Uso de Pentesting cimentado en ASVS 3.0.1 como alternativa de estandarización y protección de datos en App. Web a Universidades

Ing. Abel Isaac Leon Galeana[1], MC. Francisco Javier Gutiérrez Mata[2], Dr. Eduardo de la Cruz Gámez[3], MTI. Eloy Cadena Mendoza[4], Dr. Félix Álvarez Paliza[5], Ing. Alejandro Hernández López[6]

***Resumen*— Considerando como campo de estudio una de las principales universidades de Guerrero, se aborda una metodología de estandarización de App. Web, la cual consiste en la prevención del robo y manipulación de la información, basado en el cumplimiento del Estándar de Verificación de Seguridad en Aplicaciones 3.0.1 de Open Web Application Security Project (OWASP) usando su guía de pruebas V4.0, con el fin de obtener como resultado un panorama amplio de los escenarios que pudieran suscitarse durante un ataque externo (intrusión). Por consiguiente, es necesario descubrir, corregir y documentar la intrusión durante las pruebas realizadas.**

***Palabras clave*-- Seguridad informática, Pentesting, hacking ético, vulnerabilidades en universidades.**

Introducción

El alcance de las tecnologías de la información y comunicación (TIC) ha tenido un crecimiento desmesurado aportando con grandes contribuciones a la rama de la informática y generando una globalización del tránsito de datos entre los usuarios por medio de la red global mejor conocida como Internet, esta herramienta en su uso cotidiano puede representar un riego para usuarios inexpertos, debido a la existencia de vulnerabilidades en los sistemas que resguardan la integridad de la información (Ramires Castro, 2012), la explotación de estas brechas en muchas de las ocasiones repercute en la credibilidad y finanzas de cualquier empresa.

Se considera necesario entender que, sin importar el avance tecnológico encargado de proteger la información como puede ser el caso del hardware o software especializado en la prevención y solución de la problemática; habiendo antivirus o firewalls dedicados por mencionar algunos que se consideran de confianza por su efectividad, no es conveniente establecerse en una zona de conformidad, de lo contrario, el exceso de confianza podría generar problemáticas al no contemplar posibles brechas en el sistema, dando libre acceso a usuarios no autorizados.

Por este motivo el plantearse una metodología de Pentesting o auditoria informática como medio de prevención y protección a aplicaciones Web enlazadas a una BD, conlleva un esfuerzo de unificación de técnicas entre lo que hoy en día se denomina Hacking ético y una estandarización a las TIC relacionadas a la App. que se desea proteger. Para los inicios de toda auditoria se recomienda fijar una planeación según las necesidades de la empresa a auditar con la finalidad de evitar pérdida de tiempo debido a que muchas de las practicas con relación al tema absorben el recurso ya mencionado del personal especializado (en caso de ser una auditoria interna), así como de la infraestructura; las ventajas reflejadas al realizar estas actividades se enumeran a continuación.

- Lograr una estandarización desde la creación, reforzando las pruebas en busca de vulnerabilidades desde el tiempo de vida de la aplicación.
- En caso de no haber creado la aplicación estas pruebas determinan los niveles de seguridad existentes, cuantificándolas por prioridad de solución.
- Reforzar las vulnerabilidades encontradas para evitar que ocurra un incidente relacionado a su explotación.
- Establecer planes de contingencia a posibles intrusiones al sistema, aminorando lo menor posible las perdidas y daños a las TIC.

[1] Ing. Abel Isaac León Galeana es estudiante de Maestría en Sistemas Computacionales en un programa PNPC en el Instituto Tecnológico de Acapulco, abelleongaleana@outlook.es (**autor corresponsal**).
[2] Mc. Francisco Javier Gutiérrez Mata es docente de ingeniería en sistemas computacionales, Maestría y Jefe del centro de cómputo del Instituto Tecnológico de Acapulco, fcomata84@hotmail.com.
[3] Dr. Eduardo de la Cruz Gámez es docente de ingeniería en sistemas computacionales y Maestría del Instituto tecnológico de Acapulco, gamezeduardo@yahoo.com.
[4] MTI. Eloy Cadena Mendoza es docente de ingeniería en sistemas computacionales y Maestría del Instituto Tecnológico de Acapulco, eloy_cadena@yahoo.com.
[5] Dr. Félix Álvarez Paliza es docente en la UCLV Marta Abreu en cuba, fapaliza@uclv.edu.cu.
[6] Ing. Alejandro Hernández López es Coordinador de Servicios de Computo del Instituto Tecnológico de Acapulco alejandrohl@outlook.com.

