# CONSIDERACIONES A TENER EN CUENTA PARA EL DESARROLLO DE SITIOS WEB SEGUROS

Díaz Ovalle, Oscar.
acumashente@gmail.com
Universidad Piloto de Colombia

***Abstract*** *— There is currently high demand in developing websites because computer technologies have made it easier for ordinary people access to different content, organizations seeing this phenomenon, have migrated many of its products and service to the web, so you can capture leads.*

*Because the web is extremely unsafe, they should be important considerations when developing web sites, or implement a solution that allows organizations to establish direct connection with their customers.*

*They should have basic security considerations and be able to shield (to put it that) the site to be developed.*

***Keywords —*** *Web sites, encryption, security, CMS, confidentiality, integrity, authenticity, hash, message transactions, Certificate of authenticity.*

***Resumen*** — Actualmente existe gran demanda en el desarrollo de sitios web, debido a que las tecnologías informáticas han facilitado a las personas del común el acceso a diferentes contenidos, las organizaciones al ver este fenómeno, han migrado gran parte de sus productos y servicio a la WEB, y así poder captar potenciales clientes.

Debido a que la web es extremadamente insegura, se deben tener consideraciones importantes a la hora de desarrollar sitios web, o implementar alguna solución que le permita a las organizaciones establecer conexión directa con sus clientes.

Se deben tener consideraciones básicas de seguridad y así poder blindar (por decirlo de alguna manera) el sitio que se va a desarrollar.

***Palabras Claves —*** Sitios Web, cifrado, seguridad, CMS, confidencialidad, integridad, autenticidad, hash, mensaje, transacciones, certificado de seguridad**.**

## I. Introducción

Prácticamente cada organización o persona en el mundo tiene representación virtual en la WEB, de tal manera que ante nuestros ojos, se ha materializado un mundo paralelo a la realidad.

Desde que Larry Page y Sergey Brin inventaron Google, hoy en día para las organizaciones y personas es común utilizar sitios web para hacer compras, pagos, o adquirir algún tipo de servicio.

Estas actividades crecen a pasos agigantados, obligando a las organizaciones a establecer mecanismos para tener mayor alcance, y así ofrecer sus productos y/o servicios, la cual consiste en generar su propia representación en la WEB, un sitio WEB.

[1]Para el momento en que se escribió este artículo, existen alrededor de 958,600,000 de páginas WEB en el mundo, y crece a cada segundo, de tal manera que para finalizar el día habrán alrededor de unos 86000 nuevos sitios web.

Con lo anterior, se puede evidenciar la importante demanda para el desarrollo de sitios web, de tal manera que se deben tener en cuenta el aseguramiento y la calidad en el desarrollo de los mismos.

El tema de seguridad es muy amplio, requiere de un estudio para detallar todas aquellas vulnerabilidades que se puedan presentar en la aplicación. Es importante familiarizarse con las posibilidades de seguridad ofrecidas por las herramientas de desarrollo que se van a utilizar, el lenguaje de programación, los servidores web, etc. Con el fin de contrarrestar amenazas. A continuación, se darán conocer algunas consideraciones a tener en cuenta a la hora de desarrollar un sitio web, ya sea mediante un CMS (Gestor de Contenidos), o desarrollo a la medida.

## II. Consideraciones de Seguridad para Implementar un CMS

### *A. ACTUALIZAR PERIÓDICAMENTE EL CMS.*

[2]Es importante mantener actualizado el CMS, y sobre todo si éste es de código abierto. Estas actualizaciones en la mayoría de los casos, sirven para resolver algún tipo de vulnerabilidad.

### *B. USAR CONTRASEÑAS FUERTES.*

[3]Las contraseñas deben ser lo más robustas posibles, por lo general debe incluir números, letras y caracteres especiales y al menos con una longitud de 10 caracteres.

