¿Es seguro su sitio de WordPress?

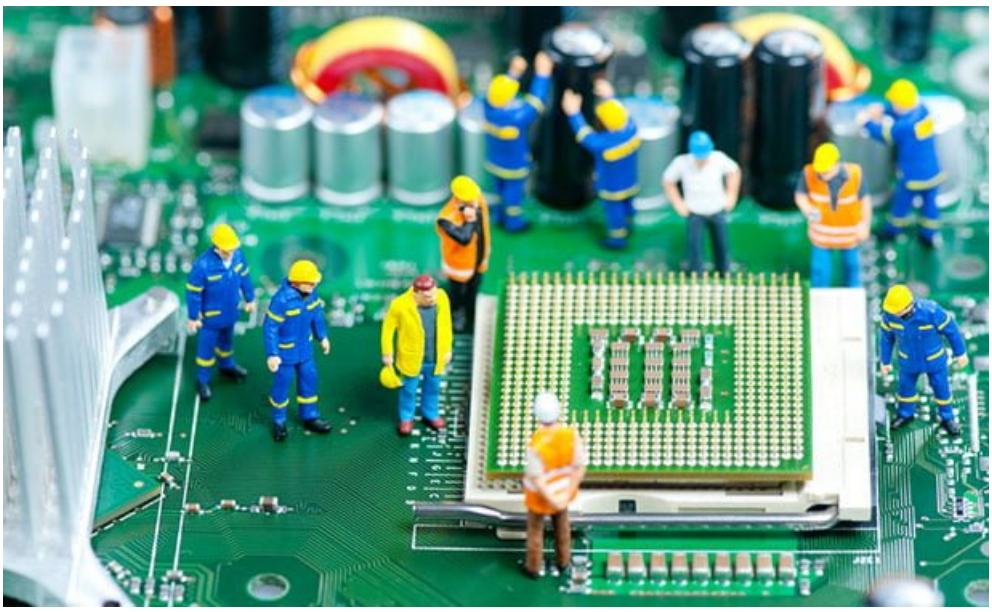
Uno de los principales problemas de la tecnología son los lapsos frecuentes en la seguridad que vivimos, una gran cantidad de información es robada todos los días y se usa para robar más información, enviar mensajes de spam, puertas traseras abiertas en los sistemas y en ocasiones incluso hacer daño a nuestros ordenadores.
Ninguna de estas cuestiones es desconocida para WordPress, un sorprendente número de sitios se han convertido en víctimas de los criminales desagradables que explotan la comunidad para su propio beneficio personal.
Con el fin de vencer esta amenaza, hemos preparado un resumen de algunas buenas herramientas y consejos sobre cómo evitar ser la próxima víctima. O si usted tiene la mala suerte de ser ya una víctima, la forma de luchar y resolver su instalación o problema.

Ataques Exploit

Es posible que haya oído hablar de él, usted puede incluso saber los detalles, pero para los que no tienen conocimiento, aquí está el problema: un exploit es una pieza de código malicioso distribuido que intenta explotar una vulnerabilidad en el código existente.

TimThumb era susceptible a un ataque de este tipo en una de sus funciones, lo que permite a los usuarios subir imágenes desde diferentes sitios y acceder a ellos libremente, las imágenes almacenadas en un directorio de caché. Esta función podría ser explotada por hackers para subir algunos archivos al servidor, lo que les permite el acceso a todos los recursos de la instalación de WordPress como deseen.
Exactamente el mismo problema afectó **Uploadify**, un plugin que permite a los usuarios subir archivos. Cuando no se controla

correctamente el plug-in permite el acceso libre a los hackers al sitio subiendo scripts PHP para conceder permisos de acceso.



El problema en estos casos, como con la mayoría de los ataques exploit, no era WordPress sino más bien los propios plugins. La solución es simple, mantener los plugins al día e informar de cualquier error que encuentre a los desarrolladores para ayudarles a arreglar posibles problemas.

Inyecciones SQL

La instalación de WordPress en sí no es inmune a los problemas. Dependiendo de la versión, la inyección SQL puede ser un gran dolor de cabeza. Una inyección SQL es un proceso por el cual un atacante trata de pasar el código SQL a través de un formulario web o script en la esperanza de que el código SQL analizará “correctamente” y recuperar datos de la base de datos. Esos datos pueden ser direcciones de correo electrónico, pero es más probable, sería nombres de usuario y contraseñas, que luego dan al usuario un mayor acceso para otros ataques.
La razón de los ataques SQL pueden ser tan irritante para combatirlos, que es necesario realizar copias de seguridad con frecuencia de la base de datos. Preferiblemente, al menos una

vez por día.

