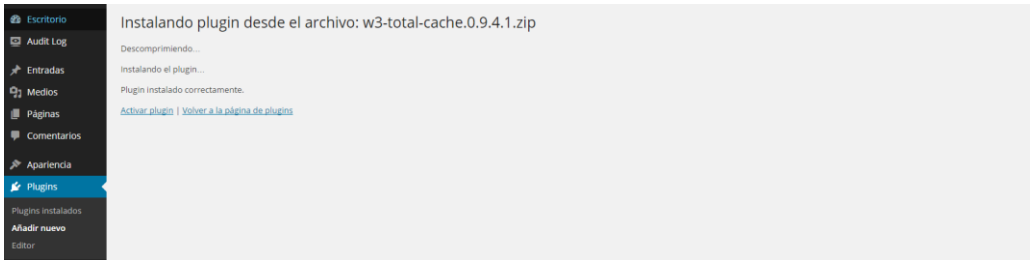
- Una vez subido e instalado hacer clic en "Activar *plugin*"

5.2. MANUAL

- En este caso, necesitaremos descargar el fichero del *plugin* desde la web del autor y seguir estos pasos:
- Ir a *Plugins* -> Añadir Nuevo -> En la parte superior escoger la opción "Subir"

- Clic en "Examinar" y seleccionar el archivo descargado.

- Clic en "Instalar Ahora" y una vez subido e instalado hacer clic en "Activar *plugin*".



NOTA:
Los *plugins* aportan nuevas capacidades a Wordpress haciéndolo más potente y flexible, sin embargo también pueden llegar a entrañar riesgos de seguridad. Por ello se recomienda siempre:

1. Instalar sólo aquellos *plugins* que sean necesarios, pues cuantos más *plugins* tengamos mayores serán las posibilidades de que un atacante encuentre un fallo de seguridad en uno de ellos que potencialmente comprometa nuestro CMS.
2. Instalar sólo *plugins* que sean confiables, es decir, que procedan de una fuente como la web de Wordpress.
3. Comprobar la fecha de última actualización de cada *plugin*, ya que une versión antigua podría contener vulnerabilidades conocidas no parcheadas.
4. Si se desarrolla o emplea un tema para personalizar la interfaz del CMS se recomienda realizar un análisis de seguridad del mismo empleando el *plugin* "Theme Check" (https://wordpress.org/plugins/theme-check/).

6. ELIMINANDO LA INFORMACIÓN DE VERSIÓN

91. Wordpress muestra, dentro del código expuesto al público, información sensible acerca de su versión, en diferentes campos HTML y funciones. A pesar de que realicemos las actualizaciones de forma automática, esto puede ofrecer información de gran valor a un posible atacante, ya que será capaz de identificar de forma única y fiable si nuestra instalación es vulnerable a algún ataque conocido.

92. A pesar de que eliminar estos datos no nos ofrecerá una protección mucho mayor, es siempre recomendable eliminar este tipo de fuga de datos. Esta información se muestra principalmente en tres áreas de nuestro sitio:

- La etiqueta ‘generator’ en la cabecera de la página:

  <meta name="generator" content="WordPress 4.0" />

- Peticiones sobre scripts o estilos de página:

  subscriptions.css?ver=4.0

- La etiqueta ‘generator’ en los feeds RSS:

  <generator>http://wordpress.org/?v=4.0</generator>

93. Para evitar mostrar la información de versión, tendremos que abrir el fichero ‘functions.php’ de nuestro tema correspondiente, que se encuentra en la ruta ‘./wp-content/themes/nombre_del_tema/functions.php’ (en nuestro caso, dado que estamos utilizando la plantilla por defecto, editamos el fichero ‘./wp-content/themes/twentyfourteen/functions.php’) e incluir el siguiente código al final del fichero:

```
/* Hide WP version strings from scripts and styles
 * @return {string} $src
 * @filter script_loader_src
 * @filter style_loader_src
 */
function fjarrett_remove_wp_version_strings( $src ) {
    global $wp_version;
    parse_str(parse_url($src, PHP_URL_QUERY), $query);
    if ( !empty($query['ver']) && $query['ver'] === $wp_version ) {
        $src = remove_query_arg('ver', $src);
    }
    return $src;
}

add_filter( 'script_loader_src', 'fjarrett_remove_wp_version_strings' );
add_filter( 'style_loader_src', 'fjarrett_remove_wp_version_strings' );

function wpmudev_remove_version() {
return '';
}

add_filter('the_generator', 'wpmudev_remove_version');
```

94. Una vez introducido correctamente, podremos comprobar que ha funcionado visualizando, por ejemplo, el código HTML de la página principal de nuestro sitio y verificando que ya no se muestra información relativa a la versión actual de Wordpress.

7. BASTIONADO DE LA CUENTA DE ADMINISTRADOR

95. La cuenta de administrador, por defecto, es la cuenta que más ataques recibirá del sistema, debido a que tiene privilegios completos para realizar cualquier tarea dentro de Wordpress, como activar o desactivar temas y plugins, crear o modificar widgets, editar el código fuente, añadir o eliminar usuarios, etc.

96. Por este motivo, es necesario establecer una serie de medidas adicionales para protegerla de posibles atacantes, entre las que incluimos las siguientes:

- Habilitar el acceso por HTTPS, de tal forma que el acceso tanto de usuarios como del administrador no pueda verse capturado o expuesto por un posible atacante en la red
- 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
- Sólo debe existir una cuenta de administrador
- Los usuarios nuevos deberán tener los mínimos permisos para poder realizar sus tareas
- No escribir contenido, tal como *posts* y páginas, desde la cuenta de administrador, ya que podría ofrecer información necesaria para que un atacante lance un ataque satisfactorio contra el sistema
- No compartir las credenciales de acceso del sistema. Si es necesario que un desarrollador acceda al panel de control, se creará una cuenta temporal con permisos de administrador, y cuando el trabajo finalice, esta se eliminará definitivamente
- Utilizar una contraseña fuerte para todos y cada uno de los usuarios del sistema:
  - Al menos 12 caracteres
  - No debe ser predecible ni encontrarse en algún diccionario
  - No debe coincidir con datos personales del usuario, tales como el nombre de un conocido, pareja, mascota, o la ciudad donde nació
  - Debe incluir mayúsculas, minúsculas y caracteres especiales como $,!,?, etc.
  - Debe ser modificada con cierta frecuencia (por ejemplo, cada dos o tres meses)
  - Para evitar olvidar la contraseña se puede utilizar los diferentes gestores disponibles, como 1Password o LastPass. Nunca debe apuntarse las credenciales en papel, en la cartera o con una nota sobre el monitor.

7.1. RENOMBRADO DE LA CUENTA

97. Una práctica habitual para asegurar la cuenta de administrador, es renombrarla con otro nombre de usuario. No es necesario utilizar una mezcla complicada de caracteres, números y símbolos, como utilizaríamos en la contraseña.

98. Para ello, accederemos por línea de consola a nuestro servidor y accederemos a MySQL:

```
$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 674
Server version: 5.5.40-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2014, 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> use guiawordpress;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from wp_users;
+----+------------+------------------------------------+---------------+---------------------+----------+---------------------+---------
------------+-------------+--------------+
| ID | user_login | user_pass                          | user_nicename | user_email          | user_url | user_registered     | user_act
ivation_key | user_status | display_name |
+----+------------+------------------------------------+---------------+---------------------+----------+---------------------+---------
------------+-------------+--------------+
|  1 | admin      | $P$BDOLt3qdR1QGWGf85k8eRwzOIIrFoB. | admin         | admin@guiawordpress.com |          | 2014-12-30 15:25:52 |
            |           0 | admin        |
+----+------------+------------------------------------+---------------+---------------------+----------+---------------------+---------
------------+-------------+--------------+
1 row in set (0.00 sec)
```

99. Como hemos visto, las credenciales de acceso se almacenan dentro de la tabla ‘wp_users’, y el campo que deseamos modificar es ‘user_login’. En nuestro ejemplo modificaremos el valor por defecto ‘admin’ por ‘localwpadmin’:

mysql> UPDATE wp_users SET user_login = 'localwpadmin' WHERE user_login = 'admin';

```
mysql> UPDATE wp_users SET user_login = 'localwpadmin' WHERE user_login = 'admin';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from wp_users;
+----+--------------+------------------------------------+---------------+---------------------+----------+---------------------+-------
-------------+-------------+--------------+
| ID | user_login   | user_pass                          | user_nicename | user_email          | user_url | user_registered     | user_a
ctivation_key | user_status | display_name |
+----+--------------+------------------------------------+---------------+---------------------+----------+---------------------+-------
-------------+-------------+--------------+
|  1 | localwpadmin | $P$BDOLt3qdR1QGWGf85k8eRwzOIIrFoB. | admin         | admin@guiawordpress.com |          | 2014-12-30 15:25:52 |
              |           0 | admin        |
+----+--------------+------------------------------------+---------------+---------------------+----------+---------------------+-------
-------------+-------------+--------------+
1 row in set (0.00 sec)
```

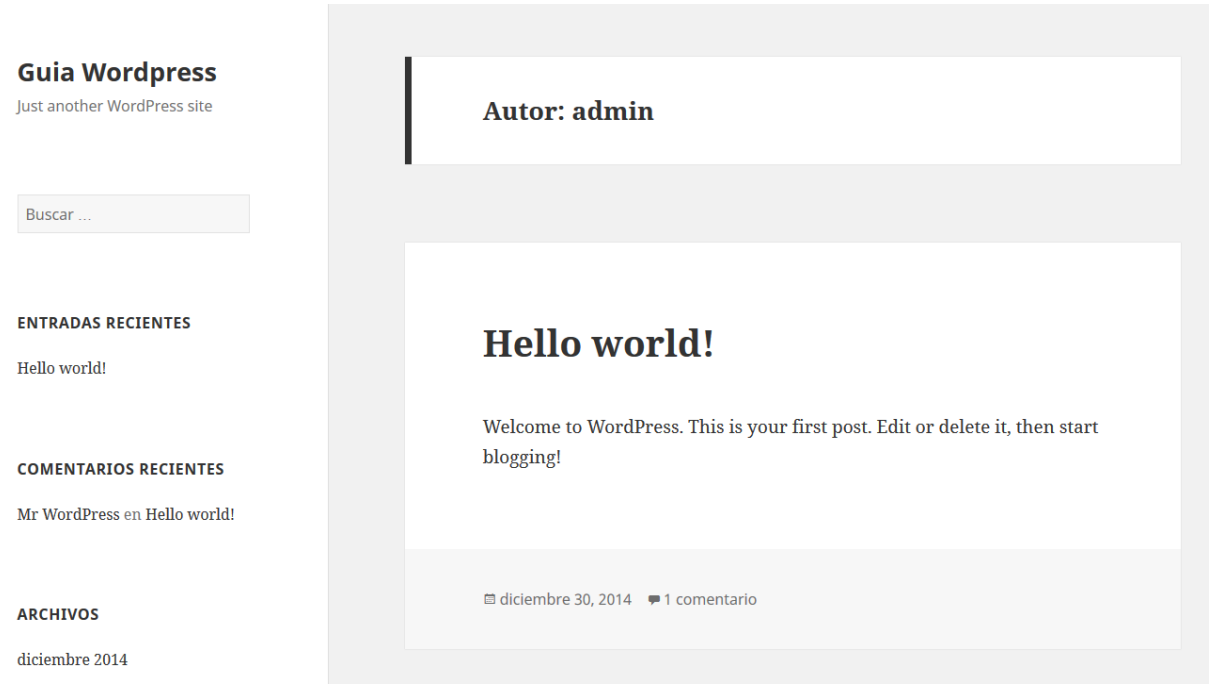
100. Desde este momento, nuestro nuevo identificador de usuario para acceder a la zona restringida de administrador habrá quedado modificado. Es posible que si nos encontrábamos registrados y actuando en el panel con nuestro navegador recibamos una alerta de que nuestra sesión ha caducado y que debemos volver a identificarnos.

7.2. MODIFICANDO EL ID POR DEFECTO

101. Por defecto, como hemos visto en las capturas anteriores, el ID asignado por defecto al administrador de Wordpress es 1. A no ser que se utilice un número aleatorio y diferente a este, si no estamos utilizando permalinks, cualquiera puede realizar una petición contra nuestro sitio web y averiguar el nombre de la cuenta de administrador:

http://sitioweb/?author=1

102. Por ejemplo, en nuestro sitio web y con la plantilla por defecto, se mostrará la siguiente información. A pesar de haber modificado el campo ‘user_login’ de la base de datos, se sigue mostrando como autor admin, debido a que el campo ‘display_name’ permanece con el valor por defecto.



103. Aunque hemos modificado correctamente el nombre de usuario de acceso al sistema, modificaremos también el valor del ID correspondiente, para reducir el riesgo a atacantes más expertos. Para realizar esta modificación, primero procederemos a realizar una copia de seguridad de nuestro sitio por si tuviéramos algún problema posterior y lo necesitáramos restaurar nuevamente.

NOTA:
Antes de modificar el ID de la cuenta de administrador, es necesario asegurarnos de que no existe ningún post o página asignado a esta cuenta. Si fuera así, necesitaremos crear de forma manual una *query* SQL para modificar los valores del ID a la cuenta correspondiente.

104. Para ello, accederemos a MySQL mediante consola y lanzaremos la siguiente petición para modificar el identificador deseado. En nuestro ejemplo modificaremos el ID actual '1' por uno más alto, para complicar su identificación; el '1534'. Además, realizaremos el mismo cambio en la tabla 'wp_usermeta' donde se almacenan los datos relacionados con los usuarios:

mysql> UPDATE wp_users SET ID = 1534 WHERE ID = 1;

mysql> UPDATE wp_usermeta SET user_id = 1534 WHERE user_id = 1;

105. La siguiente captura muestra, utilizando herramientas propias de un atacante, que con los nuevos cambios la enumeración de los usuarios del sistema no se puede realizar de forma tan sencilla:

- Antes de la modificación del ID:

```
[+] Enumerating usernames ...
[+] Identified the following 1 user/s:
    +----+-------+-------+
    | Id | Login | Name  |
    +----+-------+-------+
    | 1  | admin | admin |
    +----+-------+-------+
```

- Después de la modificación del ID:

```
[+] Enumerating usernames ...
[+] We did not enumerate any usernames
```

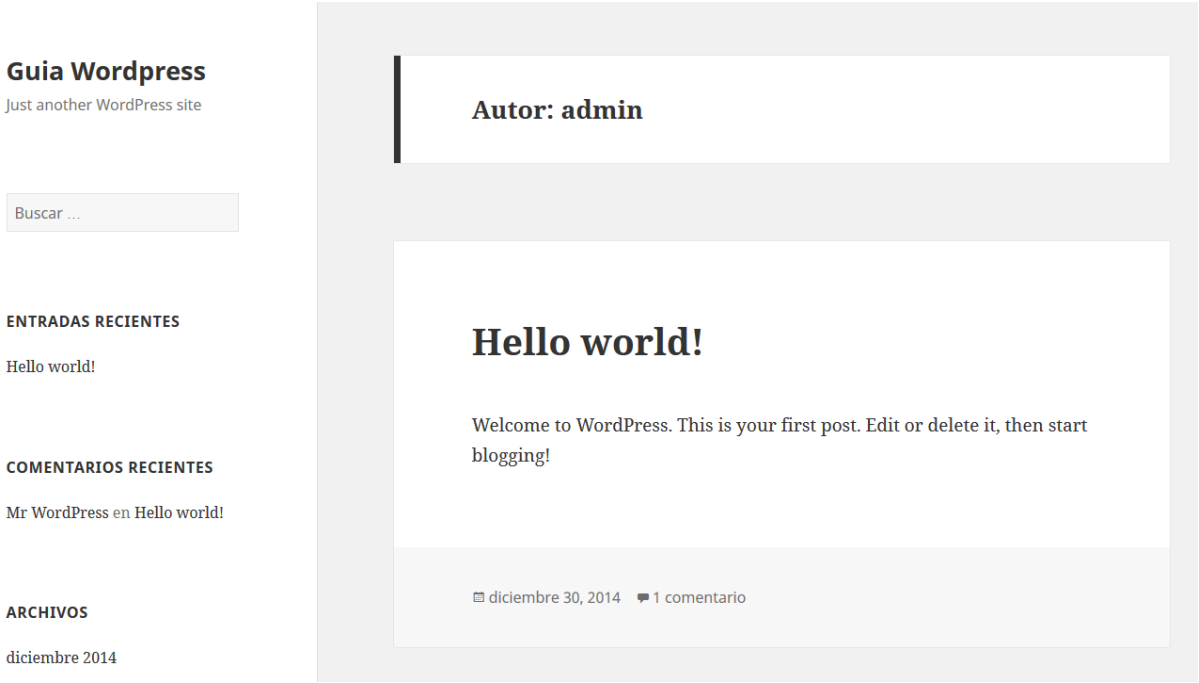
EVITAR CONFLICTOS DE ID EN WORDPRESS
Por defecto Wordpress utiliza incrementos de una unidad para asignar los ID de cada usuario nuevo. Si se van a utilizar una gran cantidad de usuarios, podría existir un conflicto en los ID si el del administrador quedara asignado a un usuario nuevo. Para ello, podemos modificar el valor de los auto incrementos de ID entre cada usuario realizando la siguiente query en la base de datos de Wordpress, dentro de MySQL:
mysql> ALTER TABLE wp_users AUTO_INCREMENT = 512;

7.3. MODIFICANDO EL NOMBRE MOSTRADO

106. En las secciones anteriores, hemos modificado el nombre de usuario y el ID correspondiente al administrador del sitio. A pesar de ello, si publicamos con la cuenta de administrador o un atacante lograra descubrirlo, se mostraría el nombre por defecto del usuario 'admin', que coincide con la cuenta de usuario.

107. Podemos comprobarlo utilizando la anterior URL, con el ID nuevo:

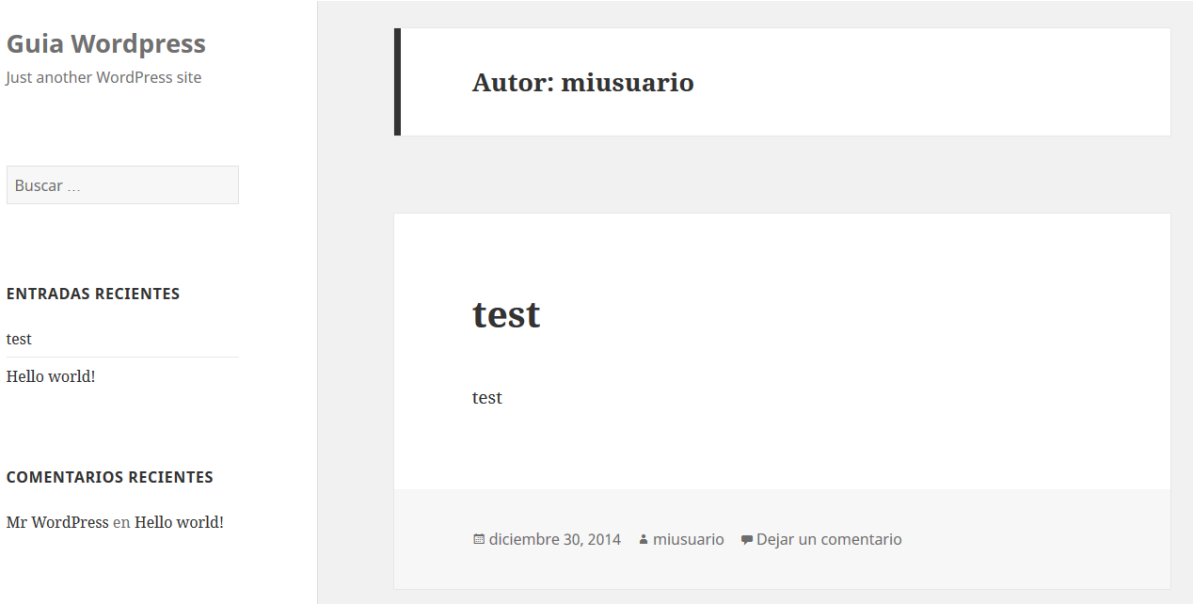
http://sitioweb/?author=1534



108. Cambiaremos el valor que se mostrará por pantalla desde MySQL con la siguiente petición (en nuestro ejemplo lo cambiaremos por el valor visible 'miusuario'):

mysql> UPDATE wp_users SET display_name = 'miusuario' WHERE display_name = 'admin';

109. Ahora, cuando entremos de nuevo en la URL anterior, veremos que el último dato que mostraba información acerca de una posible cuenta de administrador ha desaparecido:



7.4. PROTEGIENDO WP-ADMIN CON AUTENTICACIÓN HTTP

110. Proteger el directorio '/wp-admin' y el panel de control de Wordpress con un fichero .htaccess es un proceso vital para añadir una nueva capa de seguridad y proteger la instalación frente a posibles ataques 0day que sean descubiertos en un futuro.

111. 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 (sino descargaremos el paquete "apache2-utils"). Para crear el fichero, utilizaremos el siguiente comando:

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

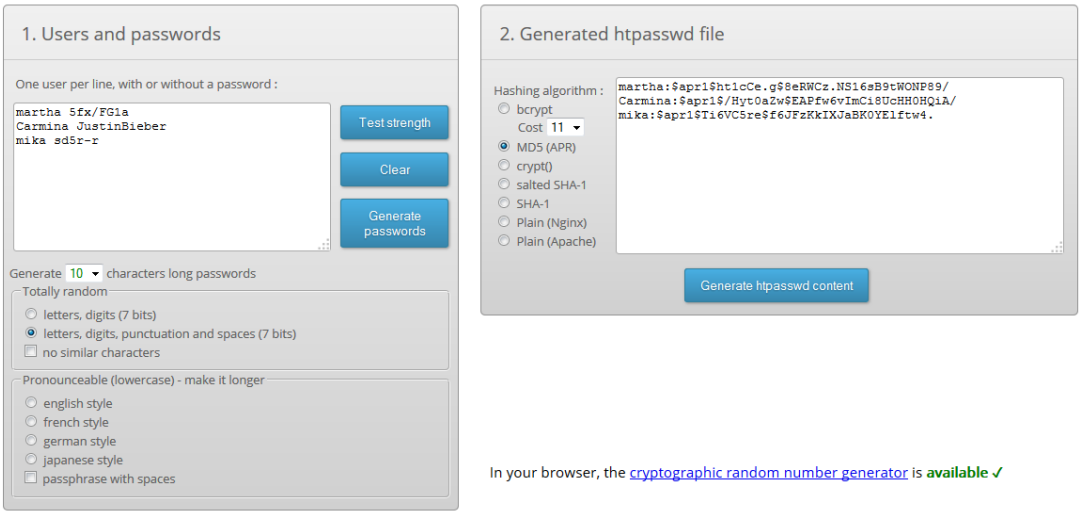
  Si deseamos crear un nuevo fichero, autorizando al usuario 'adminhttpauth', introduciremos el siguiente comando y su contraseña:

```
$ 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 caso acceso a la herramienta htpasswd dentro de nuestro servidor local o cualquier otra máquina de la que se disponga acceso.

112. 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.

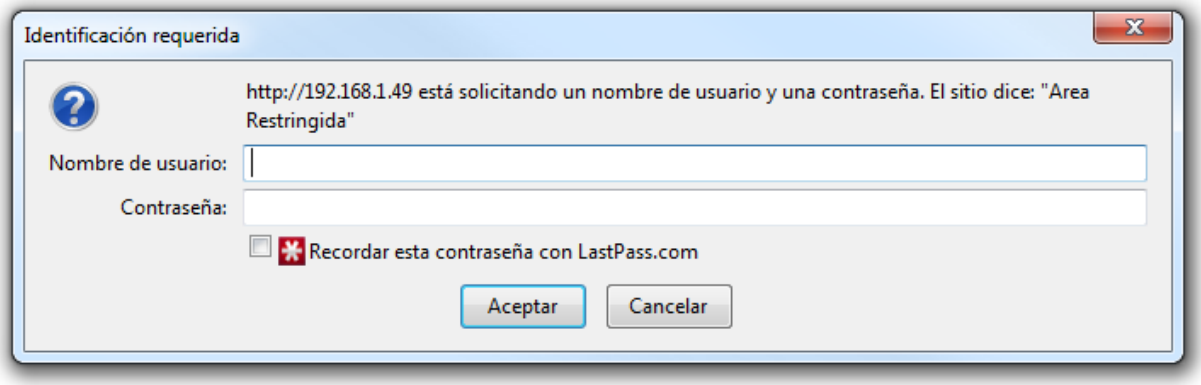
113. También generaremos otro, denominado '.htaccess', que subiremos al directorio '/wp-admin' de nuestra instalación de Wordpress. Si existiera ya el citado fichero, realizaremos una copia de seguridad de éste y procederemos a introducir los valores que se muestran a continuación:

```
# 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
```

114. 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
# Cambio de propietario y de grupo para .htaccess (cambiar usuario y grupo por apropiados)
 chown USUARIO:GRUPO /ruta/a/.htaccess
# Cambio de propietario y de grupo para .htpasswd (cambiar usuario y grupo por apropiados)
 chown USUARIO:GRUPO /ruta/a/.htpasswd
```

115. Ahora trataremos de nuevo de acceder a nuestro panel de administración de nuestro Wordpress. A diferencia de accesos anteriores, ahora encontraremos un cuadro de autenticación solicitando unas credenciales correctas 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.