### *C. USAR UN CORREO ELECTRÓNICO ADMINISTRATIVO SEGURO.*

Es importante tener en cuenta, de que el correo electrónico de administración que se utilizará para iniciar sesión, debe ser completamente diferente a cualquier otro correo electrónico que se mencione la página de contacto. Esto evitara problemas por suplantación

### *D. CAMBIAR EL PREFIJO DE LAS TABLAS DE LA BASE DE DATOS.*

Por defecto, cada CMS sugiere un prefijo, por seguridad, este prefijo se debe cambiar.

### *E. CREAR UNA CONTRASEÑA PARA LA BASE DE DATOS.*

En algunas instalaciones, por defecto la base de datos queda sin contraseña. Esto representa una gran vulnerabilidad, por lo tanto se debe asegurar de crear una contraseña para la base de

datos.

*F. ELIMINAR LA CARPETA DE INSTALACIÓN.*

Todos los CMS traen una carpeta de instalación, en ella viene configurado todo el setup del proceso de instalación, de tal manera que si esta carpeta no es eliminada, se corre el riesgo de que algún pirata informático repita el proceso de instalación generando daños en la información.

*G. USAR EXTENSIONES QUE OFREZCAN UNA CAPA ADICIONAL DE SEGURIDAD.*

De ser posible, se aconseja instalar extensiones o plugins, que optimicen el rendimiento del CMS y que por supuesto añadan seguridad adicional al sitio. Existen gran cantidad de aplicaciones modulares, o componentes que se adaptan a las necesidades del sitio.

*H. MANTENERSE ACTUALIZADO.*

El web master, debe mantenerse actualizado, leyendo contenidos relacionados a la tecnología del CMS implementado. Es importante indagar en foros, blogs, redes sociales, etc. Sobre nuevos tipos de vulnerabilidades que puedan encontrarse.

## III. Consideraciones para Desarrollos web a la Medida.

*A. UTILIZAR LA VERSIÓN MÁS ESTABLE DEL LENGUAJE DE PROGRAMACIÓN.*

De ser posible, y si no es restrictivo, es recomendable utilizar la última versión del lenguaje de programación estable. Esto permite mitigar y prevenir bugs y vulnerabilidades de versiones anteriores.

*B. USAR ALGÚN PATRÓN DE DISEÑO.*

[1]Los patrones de diseño son una solución a un problema en un contexto particular recurrente (lo que hace la solución relevante a otras situaciones) enseña (permite entender cómo adaptarlo a la variante particular del problema donde se quiere aplicar) tiene un nombre para referirse al patrón. Los patrones facilitan la reutilización de diseños y arquitecturas software que han tenido éxito.

Es recomendable aplicar en el desarrollo algún tipo de patrón de diseño, ya que facilita el mantenimiento futuro, y dado su arquitectura, agrega un nivel de seguridad.

Se recomienda utilizar algún framework de desarrollo. Éstos permiten estandarizar el código, facilitando el mantenimiento en el futuro. Existen muchos frameworks en la web de manera gratuita.

*D. IMPLEMENTAR CERTIFICADOS DE SEGURIDAD SSL.*

Actualmente, para todo sitio WEB se recomienda implementar un SSL. Este servicio permite autenticar el sitio WEB ante el público y además de cifra la información que transita en el sitio.

*E. ASEGURAR LA DISPONIBILIDAD.*

Se sugiere separar el ambiente de aplicación y la base de datos. Debido a costos, es posible tomar varias alternativas para asegurar de algún modo la disponibilidad del sitio desarrollado, tales como generar balanceadores de carga, servidores espejo, etc. En cuanto a bases de datos, también se aconseja tener un servidor dedicado, si es posible utilizar servidores espejo.

*F. NO MOSTRAR ERRORES DE EJECUCIÓN A USUARIOS.*

[5]Es inevitable impedir que nuestras aplicaciones fallen. Por esta razón, es importante establecer todos aquellos puntos de fallo y reportarlos, pero teniendo en cuenta, de que estos mensajes jamás pueden llegar a usuarios no autorizados.

Estos errores, preferiblemente deben ser guardados en archivos de logs. Mostrar los errores de ejecución a usuarios del común, genera una brecha de inseguridad, debido a que probablemente algún usuario con el conocimiento suficiente pueda entender de qué se trata el error y así poder explotarlo.