Para evitar esto, se puede tratar de proteger sus archivos usando Apache con un código como este en el archivo htaccess.:

```
<IfModule mod_rewrite.c>

RewriteEngine On

RewriteBase /

RewriteCond %{REQUEST_METHOD} ^(HEAD|TRACE|DELETE|TRACK) [NC]

RewriteRule ^(.*)$ – [F,L] RewriteCond %{QUERY_STRING} \.\.\/ [NC,OR]

RewriteCond %{QUERY_STRING} boot\.ini [NC,OR]

RewriteCond %{QUERY_STRING} tag\= [NC,OR]

RewriteCond %{QUERY_STRING} ftp\: [NC,OR]

RewriteCond %{QUERY_STRING} http\: [NC,OR]

RewriteCond %{QUERY_STRING} https\: [NC,OR]

RewriteCond %{QUERY_STRING} (\<|%3C).*script.*(\>|%3E) [NC,OR]
```

```
RewriteCond %{QUERY_STRING} mosConfig_[a-zA-Z_]{1,21}(=|%3D) [NC,OR]

RewriteCond %{QUERY_STRING} base64_encode.*\(.*\) [NC,OR]

RewriteCond %{QUERY_STRING} ^.*(\[|\]|\(|\)|<|>|ê|”|;|\?|\*|=$).* [NC,OR]

RewriteCond %{QUERY_STRING} ^.*(“|’|<|>|\|{||).* [NC,OR]

RewriteCond %{QUERY_STRING} ^.*(%24&x).* [NC,OR]

RewriteCond %{QUERY_STRING} ^.*(%0|%A|%B|%C|%D|%E|%F|127\.0).* [NC,OR]

RewriteCond %{QUERY_STRING} ^.*(globals|encode|localhost|loopback).* [NC,OR]

RewriteCond %{QUERY_STRING} ^.*(request|select|insert|union|declare).* [NC]

RewriteCond %{HTTP_COOKIE} !^.*wordpress_logged_in_.*$

RewriteRule ^(.*)$ – [F,L]

</IfModule>
```

Esto lo colocará un aficionado, pero un hacker profesional se encuentrara con otro agujero de seguridad para explotar. Afortunadamente la mayoría de los ataques son perpetrados por novatos o spammers, usando scripts como PHP o r57 Shell. La prevención de estos ataques reducirá en gran medida la cantidad de problemas que tiene que enfrentar.
<h3>Usuario Default</h3>

El de mayor agujero seguridad en todos los sistemas es que el usuario final. No importa lo complejo que sea una contraseña para esta cuenta. De hecho, cuanto más compleja sea la contraseña, peor es el riesgo para la seguridad, ya que las contraseñas muy complejas tienen que ser guardadas en algún lugar. Con Frecuencia, los usuarios guardan las contraseñas en

formato. Txt o. Doc en su computadora y que deja el sistema abierto a los ataques de phishing que utilizan archivos de virus como troyanos.

El único lugar seguro para almacenar una contraseña está dentro de su propia cabeza.

Sin embargo, incluso si sólo alguna vez guarde su contraseña en su propia memoria, usted todavía no está a salvo de los ataques de fuerza bruta. Un ataque de fuerza bruta simplemente tratará de "adivinar" la contraseña con los repetidos intentos de iniciar sesión Puede comenzar con 'aaaaaa', procediendo a 'aaaaab' y así sucesivamente hasta llegar a '000000 '.Este proceso no se limita a un único equipo, comúnmente cientos de máquinas pueden ejecutar este ataque a través de contraseñas potenciales que buscan acceso.

Una forma de manejar ataques de fuerza bruta es la instalación de un limitador de inicio de sesión que sólo permite un intento de inicio de sesión poco antes de bloquear el acceso de ese usuario durante una hora más o menos. Esto reduce el número de posibilidades de que el atacante tiene que entrar.
Hay varios plugins de WordPress que te pueden ayudar con esto: **Limit Login Attempts**, **Better WP Security** y **Login Security Solution**.

Por último, prestar atención a los nombres de usuario. El nombre de usuario por defecto para WordPress es 'Admin' y si lo deja tal como está reducirá a la mitad la cantidad de trabajo que el hacker tiene que hacer para acceder a su sitio. Si no ha cambiado su nombre de usuario mientras se instala WordPress puede hacerlo ahora. Sólo tiene que acceder a su cuenta, crear una nueva cuenta con el nombre de usuario que desea, darle permisos de administrador ya continuación, elimine la cuenta de administrador.

Acceso Directo

Otro de los problemas a nuestros sitios de WordPress se tener acceso directo a la página de inicio de sesión, lo que simplifica el proceso de hackear su sitio.