116. Algunos plugins de Wordpress utilizan Ajax, por lo que es posible que necesiten acceder al fichero 'admin-ajax.php' que se encuentra dentro del directorio protegido 'wp-admin'. Para permitir el acceso anónimo a este fichero, añadiremos el siguiente código a nuestro fichero '.htaccess' en caso necesario:

```
<Files admin-ajax.php>
    Order allow,deny
    Allow from all
    Satisfy any
</Files>
```

NOTA FINAL:
Si nuestro servidor no utiliza Apache sino Nginx, podremos proteger nuestro acceso a 'wp-login.php' utilizando HttpAuthBasicModule. Este bloque debería ser incluído:

```
location /wp-login.php {
   auth_basic "Administrator Login";
   auth_basic_user_file .htpasswd;
}
```

El formato del fichero .htpasswd debe ser el siguiente:

```
Usuario:Password
Usuario2:Password2
```

Las contraseñas deben estar cifradas con crypt. Podemos utilizar el siguiente comando en Perl para transformar de forma adecuada la contraseña elegida:

$ perl -le 'print crypt("mipassword", "salt-hash")'

Para permitir acceso a los plugins que utilicen ajax, incluiremos este código de forma adicional:

location /wp-admin/admin-ajax.php {
allow all;
}

7.5. RESTRINGIENDO EL ACCESO A WP-ADMIN POR DIRECCIÓN IP

117. Si disponemos de una dirección IP estática para acceder al panel de administración de Wordpress, incluiremos una serie de filtros para permitir que sólo determinadas direcciones puedan acceder.

118. Para ello sólo tenemos que introducir una serie de valores en el fichero .htaccess. En caso de utilizar el fichero anteriormente generado, añadiremos lo necesario al final de éste. En nuestro caso permitiremos el acceso desde la dirección IP local 192.168.5.1, pero puede introducirse cualquier dirección IP, incluidas las públicas:

```
order deny,allow
allow from 192.168.5.1
deny from all
```

119. También es posible incluir diferentes direcciones IP, añadiendo más líneas con 'allow from', como en el siguiente ejemplo:

```
order deny,allow
allow from 192.168.5.1
allow from 10.130.130.17
deny from all
```

120. Como vemos, tan sólo las direcciones IP especificadas en el archivo tendrán acceso. El resto serán bloqueadas con la orden final 'deny from all'.

NOTA:
Esta protección es también sencilla de implementar en servidores Nginx.. El siguiente bloque especifica la sintaxis para agregar direcciones IP con acceso al directorio 'wp-admin':

```
location /wp-admin {
  deny    192.168.1.1;
  allow   192.168.1.0/24;
  allow   10.1.1.0/16;
  deny    all;
}
```

7.6. USO DE AUTENTICACIÓN DE DOBLE FACTOR

121. 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.

122. 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.

123. En abril del año pasado, Wordpress anunció que iba a permitir a los usuarios iniciar sesión con autenticación de dos pasos, mejorando notablemente la seguridad.

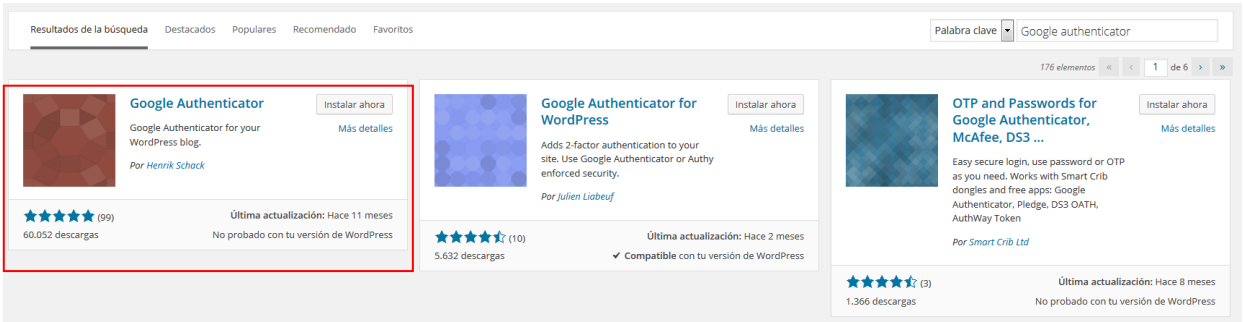
124. El inicio de una sesión con una contraseña, es la autenticación de un solo paso. Se basa únicamente en algo que una persona sabe. La autenticación de dos pasos, por definición, es un sistema en el que utiliza dos de los tres factores posibles para probar la identidad, en lugar de sólo uno:

- **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 tipos 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)

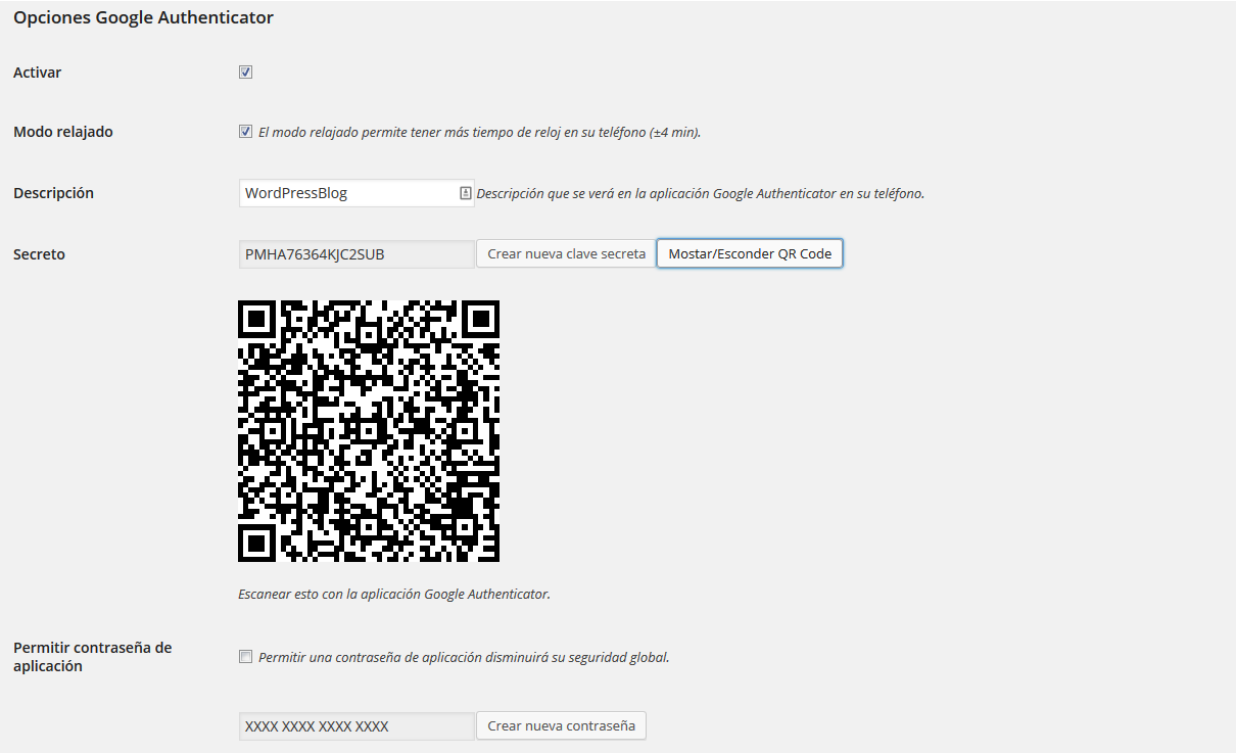
125. Existen gran cantidad de plugins en el repositorio oficial de Wordpress. Los más populares, ordenados de forma alfabética, son:

- ✓ Authy
- ✓ Clef
- ✓ Duo
- ✓ Google Authenticator
- ✓ WordFence

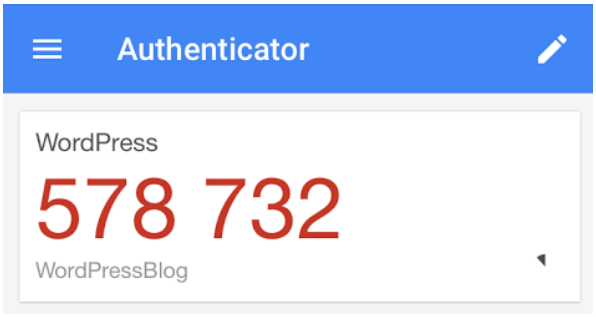
126. Cualquiera de estos plugins puede ser instalado en el sistema, aunque por sencillez mostraremos cómo realizar la instalación y configuración de 'Google Authenticator'. La instalación se realiza de la manera habitual, desde los repositorios oficiales de Wordpress. Desde 'Plugins/Añadir nuevo', buscaremos 'Google Authenticator' y pulsaremos sobre el botón 'Instalar ahora' en la primera coincidencia:



127. Una vez instalado, activaremos el plugin y podremos configurarlo desde cada perfil de cada usuario. En este ejemplo vamos a configurarlo para nuestra cuenta de administrador de Wordpress, accediendo a 'Usuarios / Tu Perfil'. Marcaremos la casilla 'Activar' de 'Opciones Google Authenticator' y pulsaremos sobre 'Mostrar/Esconder QR Code':



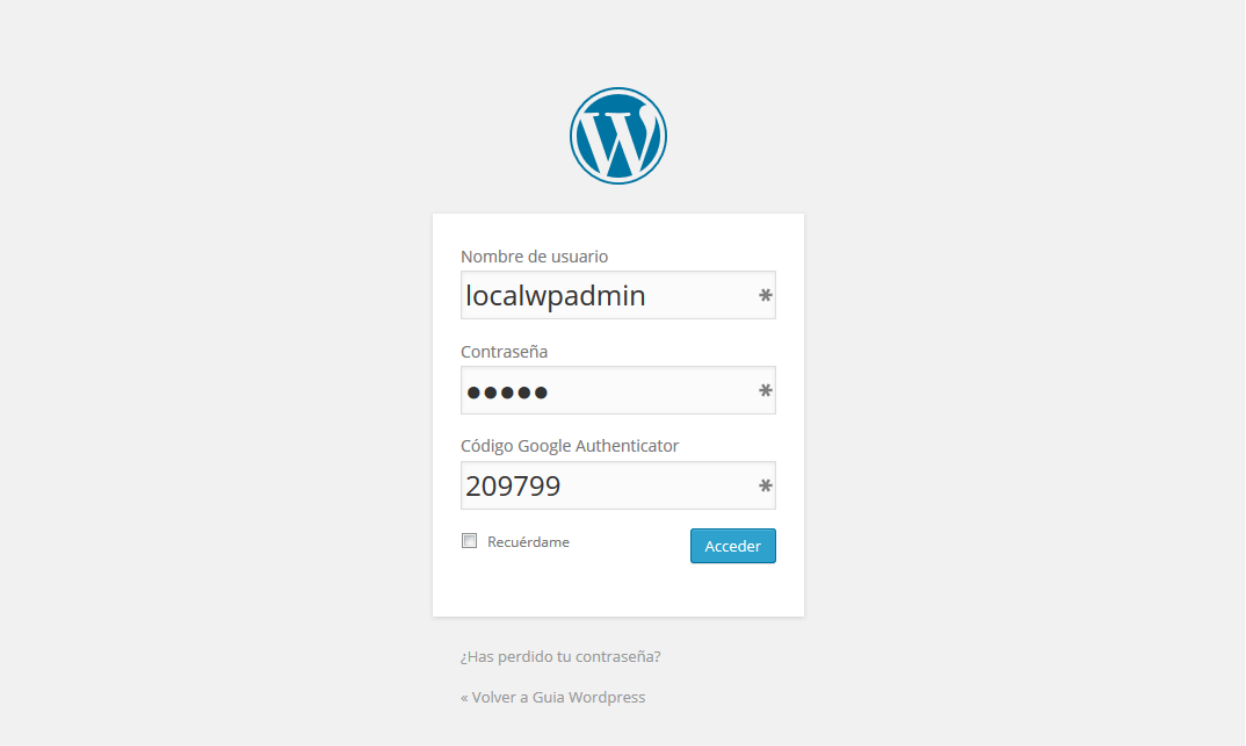
128. Una vez hayamos mostrado el código QR generado de forma automática en base a la clave secreta, deberemos descargarnos la aplicación de 'Google Authenticator' en nuestro dispositivo Smartphone y escanear el código mostrado por la pantalla. Acto seguido comenzará a mostrarnos los códigos necesarios para acceder al sistema con la autenticación de doble factor para nuestro usuario:



129. Antes de comprobar que está funcionando todo correctamente, debemos asegurarnos que la hora de nuestro reloj del Smartphone y del servidor están correctamente ajustadas y sincronizadas, debido a que los códigos que genera Google Authenticator están basados en la hora/fecha que fueron generados.

130. Podemos instalar 'ntp' en el servidor, y utilizar el ajuste de hora automática disponible en la mayoría de sistemas iOS, Android, Windows 8, etc. que se ajustará de forma automática con la facilitada por nuestro proveedor de servicios de red.

131. Una vez que hayamos comprobado la correcta sincronización de hora de todos los dispositivos involucrados, procederemos a cerrar nuestra sesión e intentar acceder de nuevo. Deberíamos ver un nuevo campo que ha sido añadido a la pantalla de acceso:



132. Para acceder, tan sólo deberemos introducir nuestras credenciales anteriores, añadiendo el código que nos facilitará 'Google Authenticator'.

NOTA:
En caso necesario, el *plugin* 'Google Authenticator' puede ser deshabilitado temporalmente o de forma definitiva entrando en la línea de comandos de nuestro servidor, y renombrando el directorio del *plugin*, dentro de 'wp-content/plugins' de 'google-authenticator' a, por ejemplo, '_google-authenticator'.

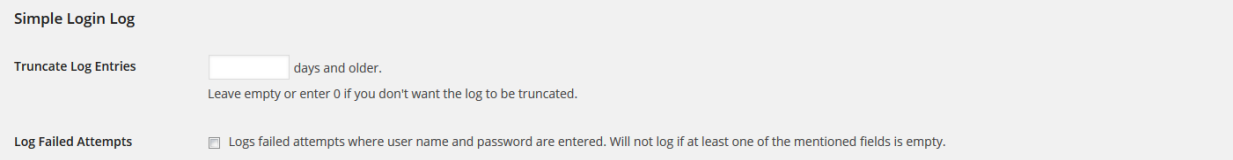
Una vez recuperada nuestra sesión como administrador, deberemos renombrar de nuevo el directorio a su nombre original, y reactivarlo desde la sección de *plugins* de Wordpress.

7.7. REGISTRO DE ACCESOS DE INICIO DE SESIÓN

133. Una de las características adicionales que instalaremos en nuestro Wordpress es la capacidad de obtener el registro de intentos de acceso al sistema. Para ello, instalaremos el plugin 'Simple Login Log' que nos permitirá obtener los intentos de acceso, junto con el nombre de usuario utilizado, la hora de acceso, dirección IP, así como el navegador utilizado.

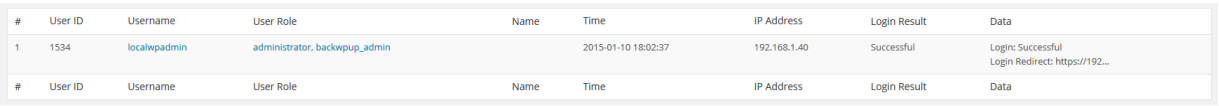
134. Para ello, buscaremos el plugin 'Simple Login Log' dentro del repositorio de Wordpress y realizaremos su instalación de la forma habitual.

135. Una vez instalado y activado, podremos acceder a su configuración dentro de 'Ajustes / Generales' y desplazaremos el scroll hasta el final de la página:



136. Como vemos, nos permitirá especificar un número de días a los que rotar el log generado, así como si preferimos que se guarde el registro solamente cuando se hayan introducido tanto el nombre de usuario como la contraseña de acceso.

137. Para visualizar el registro de los últimos accesos, pulsaremos sobre 'Usuarios / Login Log', donde tendremos incluso la opción de exportarlo a un fichero CSV para procesar ficheros de gran tamaño:



7.8. LIMITAR NÚMEROS DE INTENTOS DE INICIO DE SESIÓN

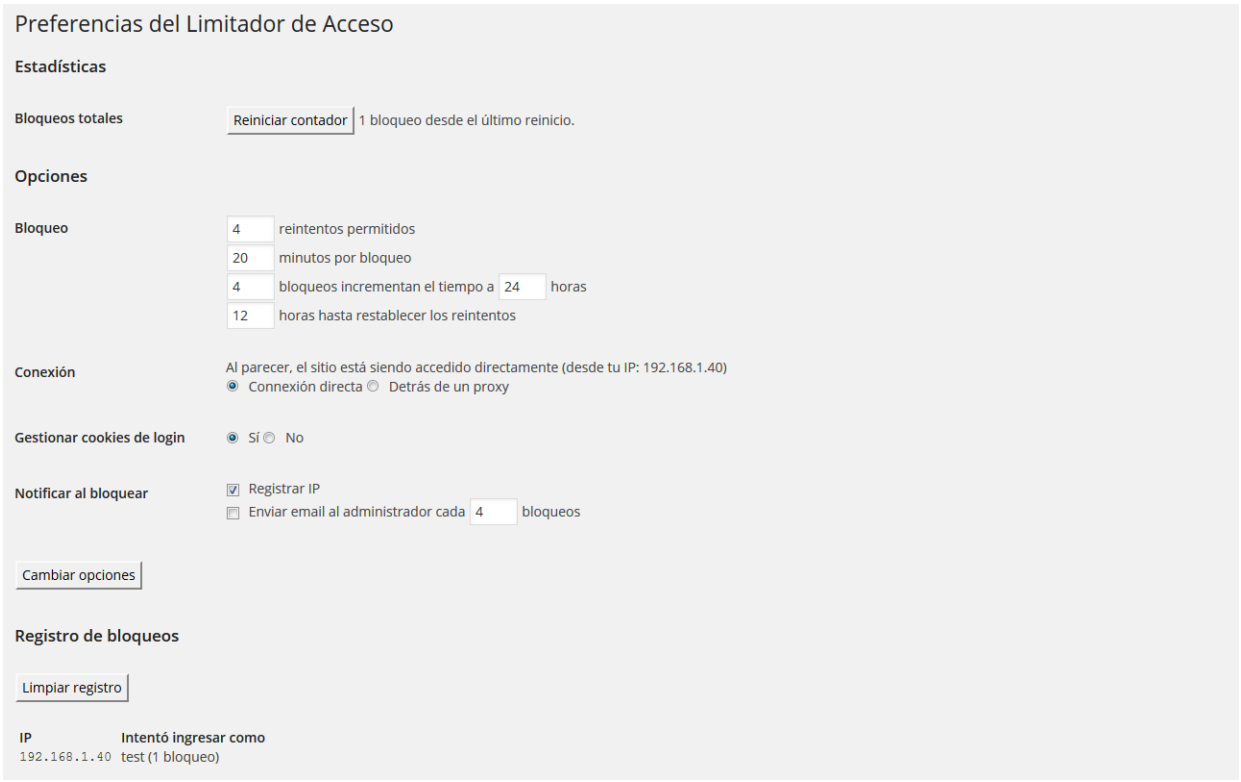
138. Según informa WordFence.com, la media de intentos fraudulentos de acceso a sitos WordPress viene a ser de unos 2.000 por minuto, utilizando ataques de fuerza bruta. Como hemos visto en los puntos anteriores, es necesario introducir diferentes medidas preventivas para evitar este tipo de ataques.

139. En este apartado, veremos un nuevo plugin, denominado 'Limit Login Attempts', que nos permitirá bloquear el acceso de posibles atacantes que intenten realizar ataques de fuerza bruta sobre nuestro sistema, además de:

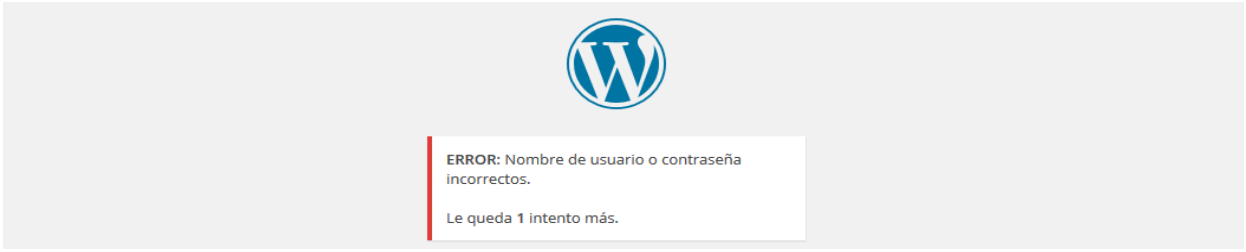
- Limitar el número de intentos por dirección IP
- Limitar el número de intentos por cookie
- Registro opcional de los intentos realizados
- Posibilidad de utilizar una lista blanca de direcciones IP

140. Realizaremos la instalación buscando el plugin 'Limit Login Attempts' dentro del repositorio de Wordpress y realizaremos su instalación y posterior activación de la forma habitual.

141. Accederemos a la configuración de los parámetros dentro de 'Ajustes / Limit Login Attempts'. Desde aquí podremos realizar diversas tareas, entre las que cabe destacar la configuración de la política de intentos para los bloqueos, liberar los bloqueos activos o la notificación de eventos al administrador, etc.



142. Una vez configurados estos datos, comprobaremos el correcto funcionamiento del plugin desde la página de inicio de sesión. Según vayamos introduciendo valores erróneos de usuario y contraseña, el sistema nos mostrará una ventana donde nos informará del número restante de intentos que nos están permitidos antes de que se ejecute la política de intentos de acceso:



143. Una vez hayamos consumido el número de intentos especificados en las opciones de configuración, nuestra dirección quedará registrada y bloqueada hasta que se cumpla el tiempo que hayamos establecido. Además, recibiremos el siguiente mensaje por pantalla:

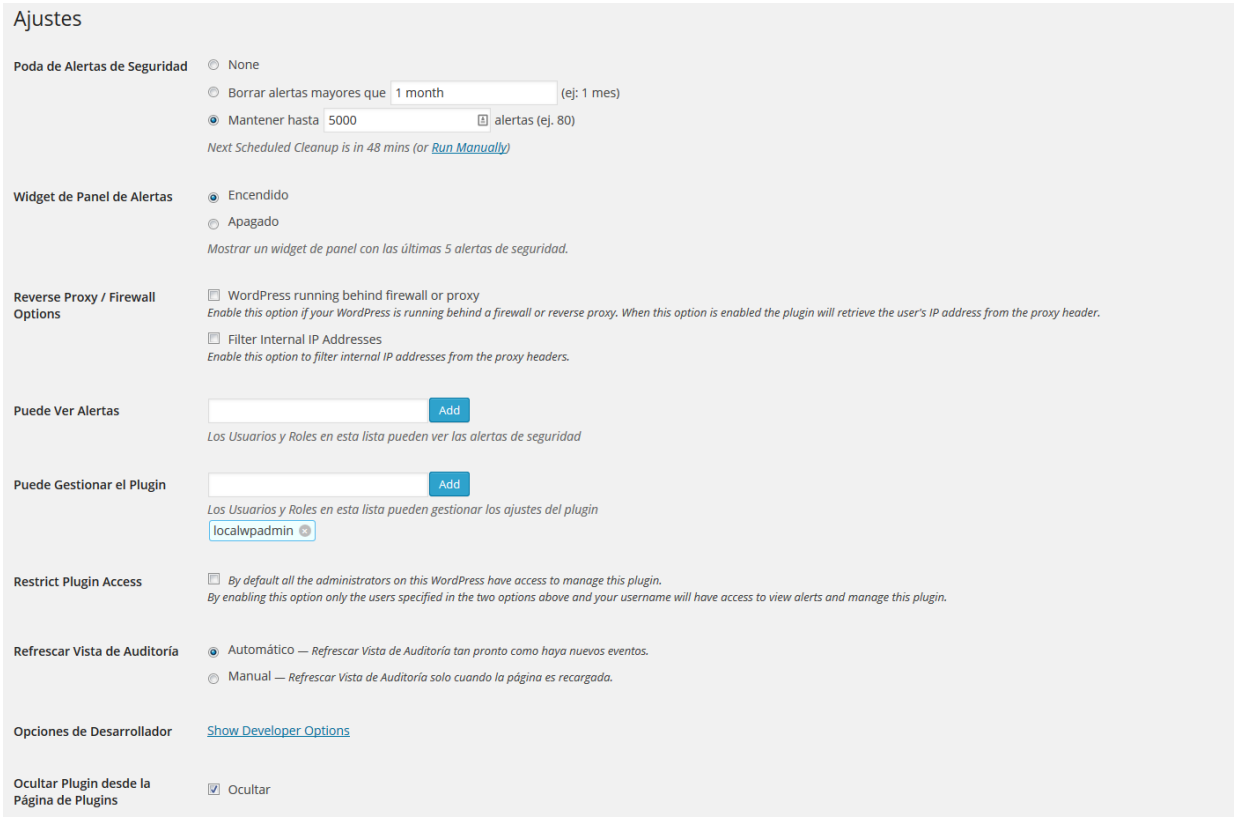
8. REGISTRO DE ACTIVIDAD DEL SISTEMA

144. A pesar de que en el apartado anterior hemos visto cómo es posible instalar el plugin 'Simple Login Log' para mantener un registro de los intentos de acceso al sistema, en este apartado veremos el plugin 'WP Security Audit Log', que nos permitirá monitorizar gran cantidad de eventos del sistema, entre los que podemos destacar:

- Creación de nuevos usuarios
- Cambios en el perfil (rol, password o cualquier otro campo) de un usuario
- Modificación de ficheros
- Creación/modificación de *posts*, páginas o categorías
- Instalación de nuevos temas
- Creación/modificación de *widgets*
- Intentos de inicio de sesión
- Actualizaciones del sistema Wordpress o derivados
- Etc.

