10 consejos para mejorar la seguridad de nuestro sitio web

La seguridad es uno de los factores más importantes para cualquier usuario a la hora de desarrollar un portal web, ya que no es plato de buen gusto darse cuenta de que nuestro sitio ha sido hackeado por un atacante. Esta situación suele producir muy mala imagen, ya que se pone en peligro la información del sitio y también a los visitantes que acceden a ella. Por muy seguros que estemos de que nuestro sitio cuenta con una programación muy robusta y segura, podemos llegar a la situación de que alguien encuentre algún agujero por donde lanzar su ataque, consiguiendo robar datos que tengamos ahí almacenados o bien utilizarlos como herramienta para infectar otros equipos.

Lo que pretendemos con este White Paper es mostraros algunas técnicas sencillas que podemos aplicar sobre nuestra web, para conseguir mejorar su seguridad y la de todos aquellos que la visiten.

1. Seguimiento de las aplicaciones instaladas

Hoy en día es muy habitual utilizar para el desarrollo de aplicaciones web CMS de código abierto como pueden ser WordPress o Joomla, por citar algunos. Estas herramientas facilitan mucho el trabajo de poner en marcha un sitio web nuevo pero también se pueden convertir en un importante problema de seguridad si no realizamos un seguimiento de las actualizaciones que van lanzando periódicamente.

En el caso de WordPress, es la propia herramienta quien te avisa cuando ha salido una nueva versión, invitándote a actualizarla. De forma habitual, el proceso de actualización suele ser algo muy sencillo, por lo que prácticamente cualquier persona podría ser capaz de realizar esta acción. Además, en la red nos podemos encontrar gran número de manuales que explican este tipo de actuaciones.

2. Recomendaciones de seguridad a la hora de programar una página web

Si en vez de apostar por una herramienta de código abierto preferimos una página web hecha a medida, es muy importante tener en cuenta una serie de recomendaciones de seguridad a la hora de programar. Entre las más importantes podemos destacar:

- Establecer filtros necesarios para determinar si cada campo de entrada de datos tiene el formato y la longitud esperada
- Eliminar caracteres especiales o escaparlos, eliminándolos si no son necesarios. Entre los caracteres más problemáticos nos podemos encontrar comillas simples, dobles, barras... eliminar estos caracteres puede ayudarnos a evitar ataques SQL Injection
- No permitir la carga de código HTML en aquellos campos que no sea necesario
- Bloquear la carga de contenido de páginas remotas

- En aquellas secciones donde sea necesario un usuario y una contraseña para acceder, hay que verificar que realmente el visitante se ha logueado y que sus datos están en la sesión

3. Archivos de configuración del servidor

Otro de los puntos importantes tiene que ver con los archivos de configuración del servidor web que hayamos contratado. Dependiendo del tipo, podemos encontrarnos el archivo .htaccess en servidores Apache, nginx.conf en servidores que tengan instalado Nginx o web.config en aquellos que utilizan Microsoft IIS. En todos estos archivos, se pueden incluir directivas que buscan mejorar la seguridad del sitio web.

Entre las cosas que sería interesante que configurásemos desde este tipo de archivos, podemos destacar:

- **Prevenir la exploración de directorios**. Con esto, estaremos evitando que los usuarios puedan acceder al contenido de cada directorio que forma parte del sitio web
- **Evitar imagen hotlinking**. No se trata en sí de una mejora de la seguridad, pero sí que estaremos evitando que otras personas puedan enlazar las imágenes que tengamos subidas en nuestro servidor en sus webs, lo que conllevaría a más consumo del ancho de banda contratado
- **Proteger los archivos sensibles**. Desde este tipo de ficheros de configuración, podemos proteger ciertos archivos y carpetas que sean importantes para el buen funcionamiento del sitio, por ejemplo los archivos de configuración. También podemos evitar la ejecución de código PHP en aquellos directorios destinados exclusivamente para almacenar imágenes o archivos CSS

Hay muchas otras cosas que se pueden hacer desde estos archivos de configuración, sólo es cuestión de investigar lo que se puede dependiendo del servidor que utilicemos.

4. Instalar SSL

En realidad, el uso de certificados de seguridad no mejoraría la seguridad de nuestro sitio, pero sí que ayudaría a mejorar la seguridad de la información que se mueve por ella, sobre todo en comercios electrónicos donde se suelen enviar datos sensibles de los usuarios.

Este tipo de certificados de seguridad, lo que hace es encriptar la información que se envía por la página web, impidiendo que alguien que intercepte el tráfico pueda descifrar los datos enviados, a no ser que también consigue la clave de encriptación.

5. Gestionar los permisos de los archivos

Otro de los aspectos al que todo el mundo debería dedicar algo de tiempo es a definir el tipo de permisos que puede tener un archivo. Con esto lo que conseguimos es definir quién puede hacer algo sobre él. Un fichero tiene disponible tres permisos y cada uno de ellos está representado con un valor:

- **Leer (4)**: Ver el contenido del archivo
- **Escribir (2)**: Modificar el contenido del archivo
- **Ejecutar (1)**: Ejecutar el archivo de programa o script

Si deseamos permitir varios permisos sólo tenemos que sumar los números. Por ejemplo, para indicar que puede leer y escribir, se establecería como permiso de usuario el valor 6.