Mientras que asegurar sus contraseñas es el problema más urgente, un usuario malicioso no podrá hacer uso de todo lo que han robado, si no pueden encontrar la página de inicio de sesión. La solución más sencilla para esto es usar un plugin como **Hide Login** para ocultar la ubicación de la página de inicio de sesión.

Ciertos archivos de nuestra instalación de WordPress también se puede acceder si no está correctamente asegurado. Podemos aclarar estos cabos sueltos por la adición de reglas a nuestro archivo htaccess:

```
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

El prefijo por defecto

Debe quedar claro que mientras más información le damos nuestro aspirante a hacker, más fácil será para que tengan éxito. El prefijo por defecto en las tablas de la base de datos de WordPress es ‘wp_’.

¿Por qué querríamos darles eso?
Vamos a cambiar el prefijo a algo más difícil de adivinar, como “oijrr58_”, por ejemplo, hará el ingreso al hacker mucho más difícil, y aumentar las posibilidades de que su sitio se mantendrá seguro.
En nuevas instalaciones, esto es una obviedad, porque el script de instalación nos pide un prefijo.

Para la mayoría sitios tienen dos opciones, puede hacer el cambio manualmente (sólo intentar esto si usted tiene un montón de tiempo y está seguro de que sabes lo que estás haciendo) o usar un plugin como la Better WP Security que se hará cargo de ella para usted.

Demasiado tarde …

Nunca es demasiado tarde. Siempre se puede luchar contra los piratas informáticos, y evitar que usted se convierta en una víctima perpetua. Si no está seguro si su sitio ha sido infectado hay herramientas que le ayudaran. Sucuri SiteCheck por ejemplo explorará su sitio y si usted está infectado, le aconsejará sobre los pasos a seguir para corregir el problema (s).

Arreglos básicos

Aquí hay algunos pasos básicos a seguir:

1. Copia de seguridad del sitio y base de datos, hackeado o no, usted no quiere perder su contenido.
2. Haga copias de los artículos que no están en su base de datos, como imágenes.
3. Descargue la última versión de WordPress.
4. Asegúrese de que todos los plugins están al día, comprobar las versiones solucionar problemas conocidos.
5. Asegúrese de que las plantillas están al día, comprobar las versiones solucionar problemas conocidos.
6. Utilice un cliente FTP o cPanel para borrar todo el directorio de WordPress.
7. Sube los nuevos archivos que ha descargado.
8. Ejecutar la actualización de base de datos.

9. Cambiar su contraseña frecuentemente.
10. Por último, compruebe todos los post, para corregir cualquier daño que se ha hecho.

Luchar contra los scripts R57

r57 es un script PHP que permite a un atacante una amplia gama de capacidades, aunque el atacante tiene estas capacidades, estos no funcionará hasta que el shell se encuentre en nuestro servidor web, con lo cual podemos evitar que se ejecute este script con los siguientes comandos:

```
find /var/www/  -name “*”.php  -type f -print0  | xargs -0 grep r57 | uniq -c  | sort -u  | cut -d”:” -f1  | awk ‘{print “rm -rf ” $2}’ | uniq
```

Este comando buscará los archivos PHP se encuentra en la carpeta public_html/, entonces dentro de los archivos encontrados se buscará cualquier mención de r57 en el nombre del archivo y su contenido. A continuación, se elimina el archivo infectado (s).

```
find /var/www/ -name “*”.txt -type f -print0 | xargs -0 grep r57 | uniq -c | sort -u | cut -d”:” -f1 | awk ‘{print “rm -rf ” $2}’ | uniq
```

Este código hace lo mismo, a excepción de los archivos TXT en lugar de. Archivos php.

Tenga en cuenta que estos códigos son para Linux, no los tratan en Windows o MacOS y ser conscientes de que son potencialmente muy destructivos ya que va a eliminar los archivos sin pedir permiso.

Código oculto

Una causa importante de preocupación está en el codigo oculto, porque el código malicioso tiende a ser más difícil de encontrar dentro de los temas. Todo tipo de daño puede hacer este codigo como redirigir a los usuarios a otros sitios, para hundir su optimización SEO.

Un arma clave en la lucha contra este tipo de problema es Theme Authenticity Checker. Este plugin no sólo comprueba el código de las líneas sospechosas sino que detecta enlaces estáticos y código oculto como el código generado en base64 que es difícil de detectar a simple vista.

Si me engañas una vez, la culpa es tuya …

El hecho de que usted ha sido atrapado en el pasado, no significa que usted tiene que mantener inquieto.
Considere la posibilidad de tomar estos pasos para asegurar aún más su wordpress:

Sólo permita PHP cuando sea estrictamente necesario.
Asegúrese de que su servidor web no permite a los clientes para modificar el archivo .htacess.
Supervisar archivos a su sitio web con una aplicación como **ConfigServer eXploit Scanner**.