145. Como vemos, este nuevo plugin añade gran cantidad de funcionalidad al anterior, permitiéndonos tener una bitácora de todas las actividades realizadas desde el panel de administración de Wordpress.

146. La instalación se realiza de la forma habitual dentro del repositorio de plugins de Wordpress, y una vez activado pulsaremos sobre 'Audit Log' en el menú lateral de nuestro panel de administración. La primera tarea que realizaremos será la configuración del plugin:



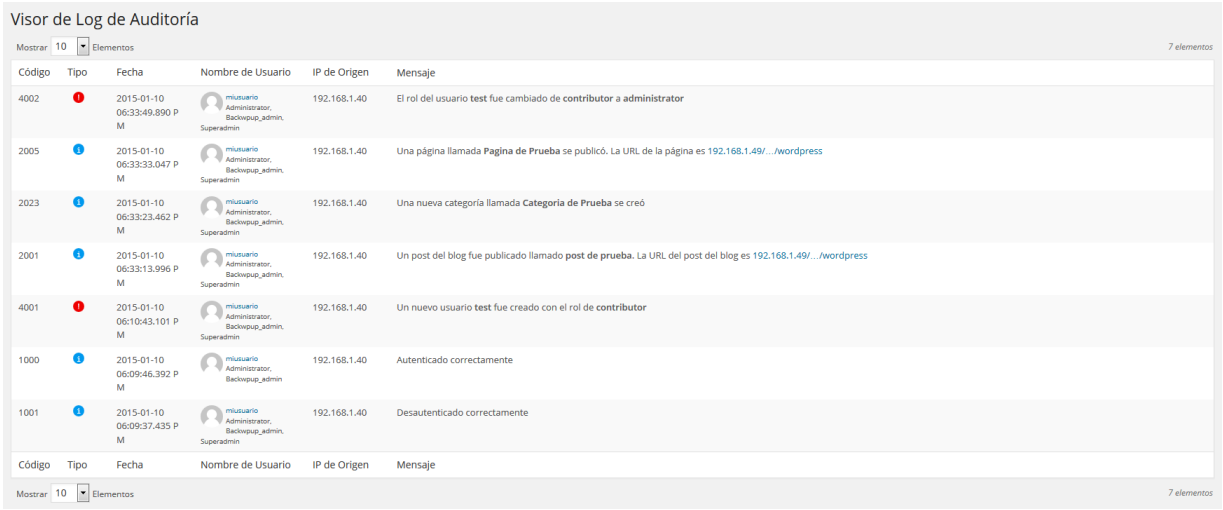
147. Dentro de las opciones de configuración, especificaremos aquellas en las que estemos interesados, y además marcaremos la opción de 'Ocultar Plugin desde la Página de Plugins', lo que evitará mostrar que el plugin está instalado desde la página de plugins habitual.

148. Dentro de 'Habilita/Deshabilita Alertas' podremos especificar qué eventos vamos a registrar en nuestra bitácora. Como podemos ver existen diferentes pestañas:

- Actividad del Sistema
- Multisitio
- Otra Actividad de Usuario
- Perfiles de Usuario
- *Plugins* y Temas
- *Posts* Personalizados
- *Posts* del Blog
- Páginas
- *Widgets*

149. En nuestro caso, mantendremos activadas todas las opciones de cada una de las pestañas para mantener un control completo de la actividad de nuestro Wordpress.

150. Finalmente, podremos observar esta bitácora de actividad dentro de 'Audit Log / Visor de Log de Auditoría', donde tendremos la información y detalle de las últimas acciones llevadas a cabo por los usuarios del sistema:



9. RESTRICCIÓN DE ACCESO A DIRECTORIOS

151. Deberemos incluir una segunda capa de protección a nuestro directorio 'wp-includes', ya que almacena todos los scripts y secuencias de comandos que, por norma general, no deberían ser accesibles por cualquier usuario.

152. Utilizaremos, de nuevo, el fichero '.htaccess' y mod_rewrite, e incluiremos el siguiente código dentro del fichero:

```
# Block the include-only files.
<IfModule mod_rewrite.c>
  RewriteEngine On
  RewriteBase /
  RewriteRule ^wp-admin/includes/ - [F,L]
  RewriteRule !^wp-includes/ - [S=3]
  RewriteRule ^wp-includes/[^/]+\.php$ - [F,L]
  RewriteRule ^wp-includes/js/tinymce/langs/.+\.php - [F,L]
  RewriteRule ^wp-includes/theme-compat/ - [F,L]
</IfModule>
```

NOTA:
Esta protección puede no funcionar correctamente en las instalaciones Multisite de Wordpress, dado que la línea 'RewriteRule ^wp-includes/[^/]+\.php$ – [F,L]' impedirá que 'ms-files.php' genere imágenes de forma correcta.

Deshabilitando esta regla, funcionará de nuevo correctamente en caso de que hubiera generado mal funcionamiento, a pesar de que ofrecerá una menor seguridad

10. RESTRICCIÓN DE ACCESO A FICHEROS SENSIBLES

153. Existen una serie de ficheros que, una vez instalado Wordpress, no deberían estar accesibles de ninguna forma al público, y que bloquearemos incluyendo las siguientes líneas en el fichero '.htaccess' de nuestro servidor:

```
## Restriccion de acceso a ficheros sensibles
Options All -Indexes

<files .htaccess>
        Order allow,deny
        Deny from all
</files>
<files readme.html>
        Order allow,deny
        Deny from all
</files>
<files license.txt>
        Order allow,deny
        Deny from all
</files>
<files install.php>
        Order allow,deny
        Deny from all
</files>
<files wp-config.php>
        Order allow,deny
        Deny from all
</files>
<files error_log>
```

```
        Order allow,deny
        Deny from all
</files>
<files fantastico_fileslist.txt>
        Order allow,deny
        Deny from all
</files>
<files fantversion.php>
        Order allow,deny
        Deny from all
</files>
```

11. PROTECCIÓN DEL DIRECTORIO 'UPLOADS'

154. El directorio 'uploads/' de nuestra instalación de Wordpress será uno de los focos principales de ataques, debido a los permisos que necesita para funcionar correctamente. Esto podría proporcionar a un atacante la posibilidad de incluir y ejecutar código malicioso, como por ejemplo, una Shell remota.

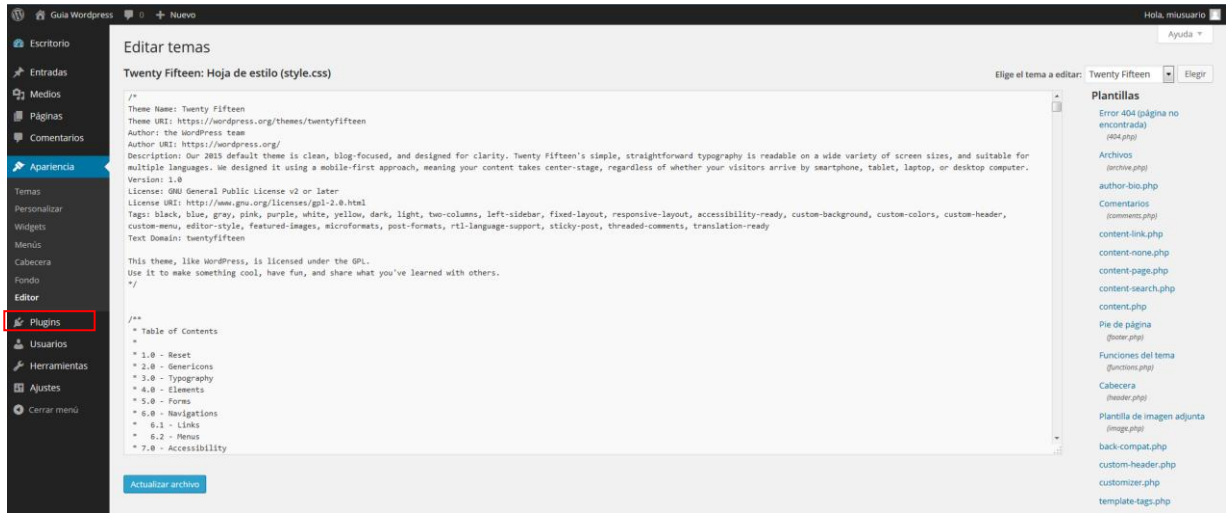
155. Para su correcta configuración, generaremos un fichero '.htaccess' dentro del propio directorio, especificando que sólo podrá contener ficheros de archivos de imagen digital con las extensiones JPG, JPEG, JPE, GIF, PNG, TIF y TIFF:

```
# secure uploads directory
<Files ~ ".*\..*">
        Order Allow,Deny
        Deny from all
</Files>
<FilesMatch "\.(jpg|jpeg|jpe|gif|png|tif|tiff)$">
        Order Deny,Allow
        Allow from all
</FilesMatch>
```

12. DESAHABILITAR EL EDITOR DE FICHEROS EN LÍNEA

156. El panel de administración de Wordpress permite a los administradores, por defecto, la posibilidad de editar los ficheros PHP, plugins o los correspondientes a los temas. Esta capacidad será algo que los atacantes utilizarán si consiguen el acceso a nuestro sistema, ya que les permitirá ejecución de código a través de los diferentes ficheros nombrados.

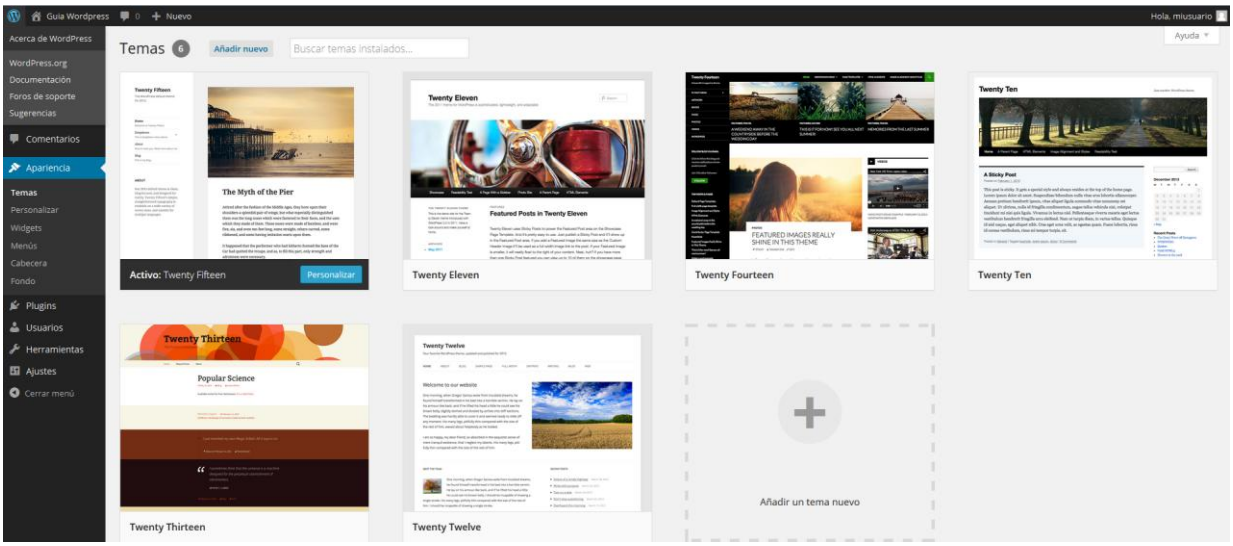
157. Wordpress pone a nuestra disposición una constante que anulará esta opción de edición desde el panel de administración, y se encuentra en 'Apariencia / Editor' como vemos a continuación:



158. Para eliminarlo introduciremos la siguiente constante en el fichero de configuración 'wp-config.php', que será equivalente a eliminar los permisos 'edit_themes', 'edit_plugins' y 'edit_files' de un usuario:

define('DISALLOW_FILE_EDIT', true);

159. Una vez introducido el cambio, veremos que desde nuestro panel ha desaparecido la opción de editor:



160. A pesar de que esta medida no impedirá a un atacante subir archivos maliciosos a nuestro sitio, será de gran utilidad para evitar otro tipo de ataques.

NOTA:
Si lo deseamos, también existe una opción más restrictiva que, además de bloquear el editor, impedirá a los usuarios instalar, modificar o actualizar *plugins* y temas a través del panel de administración. La constante a incluir en nuestro fichero 'wp-config.php' será:

define('DISALLOW_FILE_MODS', true);

13. ADMINISTRACIÓN Y NAVEGACIÓN SOBRE SSL

161. Como hemos visto anteriormente, reforzar el panel de administración de Wordpress es una tarea vital, ya que existen muchos factores externos que pueden amenazar su seguridad a pesar de las diferentes medidas que vayamos incorporando.

162. Por este motivo, vamos a activar y forzar la administración de Wordpress sobre 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 Apache2
$ 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 dias 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 adaptarlo a nuestras necesidad, 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
```

163. Existen tres constantes dentro de Wordpress que nos van a permitir definir esta situación dentro del fichero ‘wp-config.php’, son las siguientes:

- **FORCE_SSL_ADMIN**: la más importante para nuestro fin, ya que forzará a que todos los inicios de sesión de usuario y de administrador ocurran bajo SSL. Para ello, deberemos incorporar el siguiente código:

  define('FORCE_SSL_ADMIN', true);

- **FORCE_SSL_LOGIN**: nos permitirá forzar que los accesos al sistema como usuario se realicen vía SSL, de tal forma que las contraseñas no sean enviadas en texto claro, sin obligar a realizar esta tarea al administrador. El código correspondiente será:

  define('FORCE_SSL_LOGIN', true);

NOTA:
Estos valores deben ser especificados al inicio del fichero ‘wp-config.php’, por lo que si al incluirlos en la configuración no detectamos que el panel de control nos redirige a una sesión segura SSL, deberemos editar de nuevo el fichero e incluir el código correspondiente en las líneas superiores del fichero.

164. Si nuestro servidor de Wordpress está alojado detrás de un proxy inverso que proporciona SSL, pero nuestra máquina no lo dispone, estas opciones generarán un bucle infinito que impedirá su funcionamiento. Para evitar este caso, será necesario configurar Wordpress para reconocer la cabecera HTTP_X_FORWARDED_PROTO (que debemos haber configurado previamente en nuestro proxy inverso).

165. Incluiremos el siguiente código dentro de ‘wp-config.php’:

```
define('FORCE_SSL_ADMIN', true);
if ($_SERVER['HTTP_X_FORWARDED_PROTO'] == 'https')
    $_SERVER['HTTPS']='on';
```

166. Si deseamos forzar al cliente a utilizar HTTPS para la visualización del contenido de nuestro Wordpress, incluiremos la siguiente directiva dentro del fichero .htaccess del directorio raíz de la aplicación:

```
ReWriteEngine On
ReWriteCond %{SERVER_PORT} 80
ReWriteCond ^(.*)$ https://direccion_wordpress/$1 [R,L]
```

167. En el ejemplo anterior sustituiremos 'direccion_wordpress' por la ruta completa, utilizando dirección IP o dominio, de la instalación de la aplicación. Si utilizaramos el domino 'www.ejemplo.com' y Wordpress estuviera instalado en el directorio '/blog', la sentencia definitiva sería:

```
ReWriteCond ^(.*)$ https://www.ejemplo.com/blog/$1 [R,L]
```

14. PREVENCIÓN DE SPAM

14.1. AKISMET

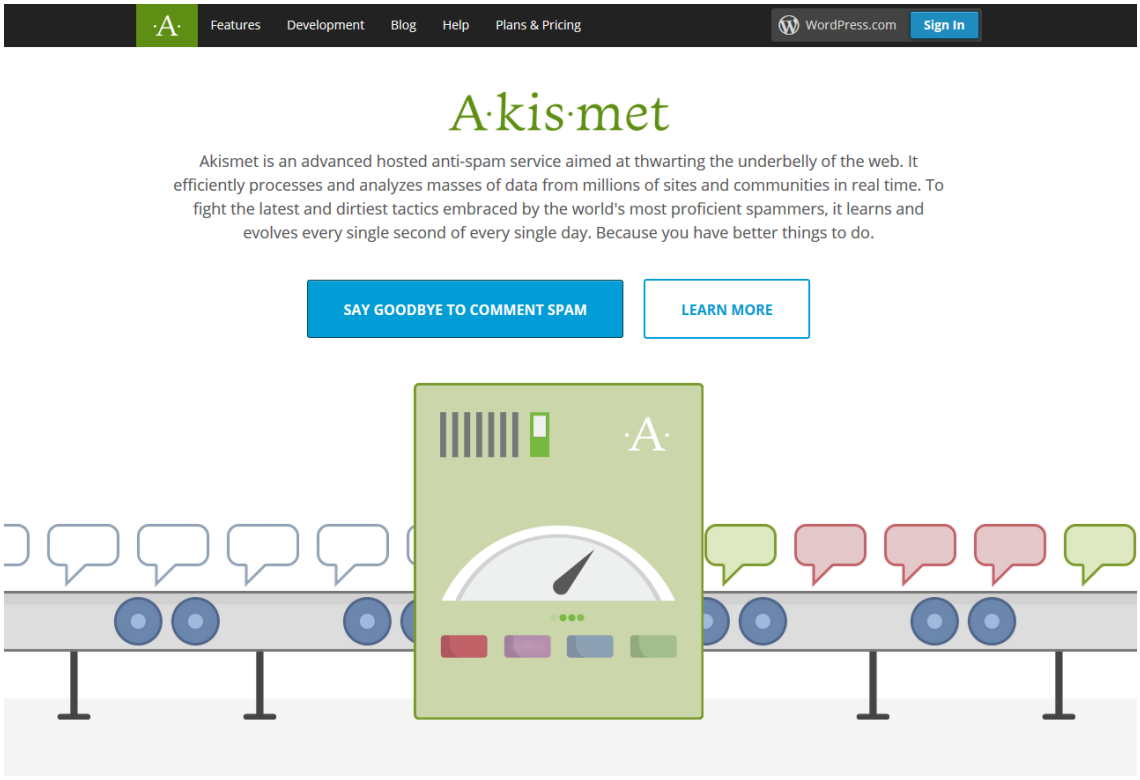
168. El spam genera diariamente un elevado tráfico de datos en Internet que acaba afectando a todos. De una u otra forma sufrimos las consecuencias y en cientos de ocasiones acabamos siendo víctimas de las mafias que ven en el spam un lucro, lo que obliga a empresas, administradores de sistemas y a los propios usuarios a implementar medidas adicionales para combatirlo.

169. No hay más que echar un vistazo a las estadísticas de spam de Akismet (7 millones y medio de bloqueos a la hora) para darse cuenta de que estamos frente a un problema cada vez más serio, que debemos frenar en nuestros sistemas.

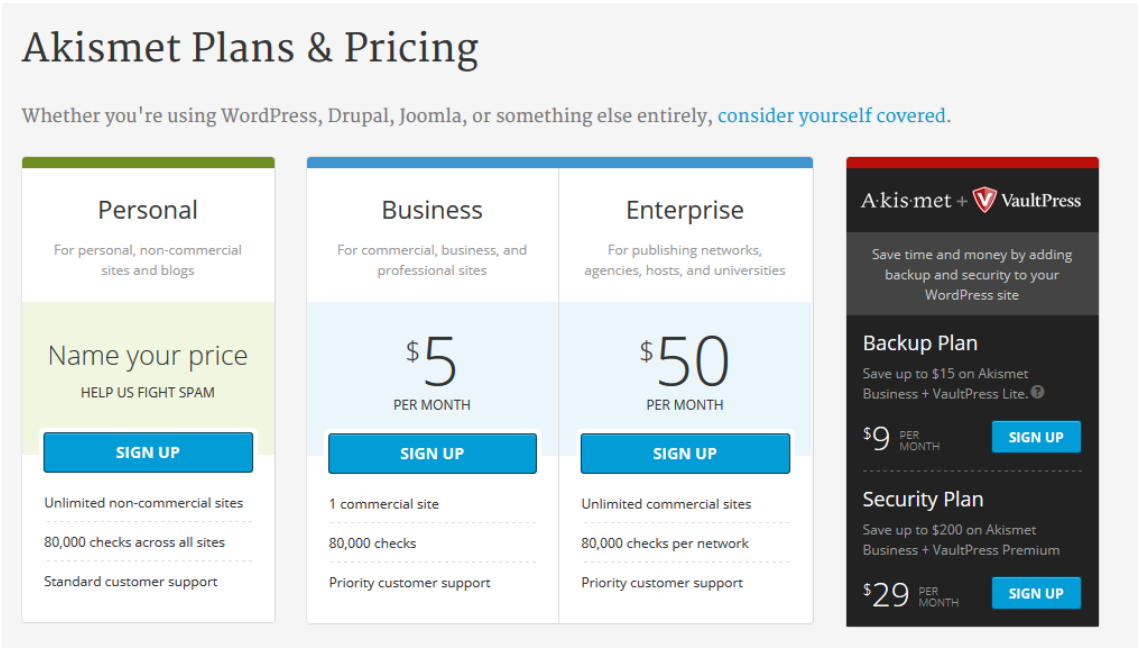
170. Esta tarea es relativamente sencilla de realizar si hacemos uso del plugin Akismet para WordPress, el cual nos permite bloquear todos aquellos comentarios de spam que pudieran llegar hasta nuestro blog a través de formularios de comentarios o cualquier otra entrada de datos.

171. Akismet viene por defecto incorporado en WordPress, por lo que en primer lugar lo más importante es asegurarnos de estar utilizando la última versión estable del plugin y crearnos una cuenta asociada y obtener una clave API para poder activar el plugin. Para ello, realizaremos los siguientes pasos:

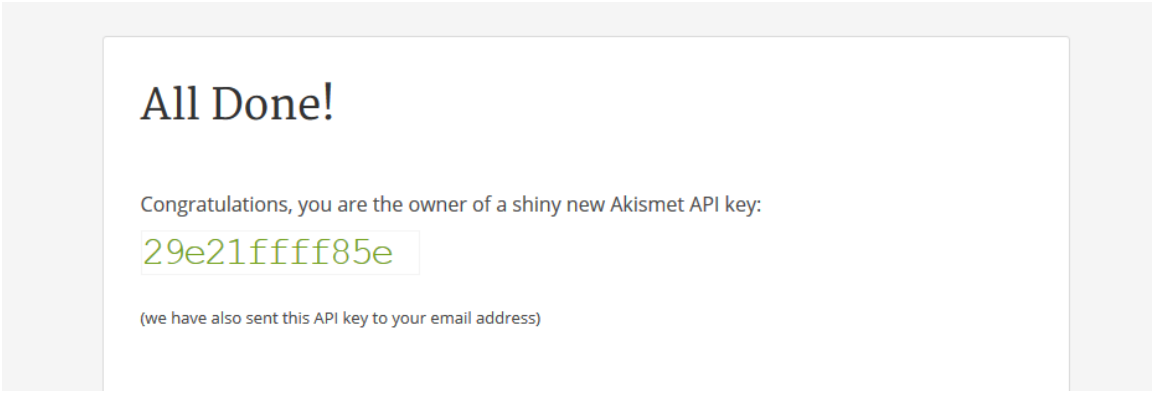
1. Accedemos a la página principal http://www.akismet.com y pulsamos sobre el botón ‘Say goodbye to comment *spam’*



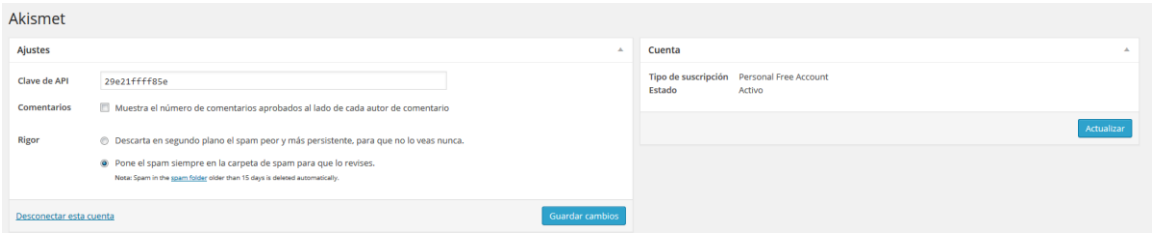
2. Deberemos generar una cuenta nueva. Seleccionaremos la cuenta más adecuada a nuestras necesidades y pulsaremos 'Sign up' (para la realización de esta guía hemos seleccionado el plan 'Personal' que es gratuíto):



3. Una vez introducidos los valores necesarios, como dirección de correo, nombre de usuario, contraseña, etc. recibiremos el valor de API necesario para activar el plugin en nuestro sistema:

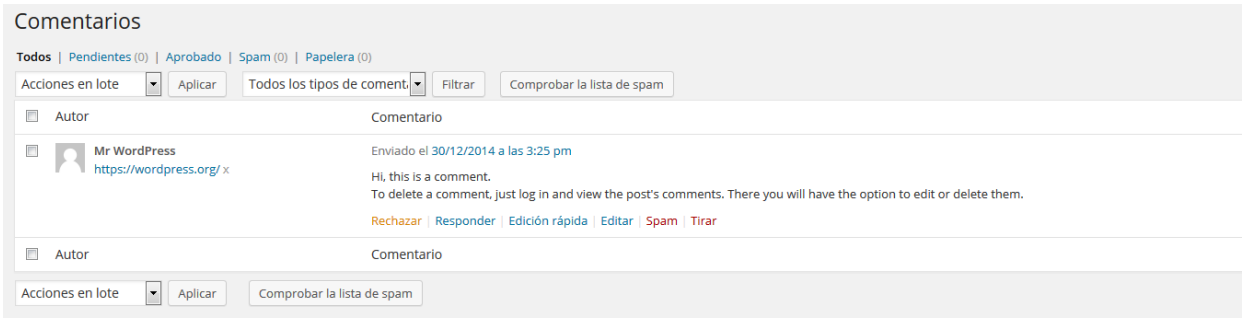


4. Volveremos a nuestro panel de administración de Wordpress, y accederemos a 'Plugins / Plugins instalados'. Procederemos a activar Akismet y pulsaremos sobre 'Ajustes' para introducir el valor de la API que nos ha proporcionado la web, y hemos recibido también por correo electrónico:

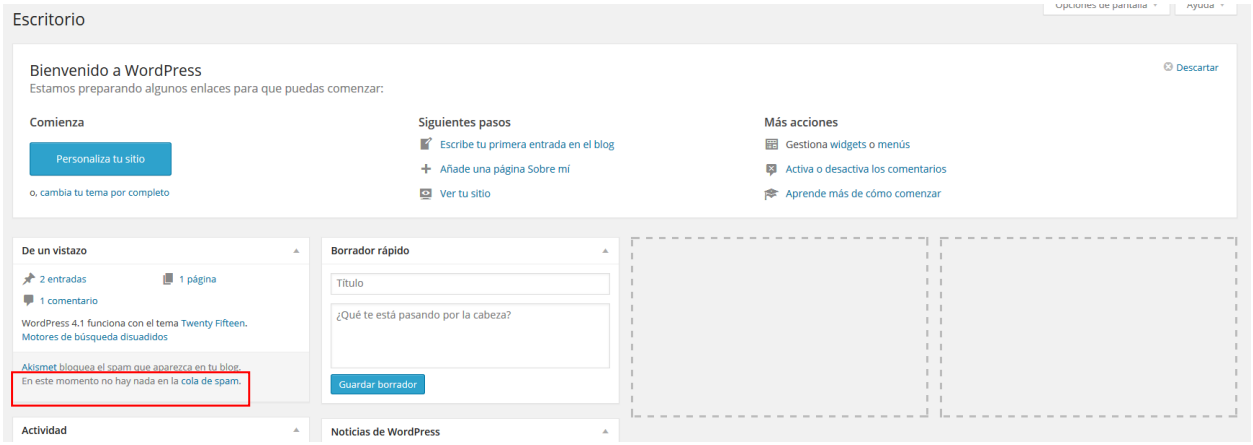


5. Una vez finalizados los pasos, el plugin estará activo y comenzará a protegernos del spam. Se recomienda mantener activada la opción 'Pone el spam siempre en la carpeta de spam para que lo revises', con el fin de poder descartar posibles falsos positivos del servicio.

172. A partir de ahora, dentro del menú 'Comentarios', tendremos una nueva opción para marcar los que consideremos como spam:



173. Además, desde el Escritorio del panel de administración, tendremos visibilidad del funcionamiento de Akismet, así como de los comentarios bloqueados:



14.2. REFERER SPAM

174. Cuando nuestros lectores realizan un comentario, el sistema accederá al fichero 'wp-comments-post.php' para realice una serie de tareas y se cree el mensaje. Durante este proceso, el navegador de nuestros visitantes enviará una petición con el campo 'Referer' incluido. 'Referer' es una cabecera HTTP que identifica la dirección de la página web (es decir, la URI o IRI) que creó el vínculo con el recurso que está siendo solicitado. A través del chequeo del campo, la página web puede determinar dónde se originó la solicitud.

175. Cuando se trata de un bot de spam, intentará incluir su información directamente en nuestro sistema sin incluir normalmente un campo 'Referer' ya que la petición no se ha realizado navegando por nuestro sitio web. Esto nos permitirá capturar este tipo de peticiones y bloquearlas de forma sencilla.

176. Para ello, tendremos que incluir el siguiente código dentro del fichero '.htaccess' de la raíz de nuestra instalación de Wordpress:

```
RewriteEngine On
RewriteCond %{REQUEST_METHOD} POST
RewriteCond %{REQUEST_URI} .wp-comments-post\.php*
RewriteCond %{HTTP_REFERER} !.*nuestrodominio.com.* [OR]
RewriteCond %{HTTP_USER_AGENT} ^$
RewriteRule (.*) http://%{REMOTE_ADDR}/$ [R=301,L]
```

177. Este código se encargará de:

a. Detectar cuando se realiza una petición POST

b. Comprobar si la petición está dirigida al fichero 'wp-comments-post.php'

c. Comprobar si existe el campo Referer dentro de la petición, si pertenece a nuestro sitio o si sencillamente no ha sido incluida

d. Enviar al *bot* de *spam* de vuelta a la dirección IP que originó la petición

NOTA:
Existe una pequeña posibilidad de que el navegador de alguno de nuestros visitantes no envíe el campo 'Referer' y sea bloqueado a la hora de enviar comentarios, aunque esto es un caso extremadamente raro.

14.3. LISTA NEGRA BASADA EN LA CABECERA 'USER-AGENT'

178. A continuación se muestra una lista negra a incluir en nuestro servidor que bloqueará gran cantidad de los bots de spam que se utilizan en la actualidad. Sólo deberemos introducirla en nuestro fichero '.htaccess' para que automáticamente localice alguno de los User-Agent utilizados para generar spam y realice el bloqueo:

```
# Deny domain access to spammers and other scumbags
RewriteEngine on
RewriteBase /
RewriteCond %{HTTP_USER_AGENT} almaden [OR]
RewriteCond %{HTTP_USER_AGENT} ^Anarchie [OR]
RewriteCond %{HTTP_USER_AGENT} ^ASPSeek [OR]
RewriteCond %{HTTP_USER_AGENT} ^attach [OR]
RewriteCond %{HTTP_USER_AGENT} ^autoemailspider [OR]
RewriteCond %{HTTP_USER_AGENT} ^BackWeb [OR]
RewriteCond %{HTTP_USER_AGENT} ^Bandit [OR]
RewriteCond %{HTTP_USER_AGENT} ^BatchFTP [OR]
RewriteCond %{HTTP_USER_AGENT} ^BlackWidow [OR]
RewriteCond %{HTTP_USER_AGENT} ^Bot\ mailto:craftbot@yahoo.com [OR]
RewriteCond %{HTTP_USER_AGENT} ^Buddy [OR]
RewriteCond %{HTTP_USER_AGENT} ^bumblebee [OR]
RewriteCond %{HTTP_USER_AGENT} ^CherryPicker [OR]
RewriteCond %{HTTP_USER_AGENT} ^ChinaClaw [OR]
RewriteCond %{HTTP_USER_AGENT} ^CICC [OR]
RewriteCond %{HTTP_USER_AGENT} ^Collector [OR]
RewriteCond %{HTTP_USER_AGENT} ^Copier [OR]
RewriteCond %{HTTP_USER_AGENT} ^Crescent [OR]
RewriteCond %{HTTP_USER_AGENT} ^Custo [OR]
RewriteCond %{HTTP_USER_AGENT} ^DA [OR]
RewriteCond %{HTTP_USER_AGENT} ^DIIbot [OR]
RewriteCond %{HTTP_USER_AGENT} ^DISCo [OR]
RewriteCond %{HTTP_USER_AGENT} ^DISCo\ Pump [OR]
RewriteCond %{HTTP_USER_AGENT} ^Download\ Demon [OR]
RewriteCond %{HTTP_USER_AGENT} ^Download\ Wonder [OR]
RewriteCond %{HTTP_USER_AGENT} ^Downloader [OR]
RewriteCond %{HTTP_USER_AGENT} ^Drip [OR]
RewriteCond %{HTTP_USER_AGENT} ^DSurf15a [OR]
RewriteCond %{HTTP_USER_AGENT} ^eCatch [OR]
RewriteCond %{HTTP_USER_AGENT} ^EasyDL/2.99 [OR]
RewriteCond %{HTTP_USER_AGENT} ^EirGrabber [OR]
RewriteCond %{HTTP_USER_AGENT} email [NC,OR]
RewriteCond %{HTTP_USER_AGENT} ^EmailCollector [OR]
RewriteCond %{HTTP_USER_AGENT} ^EmailSiphon [OR]
RewriteCond %{HTTP_USER_AGENT} ^EmailWolf [OR]
RewriteCond %{HTTP_USER_AGENT} ^Express\ WebPictures [OR]
RewriteCond %{HTTP_USER_AGENT} ^ExtractorPro [OR]
RewriteCond %{HTTP_USER_AGENT} ^EyeNetIE [OR]
RewriteCond %{HTTP_USER_AGENT} ^FileHound [OR]
RewriteCond %{HTTP_USER_AGENT} ^FlashGet [OR]
RewriteCond %{HTTP_USER_AGENT} FrontPage [NC,OR]
RewriteCond %{HTTP_USER_AGENT} ^GetRight [OR]
RewriteCond %{HTTP_USER_AGENT} ^GetSmart [OR]
RewriteCond %{HTTP_USER_AGENT} ^GetWeb! [OR]
RewriteCond %{HTTP_USER_AGENT} ^gigabaz [OR]
RewriteCond %{HTTP_USER_AGENT} ^Go\!Zilla [OR]
RewriteCond %{HTTP_USER_AGENT} ^Go!Zilla [OR]
RewriteCond %{HTTP_USER_AGENT} ^Go-Ahead-Got-It [OR]
RewriteCond %{HTTP_USER_AGENT} ^gotit [OR]
```

```
RewriteCond %{HTTP_USER_AGENT} ^Grabber [OR]
RewriteCond %{HTTP_USER_AGENT} ^GrabNet [OR]
RewriteCond %{HTTP_USER_AGENT} ^Grafula [OR]
RewriteCond %{HTTP_USER_AGENT} ^grub-client [OR]
RewriteCond %{HTTP_USER_AGENT} ^HMView [OR]
RewriteCond %{HTTP_USER_AGENT} ^HTTrack [OR]
RewriteCond %{HTTP_USER_AGENT} ^httpdown [OR]
RewriteCond %{HTTP_USER_AGENT} .*httrack.* [NC,OR]
RewriteCond %{HTTP_USER_AGENT} ^ia_archiver [OR]
RewriteCond %{HTTP_USER_AGENT} ^Image\ Stripper [OR]
RewriteCond %{HTTP_USER_AGENT} ^Image\ Sucker [OR]
RewriteCond %{HTTP_USER_AGENT} ^Indy*Library [OR]
RewriteCond %{HTTP_USER_AGENT} Indy\ Library [NC,OR]
RewriteCond %{HTTP_USER_AGENT} ^InterGET [OR]
RewriteCond %{HTTP_USER_AGENT} ^InternetLinkagent [OR]
RewriteCond %{HTTP_USER_AGENT} ^Internet\ Ninja [OR]
RewriteCond %{HTTP_USER_AGENT} ^InternetSeer.com [OR]
RewriteCond %{HTTP_USER_AGENT} ^Iria [OR]
RewriteCond %{HTTP_USER_AGENT} ^JBH*agent [OR]
RewriteCond %{HTTP_USER_AGENT} ^JetCar [OR]
RewriteCond %{HTTP_USER_AGENT} ^JOC\ Web\ Spider [OR]
RewriteCond %{HTTP_USER_AGENT} ^JustView [OR]
RewriteCond %{HTTP_USER_AGENT} ^larbin [OR]
RewriteCond %{HTTP_USER_AGENT} ^LeechFTP [OR]
RewriteCond %{HTTP_USER_AGENT} ^LexiBot [OR]
RewriteCond %{HTTP_USER_AGENT} ^lftp [OR]
RewriteCond %{HTTP_USER_AGENT} ^Link*Sleuth [OR]
RewriteCond %{HTTP_USER_AGENT} ^likse [OR]
RewriteCond %{HTTP_USER_AGENT} ^Link [OR]
RewriteCond %{HTTP_USER_AGENT} ^LinkWalker [OR]
RewriteCond %{HTTP_USER_AGENT} ^Mag-Net [OR]
RewriteCond %{HTTP_USER_AGENT} ^Magnet [OR]
RewriteCond %{HTTP_USER_AGENT} ^Mass\ Downloader [OR]
RewriteCond %{HTTP_USER_AGENT} ^Memo [OR]
RewriteCond %{HTTP_USER_AGENT} ^Microsoft.URL [OR]
RewriteCond %{HTTP_USER_AGENT} ^MIDown\ tool [OR]
RewriteCond %{HTTP_USER_AGENT} ^Mirror [OR]
RewriteCond %{HTTP_USER_AGENT} ^Mister\ PiX [OR]
RewriteCond %{HTTP_USER_AGENT} ^Mozilla.*Indy [OR]
RewriteCond %{HTTP_USER_AGENT} ^Mozilla.*NEWT [OR]
RewriteCond %{HTTP_USER_AGENT} ^Mozilla*MSIECrawler [OR]
RewriteCond %{HTTP_USER_AGENT} ^MS\ FrontPage* [OR]
RewriteCond %{HTTP_USER_AGENT} ^MSFrontPage [OR]
RewriteCond %{HTTP_USER_AGENT} ^MSIECrawler [OR]
RewriteCond %{HTTP_USER_AGENT} ^MSProxy [OR]
RewriteCond %{HTTP_USER_AGENT} ^Navroad [OR]
RewriteCond %{HTTP_USER_AGENT} ^NearSite [OR]
RewriteCond %{HTTP_USER_AGENT} ^NetAnts [OR]
RewriteCond %{HTTP_USER_AGENT} ^NetMechanic [OR]
RewriteCond %{HTTP_USER_AGENT} ^NetSpider [OR]
RewriteCond %{HTTP_USER_AGENT} ^Net\ Vampire [OR]
RewriteCond %{HTTP_USER_AGENT} ^NetZIP [OR]
RewriteCond %{HTTP_USER_AGENT} ^NICErsPRO [OR]
RewriteCond %{HTTP_USER_AGENT} ^Ninja [OR]
RewriteCond %{HTTP_USER_AGENT} ^Octopus [OR]
RewriteCond %{HTTP_USER_AGENT} ^Offline\ Explorer [OR]
RewriteCond %{HTTP_USER_AGENT} ^Offline\ Navigator [OR]
RewriteCond %{HTTP_USER_AGENT} ^Openfind [OR]
RewriteCond %{HTTP_USER_AGENT} ^PageGrabber [OR]
```

```
RewriteCond %{HTTP_USER_AGENT} ^Papa\ Foto [OR]
RewriteCond %{HTTP_USER_AGENT} ^pavuk [OR]
RewriteCond %{HTTP_USER_AGENT} ^pcBrowser [OR]
RewriteCond %{HTTP_USER_AGENT} ^Ping [OR]
RewriteCond %{HTTP_USER_AGENT} ^PingALink [OR]
RewriteCond %{HTTP_USER_AGENT} ^Pockey [OR]
RewriteCond %{HTTP_USER_AGENT} ^psbot [OR]
RewriteCond %{HTTP_USER_AGENT} ^Pump [OR]
RewriteCond %{HTTP_USER_AGENT} ^QRVA [OR]
RewriteCond %{HTTP_USER_AGENT} ^RealDownload [OR]
RewriteCond %{HTTP_USER_AGENT} ^Reaper [OR]
RewriteCond %{HTTP_USER_AGENT} ^Recorder [OR]
RewriteCond %{HTTP_USER_AGENT} ^ReGet [OR]
RewriteCond %{HTTP_USER_AGENT} ^Scooter [OR]
RewriteCond %{HTTP_USER_AGENT} ^Seeker [OR]
RewriteCond %{HTTP_USER_AGENT} ^Siphon [OR]
RewriteCond %{HTTP_USER_AGENT} ^sitecheck.internetseer.com [OR]
RewriteCond %{HTTP_USER_AGENT} ^SiteSnagger [OR]
RewriteCond %{HTTP_USER_AGENT} ^SlySearch [OR]
RewriteCond %{HTTP_USER_AGENT} ^SmartDownload [OR]
RewriteCond %{HTTP_USER_AGENT} ^Snake [OR]
RewriteCond %{HTTP_USER_AGENT} ^SpaceBison [OR]
RewriteCond %{HTTP_USER_AGENT} ^sproose [OR]
RewriteCond %{HTTP_USER_AGENT} ^Stripper [OR]
RewriteCond %{HTTP_USER_AGENT} ^Sucker [OR]
RewriteCond %{HTTP_USER_AGENT} ^SuperBot [OR]
RewriteCond %{HTTP_USER_AGENT} ^SuperHTTP [OR]
RewriteCond %{HTTP_USER_AGENT} ^Surfbot [OR]
RewriteCond %{HTTP_USER_AGENT} ^Szukacz [OR]
RewriteCond %{HTTP_USER_AGENT} ^tAkeOut [OR]
RewriteCond %{HTTP_USER_AGENT} ^Teleport\ Pro [OR]
RewriteCond %{HTTP_USER_AGENT} ^URLSpiderPro [OR]
RewriteCond %{HTTP_USER_AGENT} ^Vacuum [OR]
RewriteCond %{HTTP_USER_AGENT} ^VoidEYE [OR]
RewriteCond %{HTTP_USER_AGENT} ^Web\ Image\ Collector [OR]
RewriteCond %{HTTP_USER_AGENT} ^Web\ Sucker [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebAuto [OR]
RewriteCond %{HTTP_USER_AGENT} ^[Ww]eb[Bb]andit [OR]
RewriteCond %{HTTP_USER_AGENT} ^webcollage [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebCopier [OR]
RewriteCond %{HTTP_USER_AGENT} ^Web\ Downloader [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebEMailExtrac.* [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebFetch [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebGo\ IS [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebHook [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebLeacher [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebMiner [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebMirror [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebReaper [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebSauger [OR]
RewriteCond %{HTTP_USER_AGENT} ^Website [OR]
RewriteCond %{HTTP_USER_AGENT} ^Website\ eXtractor [OR]
RewriteCond %{HTTP_USER_AGENT} ^Website\ Quester [OR]
RewriteCond %{HTTP_USER_AGENT} ^Webster [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebStripper [OR]
RewriteCond %{HTTP_USER_AGENT} WebWhacker [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebZIP [OR]
RewriteCond %{HTTP_USER_AGENT} ^Wget [OR]
RewriteCond %{HTTP_USER_AGENT} ^Whacker [OR]
```

```
RewriteCond %{HTTP_USER_AGENT} ^Widow [OR]
RewriteCond %{HTTP_USER_AGENT} ^WWWOFFLE [OR]
RewriteCond %{HTTP_USER_AGENT} ^x-Tractor [OR]
RewriteCond %{HTTP_USER_AGENT} ^Xaldon\ WebSpider [OR]
RewriteCond %{HTTP_USER_AGENT} ^Xenu [OR]
RewriteCond %{HTTP_USER_AGENT} ^Zeus.*Webster [OR]
RewriteCond %{HTTP_USER_AGENT} ^Zeus
RewriteRule ^.* - [F,L]
```

NOTA:
Podemos también añadir a la lista de bloqueos aquellas peticiones que no incluyen un User-Agent, e intenten realizar peticiones POST sobre nuestro sitio con este código adicional:

```
RewriteEngine On
RewriteCond %{REQUEST_METHOD} POST
RewriteCond %{HTTP_USER_AGENT} ^$
RewriteRule .* - [F]
```

15. DESHABILITAR EL REPORTE DE ERRORES

179. Si alguno de los plugins o temas instalados en nuestro servidor genera algún error en su ejecución a un visitante, es posible que se filtre cierta información sensible de nuestro servicio, como podría ser el path de ejecución, que podría ayudar a un atacante a elaborar un ataque más preciso y dirigido.

180. Por este motivo, deshabilitaremos la opción de mostrar estos mensajes de error a los visitantes añadiendo el siguiente código al fichero 'wp-config.php':

```
error_reporting(0);
@ini_set('display_errors', 0);
```

181. Si fuera necesario acceder a esta información, podemos generar un fichero de log o debug de Wordpress añadiendo el siguiente código al fichero anteriormente mencionado:

```
@ini_set( 'log_errors', 'On' );
@ini_set( 'display_errors', 'Off' );
define( 'WP_DEBUG', false );
define( 'WP_DEBUG_LOG', false );
define( 'WP_DEBUG_DISPLAY', false );
```

182. Dado que la ruta por defecto para este fichero será '/wp-content/debug.log', puede que quede expuesto de forma pública y un atacante pueda acceder a información sensible de nuestra instalación. En caso de no poder generarlo en otra ruta inferior a nuestro directorio público, configuraremos los permisos del fichero a 600 y añadiremos lo siguiente al fichero '.htaccess' de nuestra instalación:

```
<Files debug.log>
   Order allow,deny
   Deny from all
</Files>
```

183. Esto evitará que cualquier usuario pueda acceder al fichero utilizando HTTP.

16. BORRADO AUTOMÁTICO DEL PLUGIN HELLO DOLLY

184. Hello Dolly es uno de los plugins que vienen por omisión en una instalación nueva de WordPress (junto con Akismet), y vuelve a instalarse en cada actualización que hacemos, sin importar si lo queremos o no. La función de este plugin para WordPress es solamente poner frases de la canción "Hello Dolly" de Louis Armstrong de forma aleatoria en la administración de WordPress.

185. Para evitar tener que borrarlo de cada uno de nuestros sitios Web de forma manual cada vez que realicemos una instalación o que actualicemos la plataforma Wordpress, incluiremos un código para que cuando detecte su presencia, lo elimine de forma automática.

186. Para ello, tendremos que abrir el fichero 'functions.php' de nuestro tema correspondiente, que se encontrará en la ruta './wp-content/themes/nombre_del_tema/functions.php' (en nuestro caso, dado que estamos utilizando la plantilla por defecto, editamos el fichero './wp-content/themes/twentyfourteen/functions.php') e incluir el siguiente código al final del fichero:

```
// Borrado automatico del plugin Hello Dolly
function eliminar_dolly() {
   if (file_exists(WP_PLUGIN_DIR.'/hello.php')) {
      require_once(ABSPATH.'wp-admin/includes/plugin.php');
      require_once(ABSPATH.'wp-admin/includes/file.php');
      delete_plugins(array('hello.php'));
   }
}
add_action('admin_init','eliminar_dolly');
```

187. Guardamos los cambios y a partir de este momento, Wordpress se encargará de comprobar si el plugin existe en nuestro sistema y eliminarlo de forma correcta en caso afirmativo.

17. INSTALACIÓN DE 'ITHEMES SECURITY'

188. En esta sección veremos el plugin 'iThemes Security', que nos permitirá aplicar de forma automática más 30 modificaciones y comprobaciones a nuestro sistema para comprobar su seguridad. Algunas de las características que veremos a continuación son:

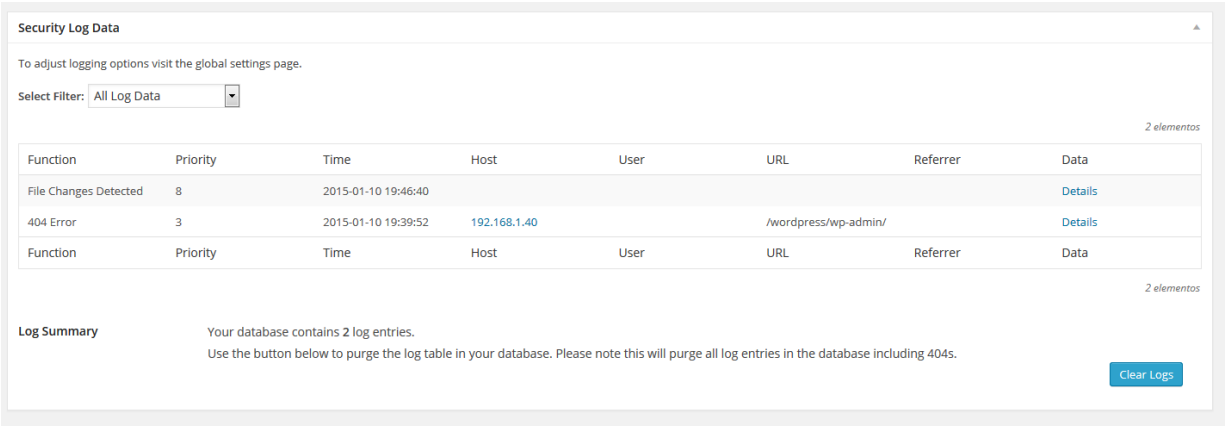
- Generación de contraseñas seguras
- Eliminación de las etiquetas de Wordpress
- Cambio del directorio 'wp-content'
- Renombrado de la cuenta de administrador
- Forzado de SSL en inicio de sesión o panel de administración
- Etc.

189. Algunas de estas características han sido anteriormente vistas, pero se mostrará una forma gráfica y unificada de realizarlas.

190. Comenzaremos con la instalación del plugin de la forma habitual, buscando desde 'Plugins / Añadir nuevo' y buscando 'iTheme Security' en el repositorio de Wordpress. Una vez localizado, procederemos a su instalación y activación.

191. Podremos acceder a su propio panel pulsando en la etiqueta 'Security' de la barra de navegación izquierda. Una vez dentro, nos mostrará aquellas alertas de seguridad que podemos solucionar, desde las más críticas a las más bajas (algunas probablemente estén solucionadas con el uso de otro plugin que hayamos instalado en el sistema):

192. Si deseamos solucionar alguno de estos problemas de seguridad, deberemos pulsar sobre 'Fix it' y automáticamente nos redirigirá a la parte de configuración donde podemos tratar con la solución disponible. También podremos acceder al registro de las acciones generadas en el sistema desde la opción de 'Logs':



193. Procederemos a configurar las distintas opciones que nos permite el plugin desde el menú 'Settings'. Dentro de la configuración tendremos diferentes secciones con valores que vamos a configurar.

17.1. GLOBAL SETTINGS

- Write to files: permitirá al *plugin* escribir en los ficheros necesarios del sistema (wp-config.php y .htaccess). Si no activamos esta opción, el *plugin* nos mostrará los cambios que deberemos introducir de forma manual dentro de estos ficheros cada vez que realicemos una modificación en la configuración.
- Notification Email: direcciones de email donde recibiremos las diferentes alertas de seguridad que se generen.
- Send Digest Email. Permite enviar notificaciones agrupadas por correo. Esta opción es útil si nuestro sitio recibe gran cantidad de ataques, ya que recibiremos una menor cantidad de emails de alerta.
- Backup Delivery Mail: dirección de correo donde se enviarán los *backups* de nuestro sitio.
- Host Lockout Message: código HTML que será mostrado cuando una dirección sea bloqueada de forma automática.

- User Lockout Message: código HTML mostrado cuando un usuario sea bloqueado por intentar realizar un ataque de fuerza bruta.
- Blacklist Repeat Offender: si activamos esta opción, la dirección IP de los equipos atacantes serán bloqueados y además includos en la lista de baneos del sistema.
- Blacklist Threshold: número de bloqueos permitidos antes de que la direccion IP sea baneada de forma definitiva.
- Blacklist Lookback Period: número de días durante los que se recordará el número de bloqueos para proceder al baneo.
- Lockout Period: número de minutos en los que una dirección IP o usuario será bloqueada por intentos erróneos de acceso.
- Lockout White List: direcciones IP que introduciremos en la lista blanca que no se verá afectado por los bloqueos.
- Email Lockout Notifications: envio de alerta por correo electrónico por cada bloqueo que se realice de forma automática
- Log Type: gestiona la forma en la que los logs internos del *plugin* serán almacenados. La opción por defecto, así como la más recomendada, será 'Database Only'.
- Days to Keep Database Logs: número de días durante los cuales se almacenará la información de los logs. Podremos incrementar este valor en función de las visitas y el tráfico que recibamos en nuestro sitio.
- Path to Log Files: ruta donde se guardarán los *logs* si seleccionamos la opción de local en vez de utilizar la base de datos. Esta ruta debería estar fuera de la ruta de nuestro sitio web para evitar que posibles atacantes obtengan acceso por HTTP.
- Allow Data Tracking: permite enviar información de nuestro sistema para generar estadísticas. Debe ser deshabilitada.
- Hide Security Menu in Admin Bar: esconde la posibilidad de acceder al panel de 'iTheme Security' desde el menú del panel de administrador.

17.2. 404 DETECTION

194. Permite detectar cuando un usuario está intentando escanear el sitio en busca de páginas o aplicaciones vulnerables, generando gran cantidad de errores 404. Cuando se detecta este tipo de actividad, se presupone que nos encontramos frente a un atacante y podremos bloquearle de forma efectiva.

195. Para activar esta característica, pulsaremos sobre 'Enable 404 detection', encontrando las siguientes características:

- Minutes to Remember 404 Error: número de minutos que el sistema recordará el número de errores 404 generados por el visitante.
- Error Threshold: el número de errores permitidos por visitante antes de realizar el bloqueo.
- 404 File/Folder White List: listado de ficheros permitidos en la generación de errores 404, como el fichero robots.txt, favicon.ico, etc.

- Ignores File Types: tipos de ficheros que no generarán una alerta 404 en el sistema. Generalmente se trata de hojas de estilo o imágenes.

17.3. AWAY MODE

196. Esta opción nos permitirá deshabilitar el acceso al panel de administración del sistema durante un periodo concreto. Esto evitará la exposición del panel a posibles atacantes dentro de horarios en los que no se encuentre ningún administrador trabajando.

197. Para su activación pulsaremos sobre 'Enable away mode' y configuraremos los parámetros mostrados:

- Type of Restriction: diaria o puntual en la fecha seleccionada
- Start Time: hora de la desconexión del panel
- End Time: hora para la recuperación del panel

17.4. BANNED USERS

198. Nos permitirá banear determinadas direcciones IP o User-Agents de nuestro sistema de forma automática. No será necesario activarlo si hemos realizado las tareas explicadas en el apartado de 'Prevención de Spam' de esta guía. Encontraremos las siguientes opciones:

- Default Blacklist: permite utilizar la lista de *bots* creada por Jim Walker (HackRepair.com).
- Ban Users: permite activar el bloqueo de usuarios.
- Ban User-Agents: permite bloquear las peticiones realizadas con el User-Agent que especifiquemos en el cuadro de texto.

17.5. BRUTE FORCE PROTECTION

199. Nos permitirá bloquear los intentos de ataques de fuerza bruta que recibamos en el sistema. No es necesario activar la opción si tenemos el plugin 'Limit Login Attempts' correctamente instalado.

17.6. DATABASE BACKUP

200. Permite generar copias de seguridad de nuestra actual instalación de Wordpress. Esta actividad puede realizarse desde este plugin, o con los diversos métodos expuestos en la sección 'Copias de Seguridad' de esta guía.

17.7. FILE CHANGE DETECTION

201. Esta característica nos permitirá conocer si se ha realizado alguna modificación de los ficheros del sistema por un posible atacante. Comparará los ficheros de nuestro sistema con cada comprobación realizada, detectando posibles modificaciones de código o ficheros nuevos que hayan podido ser inyectados de forma remota y que pongan en riesgo nuestro equipo.

202. Pulsaremos sobre 'Enable File Change detection' para activar esta característica y configuraremos:

- Split File Scanning: permite dividir los chequeos en 7 tareas diferentes. Útil para sitios con gran cantidad de contenidos y de ficheros accesibles por el usuario final.
- Include/Exclude Files and Folders: permite especificar qué ficheros o directorios no serán analizados por el sistema en búsqueda de cambios.
- Ignore File Types: extensiones de ficheros donde no se realizará el escaneo.
- Email File Change Notifications: una vez activado enviará un email de alerta a la dirección de correo configurada en el *plugin* con cada una de los cambios detectados.
- Display File Change Admin Warning: deshabilita la visualización de alertas dentro del panel de administración.

17.8. HIDE LOGIN AREA

203. Permite esconder la página de login de Wordpress, haciendo más complicados los ataques automatizados sobre el sistema.

204. Para habilitar esta característica pulsaremos sobre 'Enable the hide backend feature' y configuraremos:

- Login Slug: será el alias que utilizaremos para acceder a nuestra página de *login*. No podrá ser igual a 'login', 'admin', 'dashboard', o 'wp-admin', ya que son valores por defecto en Wordpress.
- Enable Theme Compatibility: marcaremos la opción si al activar 'Hide Login Area' tenemos problemas para acceder desde el tema que esté actualmente instalado. ☑

17.9. MALWARE SCANNING

205. Permite escanear los diferentes ficheros que se encuentren en nuestro servidor web en búsqueda de malware, utilizando el servidor proporcionado por VirusTotal.

206. Para acceder a esta característica, será necesario disponer de una clave para acceder al servicio de VirusTotal e introducirla en el campo correspondiente.

17.10. SECURE SOCKET LAYER (SSL)

207. Nos permite configurar las diferentes opciones de acceso por SSL vistas en la sección 'Administración y navegación sobre SSL' de esta misma guía.

17.11. STRONG PASSWORDS

208. Permite activar la opción de contraseñas robustas en el rol de usuario que deseemos. Para ello, pulsaremos sobre 'Enable strong password enforcement' y seleccionaremos qué tipo de usuario recibirá la restricción.

17.12. SYSTEM TWEAKS

209. Diferentes medidas de protección sobre el sistema, entre las que cabe destacar:

- System Files: activa la protección de ficheros como 'readme.html', 'install.php', 'wp-includes', etc. No es necesaria su activación, debido a que este tipo de ficheros han sido correctamente configurados a lo largo de la guía.

- Directory Browsing: impide el listado de ficheros dentro de cada directorio.
- Request Methods: evita que se utilicen diferentes métodos HTTP, como TRACE, DELETE o TRACK.
- Suspicious Query Strings: filtra peticiones que reciba el sistema y contengan cadenas de texto sospechosas.
- Long URL Strings: filtra aquellas peticiones que sean sospechosamente largas. No debe ser utilizado si desconocemos la existencia de algún *plugin* o aplicación instalada que haga este tipo de usos.
- File Writing Permissions: configura los permisos de ficheros y directorios de forma adecuada. No es necesario si se han seguido las directrices de 'Configuración de permisos' de esta misma guía.
- Uploads: permite deshabilitar la ejecución de código PHP en el directorio 'uploads', evitando que un atacante pueda subir una Shell o realizar algún ataque contra el sistema.

17.13. WORDPRESS TWEAKS

210. Modificaciones relativas a Wordpress para asegurar nuestro sitio. Destacan las siguientes posibilidades:

- Generator Meta Tag: elimina el rastro de la versión actual de Wordpress en la generación de páginas. No es necesario si se ha seguido lo expuesto en la sección 'Eliminando la información de versión' de esta misma guía.
- Display Random Version: mostrará una versión errónea y diferente a la actual cuando se generen páginas de forma dinámica.
- File editor: Deshabilita el editor para *plugins* y temas del sistema que permite a los usuarios escribir código propio. No necesario si se han seguido las pautas mostradas en 'Deshabilitar el editor de ficheros en línea' de esta misma guía. ☑
- XML-RPC: permite mantener el servicio XMLRPC funcionando, sólo deshabilitar los Trackbacks/Pingsbacks o deshabilitarlo completamente. Esta última opción es la más segura, aunque evitará que ciertas funcionalidades y *plugins* como Jetpack funcionen correctamente.
- Login Error Messages: permite evitar mostrar un mensaje de error cuando se introducen unas credenciales erróneas de acceso al sistema. Puede ayudar a mitigar intentos de ataque de fuerza bruta.
- Disable Extra User Archives: deshabilita la página de autor si este no ha escrito ningún post, complicando la tarea de enumerar usuarios por parte de un posible atacante.

17.14. OPCIONES AVANZADAS

211. Dentro de las opciones avanzadas, nos permitirá tres tareas:

- Admin User: no es necesario si hemos aplicado las diferentes directrices mostradas a lo largo de esta guí.

- Change Content Directory: nos permite modificar la ruta por defecto del directorio 'wp-content'. Este cambio debe realizarse siempre con una instalación limpia, ya que si hemos publicado algún post o existe alguna imagen en el sistema, dejará de funcionar de forma correcta. Una vez realizado este cambio, no es posible volver al estado inicial.
- Change Database Prefix: permite modificar el prefijo de las tablas creadas por Wordpress, como hemos visto anteriormente en esta guía.

18. ACTUALIZACIONES AUTOMÁTICAS

212. Desde la versión 3.7. WordPress incorpora un sistema automático para las actualizaciones. Por defecto se realizan automáticamente las actualizaciones menores de mantenimiento y seguridad, de X.Y.Z a X.Y.Z+1 (por ejemplo de 3.8.1 a 3.8.2) y las actualizaciones de los ficheros de traducción

213. En WordPress existen 4 tipos de actualizaciones:

- Actualizaciones del núcleo. A su vez estas se dividen en tres categorías:
  a. Actualizaciones de desarrollo (versiones beta o *release candidate* RC)
  b. Actualizaciones menores de mantenimiento y seguridad (por ejemplo de 3.8.1 a 3.8.2)
  c. Actualizaciones mayores
- Actualizaciones de *plugins*
- Actualizaciones de temas
- Actualizaciones de ficheros de traducción

214. Las opciones para la configuración del actualizador automático son bastante completas y flexibles. Podemos hacerlo a través del fichero 'wp-config.php', con el uso de filtros o, si no queremos tocar código, con un plugin como 'Advanced Automatic Updates'.

215. Recomendaciones sobre las actualizaciones automáticas:

- Es recomendable desactivar todas las actualizaciones automáticas si tenemos en el WordPress una tienda. Básicamente, si tenemos ingresos sustanciales a través de la web o si se lleva un control de stock o cupo de reservas, no es recomendable que se actualice automáticamente ya que conviene tener la precaución de hacer antes una copia de seguridad y probar las actualizaciones en un entorno de desarrollo.
- También es recomendable desactivar las actualizaciones si usamos un *plugin* de multi-idioma. Sobre todo en el caso del *plugin* 'qtranslate' ya que, si la versión de WordPress es superior a la que soporta qtranslate este se desactiva automáticamente y deberemos activarlo manualmente tras la actualización.
- También es recomendable desactivarlas si tenemos algún *plugin* muy complejo que afecte al funcionamiento principal y general de la web (por ejemplo si nuestra web se basa en una agenda de eventos y usamos un *plugin* complejo para gestión de eventos).

- Podemos mantener la configuración por defecto (actualizaciones menores) si tenemos una instalación de WordPress estándar con *plugins* simples. En este caso, si lo deseamos, podemos activar también la actualización de *plugins*.
- La actualización del tema solo es recomendable activarla si no hemos realizado ninguna modificación en el código del tema y tenemos una instalación de WordPress estándar con *plugins* simples.

216. En ningún caso deben activarse las actualizaciones automáticas:

- Si al añadir *plugins* o actualizar manualmente nos pide las credenciales del FTP del servidor.
- Si el servidor no tiene activado el OpenSSL para PHP, ya que el instalador necesita comunicarse de forma segura a través de HTTPS con WordPress.org
- Si usamos un sistema de control de versiones o repositorio de código tipo SVN o GIT (este caso solo afecta a desarrolladores y se puede corregir con el filtro automatic_updates_is_vcs_checkout).
- Si WP- Cron no funciona correctamente en nuestra instalación o está desactivado.

18.1. CONFIGURACIÓN A TRAVÉS DE 'WP-CONFIG.PHP'

217. Podemos configurar cómo queremos que se comporte el actualizador automático añadiendo constantes en el fichero de configuración 'wp-config.php' situado en la raíz del directorio donde tenemos instalado WordPress.

218. Para desactivar completamente las actualizaciones automáticas de cualquier tipo añadiremos la siguiente constante:

```
define( 'AUTOMATIC_UPDATER_DISABLED', true );
```

219. Para configurar las actualizaciones automáticas del núcleo, lo podemos hacer con la constante WP_AUTO_UPDATE_CORE. La definición de este constante puede tener tres valores:

1. Desactiva todas las actualizaciones del núcleo:

   ```
   define( 'WP_AUTO_UPDATE_CORE', false );
   ```

2. Activa todas las actualizaciones del núcleo, incluidas las mayores y las versiones de desarrollo:

   ```
   define( 'WP_AUTO_UPDATE_CORE', true );
   ```

3. Activa solo las actualizaciones menores del núcleo:

   ```
   define( 'WP_AUTO_UPDATE_CORE', 'minor' );
   ```

220. Para una configuración más flexible, como por ejemplo activar actualizaciones mayores sin activar las de desarrollo o activar las actualizaciones de plugins o temas tendremos que usar filtros.

18.2. CONFIGURACIÓN A TRAVÉS DE FILTROS

221. La configuración a través de filtros es más completa y flexible que la configuración con constantes. Deberemos añadir los filtros en el fichero functions.php de nuestro tema o en un fichero que hayamos creado para este propósito.

222. Para desactivar completamente las actualizaciones automáticas de cualquier tipo añadiremos el siguiente filtro:

add_filter('automatic_updater_disabled', '__return_true');

223. Para configurar las actualizaciones del núcleo de una forma más selectiva disponemos de varios filtros. Dependiendo de si vínculamos el filtro con la directiva __return_true o __return_false activa o desactiva las actualizaciones respectivamente.

1. Activar las actualizaciones automáticas de las versiones mayores:

add_filter('allow_major_auto_core_updates', '__return_true');

2. Activar las actualizaciones automáticas de las versiones de desarrollo:

add_filter('allow_dev_auto_core_updates', '__return_true');

3. Desactivar las actualizaciones de las versiones menores:

add_filter('allow_minor_auto_core_updates', '__return_false');

224. En el caso de que nuestro WordPress no sea compatible con las actualizaciones automáticas porque usemos un sistema de control de versiones (svn o git) podemos forzar a que se actualice automáticamente con el siguiente filtro:

add_filter('automatic_updates_is_vcs_checkout', '__return_false', 1);

225. Para activar la actualización automática de plugins disponemos del siguiente filtro:

add_filter('auto_update_plugin', '__return_true');

226. De la misma forma podemos activar la actualización automática de temas con el filtro:

add_filter('auto_update_theme', '__return_true');

227. Y por último podemos desactivar la actualización de los ficheros de traducción con este filtro:

add_filter('auto_update_translation', '__return_false');

228. A parte de estos filtros, disponemos de filtros para configurar el email de notificación que envía el actualizador automático. El actualizador automático, después de ejecutarse, envía un email al administrador del WordPress con el resultado tanto si ha ido bien como si ha ocurrido un error. Podremos desactivar el envío de este email o que solo envíe el email según el resultado y tipo de actualización que queramos con un filtro:

add_filter('auto_core_update_send_email', '__return_false');

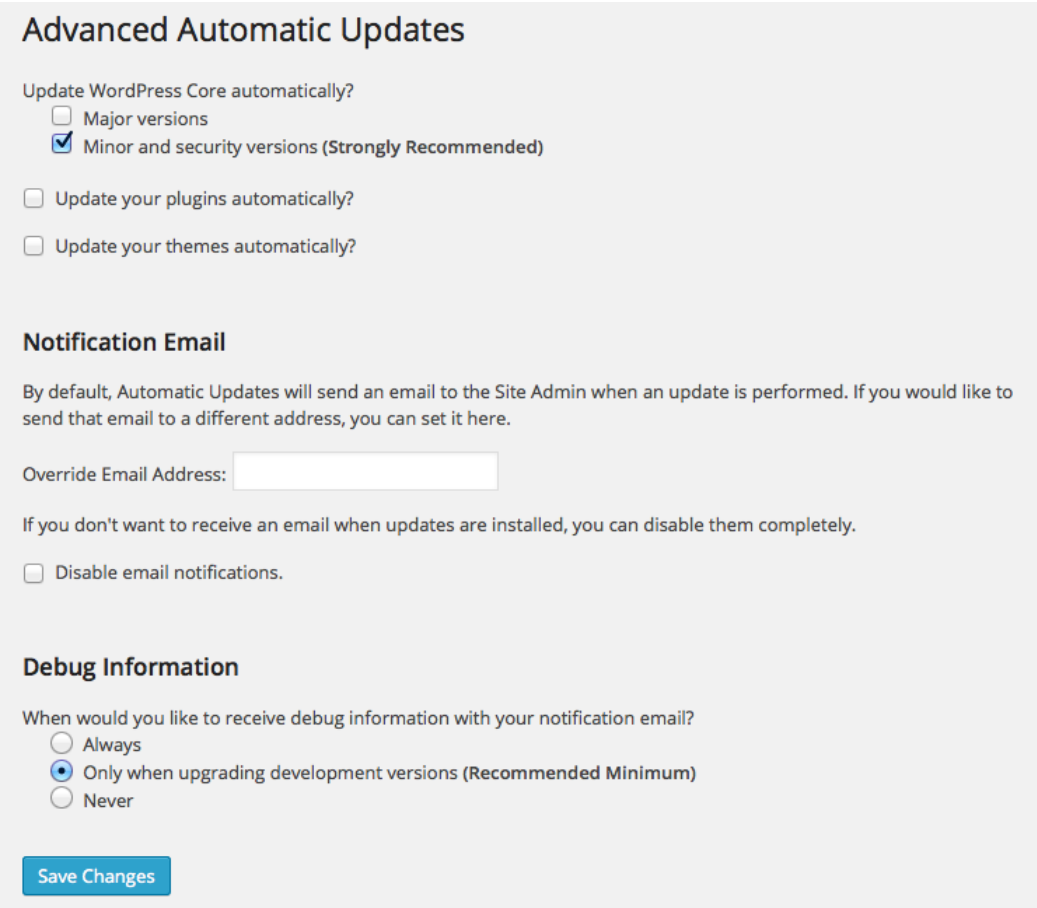
229. También disponemos de un filtro para modificar la dirección de email a la que envía según el resultado y tipo de actualización

/ @param bool $send Whether to send the email. Default true.*
** @param string $type The type of email to send.*
** Can be one of 'success', 'fail', 'critical'.*
** @param object $core_update The update offer that was attempted.*
** @param mixed $result The result for the core update. Can be WP_Error.*
**/*
apply_filters('auto_core_update_email', $email, $type, $core_update, $result);

18.3. CONFIGURACIÓN A TRAVÉS DE UN PLUGIN

230. Si no queremos tocar código podemos cambiar la configuración del actualizador con un plugin como 'Advanced Automatic Updates'.

231. Este plugin nos añade una página en el administrador de Ajustes de WordPress desde donde podemos configurar cómo queremos que se comporte el actualizador automático o desactivarlo completamente:



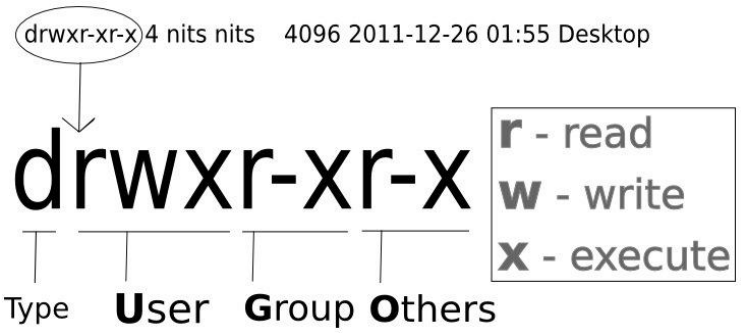
19. CONFIGURACIÓN DE PERMISOS

232. Los distintos ficheros y directorios poseen permisos que especifican quién y qué puede leer, escribir, modificar y accederlos. Esto es importante, puesto que WordPress puede necesitar acceso de escritura a ficheros en tu directorio 'wp-content' para activar ciertas funcionalidades.

233. 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

234. 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.



235. 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)

236. 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.

237. 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

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

239. 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).

240. Cualquier fichero que necesite acceso de escritura desde WordPress debería pertenecer al grupo o al usuario de la cuenta de usuario utilizada por WordPress (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 éste se encuentre utilizando un usuario separado, en un grupo de usuario aparte, como apache, www-data o nobody. Si WordPress se ejecuta con 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
- 664 en vez de 644 para ficheros

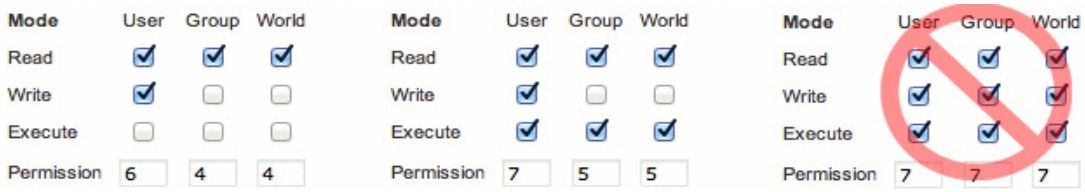
241. Los permisos de fichero y directorio de WordPress deberían ser los mismos para la mayoría de usuarios, dependiendo del tipo de instalación que hayas realizado y la configuración de *umask* del entorno de sistema en el momento de la instalación.

242. 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 Wordpress deberían poseer permisos de escritura solamente por la cuenta de usuario utilizada
2. Si se desea utilizar el editor incorporado de temas, todos los ficheros necesitarán tener permisos de escritura de grupo
3. Para garantizar la seguridad de los archivos, es necesario mantenerlos en 644:

   $ find /ruta_absoluta/wordpress/ -type d -exec chmod 755 {} \;
4. Para garantizar la seguridad de los directorios, es necesario mantenerlos en 755.

$ find /ruta_absoluta/wordpress/ -type f -exec chmod 644 {} \;



5. 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.
6. Respetar las excepciones que se muestran a continuación
   a. /wp-config.php: manter permisos en 600
   b. /cgi-bin/.htaccess: mantener permisos en 604
   c. /cgi-bin/php.ini: mantener permisos en 600
   d. /cgi-bin/php.cgi: mantener permisos en 711
   e. /cgi-bin/php5.cgi: mantener permisos en 100

19.1. PERMISOS PARA ENTORNOS COMPARTIDOS CON SUEXEC

243. Lo arriba descrito puede que no se aplique a sistemas de hosting compartido que usen 'suexec' para ejecutar binarios PHP. Para estos sistemas, el proceso PHP corre como el propietario de los propios ficheros, permitiendo una configuración más simple y un entorno más seguro en el caso específico de hosting compartido.

NOTA:
Los métodos 'suexec' NUNCA deberían ser utilizados en una configuración de un sitio único, ya que son más seguros solamente para el caso concreto de *hosting* compartido.

244. En tal configuración 'suexec', el esquema de permisos correctos es simple de entender.

- Todos los ficheros deberían pertenecer a la verdadera cuenta de usuario FTP, no a la cuenta de usuario usada por el proceso demonio httpd (nobody, apache, www-data, etc.).
- La propiedad de grupo es irrelevante, a menos que haya requisitos específicos de grupo para el proceso de comprobación de permisos del servidor web. Esto no es habitualmente el caso.
- Todos los directorios deberían tener los permisos 755 o 750.
- Todos los ficheros deberían tener los permisos 644 o 640, a excepción del fichero 'wp-config.php', que debería ser 600 para prevenir que otros usuarios del servidor tengan acceso.

- Ningún directorio debería tener nunca los 777, incluso los directorios de carga. Ya que el proceso PHP se ejecuta como el propietario de los ficheros, consigue que los permisos del usuario puedan escribir incluso en un directorio 755.

245. En este tipo de configuración específica, WordPress detectará que puede crear directamente ficheros con la propiedad adecuada, y de ese modo no preguntará por las credenciales de FTP cuando se actualicen o instalen plugins.

20. COPIAS DE SEGURIDAD

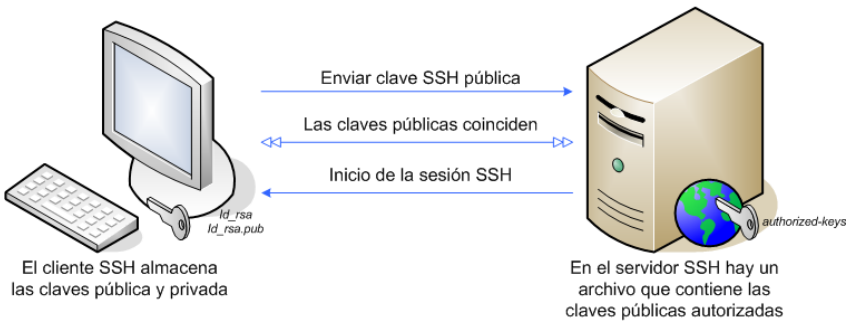
246. La base de datos de Wordpress contiene cada entrada, página, comentario, enlace o plugin que tengamos instalado en nuestra aplicación Web, por lo que si esta se borrara accidentalmente o acabara corrupta, se perderían los datos en ella contenidos. Existen, además, muchos motivos por los que esto puede ocurrir, y no todos pueden ser controlados por el administrador. Con una copia de seguridad adecuada, estos datos pueden ser restablecidos de forma sencilla.

247. Existen varios métodos para realizar estas copias de seguridad, aunque vamos a centrarnos en solamente dos:

1. Copias de seguridad automatizadas a nivel de consola
2. BackWPup: *Plugin* para Wordpress

20.1. COPIAS DE SEGURIDAD AUTOMATIZADAS A NIVEL DE CONSOLA

248. 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.



249. 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':