Estas ventajas sugieren que es ampliamente recomendable la ejecución de los test o auditorias de seguridad, sin embargo, estas deben estar siempre respaldadas por una metodología y estándar las cuales contengan una cimentación y actualización de expertos relacionados al tema, en caso contrario los resultados obtenidos en todas las pruebas no pueden considerarse fiables al carecer de sustento y por consiguiente pueden ser desechadas.

Atreves del estudio con base a un análisis de pruebas, se plantea una metodología a seguir para el blindaje de las aplicaciones en entorno Web por medio de pruebas de explotación, con el fin de proteger y salvaguardar los activos contenidos, de esta forma garantizando la prevalencia de la empresa, con el fin de demostrar la eficiencia de las prácticas como auxiliares para la toma de decisiones anticipadas ante la presencia de vulnerabilidades en las aplicaciones Web.

Descripción del método

Ámbitos generales de las amenazas a la seguridad informática.

Debido a que no es posible determinar de manera concreta, cuando y como se puede presentar un ataque a las TIC y el robo de la información, la cual representa uno de los principales rubros de toda empresa, es necesario pensar que se está expuesto a este tipo de actividades ilícitas en todo momento, la protección a las tecnologías Web de las entidades debe ser consideradas de alta prioridad, dela misma forma la protección a las BD enlazadas a App. Web, que en su mayoría se encuentra montadas en gestores comerciales como lo son MySQL, Access, PostgreSQL entre otras (Villalobos Murillo, 2012), esta ideología queda sustentada debido a que son amenazas con un tiempo prolongado entre los estudios elaborados por organizaciones especializadas en el tema como lo son *ESET, Norton y CCN-CERT* por mencionar algunas, sin embargo a pesar de los esfuerzos por solucionar estas problematicas, los usuarios mal intencionados, siempre hayan la manera de encontrar o crear brechas en los sistemas de seguridad. Esto es visible en los estudios realizados por Open Web Application Security Project por sus siglas OWASP los cuales se muestran en la tabla 1. Esta fundación la cual no tiene ánimos de lucro a liderado desde el 2001 la promulgación de la seguridad del Software en general y las aplicaciones Web, en la actualidad sus trabajos y publicaciones, así como los manuales proporcionados por dicha fundación son identificados como los de mayor valor en relación a las auditorias de seguridad enfocadas a App. Web (OWASP, 2017).

	2007	2010	2013	2017
A-1	Cross Site Scripting (XSS)	Injection	Injection	Injection
A-2	Injection Flaws	Cross-Site Scripting (XSS)	Broken Authentication and Session Management	Broken Authentication and Session Management
A-3	Malicious File Execution	Broken Authentication and Session Management	Cross-Site Scripting (XSS)	Cross-Site Scripting (XSS)
A-4	Insecure Direct Object Reference	Insecure Direct Object References	Insecure Direct Object References	Broken Access Control
A-5	Cross Site Request Forgery (CSRF)	Cross-Site Request Forgery (CSRF)	Security Misconfiguration	Security Misconfiguration
A-6	Information Leakage and Improper Error Handling	Security Misconfiguration	Sensitive Data Exposure	Sensitive Data Exposure
A-7	Broken Authentication and Session Management	Insecure Cryptographic Storage	Missing Function Level Access Control	Insufficient Attack Protection
A-8	Insecure Cryptographic Storage	Failure to Restrict URL Access	Cross-Site Request Forgery (CSRF)	Cross-Site Request Forgery (CSRF)
A-9	Insecure Communications	Insufficient Transport Layer Protection	Using Known Vulnerable Components	Using Components with Known Vulnerabilities
A-10	Failure to Restrict URL Access	Unvalidated Redirects and Forwards	Unvalidated Redirects and Forwards	Inderprotected APIs

Tabla 1.- Tabla comparativa de las principales vulnerabilidades publicadas por OWASP entre los años 2007 a 2017 (OWASP, 2010) (OWASP, 2013) (OWASP, 2017).