También nos encontramos tres tipos de usuarios diferentes:

- **Propietario**: Por lo general hace referencia a quien lo ha creado, aunque esto se puede cambiar
- **Grupo**: A cada archivo se le asigna un grupo y cualquier usuario que forme parte de ese grupo obtendrá esos permisos
- **Público**: El resto de usuarios

Así tendríamos un valor con 3 cifras, puestas este orden: propietario-grupo-público. Por ejemplo, queremos que el propietario tenga acceso a leer y escribir (4+2=6), el grupo que tenga acceso de sólo lectura (4), y el público que no tenga acceso (0), el archivo de configuración de permisos debería tener el valor de 640.

6. Encriptar la información sensible

Hoy en día la mayoría de aplicaciones webs hacen uso de base de datos donde almacenar la información que se muestra en el sitio o bien para almacenar todos los datos que los usuarios envían por medio de los formularios que hay en el sitio. Hay ocasiones en las que parte de esa información es de gran importancia. En esos casos es buena idea encriptar los datos que se almacenan, para que en caso de que alguien entre a la base de datos de nuestro servidor no pueda leer la información, sino que únicamente verá letras y números sin sentido. Es el caso de las contraseñas o de los números de cuentas bancarias, información que siempre debería estar encriptada utilizando algún algoritmo creado para ello.

7. Realizar copias de seguridad periódicas

Por raro que pueda parecer no todo el mundo las realiza. En caso de sufrir algún tipo de ataque que haya infectado nuestro sitio, la forma más segura de acabar con él es recurriendo a alguna copia de seguridad que tengamos realizada de nuestro portal.

Estos backups también nos garantizan tener salvada nuestra información en caso de sufrir algún tipo de evento catastrófico. Eso sí, no hagáis la copia en vuestro propio equipo, sino en algún soporte externo que esté guardado en un lugar diferente a donde tengamos el servidor.

8. Buscar vulnerabilidades

Es muy importante realizar algún tipo de auditoría en nuestro sitio en busca de cualquier tipo de vulnerabilidades. Hay varias herramientas que se encargan de realizar de forma automática este tipo de análisis, como pueden ser Nikto o W3AF.

```
larry@sobremesa ~ $ nikto -h NOMBRE_DEL_HOST
- Nikto v2.1.4
---------------------------------------------------------------------------
+ Target IP:          XXX.XXX.XXX.XXX
+ Target Hostname:    NOMBRE_DEL_HOST
+ Target Port:        80
+ Start Time:         2014-05-07 21:14:16
---------------------------------------------------------------------------
+ Server: nginx
+ No CGI Directories found (use '-C all' to force check all possible di
+ robots.txt contains 1 entry which should be manually viewed.
+ ETag header found on server, fields: 0x52af7c7c 0x21
+ OSVDB-3093: /.htaccess: Contains authorization information
+ OSVDB-3092: /xmlrpc.php: xmlrpc.php was found.
+ /wp-content/plugins/akismet/readme.txt: The WordPress Akismet plugin
+ OSVDB-3092: /license.txt: License file found may identify site softwa
+ /wordpress/: A WordPress installation was found.
+ 6448 items checked: 41 error(s) and 7 item(s) reported on remote host
+ End Time:           2014-05-07 22:05:36 (3080 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Este tipo de herramientas, suelen lanzar gran cantidad de llamadas HTTP para tratar de obtener información que mostrar al usuario, indicando aquellas vulnerabilidades que hayan sido encontradas.

9. Configurar las cookies para que sean httponly y secure

Si configuramos las cookies como "secure" estaremos consiguiendo que sólo se intercambien entre el navegador y tu aplicación por medio del protocolo HTTPS. Por otro lado, al marcarlas como "httponly", evitaremos que haya scripts que puedan acceder a la información almacenadas en ellas, reduciendo las posibilidades de sufrir un ataque de tipo cross-site scripting.

10. Apostar por un proveedor de hosting de confianza

En el mercado nos podemos encontrar con cientos de empresas que dan servicios de alojamiento, pero no todos ellos ofrecen el mismo nivel de seguridad. Es muy importante que este proveedor de hosting tenga algún sistema de detección y prevención de intrusos, incluso que ofrezca barreras de seguridad para bloquear posibles ataques como puede ser el uso de firewall en sus sistemas.

Una buena práctica, consiste en bloquear los protocolos considerados de administración (Remote Desktop o Terminal Service, telnet, ssh, webmin, usermin y si es posible ftp) de tal forma que sólo se pueda acceder desde las direcciones IP que normalmente emplee el usuario para su gestión. De esta forma, aunque un atacante pudiera conseguir nuestras contraseñas, al no disponer de una IP autorizada, no podría conectarse.