```
$ ssh-keygen -b 4096 -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/wpuser/.ssh/id_rsa):
Created directory '/home/wpuser/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/wpuser/.ssh/id_rsa.
Your public key has been saved in /home/wpuser/.ssh/id_rsa.pub.
The key fingerprint is:
ce:d0:67:26:bf:78:0f:1d:4a:d8:4f:6f:4b:82:16:a3 wpuser@GuiaWordpress
The key's randomart image is:
+--[ RSA 4096]----+
|                 |
|                 |
|                 |
|       . o       |
|      . S O o    |
|       + O O o   |
|        E * + +  |
|         o.o + . |
|         ..... . |
+-----------------+
```

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:

```
$ ssh-copy-id -i ~/.ssh/id_rsa.pub wpbackup@212.71.237.171
The authenticity of host '212.71.237.171 (212.71.237.171)' can't be established.
ECDSA key fingerprint is 84:ee:e9:2e:1c:7c:84:3e:47:be:da:54:ce:ce:f2:91.
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
wpbackup@212.71.237.171's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'wpbackup@212.71.237.171'"
and check to make sure that only the key(s) you wanted were added.
```

4. Comprobamos que podemos realizar la conexión sin contraseña:

```
$ ssh wpbackup@212.71.237.171
Linux (none) 3.15.4-x86_64-linode45 #1 SMP Mon Jul 7 08:42:36 EDT 2014 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.
Last login: Wed Dec 31 08:25:38 2014 from 52.red-83-56-233.dynamicip.rima-tde.net
wpbackup@(none):~$
```

250. 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 Wordpress 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*

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

```
#!/bin/bash
#
# Script para la realizacion de copias de seguridad en Wordpress por SSH
#

BLOG_DIR=/var/www//wordpress
BACKUP_DIR=/var/backups/wordpress
```

```
REMOTE_USER=wpbackup
REMOTE_HOST=212.XX.XXX.XX
REMOTE_BACKUP_DIR=/backups/wp_principal

DB_NAME=`echo "<?php require_once(\"${BLOG_DIR}/wp-config.php\"); echo DB_NAME;" | php`
DB_USER=`echo "<?php require_once(\"${BLOG_DIR}/wp-config.php\"); echo DB_USER;" | php`
DB_PASS=`echo "<?php require_once(\"${BLOG_DIR}/wp-config.php\"); echo DB_PASSWORD;" | php`
DB_HOST=`echo "<?php require_once(\"${BLOG_DIR}/wp-config.php\"); echo DB_HOST;" | php`

BLOG_DIR=`dirname "$BLOG_DIR"`/`basename "$BLOG_DIR"`
BACKUP_DIR=`dirname "$BACKUP_DIR"`/`basename "$BACKUP_DIR"`

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 Wordpress... "
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 "error!"
    exit 3
fi
echo "Copia de seguridad finalizada!"
```

252. 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:

```
$ ./backup.sh
Realizando volcado de la base de datos...
Realizando copia de seguridad de los ficheros de Wordpress...
tar: Eliminando la `/' inicial de los nombres
Enviando copia de seguridad de BBDD/ficheros a host remoto...
guiawordpress-20141230.sql.bz2                100%   84KB  83.9KB/s   00:00
wordpress-20141230.tar.bz2                    100%   16MB   1.4MB/s   00:11
Copia de seguridad finalizada!
$
```

253. 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
$
```

I. RESTAURANDO LA COPIA DE SEGURIDAD

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

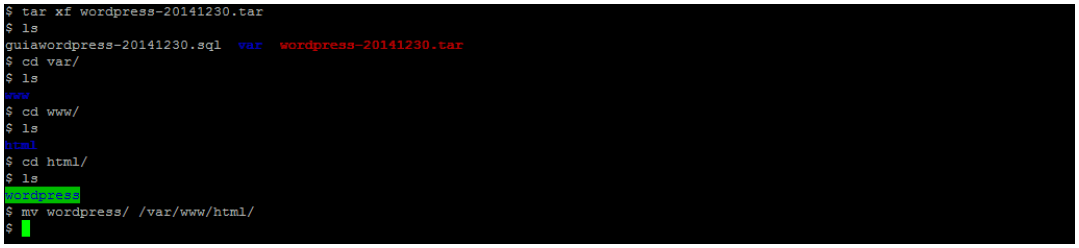
1. Descargar los ficheros correspondientes a la copia de seguridad de la base de datos y de los ficheros de Wordpress a nuestro servidor
2. Descomprimimos los ficheros con el comando ‘bzip2’

```
$ ls
guiawordpress-20141230.sql.bz2  wordpress-20141230.tar.bz2
$ bunzip2 *
$ ls
guiawordpress-20141230.sql  wordpress-20141230.tar
$
```

3. Importamos el *backup* en formato .sql a MySQL y comprobamos que se restaura correctamente

```
$ mysql -u root -p < guiawordpress-20141230.sql
Enter password:
$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 591
Server version: 5.5.40-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2014, 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 |
| guiawordpress      |
| mysql              |
| performance_schema |
+--------------------+
4 rows in set (0.00 sec)

mysql>
```

4. Descomprimimos el archivo correspondiente a la copia de seguridad de los ficheros. Es importante saber que se expandirá la ruta completa donde el anterior Wordpress 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



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

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

20.2. BACKWPUP: PLUGIN PARA WORDPRESS

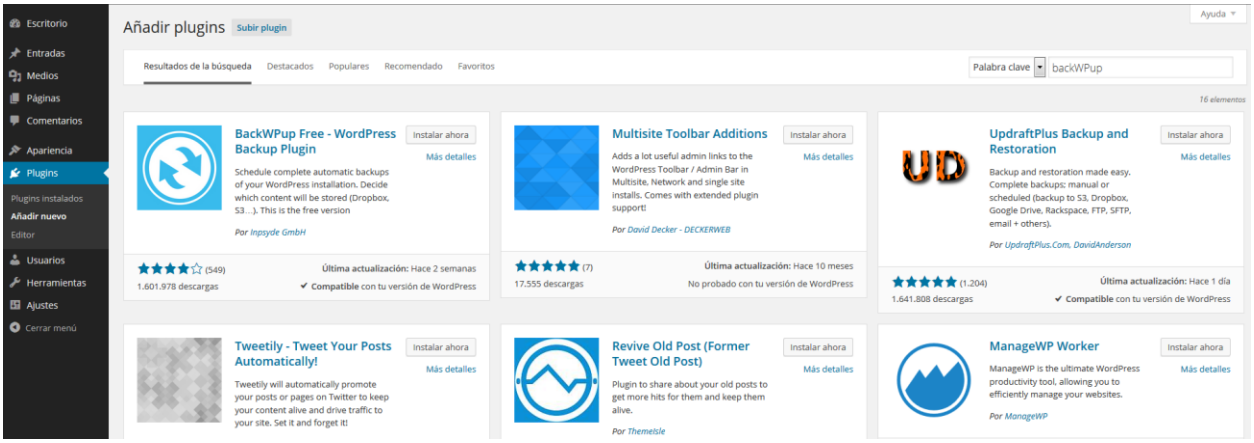
256. Otra forma de realizar copias de seguridad es utilizar los diferentes plugins ya existentes en el repositorio oficial de Wordpress. En nuestro caso, vamos a instalar y configurar BackUPup.

257. BackWPup se encuentra entre los 20 plugins más descargados, y nos ayudará a realizar la copia de seguridad de nuestros ficheros locales y la base de datos. Además es capaz de realizar otras muchas tareas interesantes, como:

- Optimizar, chequear y reparar la base de datos
- Realizar el *backup* de los ficheros en diferentes formatos comprimidos
- Enviar los logs y *backups* por correo
- Generar un fichero con los *plugins* instalados en el sistema
- Enviar el contenido final a servicios en la nube como:
  - Dropbox
  - Amazon S3
  - Microsoft Azure
  - RackSpaceCloud
- Y muchas otras.

258. BackUPup no soporta la transmisión de ficheros por SSH, por lo que si deseamos utilizar esta forma de copia de seguridad, deberemos configurar un backup local y luego crear un script en el cron de forma que envíe automáticamente los datos a nuestra máquina remota. En nuestro ejemplo configuraremos un backup local, una copia remota por FTP, así como otra de respaldo en una cuenta de Dropbox.

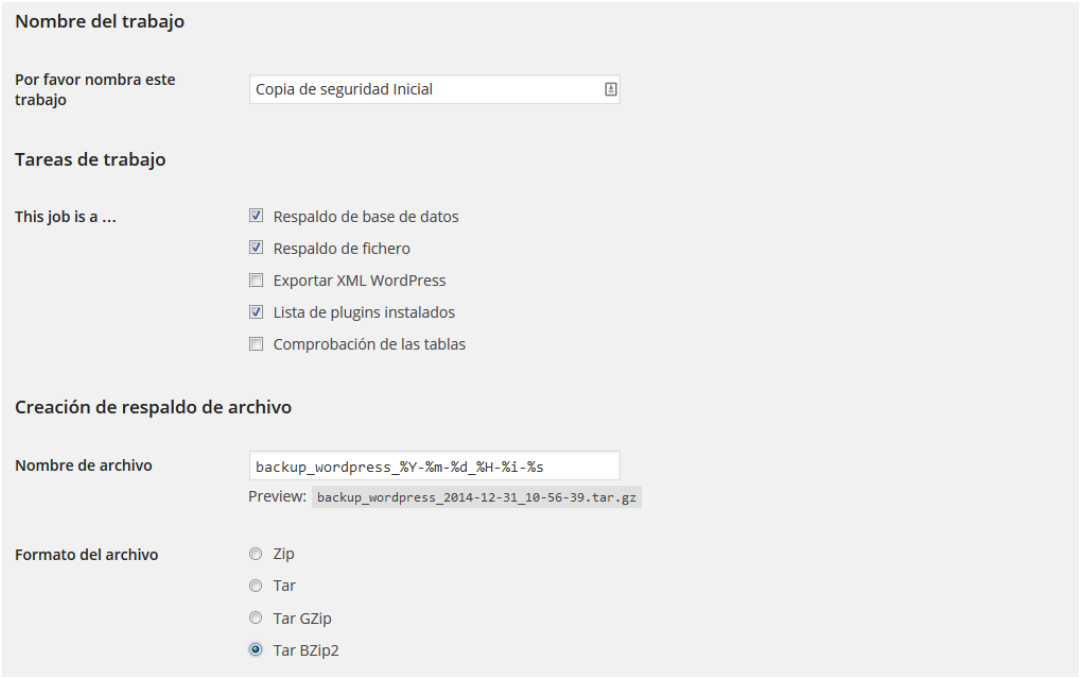
259. La instalación se realiza de la manera habitual, desde los repositorios oficiales de Wordpress. Desde 'Plugins/Añadir nuevo', buscaremos BackWPup y haremos click en el botón 'Instalar ahora' de BackUPup Free:



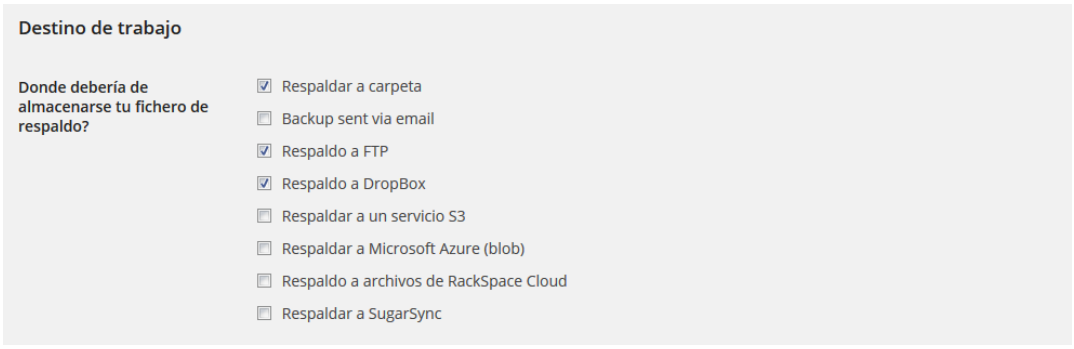
260. Una vez instalado, sólo tendremos que pulsar sobre ‘Activar el plugin’:

261. Dentro de la configuración del programa, añadiremos una nueva tarea con ‘Add new job’. Deberemos introducir una serie de valores correspondientes a la configuración de la nueva copia de seguridad que vamos a realizar. Incluiremos las siguientes opciones en la pestaña General:

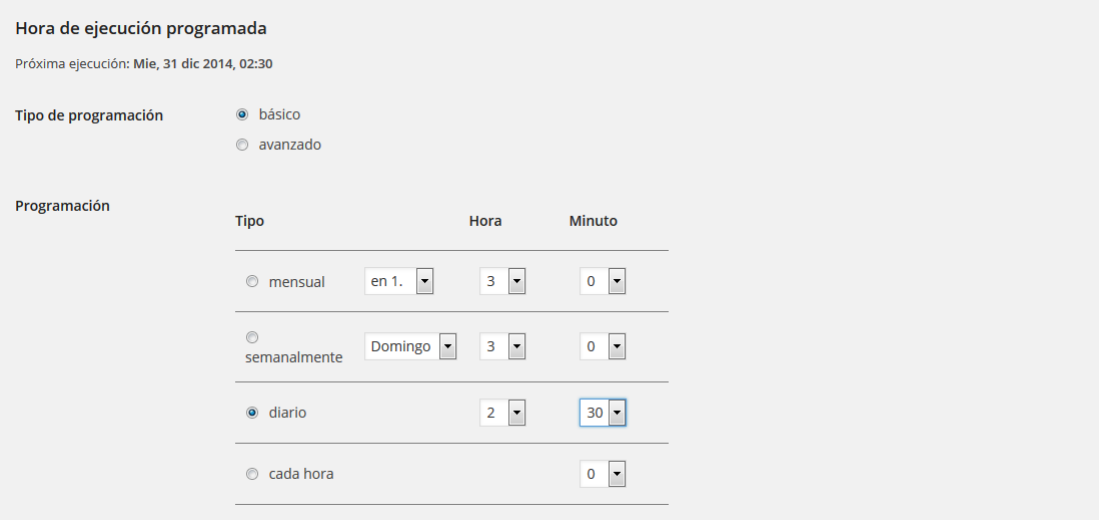
- Nombre del trabajo: Copia de seguridad inicial.
- Respaldo de base de datos.
- Respaldo de ficheros.
- Lista de *plugins* instalados.
- Nombre del archivo: nombre del fichero que deseamos que se cree. Podemos incluir los valores correspondientes de día, año, mes, hora etc. utilizando los valores %d, %Y, %m, %H.
- Formato del archivo: seleccionaremos el formato Tar BZip2 que conseguirá la máxima compresión, optimizando el espacio necesario.



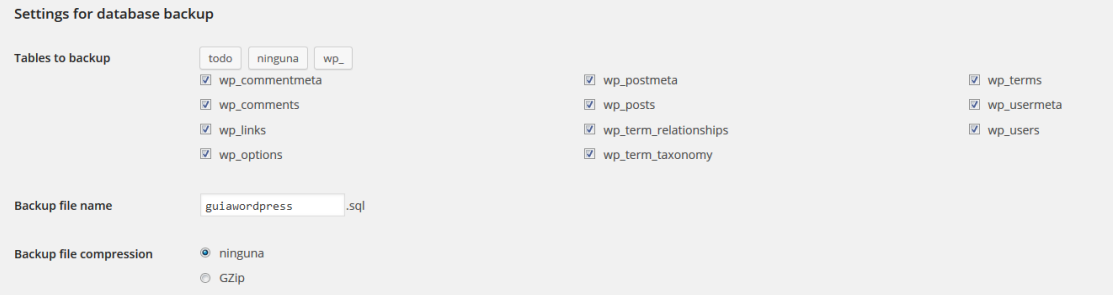
- Destino de trabajo: nos permite seleccionar dónde van a almacenarse los *backups* que se realicen de forma automática. En nuestro caso activaremos, como ejemplo, los siguientes:
    - Respaldar a carpeta
    - Respaldo a FTP
    - Respaldo a Dropbox



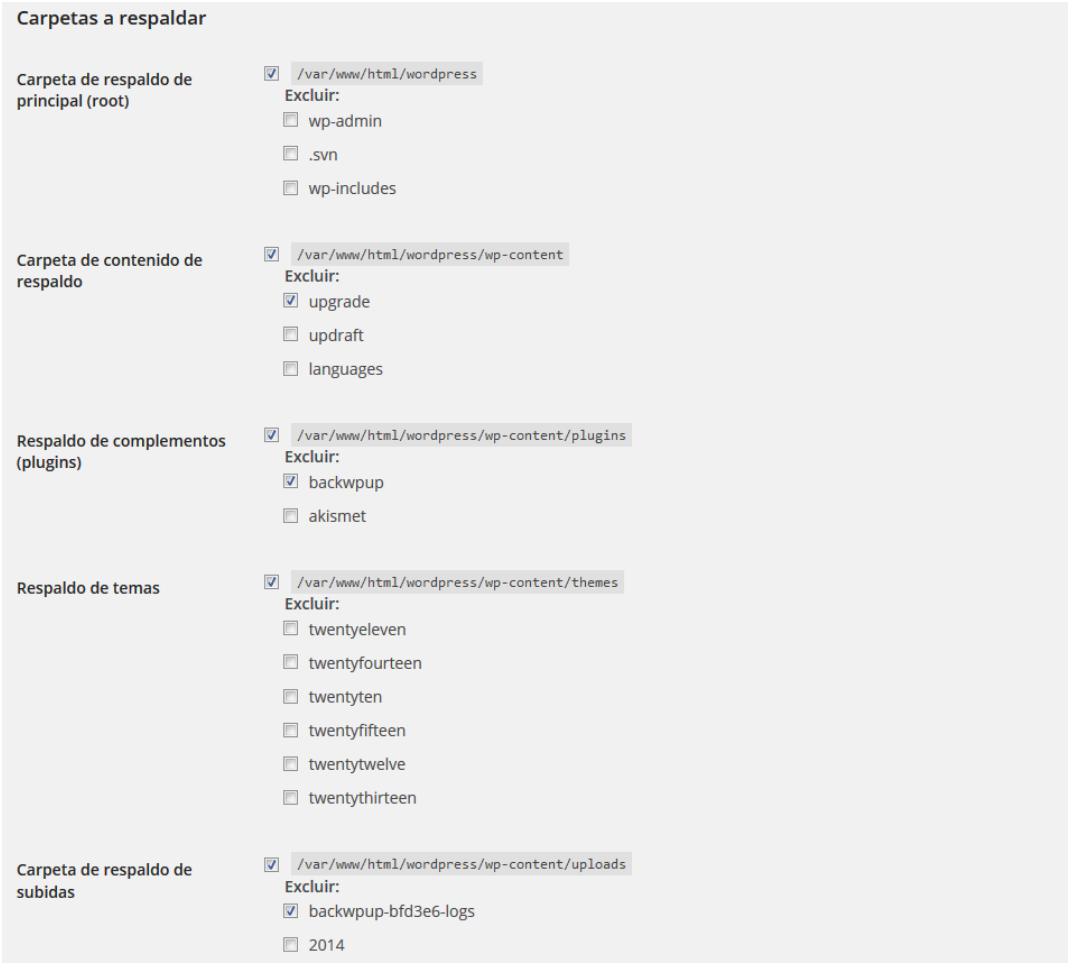
- Dentro de la pestaña 'Programación' deberemos especificar cuando queremos que el trabajo se realice de forma automática. Seleccionaremos la opción de 'Comenzar el trabajo / con cron de Wordpress', y en 'Programación' elegimos el momento para la realización de la copia (como ejemplo especificaremos realizar una copia diaria a 2.30 AM):



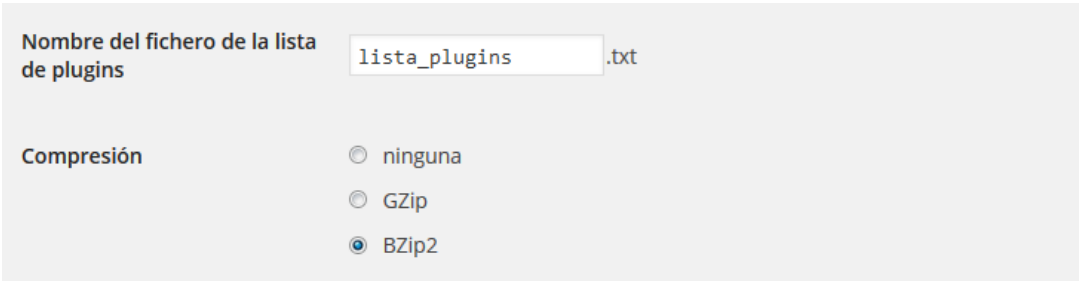
- Ahora, procederemos a la configuración del *backup* en sí, tanto en la parte de la base de datos, como la correspondiente a los ficheros. Dentro de la pestaña 'Respaldo de DB', marcaremos la opción 'Tables to backup / todo' y activaremos la compresión en formato 'GZip'.



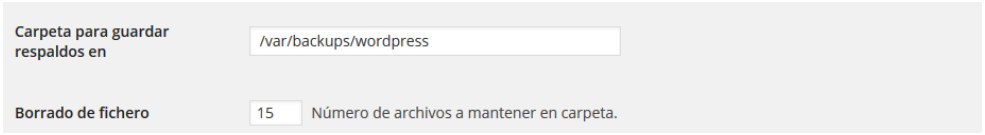
- Dentro de la pestaña 'Ficheros', marcaremos las carpetas y ficheros que deseamos que se incluyan en la copia de seguridad. La configuración por defecto será suficiente, y sólo deberemos añadir si deseamos respaldar algún tema propio, o los incluídos por defecto, donde hayamos realizado alguna modificación sustancial.