Si es necesario mostrar errores al usuario, tenga en cuenta lo siguiente:

- No muestre información que pueda resultar útil a usuarios malintencionados.
- No mostrar información detallada, esta debe ser exclusivamente para usuarios con el rol apropiado para tratar la excepción.

*G. IDENTIFICAR LOS FORMULARIOS Y ASEGURARLOS.*

Este ítem, es el más importante a tener en cuenta a la hora de desarrollar un sitio web. Los formularios son entradas al sistema, ya sea para realizar algún registro o generar consultas. En este punto, es importante tener en cuenta a nivel de seguridad, dos tipos de ataques: ataques humanos y ataques automáticos o robots.

Los ataques humanos, por lo general son aquellos que basan su conocimiento para hallar huecos de seguridad, ya sea para acceder al sistema y o explotar vulnerabilidades.

Los ataques automatizados o robots, son muy peligrosos, ya que por su eficiencia en su automatización, pueden realizar estragos en un sitio de diferentes maneras.

Teniendo en cuenta lo anterior, se recomienda lo siguiente.

*1) Generar un token de seguridad:* Por cada vez que se ingrese al formulario. Este token, debe ser generado aleatoriamente y además aplicar algún algoritmo de cifrado no reversible como SHA o MD5, para validar su autenticidad a la hora de recibir los datos para ser procesados.

*2) Implementar un Captcha:* Para verificar si el que envía el formulario es humano.

*3) Evitar usar el método GET:* Preferiblemente para el envió de formularios, utilizar POST es más seguro, ya que las variables GET son visibles y de alguna manera son más fáciles de manipular que las variables utilizadas mediante el método POST.

*4) Verificación de valores de formularios:* Cada valor que es enviado mediante el formulario, debe ser inspeccionado.

Este proceso sirve para verificar la información y evitar posible inyecciones de código y o Cross-Site Scripting. Existen muchas técnicas para evitar código malicioso, tales como activar el modsecurity en el servidor de aplicaciones, limpiar los valores recibidos mediante un algoritmo que detecte el código malicioso, etc.

*5) Archivos adjuntos en formularios:* Si el formulario debe llevar archivos adjuntos, es importante validar el tipo de archivo que se va a subir. Se debe restringir, por ejemplo, si es necesario únicamente subir archivos tipo doc y pdf, de tal manera que no se pueda subir otro tipo de archivo, esto para evitar que algún atacante suba algún script que pueda invocar remotamente. El paso a seguir es validar que el archivo sea completamente legible, posteriormente cifrar el contenido del archivo mediante un algoritmo simétrico para asegurar la confidencialidad y finalmente guardar el archivo, ya sea en la base de datos, o en algún directorio.

*6) Seguridad en base de datos: [5]Se deben verificar cada una de las entradas que provienen de los formularios.*

Es importante establecer criterios de aceptación, tales como palabras reservadas, etiquetas html, con el fin de evitar código malintencionado.

Si requiere guardar etiquetas html, scritps javascript, xml, etc., es recomendable codificar estos contenidos, de tal manera que puedan ser vistos pero no ejecutados a voluntad por los usuarios.

Los accesos a base de datos, deben estar limitados en sus funciones. Es importante establecer qué roles son necesarios para el usuario que será utilizado en la aplicación.

En cuanto a las consultas SQL, es importante no generarlas de forma concatenada con información que provenga del usuario, se puede optar por crear consultas parametrizadas, usando la entrada del usuario para determinar los valores a parametrizary finalmente generar la consulta. En la web, existen muchísimos frameworks que facilitan este proceso.

Se recomienda realizar copias de seguridad o back ups periódicamente.