Como se muestra en la tabla 1 la evolución de las vulnerabilidades mostrada por OWASP en su informe de Top10 de las principales vulnerabilidades a las App. Web ha estado en un cambio constante desde el 2007 hasta el 2017, en estos informes enumeran las debilidades por orden de prioridad, tomando como estudio las empresas en un

tiempo de por lo general tres años entre un informe y otro, como se observa las vulnerabilidades han ido cambiando sustituyendo anteriores o subiendo en el nivel de explotación y en otros casos se han ido sustituyendo por nuevas.

Tomando como base este estudio, es de carácter prioritario el crear un plan de acción para el fortalecimiento de las App. de este modo se determina que si bien es posible usar una ISO para la normalización no es despreciable usar un estándar Open Source para verificar el funcionamiento de los módulos y fases que forman las App. un ejemplo de esto es el Estándar de Verificación de Seguridad en Aplicaciones en su versión 3.0.1 creada por OWASP al ser un estándar creado netamente para el desarrollo de App. de entorno Web, el trabajo por módulos incrementa su eficiencia en cuestión de seguridad y prevención de errores de código. En el caso de trabajar con una App. que no fue desarrollada por la empresa permite el chequeo de la seguridad de estas, al combinar el estándar con el uso de la Guía de pruebas versión 4.0.

Factores de decisión para la realización de un Pentesting.

Hoy en día es destacable que las App. Web son el portal de relación entre empresa, usuario y colaboradores según el rubro que se maneje de igual forma las App. enlazadas a un servidor de BD representan un riesgo al no contar con una certeza de los niveles de seguridad, por esto muchas de las empresas empiezan a girar hacia este tema, ya no es solo un tema de interés para empresas cuyo rubro sea de carácter, bancario, gubernamental, etc., las universidades también cuentan con información importante que debe ser protegida y se debe tener una auditoria constante (Rando, González, Aparicio, Martín, & Alonso Cebián, 2016)

Las organizaciones al emprender auditorias informáticas internas en busca de vulnerabilidades pretenden cuantificar las brechas y establecer posibles escenarios de ataque para crear planes de contingencia; dependiendo las necesidades que la empresa desee cubrir el estudio y/o método de acción puede variar, por eso, es necesario el instaurar metodologías de seguimiento y determinar los límites a alcanzar para el final del proyecto de Pentesting.

Selección de un estándar de seguimiento y normalización.

El establecer o seguir una estandarización permite garantizar la funcionalidad y resultados satisfactorios, el Estándar de Verificación de Seguridad en Aplicaciones 3.0.1, por sus siglas en inglés *ASVS* el cual en su contenido establece un marco de referencia para los requisitos de seguridad, los requisitos de funcionalidad y no funcionalidad del diseño, desarrollo y testeo de App. Web. El estándar proporciona una visión general, pero a su vez detallada de las funciones que debe desempeñar toda App. Web, en el caso de estudio al no encontrarse en la etapa de diseño ni implementación de la App., es necesario verificar que esta cumple con los puntos descritos en el estándar anteriormente mencionado, como parte de una estrategia de seguridad, entre los requisitos básicos de verificación más competentes que debe cumplir toda App. Web se encuentran los mostrados en la *tabla 2.*

V1.- Arquitectura, diseño y monitoreo de amenazas.	Requisito relacionado a la funcionabilidad de cada una de las partes de la App. Web, en esta fase se revisa que todas las partes de la App. Cumplen con su función establecida y de acuerdo a las funciones del negocio y su seguridad para ello es necesario establecer una investigación previa del negocio.
V2.- Autenticación.	Requisito que consta en verificar lo que es verdadero, establecer pruebas a las claves de usuarios es uno de los requisitos fundamentales de cualquier pentesting, de esta forma es factible determinar los niveles de complejidad de dichas claves así como poder determinar si es posible acceder a un usuario privilegiado con acceso a permisos del sistema.
V3.- Gestión de sesiones	Requisito que hace alusión a todos los controles que rigen y permiten la interacción entre el usuario y la App. Web, en esta fase se debe de verificar que la App. Invalida toda sesión después de un periodo de inactividad y cierre de sesión.
V4.- Control de acceso	Requisito que se encarga de definir las limitantes de acceso a las herramientas, funciones y visualización de la BD de los usuarios, verificar que los controles de acceso fallen de forma segura, los atributos de usuario en su control de acceso no deben ser manipulados pos