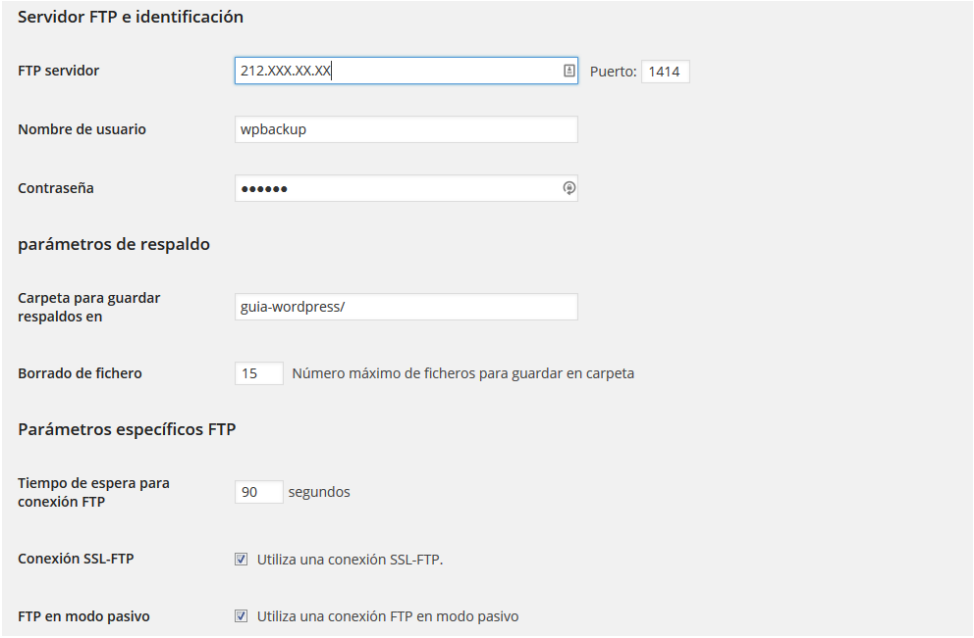
- La pestaña 'Extensiones' nos permitirá activar la opción de generar una lista de forma automática con los *plugins* que estén instalados en el sistema. Tan sólo indicaremos el nombre para este fichero y activaremos la compresión en 'BZip2':



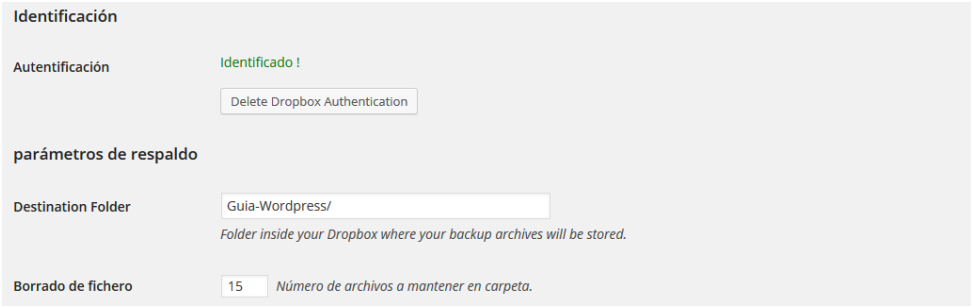
- Finalmente, sólo nos restará configurar los diferentes métodos de respaldo que hemos activado para esta copia de seguridad:
    - Dentro de la pestaña 'A: Carpeta', tan sólo deberemos indicar la ruta local donde guardar los ficheros generados:



- Dentro de la pestaña ‘A:FTP’ deberemos especificar los parámetros necesarios para la conexión al servidor remoto por FTP, como dirección, nombre de usuario y contraseña. Además, nos permitirá realizar conexiones tipo SSL-FTP si lo deseamos:

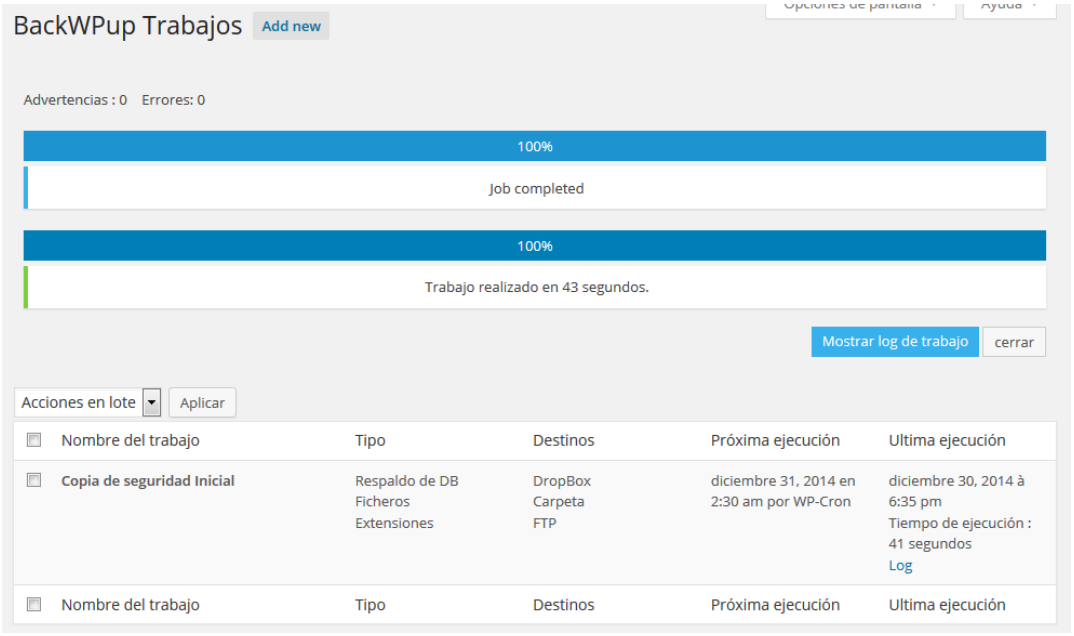


- Dentro de la pestaña ‘A: DropBox’, pulsaremos sobre ‘Get DropBox App auth code’ para permitir a BackWPup conectarse al servicio de forma automática. Una vez autoricemos a BackWPup, copiaremos el auth code que nos proporcione Dropbox y la introduciremos en el campo correspondiente:



- Cuando hayamos realizado la configuración anterior, pulsaremos sobre ‘Guardar cambios’ y ya tendremos lista nuestra copia de seguridad. Si deseamos lanzar la copia de forma inmediata, pulsaremos sobre ‘Arrancar ahora’:

- Automáticamente seremos redirigidos al menú lateral ‘BackWPup / Trabajos’, donde podremos ver el log generado durante la copia de seguridad, así como el proceso completo, próxima ejecución y los destinos seleccionados:



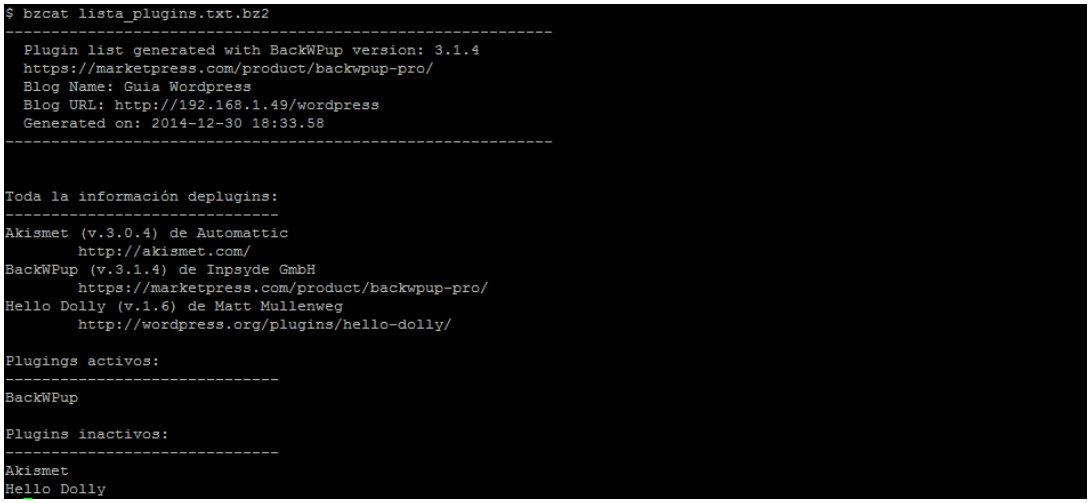
II. RESTAURANDO LA COPIA DE SEGURIDAD

262. Para restaurar la copia de seguridad realizada de forma automática con BackWPup, deberemos seguir los siguientes pasos:

1. Descargar el fichero de la última copia de seguridad que deseemos restaurar en nuestro servidor al directorio en el que wordpress fue instalado y descomprimir el fichero con el comando 'tar xvjf nombre_copia_seguridad.tar.bz2'.
2. Utilizar la consola de administración para crear la base de datos de Wordpress y después importar el fichero .sql de la copia de seguridad:

```
$ ls
backwpup_readme.txt  lista_plugins.txt.bz2  wp-admin             wp-config-sample.php  wp-links-opml.php  wp-settings.php
guiawordpress.sql    manifest.json          wp-blog-header.php   wp-content            wp-load.php        wp-signup.php
index.php            readme.html            wp-comments-post.php wp-cron.php           wp-login.php       wp-trackback.php
license.txt          wp-activate.php        wp-config.php        wp-includes           wp-mail.php        xmlrpc.php
$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 615
Server version: 5.5.40-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2014, 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 guiawordpress;
Query OK, 1 row affected (0.00 sec)

mysql> Bye
$ mysql -u root -p guiawordpress < guiawordpress.sql
Enter password:
$
```

3. Eliminamos el fichero de respaldo .sql, así como 'backwpup_readme.txt'.
4. Los *plugins* activados en la anterior instalación se encuentran dentro del fichero .bz2 con el nombre que indicamos en el momento de la generación de la copia. Dado que BackWPup no realiza copias de seguridad de estos *plugins*, será necesario visualizar los *plugins* que estuvieron instalados e instalarlos de forma manual.



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

263. Una vez finalizado este proceso, nuestra copia de seguridad quedará completamente restaurada y dispondremos una réplica exacta al momento de la última copia de seguridad que se realizó en el sistema.

21. RECUPERACIÓN ANTE UN COMPROMISO DE SEGURIDAD

264. A pesar de las medidas anteriormente descritas, es posible que en algún momento suframos un compromiso en nuestro sistema. De forma general, describiremos unas pautas de actuación de cara a recuperar nuestro sitio Wordpress de forma segura:

1. **Realizar un *backup***: 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.

2. **Resetear contraseña de administrador**: en el caso de que no podamos acceder a nuestra cuenta de administrador, procederemos a recuperarla. En este caso utilizaremos WP CLI, que nos permitirá gestionar nuestra instalación de Wordpress desde la línea de comandos. La descargaremos desde la ruta donde esté instalado Wordpress con el siguiente comando:

   $ curl -O https://raw.githubusercontent.com/wp-cli/builds/gh-pages/phar/wp-cli.phar

   Una vez descargada, comprobaremos que cumplimos todos los requisitos de dependencias en el sistema ejecutando:

   $ php wp-cli.phar --info

   Podremos comprobar los usuarios generados en nuestra instalación con el siguiente comando:

   $ php wp-cli.phar user list

   Para cambiar la contraseña de cualquier usuario del sistema, procederemos de la siguiente forma:

   $ php wp-cli.phar user update <id_del_usuario> --user_pass=nueva_contraseña

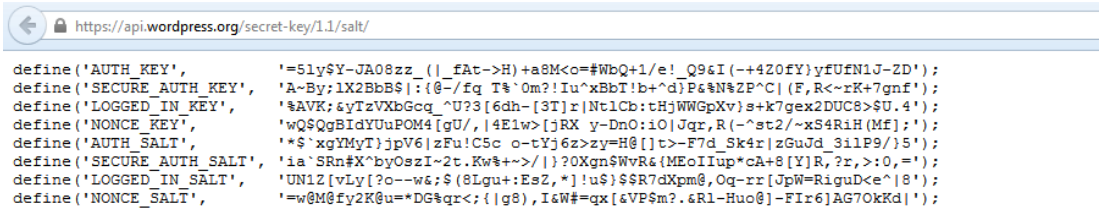
3. **Modificar la contraseña de acceso a la base de datos**: debido a que en el momento inicial no conoceremos el alcance del ataque, será necesario modificar también la contraseña de nuestro usuario de MySQL. Entraremos en la consola de administración de la base de datos e introduciremos:

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

   Deberemos modificar esta contraseña también en el fichero de configuración de Wordpress 'wp-config.php'.

4. **Modificar las Secret Keys**: Wordpress almacena un conjunto de variables aleatorias que mejoran el cifrado de la información almacenada en las cookies del usuario. Existen un total de cuatro claves de seguridad: AUTH_KEY, SECURE_AUTH_KEY, LOGGED_IN_KEY y NONCE_KEY.

   Ante un incidente de seguridad, el valor de estas variables debe modificado. Para ello visitaremos la siguiente URL con el navegador https://api.wordpress.org/secret-key/1.1/salt/:

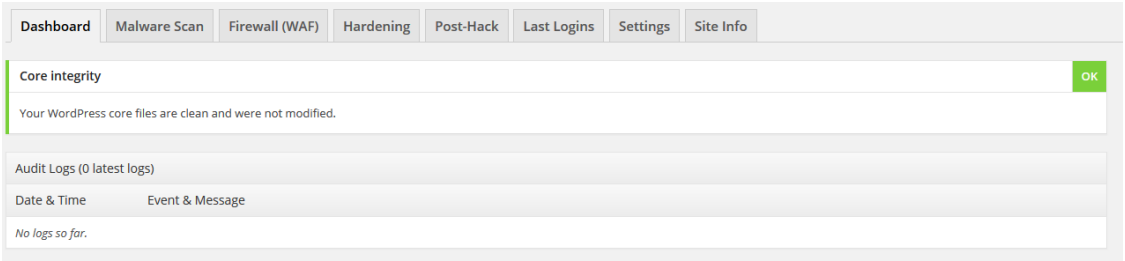
   

   O lanzaremos una consulta desde la línea de comandos de nuestro servidor:

   $ curl https://api.wordpress.org/secret-key/1.1/salt/

   Finalmente, sustituiremos estos valores en el fichero de configuración 'wp-config.php'.

5. **Escanear en busca de malware**: existen algunos scanner que nos permitirán descubrir de forma sencilla y rápida dónde se ha producido el compromiso en nuestro sistema, incluso analizando la información contenida en la base de datos. Esta información, a pesar de ser preliminar y no concluyente, nos ayudará a avanzar en la restauración del servicio. Uno de los *plugins* recomendados para esta terea será 'Sucuri Security - Auditing, Malware Scanner and Security Hardening', que podremos instalar de la forma habitual desde el repositorio oficial de Wordpress.

6. **Reinstalar Wordpress**: ahora procederemos a borrar TODOS los ficheros dentro de nuestro directorio de instalación de Wordpress, excepto el fichero 'wp-config.php' y el directorio 'wp-content'. A continuación, descargaremos una copia del software original de Wordpress. Si somos capaces de identificar si nuestro fichero de configuración no ha sido comprometido, utilizaremos este para la nueva instalación. En caso contrario, modificaremos los valores necesarios de configuración de base de datos sobre el nuevo, y procederemos a borrar el antiguo.

Si hemos realizado la instalación de Wordpress utilizando SVN, podremos comprobar de forma sencilla, y desde la línea de comandos, qué ficheros han sido modificados respecto a los originales utilizando el siguiente comando:

$ svn diff .

7. **Revisión del directorio 'wp-content'**: será necesario inspeccionar los ficheros dentro de este directorio en busca de posibles puertas traseras o código malicioso que el atacante haya podido incluir. En caso de no disponer de esa capacidad, podemos restaurar el contenido de una copia de seguridad anterior al compromiso.
8. **Revisión y reinstalación de *plugins***: ahora procederemos a localizar los *plugins* que estaban anteriormente instalados, los desactivaremos, eliminaremos y reinstalaremos del repositorio oficial de Wordpress de nuevo.
9. **Revisión de temas**: eliminaremos cualquier tema que no esté en uso, incluyendo aquellos que vienen por defecto con la instalación. Será necesario revisar el código PHP y los ficheros y referencias a Javascript, por si estos hubieran sido comprometidos. Generalmente, si existe este código malicioso, estará ubicado en los ficheros 'header.php' o 'footer.php'. Si tenemos acceso al tema original o disponemos de una copia de nuestra modificación propia, procederemos a reinstalarla de nuevo.
10. **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.
11. **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.

22. SEGURIDAD PERSONAL

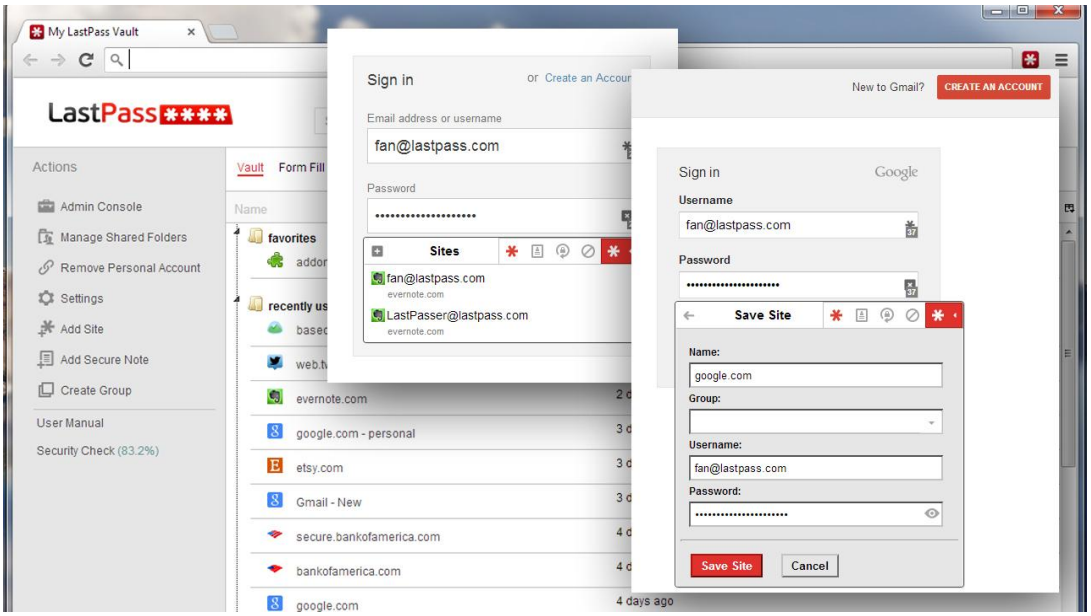
22.1. USO DE GESTORES DE CONTRASEÑAS

265. Un gestor de contraseñas es un programa que se utiliza para almacenar una gran cantidad de parejas usuario/contraseña. La base de datos donde se guarda esta información está cifrada mediante una única clave, de forma que el usuario sólo tenga que memorizar una clave para acceder a todas las demás.

266. Además de generar contraseñas robustas, estos gestores se integran en nuestro navegador para rellenar los formularios de login de los sitios web, de tal forma que con sólo pulsar un botón se copie el usuario y contraseña. También son capaces de rellenar automáticamente perfiles cuando nos registramos, o de guardar las contraseñas cuando entramos en un sitio que no teníamos guardado. En muchos casos también podremos guardar otro tipo de credenciales, aunque no estén ligadas a páginas web.

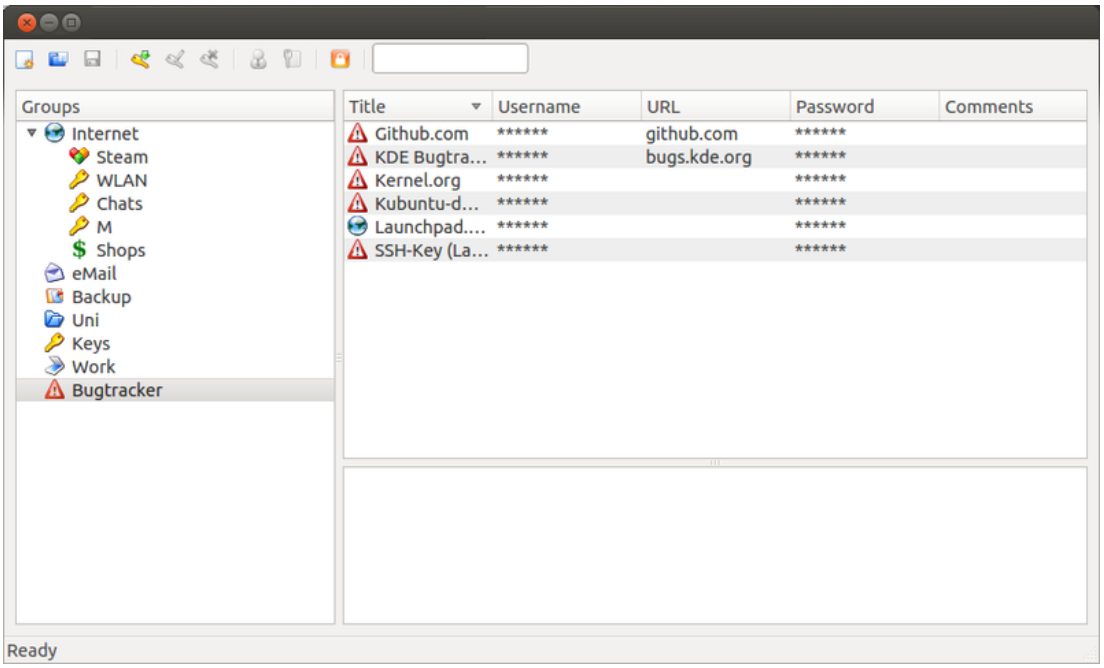
267. En esta guía, recomendaremos el uso de los siguientes gestores:

- **LastPass**: un servicio de gestión de contraseñas multiplataforma, además disponible como complemento para varios navegadores como Google Chrome, Firefox, Opera, Safari e Internet Explorer. Su objetivo es resolver el problema de la fatiga que crea en el usuario el recordar contraseñas centralizando la gestión de las mismas en la nube. Para usarlo solo debes crear y recordar un password maestro con el que abres la aplicación y accedes al resto. Las contraseñas son cifradas localmente y sincronizadas a los navegadores soportados.

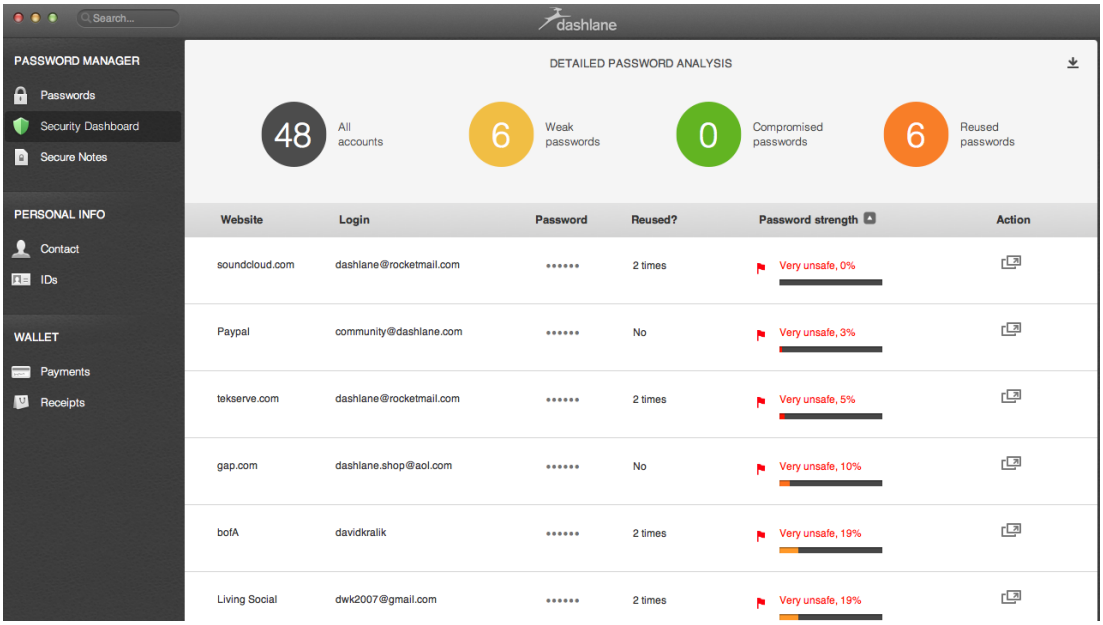


- **1Password**: un gestor de contraseñas disponible para varias plataformas que te permite administrar todas tus credenciales desde un mismo sitio. De igual manera que LastPass, su acceso está sujeto a una contraseña única o maestra que protege al resto. También poseen complemento para Google Chrome, pudiendo así sincronizar y acceder a tus contraseñas desde donde desees. Las mismas son cifradas con AES 256 bits o Rijndael, un esquema muy seguro.

- **KeePassX**: un gestor de contraseñas multiplataforma, open source y cifrado. Almacena tus contraseñas de una manera segura para que no tengas que recordarlas más. Solo una contraseña maestra es requerida para acceder a la aplicación; genera archivos cifrados con tus contraseñas que puedes guardar en una carpeta compartida o enviar por e-mail si lo necesitas. Su base de datos siempre está cifrada, bien sea con AES o el algoritmo cifrado Twofish con una llave de 256 bits. KeePassX es un port de KeePass Password Safe, un gestor de contraseñas de Windows, por lo que son compatibles.



- **DashLane**: un gestor de contraseñas multiplataforma que almacena todo tipo de credenciales, desde contraseñas hasta cuentas bancarias. Posee una función de auto-completado de formularios que muchos estiman. Su interfaz es agradable y aunque tiene opción de sincronizarse en la nube, es opcional.



22.2. ANTIVIRUS

268. Los virus, los gusanos y los caballos de Troya son programas creados por atacantes, que utilizan Internet para infectar equipos vulnerables. Los virus y los gusanos pueden auto-replicarse de un equipo a otro, mientras que los caballos de Troya entran en un equipo ocultándose dentro de un programa aparentemente de confianza, por ejemplo, un protector de pantalla. Pueden borrar información del disco duro o deshabilitar completamente el equipo. Otros no causan ningún daño directo, pero empeoran el rendimiento o la estabilidad de nuestro sistema.