*7) Utilizar cookies seguras:* [5]Las cookies son muy útiles a la hora de guardar datos temporales del usuario, sin embargo son muy vulnerables a la suplantación, ya que se alojan en el explorador del equipo del cliente. Se debe tener en cuenta lo siguiente:

- No guardar información crítica ni siquiera de forma temporal, como por ejemplo contraseñas. Por norma, se recomienda no guardar nada en una cookie, que de tal manera de que si llegara a ocurrir una suplantación, ésta información no represente riesgo alguno para la aplicación.
- Se debe establecer perdidos de vencimiento cortos para las cookies.
- Se recomienda cifrar la información que guarde en las cookies.

## IV. Consideraciones para Desarrollo Basados en Comercio Electrónico

[4][6]Hoy en día se realizan procesos en donde se envía información que para organizaciones o personas se considera privada o restringida para el resto de los usuarios, estos procesos son:

- Envío de orden de pedido al comerciante, junto con información del método de pago.
- Solicitud de acceso del comerciante a la entidad financiera del comprador.
- Confirmación de orden por parte del comerciante al comprador.
- Solicitud de reembolso por parte del comerciante a la institución financiera del comprador.

Dentro de los límites establecidos en los procesos de comercio electrónico son:

- Proporcionar la correcta autenticación entre los participantes del comercio.
- Garantizar la confidencialidad de la información sensible (número de tarjeta, fecha de caducidad, etc).
- Preservar la integridad de la información proporcionada por el comprador que contiene tanto la orden de pedido como las instrucciones de pago.

Cada uno de los aspectos anteriormente descritos son importantes en una red de transmisión de datos, pero son vulnerables. Esta característica aumenta la preocupación cuando se relaciona con internet u otra red pública, por su ubicación y desconocimiento entre las partes que intervienen en el comercio electrónico.

A continuación, nombraré algunas consideraciones a tener en cuenta para el aseguramiento entre los procesos de comercio electrónico.

*1) La confidencialidad:* Se debe cifrar el mensaje, que contiene datos como: datos de la tarjeta, fecha caducidad, y proceso de pago. Con el fin de evitar que esta información pueda ser accedida por alguien no deseado.

*2) La Integridad:* Se debe garantizar que la información suministrada entre los actores del comercio electrónico, sea la misma siempre. Se recomienda entonces, implementar algún algoritmo de cifrado no reversible, para establecer una firma digital y así, asegurar que la información llegue a su destino sin modificaciones.

*3) Autenticación del comprador:* Se debe verificar la autenticidad del comprador, como usuario legítimo de la cuenta bancaria o tarjeta. Esto se realiza mediante la emisión de certificados y la generación de firmas digitales.

*4) Autenticación del comerciante:* Al igual que la autenticación del comprador, autenticar al comerciante garantiza que se mantiene la relación con la entidad financiera que aceptará el pago. Esto se realiza mediante la emisión de certificados y la generación de firmas digitales.

*5) Confidencialidad del mensaje:* Para la confidencialidad de mensajes, se debe utilizar llaves simétricas para el cifrado del mismo.

Las claves se deben generar aleatoriamente, y se debe cifrar con la llave pública del destinatario. La unión entre la llave cifrada y el mensaje cifrado se llama sobre electrónico.

El destinatario, al recibir el mensaje, con la clave privada descifra la clave simétrica, la cual permitirá descifrar el contenido del mensaje. La programación implementada en estos procesos, debe asegurar la integridad de las llaves, y que las mismas no sean interferidas del contenido del mensaje.

*6) Implementación de Firmas Electrónicas para la integridad y autenticidad de los mensajes:* Las firmas digitales garantizan la integridad del mensaje, es decir, que el mensaje no sea modificado de manera fraudulenta en su tránsito, también permite autenticar, que permite validar si los actores que intervienen son los que dicen ser. Consiste en la generación de un par de llaves entre las partes, una pública y otra privada. De tal manera que, para cada actor que interviene en el sistema, tiene un par de llaves (la llave privada, generada por él, y la pública, que es brindada por su contraparte) con la llave privada se cifra el mensaje, y el destinatario descifra el mensaje con la llave pública, esto permite autenticar el origen del mensaje.