	usuarios finales sin previa autorización.
V5.- Manejo de entrada de datos maliciosos	Requisito importante de evaluar debido a que del fallo de la validación apropiada de los datos antes de ser utilizados, se derivan todas las vulnerabilidades relacionadas a App. Web antes enumeradas en la tabla 1

Tabla 2.- Requisitos de Verificación Detallada (Manico, Jim, 2017)

Fase de penetración

Ya que se ha planeado el seguimiento de un estándar para la evaluación de una de las principales App. Web en un campo de estudio se procede a establecer pruebas de penetración auxiliadas por la Guía de Pruebas 4.0 de OWASP, las pruebas de Pentesting se encuentran basadas en un enfoque de caja negra (Black-box) y caja blanca (White-box), cuyo objetivo principal consiste en evaluar la App. Poniéndola a prueba bajo diferentes escenarios de búsqueda de vulnerabilidades para determinar su nivel de seguridad, así como detectar vulnerabilidades y poder corregirlas.

- Las pruebas de caja negra (Black-box) se basan en que el auditor toma el papel de una persona ajena a la empresa el cual carece de información previa, estas pruebas, se basan en su totalidad en la recolección de la información pública de la entidad dicha actividad es denominada Footprinting, permitiendo determinar posibles vías de ataque (González Pérez, 2014).
- Las pruebas de caja blanca (White-box) consisten en que el auditor tiene acceso previo a toda la información del sistema desde el diseño, código y BD, durante el tiempo que perduren las pruebas se mantiene en constante revisión el código de la App. Web, para determinar posibles fallos, esta parte del Pentesting está estrechamente relacionada al ciclo de vida de desarrollo de Software (SDLC), por ello es recomendable el uso en conjunto del ASVS y la Guía de Pruebas 4.0 de OWASP (Guevara Soriano, 2012).

Cabe mencionar que las pruebas de Pentesting deben seguir orden para poder documentarlas de manera consecutiva para poder ordenarlas por nivel de prioridad de solución dentro de un informe, las faces genéricas que debe abordar cualquier Pentesting son descritas a continuación:

- Alcance y términos de la auditoria de seguridad: en esta etapa se llega a acuerdos de trabajo entre Pentester y la empresa, en cuyo caso se le debe de indicar en qué consistirá cada prueba y los efectos que estos conllevan para que de esta forma la empresa decida si le favorece o no, por otro lado, la empresa también puede delimitar hasta qué punto desea auditar o en otros casos imponer condiciones para que el auditor decida qué acciones tomar. Todos estos acuerdos deben ser anotados en un contrato el cual deberá ser firmado tanto por la empresa como por el auditor, este sirve como protección (para ambas partes) y garantía de confidencialidad.
  Ejemplificando esto en el caso de estudio se delimito que la App. Web a auditar debe estar funcionando en todo momento para evitar problemáticas tanto con el personal docente, así como el alumnado, en consecuencia, se tomó la iniciativa de duplicar el servidor de BD y sus características para de esta forma hacer pruebas sin ninguna limitante para obtener resultados reales y de interés para la empresa.
- *Recolección de información:* en esta etapa el auditor se encarga de recolectar toda la información que sea posible de la empresa a auditar, muchos Auditores se valen de la Ingeniería social, redes sociales, Google hacking y Footprinting, una vez que se tiene toda la información el auditor se puede dar una idea general sobre cómo se comporta el objetivo, como está constituido y de esta manera determinar cómo atacarlo.
- *Análisis de las vulnerabilidades:* en esta etapa el auditor determina por medio de la información recabada en la fase anterior cual es el mejor vector de ataque debido para esta fase ya debe de contar con un panorama general de los medios para acceder al sistema y la información.
- *Explotación*: el objetivo de esta etapa es la de establecer los ataques con la finalidad de obtener resultados que determinen el nivel de seguridad de la App. Web, en esta fase se debe de tener cuidado y certeza de lo que se está haciendo de caso contrario la auditoria podría comprometer la información de la empresa.
  Una ejemplificación en el campo de estudio es la búsqueda y explotación de vulnerabilidades presentada por OWASP en el año2017 como se puede visualizar en la tabla 1, entre las cuales se encuentran Inyección SQL que va relacionada al punto V5 de la estandarización, XSS, autenticación y administración de sesión relacionado al punto V2 y V3, para de esta forma obtener resultados que sean favorables al plantel educativo.