269. Cada día se identifican nuevos virus, por lo que es importante usar un programa antivirus que pueda actualizarse de manera automática. Cuando actualiza el programa, los virus nuevos se agregan a una lista de virus que se deben comprobar, lo que ayuda a proteger el equipo de los nuevos ataques. Si la lista de virus está obsoleta, su equipo será vulnerable a las nuevas amenazas.

270. En la actualidad existe gran cantidad de productos de antivirus, entre los que destacaremos los siguientes:

- **Avast! Free Antivirus**: es un antivirus gratuito que no sólo cuenta con versión para Windows, sino que también dispone de una para equipos con sistema operativo Linux. Ofrece los niveles de protección básica para un ordenador: antivirus y antispyware. Además, cuenta con una opción de *sandbox* que permite ejecutar cualquier archivo sospechoso en la nube antes de que se cargue en el ordenador, evitando así problemas posteriores.
- **Windows Defender**: es el nuevo antivirus gratuito de Microsoft que viene a sustituir a Microsoft Windows Essentials, heredando muchas de sus opciones. Su funcionamiento es realmente bueno a la hora de poner coto a los ataques de virus en tu PC. Sobretodo destaca en lo que se refiere a la detección de virus en archivos comprimidos, tanto online como offline, donde se postula como uno de los mejores en este sentido. Windows Defender viene preinstalado con Windows 7 y 8.

22.3. PHISHING

271. El *phishing* es un método de estafa en línea, y los autores de estos fraudes, conocidos como "phishers", son artistas del engaño con conocimientos técnicos. En una típica estafa electrónica, los phishers envían correos electrónicos, que parecen provenir de una compañía legítima, con los cuales intentan engañar a los usuarios para que proporcionen información privada que se usará para robar la identidad. Se utilizan frecuentemente dispositivos sofisticados para robar información, inclusive ventanas emergentes, máscaras de URL que aparentan ser direcciones reales de sitios Web y programas de registro de pulsaciones de teclado, que capturan nombres y contraseñas de cuentas.

272. Para protegerse del phishing, será necesario seguir estas pautas básicas:

- Desconfíe de los correos electrónicos que le solicitan información confidencial, en particular de aquellos de índole financiera. Las organizaciones legítimas nunca solicitarán información confidencial por correo electrónico.
- No se sienta obligado a revelar información confidencial. A los *phishers* les gusta aplicar tácticas alarmistas, y pueden amenazar con desactivar una cuenta o retrasar los servicios hasta que actualice determinada información. Asegúrese de contactar directamente con la empresa para confirmar la autenticidad de la solicitud.
- Tenga cuidado con las solicitudes genéricas de información. Los correos electrónicos fraudulentos no suelen estar personalizados, mientras que los correos bancarios auténticos suelen mencionar alguna cuenta que tiene en ese banco.
- Nunca envíe información confidencial a través de formularios integrados en los mensajes de correo.
- Nunca utilice vínculos que aparecen en un correo electrónico para conectarse a un sitio Web. Abra una ventana nueva del navegador y escriba la URL directamente en la barra de direcciones.

22.4. FIREWALL

273. Un firewall es un programa que funciona en su ordenador de forma permanente. El programa monitoriza las conexiones que entran y salen del ordenador y es capaz de distinguir las que son legítimas de las que son realizadas por intrusos. El firewall, junto con un antivirus, proporciona el grado más elevado posible de seguridad con herramientas comerciales.

274. La mayoría de sistemas operativos integran entre sus características un firewall, simplemente tienes que asegurarte que está activo y configurado. Si no te convence el firewall que incorpora tu sistema operativo, puedes hacer uso de firewalls desarrollados por otros fabricantes instalándolos como cualquier otra aplicación (muchos de los antivirus integran funciones de cortafuegos). Sin embargo, es muy importante que tengas en cuenta que sólo puede haber uno funcionando en tu equipo; por lo que, si quieres utilizar un firewall que no sea el del sistema operativo, antes de empezar a ejecutar el nuevo, deshabilita el de tu sistema operativo primero. Estas herramientas generalmente aportan más información y permiten más control sobre las conexiones, aunque también son más complejas de manejar.

275. Un sistema firewall contiene un conjunto de reglas predeterminadas que le permiten al sistema:

- Autorizar la conexión (permitir)
- Bloquear la conexión (denegar)
- Rechazar la petición de conexión sin informar al que lo envió (negar)

276. Durante los primeros días de funcionamiento, el firewall personal enviará un gran número de mensajes. Esos avisos serán básicamente de dos tipos:

- Peticiones de conexión de programas: cuando se utiliza un programa, normalmente se establecen conexiones a internet. El firewall detectará esa conexión y advertirá de ello al usuario.
- Ataques detectados: el sistema le advertirá que su sistema está siendo atacado. La frecuencia de los ataques puede llegar a ser de hasta 3 o 4 por hora, aunque la mayoría de las veces se trata de ataques que se dirigen a redes enteras y que no afectan a ninguna máquina.

277. Se recomienda utilizar todos los programas con conexión a la red disponibles durante los primeros días hasta que no existan mensajes de aviso y el firewall personal los tenga todos registrados.

22.5. SEGURIDAD WI-FI

278. A continuación se proponen una serie de acciones con las que poder aumentar la seguridad de las conexiones a través de una red pública:

- Sospecha de todas las redes públicas: No asumas sin más que el enlace Wi-Fi es legítimo. Podría ser un enlace falso configurado por un atacante que está tratando de capturar información personal valiosa de usuarios ingenuos. Duda de todo y no te conectes a un punto de acceso inalámbrico desconocido o no reconocido.
- Uso de una VPN: Es indispensable contar con una conexión de red privada virtual (VPN) cuando nos conectarnos a nuestra organización a través de una conexión no segura, como un punto de acceso de una red Wi-Fi. Incluso si un atacante logra posicionarse en medio de la conexión, los datos estarán cifrados a conciencia. Como la mayoría de los atacantes persiguen objetivos fáciles, probablemente se desharán de la información robada en lugar de comenzar un largo proceso de descifrado.
- Uso de conexiones SSL: Activa la opción "Usar siempre HTTPS" en los sitios web que visites con frecuencia, o que requieran que introduzcas algún tipo de credenciales. Recuerda que los atacantes saben cómo reutiliza la gente las contraseñas, así que es probable que el nombre de usuario y la contraseña usados para un foro sean los mismos que los usados para una página bancaria o red corporativa. De este modo, enviar dichas credenciales de forma no cifrada podría abrir la puerta a un hacker inteligente. La mayoría de los sitios web que requieren una cuenta o credenciales cuentan con la opción "HTTPS" en alguna parte de su configuración.

- Desactiva los recursos compartidos: Cuando te conectas a Internet en un lugar público, es poco probable que sea necesario compartir algo. Puedes desactivar el uso compartido de datos en las preferencias del sistema o en el Panel de Control, en función del sistema operativo.
- Desactiva la Wi-Fi cuando no esté en uso: Incluso si no te has conectado de forma activa a una red, el hardware de conexión inalámbrica de tu equipo continúa transmitiendo datos con cualquier red dentro de su rango de alcance. Existen medidas de seguridad para evitar que esta comunicación sin importancia llegue a ser peligrosa, pero no todos los routers inalámbricos son iguales.
- Parches de seguridad: Es conveniente actualizar las aplicaciones que están en los dispositivos ya que de no hacerlo, un atacante puede utilizar los fallos de seguridad que las aplicaciones puedan tener y tener por tanto acceso a los dispositivos.
- Utiliza redes con cifrados seguros: ante la opción de elegir entre diferentes redes, es mejor utilizar la que tenga el mejor protocolo de seguridad. De más a menos segura sería WPA2, WPA y WEP.
- Limpieza de lista de redes Wi-Fi: Una vez que el dispositivo se conecta a una red, ésta queda memorizada. Es interesante poder realizar una limpieza de todas aquellas redes que han sido utilizadas en un momento puntual y que son inseguras para dejar solo aquellas que son de confianza.

ANEXO A. LISTA DE CONSTANTES DE WORDPRESS

GENERAL

- AUTOSAVE_INTERVAL

  Define un intervalo en el que WordPress debería hacer un autoguardado.

  Valor: tiempo en segundos (Por defecto: 60)

- CORE_UPGRADE_SKIP_NEW_BUNDLED

  Permite saltar nuevos archivos en paquete como en *plugins* y/o temas en las actualizaciones.

  Valores: true|false

- DISABLE_WP_CRON

  Desactiva la función cron de WordPress.

  Valor: true

- EMPTY_TRASH_DAYS

  Controla el número de días antes de que WordPress borre permanentemente entradas, páginas, adjuntos y comentarios de la papelera de reciclaje.

  Valor: tiempo en días (Por defecto: 30)

- IMAGE_EDIT_OVERWRITE

  Permite a WordPress sobreescribir una imagen antes de editar o guardar la imagen como copia.

  Valores: true|false

- MEDIA_TRASH

  Activa/Desactiva la función de papelera de reciclaje para los medios.

  Valores: true|false (Por defecto: false)

- WPLANG

  Define el idioma en que se usará WordPress.

  Valores: Para Español es_ES

- WP_DEFAULT_THEME

  Define el tema por defecto para los sitios nuevos, también sirve como respaldo en caso de fallo del tema activo.

  Valor: nombre del tema (Por defecto: twentyeleven)

- WP_CRON_LOCK_TIMEOUT

  Define un periodo de tiempo en el que se finalizará un único “cronjob”. Desde WordPress 3.3.

  Valor: tiempo en segundos (Por defecto: 60)

- WP_MAIL_INTERVAL

  Define un periodo de tiempo en el que se podrá hacer una única petición de email.

  Valor: tiempo en segundos (Por defecto: 300)

- WP_POST_REVISIONS

  Activa/desactiva la función de revisión de entradas . Un numero mayor que 0 define el número de revisiones para las entradas.

  Valores: true|false|número (Por defecto: true)

- WP_MAX_MEMORY_LIMIT

  Te permite cambiar el límite máximo de memoria para algunas funciones de WordPress.

  Valores: Ver en PHP docs (Por defecto: 256M)

- WP_MEMORY_LIMIT

  Define el límite de memoria para WordPress.

  Valores: Ver en PHP docs (Por defecto: 32M, para Multisitio 64M)

ESTADO

- APP_REQUEST

  Se define si hay una petición del protocolo de publicación Atom.

  Valor: true

- COMMENTS_TEMPLATE

  Se define si se carga la plantilla de comentarios.

  Valor: true

- DOING_AJAX

  Se define si hay una petición AJAX.

  Valor: true

- DOING_AUTOSAVE

  Se define si WordPress está autoguardando entradas.

  Valor: true

- DOING_CRON

  Se define si WordPress está realizando un cronjob.

  Valor: true

- IFRAME_REQUEST

  Se define si hay una petición inlineframe.

  Valor: true

- IS_PROFILE_PAGE

Se define si el usuario cambia los ajustes de su perfil.

Valor: true

- SHORTINIT

  Puede definirse para cargar solo la mitad de WordPress.

  Valor: true

- WP_ADMIN

  Se define si hay una petición en el escritorio de WordPress.

  Valor: true

- WP_BLOG_ADMIN

  Se define si hay una petición en /wp-admin/.

  Valor: true

- WP_IMPORTING

  Se define si WordPress está importando datos.

  Valor: true

- WP_INSTALLING

  Se define en nuevas instalaciones o actualizaciones.

  Valor: true

- WP_INSTALLING_NETWORK

  Se define si hay una petición en la administración o instalación de la red. Desde WordPress 3.3, antes era WP_NETWORK_ADMIN_PAGE.

  Valor: true

- WP_LOAD_IMPORTERS

  Se define si visitas el Importador (Herramientas → Importar).

  Valor: true

- WP_NETWORK_ADMIN

  Se define si hay una petición en /wp-admin/network/.

  Valor: true

- WP_REPAIRING

  Se define si hay una petición en /wp-admin/maint/repair.php.

  Valor: true

- WP_SETUP_CONFIG

  Se define si se va a instalar o configurar WordPress.

  Valor: true

- WP_UNINSTALL_PLUGIN

Se define si se va a desinstalar un *plugin* (para uninstall.php).

Valor: true

- WP_USER_ADMIN

  Se define si hay una petición en /wp-admin/user/.

  Valor: true

- XMLRPC_REQUEST

  Se define si hay una petición en la API XML-RPC.

  Valor: true

RUTAS, DIRECTORIOS Y ENLACES

- ABSPATH

  Ruta absoluta al directorio raíz de WordPress.

  Por defecto: path to wp-load.php

- WPINC

  Ruta relativa a /wp-includes/. No puedes cambiar esto.

  Por defecto: wp-includes

- WP_LANG_DIR

  Ruta absoluta al directorio con los archivos de idioma.

  Por defecto: WP_CONTENT_DIR /languages or WP_CONTENT_DIR WPINC /languages

- WP_PLUGIN_DIR

  Ruta absoluta al directorio de *plugins*.

  Por defecto: WP_CONTENT_DIR /*plugins*

- WP_PLUGIN_URL

  URL al directorio de *plugins*.

  Por defecto: WP_CONTENT_URL /*plugins*

- WP_CONTENT_DIR

  Ruta absoluta a wp-content.

  Por defecto: ABSPATH wp-content

- WP_CONTENT_URL

  URL al directorio wp-content.

  Por defecto: {Site URL}/wp-content

- WP_HOME

  URL de la portada de tu WordPress.

- WP_SITEURL

  URL al directorio raíz de tu WordPress.

- WP_TEMP_DIR

  Ruta absoluta a un directorio donde se puedan guardar los archivos temporales.

- WPMU_PLUGIN_DIR

  Ruta absoluta al directorio de *plugins* disponibles en multisitio.

Por defecto: WP_CONTENT_DIR /mu-*plugins*

- WPMU_PLUGIN_URL

  URL al directorio de *plugins* disponibles en multisitio.

  Por defecto: WP_CONTENT_URL /mu-*plugins*

BASE DE DATOS

- DB_CHARSET

  Define el mapa de caracteres de la base de datos.

  Valores: Ver los MySQL docs (Por defecto: utf8)

- DB_COLLATE

  Define el cotejo de la base de datos.

  Valores: Ver los MySQL docs (Por defecto: utf8_general_ci)

- DB_HOST

  Define el servidor de la base de datos.

  Valores: dirección IP, dominio y/o puerto (Por defecto: localhost)

- DB_NAME

  Define el nombre de la base de datos.

  Valor: nombre de la base de datos

- DB_PASSWORD

  Define la contraseña de la base de datos.

- DB_USER

  Define el usuario de la base de datos.

- WP_ALLOW_REPAIR

  Te permite reparar y optimizar automáticamente las tablas de la base de datos con /wp-admin/maint/repair.php.

  Valor: true

- CUSTOM_USER_TABLE

  Te permite definir un usuario personalizado para la base de datos.

Valor: nombre de la tabla

- CUSTOM_USER_META_TABLE

  Te permite definir una tabla meta de usuario personalizada.

  Valor: nombre de la tabla

MULTISITIO

- ALLOW_SUBDIRECTORY_INSTALL

  Te permite instalar Multisitio en un subdirectorio.

  Valor: true

- BLOGUPLOADDIR

  Ruta absoluta al directorio de cargas del sitio concreto.

  Por defecto: WP_CONTENT_DIR /blogs.dir/{Blog ID}/files/

- BLOG_ID_CURRENT_SITE

  ID del blog del sitio principal.

  Por defecto: 1

- DOMAIN_CURRENT_SITE

  Dominio del sitio principal.

  Por defecto: dominio

- DIEONDBERROR

  Cuando se define se muestran en pantalla los errores de la base de datos.

  Valor: true

- ERRORLOGFILE

  Cuando se define se guardan en un archivo de registro los errores de la base de datos.

  Valor: ruta absoluta a un archivo con permisos de escritura

- MULTISITE

  Se define si se va a usar Multisitio.

  Valor: true

- NOBLOGREDIRECT

  Define una URL de un sitio al que WordPress debería redirigir si está cerrado el registro o un sitio no existe.

  Valores: %siteurl% para el sitio principal o URL personalizada

- PATH_CURRENT_SITE

  Ruta al sitio principal.

- UPLOADBLOGSDIR

Ruta al directorio base de subidas, relativo a ABSPATH.

Por defecto: wp-content/blogs.dir

- SITE_ID_CURRENT_SITE

  ID de la red del sitio principal.

  Por defecto: 1

- SUBDOMAIN_INSTALL

  Define si se instalará un subominio o no.

  Valores: true|false

- SUNRISE

  Cuando se define WordPres cargará el archivo /wp-content/sunrise.php.

  Valor: true

- UPLOADS

  Ruta al directorio de subidas específico de un sitio, relativo a ABSPATH.

  Por defecto: UPLOADBLOGSDIR /{blogid}/files/

- WPMU_ACCEL_REDIRECT

  Activa/Desactiva soporte para X-Sendfile Header.

  Valores: true|false (Por defecto: false)

- WPMU_SENDFILE

  Activa/Desactiva soporte para X-Accel-Redirect Header.

  Valores: true|false (Por defecto: false)

- WP_ALLOW_MULTISITE

  Cuando se define estará disponible la función de Multisitio (Herramientas → Configurar Red).

  Valor: true

CACHE Y COMPRESIÓN DE SCRIPTS

- WP_CACHE

  Cuando se define WordPress cargará el archivo /wp-content/advanced-cache.php.

  Valores: true|false (Por defecto: false)

- COMPRESS_CSS

  Activa/Desactiva la compresión de las hojas de estilo.

  Valores: true|false

- COMPRESS_SCRIPTS

  Activa/Desactiva la compresión de archivos Javascript.

Valores: true|false

- CONCATENATE_SCRIPTS

  Activa/Desactiva la consolidación de archivos CSS y Javascript antes de comprimirlos.

  Valores: true|false

- ENFORCE_GZIP

  Activa/Desactiva la salida gzip.

  Valores: true|false

SISTEMA DE FICHEROS Y CONEXIONES

- FS_CHMOD_DIR

  Define los permisos de lectura y escritura de los directorios.

  Valores: Ver PHP Handbuch (Por defecto: 0755)

- FS_CHMOD_FILE

  Define los permisos de lectura y escritura de los archivos.

  Valores: Ver PHP Handbuch (Por defecto: 0644)

- FS_CONNECT_TIMEOUT

  Define el tiempo máximo para establecer una conexión.

  Valores: tiempo en segundos (Por defecto: 30)

- FS_METHOD

  Define el método para conectarse al sistema de archivos.

  Valores: direct|ssh|ftpext|ftpsockets

- FS_TIMEOUT

  Define el tiempo máximo para una conexión perdida.

  Valores: tiempo en segundos (Por defecto: 30)

- FTP_BASE

  Ruta al directorio raíz de WordPress.

  Por defecto: ABSPATH

- FTP_CONTENT_DIR

  Ruta al directorio /wp-content/.

  Por defecto: WP_CONTENT_DIR

- FTP_HOST

  Define el servidor FTP.

  Valores: Dirección IP, dominio y/o puerto

- FTP_LANG_DIR

  Ruta al directorio con los archivos del idioma.

  Por defecto: WP_LANG_DIR
- FTP_PASS

  Define la contraseña FTP.
- FTP_PLUGIN_DIR

  Define el directorio de *plugins*.

  Por defecto: WP_PLUGIN_DIR
- FTP_PRIKEY

  Define una clave privada para SSH.
- FTP_PUBKEY

  Define una clave pública para SSH.
- FTP_SSH

  Activa/Desactiva SSH.

  Valores: true|false
- FTP_SSL

  Activa/Desactiva SSL.

  Valores: true|false
- FTP_USER

  Define el usuario FTP.
- WP_PROXY_BYPASS_HOSTS

  Te permite definir algunas direcciones que no pasarán por el proxy.

  Valores: www.ejemplo.com, *.ejemplo.org
- WP_PROXY_HOST

  Define la dirección del proxy.

  Valores: Dirección IP o dominio
- WP_PROXY_PASSWORD

  Define la contraseña del proxy.
- WP_PROXY_PORT

  Define el puerto del proxy.
- WP_PROXY_USERNAME

  Define el usuario del proxy.
- WP_HTTP_BLOCK_EXTERNAL

  Te permite bloquear peticiones externas.

Valores: true|false

- WP_ACCESSIBLE_HOSTS

  Si se define WP_HTTP_BLOCK_EXTERNAL puedes añadir servidores que no deberían bloquearse.

  Valores: www.ejemplo.com, *.ejemplo.org

TEMAS

- BACKGROUND_IMAGE

  Define una imagen de fondo por defecto.

- HEADER_IMAGE

  Define una imagen de cabecera por defecto.

- HEADER_IMAGE_HEIGHT

  Define la altura de la imagen de cabecera.

- HEADER_IMAGE_WIDTH

  Define el ancho de la imagen de cabecera.

- HEADER_TEXTCOLOR

  Define el color de fuente del texto de la cabecera.

- NO_HEADER_TEXT

  Activa/Desactiva el soporte para texto en la cabecera.

  Valores: true|false

- STYLESHEETPATH

  Define la ruta absoluta a la hoja de estilos del tema actual.

- TEMPLATEPATH

  Define la ruta absoluta a los archivos de plantilla del tema actual.

- WP_USE_THEMES

  Activa/Desactiva la activación de temas.

  Valores: true|false

DEBUG

- SAVEQUERIES

  Activa/Desactiva el guardado de las queries de la base de datos en un array ($wpdb->queries).

  Valores: true|false

- SCRIPT_DEBUG

  Activa/Desactiva la activación de archivos comprimidos CSS y Javascript.

Valores: true|false

- WP_DEBUG

  Activa/Desactiva el modo debug en WordPress.

  Valores: true|false (Por defecto: false)

- WP_DEBUG_DISPLAY

  Activa/Desactiva la visualización de errores en pantalla.

  Valores: true|false|null (Por defecto: true)

- WP_DEBUG_LOG

  Activa/Desactiva la escritura de errores en el archivo /wp-content/debug.log.

  Valores: true|false (Por defecto: false)

SEGURIDAD Y COOKIES

- ADMIN_COOKIE_PATH

  Ruta al directorio /wp-admin/.

  Por defecto: SITECOOKIEPATH wp-admin o para Multisitio en subdirectorio SITECOOKIEPATH

- ALLOW_UNFILTERED_UPLOADS

  Permite subidas sin filtrado para los administradores.

  Valor: true

- AUTH_COOKIE

  Nombre de la cookie para la identificación.

  Por defecto: wordpress_ COOKIEHASH

- AUTH_KEY

  Clave secreta.

  Valores: Ver el generador

- AUTH_SALT

  Clave secreta.

  Valores: Ver el generador

- COOKIEHASH

  Hash para generar nombres de las cookies.

- COOKIEPATH

  Ruta al directorio raíz de WordPress.

  Por defecto: URL de la portada sin http(s)://

- COOKIE_DOMAIN

Dominio de la instalación de WordPress.

Por defecto: false o para Multisite con subdominios .dominio el sitio principal

- CUSTOM_TAGS

  Te permite sobreescribir la lista de tags HTML seguras. Echa un vistazo al archivo /wp-includes/kses.php.

  Valores: true|false (Por defecto: false)

- DISALLOW_FILE_EDIT

  Te permite desactivar la edición de archivos de temas y *plugins* con el editor de WordPress.

  Valor: true

- DISALLOW_FILE_MODS

  Te permite desactivar la edición, actualización, instalación y borrado de *plugins*, temas y archivos del núcleo desde el escritorio de WordPress.

  Valor: true

- DISALLOW_UNFILTERED_HTML

  Te permits desactivar el HTML sin filtrado para todos los usuarios, administradores incluidos.

  Valor: true

- FORCE_SSL_ADMIN

  Activa SSL para los accesos y el escritorio.

  Valores: true|false (Por defecto: false)

- FORCE_SSL_LOGIN

  Activa SSL para los accesos.

  Valores: true|false (Por defecto: false)

- LOGGED_IN_COOKIE

  Nombre de la cookie para los accesos.

  Por defecto: wordpress_logged_in_ COOKIEHASH

- LOGGED_IN_KEY

  Clave secreta.

  Valores: Ver el generador

- LOGGED_IN_SALT

  Clave secreta.

  Valores: Ver el generador

- NONCE_KEY

  Clave secreta.

Valores: Ver el generador

- NONCE_SALT

  Clave secreta.

  Valores: Ver el generador

- PASS_COOKIE

  Nombre de la cookie para la contraseña.

  Por defecto: wordpresspass_ COOKIEHASH

- PLUGINS_COOKIE_PATH

  Ruta al directorio de *plugins*.

  Por defecto: WP_PLUGIN_URL sin http(s)://

- SECURE_AUTH_COOKIE

  Nombre de la cookie para la identificación SSL.

  Por defecto: wordpress_sec_ COOKIEHASH

- SECURE_AUTH_KEY

  Clave secreta.

  Valores: Ver el generador

- SECURE_AUTH_SALT

  Clave secreta.

  Valores: Ver el generador

- SITECOOKIEPATH

  Ruta de tu sitio.

  Por defecto: URL del sitio sin http(s)://

- TEST_COOKIE

  Nombre de la cookie para la cookie de prueba.

  Por defecto: wordpress_test_cookie

- USER_COOKIE

  Nombre de la cookie para los usuarios.

  Por defecto: wordpressuser_ COOKIEHASH