Para garantizar el contenido del mensaje, es importante implementar algún algoritmo de cifrado no reversible, como por ejemplo MD5, que produce un valor has único para el mensaje cifrado, el hash se cifra con la llave privada y se le envía al destinatario junto con el mensaje original, esto para que el destinatario aplique de la misma manera el algoritmo de cifrado no reversible, y así comprobar con el hash de origen que el mensaje no ha sido modificado.

*7) Certificados de Autenticidad:* Estos certificados, son emitidos por entidades de confianza o Autoridades Certificadoras o CA, la cual contiene la clave pública de la persona o entidad, junto con información propia, en donde la Autoridad Certificadora pone su estampa o firma digital. Como la clave pública de la CA es pública.

*8) Transacciones Basadas en SET:* Visa y MasterCard han hecho grandes esfuerzos para generar buenas prácticas en cuanto a comercio electrónico. Básicamente SET ofrece un protocolo que emula de forma electrónica, mediante el uso de certificados y firmas digitales, y la realización de pagos electrónicos, mediante la utilización de tarjetas de crédito.

## V. Detección y Pruebas de Vulnerabilidad

Básicamente consiste en la búsqueda y detección de vulnerabilidades, tratando de hacerlo como un intruso probablemente lo haría pero de forma ética.

Existen muchas herramientas en el mercado, especializadas para realizar este proceso. También las hay gratuitas, pero el uso de estas herramientas requiere de conocimiento profundo sobre la misma. Entiéndase que esta actividad es conocida como Hacking Ético.

Actualmente existe soluciones, bastante robustas especializadas en seguridad informática, y además gratuita en la gran mayoría de sus componentes. Estas herramientas, son distribuciones GNU/Linux, estas son: BackTrack y Kali, ambos son muy parecidos.

Se debe tener en cuenta, que la principal herramienta ha de ser nuestro conocimiento. Debemos aprender a pensar como lo haría una atacante, con el objetivo de detectar vulnerabilidades y así poder tomar acciones correctivas antes de que se presente algún incidente.

A continuación, mencionaré algunos componentes existentes tanto en Kali, como en BackTrack, y de qué manera usarlos.

*1) Nmap:* [7][8]Sirve para escanear redes, que nos permite identificar servicios activos en la red, tales como puertos, identificación de equipos activos, si existe firewall, sistemas operativos ect. Nmap es una herramienta que tiene mucha flexibilidad a la hora de hacer un ataque, se pueden establecer varios objetivos y una serie de parámetros para realizar el ataque.

Originalmente Nmap está basado en línea de comandos, existen algunas versiones con interfaz gráfica, pero se puede explotar mejor desde línea de comandos.

Nmap tiene la siguiente estructura:

>nmap ***,***,***,***

En donde ***,***,***,*** es un objetivo cualquiera (también puede incluir un dominio), y nmap es el comando. Al ejecutar este comando, realiza un escaneo sencillo mediante ping, ver Fig. 1, y así revisar los mil puertos más comunes.

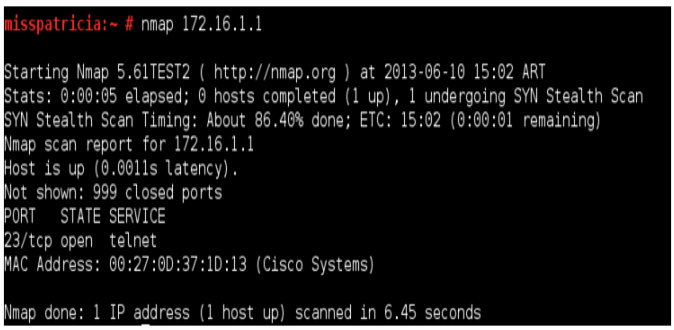


Fig. 1 Ejemplo de escaneo de ip utilizando el comando nmap en Kali Linux

Para explotar mejor esta herramienta, aquí se darán a conocer los parámetros disponibles para utilizar en nmap:

- [-iL] indicar una lista de equipos o redes a escanear. > nmap –iL hosts.txt
- [-sP] solo escanea con un ping. Es una buena forma de ver cuántas direcciones IP se pueden escanear. Una vez que se tienen enlistadas, se podrá ir solo con las que están vivas.
- [-P0] es la forma de omitir el ping e ir directo al escaneo de puertos. Muchos sistemas no responden el ping como método de seguridad, por lo que, escanearlos de todas formas, también puede ser útil en entornos más reales (no es necesario para los entornos de aprendizaje inicial).