- *Generación de informes*: última etapa de una auditoria informática, en la cual el Pentester creará un escrito o manual de los pasos, métodos y herramientas utilizadas durante el Pentesting, priorizando las vulnerabilidades encontradas desde la que presenta mayor riesgo y por consiguiente es la más importante a resolver hasta la que representa un menor riesgo.

Comentarios Finales.

En la actualidad la actividad de robo o suplantación de la información ha estado en constante crecimiento, tanto es así que las empresas han modificado sus hábitos de resguardo de datos, ya no les resulta satisfactorio el invertir solo en antivirus los cuales los mantienen controlados pero no en un resguardo total, el campo de estudio fue orientado al campo educativo universitario debido a que estas entidades actualmente manejan una serie de datos importantes y de alto interés para el público en general y por esta razón se deben determinar las fortalezas de sus TIC en especial a los sistemas de difusión masiva en caso del campo de estudio una App. de entorno Web enlazada a una BD.

Debido a esto se propone la implementación de auditorías de seguridad o Pentesting a App. Web, las cuales están en constante uso tanto de usuarios interno y externos a la empresa, apoyada con el uso de una metodología basada en el Estándar de Verificación de Seguridad en Aplicaciones versión 3.0.1, propuesto por OWASP por la cual se basa el cumplimiento de las funcionalidades de los componentes según se estén evaluando durante el Pentesting. Las pruebas de penetración por otro lado están sustentadas en la Guía de pruebas 4.0 de OWASP.

Los cuales han arrojado resultados significativos a pesar de que están orientados a una App. Web específica de una entidad universitaria, el estándar y manuales anteriormente mencionados generan un índice de confianza alto para el cual debe de tenerse en cuenta para proyectos personales, aunque las técnicas y objetivos varíen según las necesidades de las empresas que quieran recurrir a establecer auditorías internas en busca de vulnerabilidades, se concluye que este es una actividad de alto valor para la cuantificación de los índices de seguridad de cualquier App. Así como de las tecnologías TIC.

Referencias

González Pérez, P. (2014). *Ethical Hacking.Teoria y practica para la realizacion de un pentesting.* Madrid: 0xWORD.

Guevara Soriano, A. (2012). Haking ètico: Mitos y realidades . *.Seguridad, Cultura de prevención para TI*(12), 8-14.

Manico, Jim. (2017). *Estándar de Verificación de Seguridad en Aplicaciones 3.0.1.* OWASP.

OWASP. (2010). *The Ten Most Critical Web Application Security Risks.* OWASP.

OWASP. (2013). *Top10: The Ten Most Critical Web Application Security Risks.* OWASP.

OWASP. (04 de Abril de 2017). *About The Open Web Application Security Project*. Obtenido de https://www.owasp.org/index.php/About_The_Open_Web_Application_Security_Project

OWASP. (2017). *Top10: The Ten Most Critical Web Application Security Risks.* OWASP.

Ramires Castro, A. (2012). Riesgo Tecnológico y su impacto para las organizaciones. *.Seguridad, cultura de prevención para TI*(14), 13-16.

Rando, E., González, P., Aparicio, A., Martín, R., & Alonso Cebián, J. (2016). *Hacking Web Technologies.* Madrid: 0xWORD.

Villalobos Murillo, J. (2012). Principios de seguridad en bases de datos. *.Seguridad*(12), 4-7.