- [-p] lista los puertos que se desean escanear. > nmap –iL hosts.txt –p 22,25,80,445
- [-sV] intenta determinar la versión del servicio en el objetivo.
- [-O] informa el sistema operativo en el objetivo.

*2) Nessus:* [10]Es una herramienta muy versátil, gratuita únicamente si no se utiliza en modo corporativo. Nessus es muy intuitivo, permite configurar escenarios de ataque ya que tiene una gran base de datos de vulnerabilidades, de tal manera de que si ejecutamos Nessus sin parametrizar, probara todas las vulnerabilidades existentes en la base de datos, ver Fig. 2.

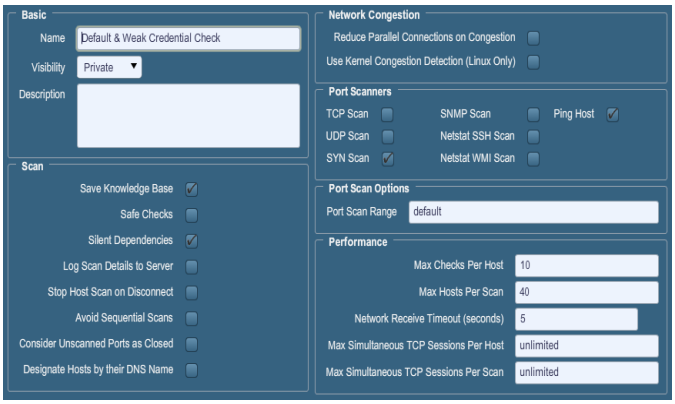


Fig.2 Sección de configuraciones generales de exploración de Nessus

Baicamente, el funcionamiento de Nessus es similar al de Nmap, se deben establecer los objetivos, y por supuesto parametrizar el escenario de ataque, esto permite limitar el alcance de Nessus en cuanto sus escaneos.

*3) Metasploit Framework:* [11]Una vez descubiertas las vulnerabilidades con cualquiera de las herramientas nombradas en los puntos anteriores, Metasploit Framework permite explotar dichas vulnerabilidades para conocer cuál sería el daño potencial de dicha vulnerabilidad. Metasploit, tiene una gran base de datos de vulnerabilidades con su respectivo programa especializado para explotar la vulnerabilidad.

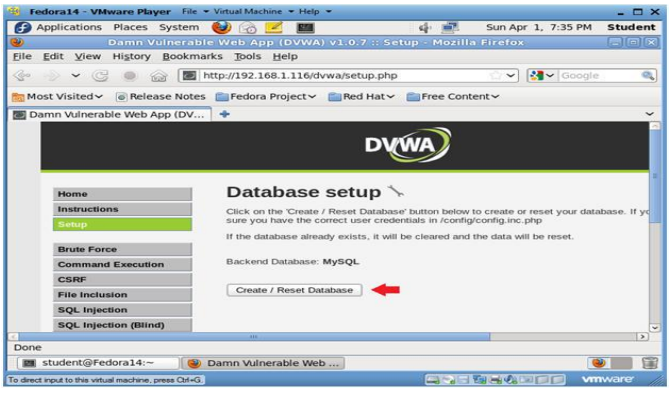


Fig. 3 Damn Vulnerable Web Application (computersecuritystudent.com)

*4) DVL – DVWA:* [9]Es un entorno de pruebas, dedicado exclusivamente para testear vulnerabilidades. Es un escenario de investigación con la ventaja de que es un sitio en donde jamás se podrá hacer daño a nadie si probamos las herramientas descritas en los puntos anteriores. Existe una versión instalable, y otra existente en la web, ver Fig. 3.

*5) Kali Linux (Backtrack):* Kali y Backtrack, son sistemas opertivos GNU/Linux, especializados en seguridad informática. Las herramientas antes descritas (Nmap, Metasploit y Nessus) vine incorporadas en estos sistemas operativos junto con muchas otras herramientas.

Las herramientas vienen presentadas en categorías, la cual permite experimentar de manera ordenada cada una de ellas, aquí una breve descricion de las categorías existentes tanto en Kali como en Backtrack:

- Information gathering: Permite recolectar datos del objetivo, especialmente herramientas DNSHerramientas, Nmap está en esta categoría.
- Aplicaciones web: Sirve para realizar análisis de sitios web. Se recomienda Nikto y w3af para encontrar vulnerabilidades en los sitios.
- Ataques a contraseñas: Se prueban ataques de fuerza fruta o diccionario.
- Ataques inalámbricos: Permite espiar la red a la que se está conectado.
- Herramientas de explotación: Permite explotar vulnerabilidades. Metasploit Framework esta en esta sección.
- Sniffing/Spoofing: Permite analizar el trafico de una red. Wireshark y Ettercap son las herramientas más recomendables.
- Ingeniería inversa: Por medio de un proceso de ingeniería inversa ,Ollydbg es uno de los mejores debuggers que ayuda a comprender el funcionamiento de una archivo de sistema mediante ingeniería inversa.
- Forense: Es una serie de herramientas, dedicadas a la disciplina forence, para analizar el estado de un sistema en determinado incidente.

## VI. Conclusiones

Es importante tener en cuenta que es imposible alcanzar seguridad absoluta, pero se pueden implementar buenas prácticas orientadas a la seguridad a la hora de desarrollar.

Se recomienda que la seguridad sea parte del desarrollo, y que no se considere como un asunto que se aplique al final del proceso del desarrollo, de tal manera, que la seguridad sea homogenizada en el proceso de desarrollo.

Siempre se debe considerar fortalecer la confidencialidad, autenticidad y la integridad de la información, se deben aplicar, según sea el caso, el cifrado de la información y la generación de llaves para garantizar integridad y autenticación de origen. De ser posible, se recomienda implementar un CA para aumentar la confianza al sitio web.

## REFERENCIAS BIBLIOGRÁFICAS


[1] Internet Live Stats-Internet Usage & Social Media Statistics[Online]. Disponible: http://www.internetlivestats.com/

[2] La importancia de las actualizaciones generales desoftware y los parches [Online]. Disponible:http://mx.norton.com/importancia-de-las-actualizaciones/article

[3] Recomendaciones de Seguridad Informática para el Internauta[Online]. Disponible: http://cms.ual.es/UAL/universidad/serviciosgenerales/stic/servicios/recomendaciones/contrase%C3%B1as/index.htm

[4] Mercadeo en Internet, negocios del siglo xxi[Online]. Disponible: http://www.icesi.edu.co/blogs_estudiantes/sibunmarketing/

[5] Microsoft, Procedimientos de seguridad básicos para aplicaciones Web [Online]. Disponible: https://msdn.microsoft.com/es-es/library/zdh19h94(v=vs.100).aspx#cpconbestsecuritypracticesforwebapplicationsanchor1

[6] Criptografía y los sistemas de seguridad en las páginas web [Online]. Disponible: https://sites.google.com/site/abigailalcocer/Home/paypal/check-list/criptografia

[7] Nmap Security Scanner [Online]. Disponible: http://nmap.org/

[8] Seguridad de la información [Online]. Disponible: http://revista.seguridad.unam.mx/numero-18/pruebas-de-penetraci%C3%B3n-para-principiantes-5-herramientas-para-empezar

[9] Damn Vulnerable Web Application [Online]. Disponible: http://www.dvwa.co.uk/

[10] Nessus [Online]. Disponible: http://www.tenable.com/products/nessus/

[11] Metasploit [Online]. Disponible: http://www.metasploit.com/