SEGURIDAD WEB CON HERRAMIENTAS OPEN SOURCE

PACHÓN QUINTERO PEDRO GERMÁN
Especialización en Seguridad Informática, Universidad Piloto de Colombia
Bogotá, Colombia
german.pachon@gmail.com

Abstract-The evolution of the Internet and the expansion of connections worldwide, has enabled most markets and businesses move towards e-commerce and online transactions where sensitive data and information are managed, before this growth have also increased the cyber attacks against web applications and in a scenario of uncertainty about the quality and safety of the code of the applications the best option is to ensure through layers of security, independent of code web applications in which it is developed, ensuring at all times the three pillars of security: availability, confidentiality and integrity.

Keywords: *Web Security, controls, Open Source, Layers, vulnerabilities.*

Resumen- La evolución de la Internet y la expansión de las conexiones en todo el mundo, ha permitido que la mayoría de mercados y empresas se muevan hacia el comercio electrónico y las transacciones en línea, donde se manejan datos e información muy sensible. Ante este crecimiento también han aumentado los ataques informáticos a las aplicaciones web y ante un escenario de incertidumbre sobre la calidad y seguridad del código de los sitios web, la mejor opción es asegurar por medio de capas de seguridad las aplicaciones Web, independiente del código con el que se haya desarrollado, garantizando en todo momento los tres pilares de la seguridad, entendidos como disponibilidad, confidencialidad e integridad.

Palabras clave: *Seguridad web, controles, OpenSource, capas, vulnerabilidades.*

I. INTRODUCCIÓN

Los servidores Web, están expuestos a distintos vectores de ataque, que podrían convertirse en serias vulnerabilidades de seguridad, que afecten no solo el servicio de la aplicación sino que también pueden, dependiendo su criticidad, afectar toda la infraestructura de una organización[1].

La adopción de controles que permitan mitigar estos riesgos, será el punto de partida para garantizar y dar confianza a los usuarios que utilicen las aplicaciones web, de que su información y datos personales serán tratados bajo estrictas medidas de seguridad.

A través de este documento, se pretende mostrar un modelo de seguridad para aplicaciones Web a través de capas y utilizando únicamente herramientas libres, lo que permitirá ser empleado tanto para grandes como para pequeñas empresas que no contarían con el suficiente capital para invertir en controles licenciados.

II. SEGURIDAD WEB

Si una aplicación Web, tiene por objeto ser accedida por múltiples usuarios y conexiones siendo visible a través de internet, se debe asegurar en todo momento que solamente sea utilizada para los fines de su

[1] http://www.imperva.com/docs/hii_web_application_attack_report_ed5.pdf

desarrollo y cumpla con los mínimos requisitos de seguridad, de acuerdo a esto, la seguridad se convierte en el pilar principal de diseño de aplicaciones Web con el fin de garantizar la disponibilidad, confidencialidad e integridad de toda la información del sitio.

Muchas veces la cantidad de código que puede tener un sitio en internet, no permite auditar todas las líneas de desarrollo en cuanto a protección y quedan huecos de seguridad que podrían ser vulnerados por un hacker con experiencia, alterando la información del sitio y el buen nombre de la organización[2].

Es muy común que las empresas piensen que implementar seguridad a su infraestructura informática y de servicios, tiene un costo muy alto y no aporta rentabilidad al negocio, sin embargo existen herramientas Open Source que garantizan la misma efectividad con una buena configuración, aportando robustez en la seguridad y garantizando que el daño en un ataque informático sea menor.

III. CONTROLES DE SEGURIDAD WEB

Aunque existen muchos tipos de controles de seguridad a nivel web, se pueden agrupar en diferentes categorías dependiendo su alcance, como vemos a continuación:

A. Controles de Aplicación

Estos controles están enfocados a proteger la capa de aplicación y es la primera línea de defensa donde llegan las peticiones del usuario y donde viajan datos personales y contraseñas de acceso. Los controles más efectivos son el cifrado SSL[3] y un WAF[4].

B. Controles en el sistema operativo

La instalación inicial de cualquier sistema operativo tiene muchas configuraciones por defecto, generando un alto índice de riesgo si el atacante quiere obtener información del sistema, por ejemplo que sería fácilmente adquirida con una configuración por defecto de los servicios web. Allí, es posible implementar controles de Hardening del SO, Base de datos y Servidor Web, adicionalmente se pueden incluir detectores de intrusos a nivel de host que pueden identificar peticiones sospechosas o cambios en la integridad del sistema[5].

C. Controles de red

Estos controles van enfocados a proteger el canal por donde pasan las peticiones hacia el servidor web, bloqueando puertos y filtrando el tráfico de red. En este caso, se utiliza el firewall ya sea local o del perímetro.

D. Controles físicos

Busca garantizar que el acceso físico a los servidores este protegido de personal no autorizado, adicionalmente garantizar la seguridad ante apagones eléctricos, incendios, inundaciones y desastres naturales. En esta categoría se definen controles físicos de acceso como la ubicación de los Data Center, el registro de acceso, el monitoreo, entre otros[6].

E. Políticas y guías de desarrollo seguro

Las políticas y guías de desarrollo seguro deben acompañar todo el ciclo de vida de la aplicación, desde la definición de los requisitos no funcionales, pasando por su desarrollo y despliegue.

---

2 http://www.welivesecurity.com/la-es/2015/03/12/10-consejos-desarrollo-seguro-de-aplicaciones/

3 https://www.digicert.com/es/ssl.htm

4 http://softsecuritycorp.com/index.php/soluciones-y-productos/proteccion-de-redes-y-aplicaciones/firewall-de-aplicaciones-waf

5 http://es.ccm.net/contents/162-sistema-de-deteccion-de-intrusiones-ids

6 http://www.datacenterdynamics.es/focus/archive/2012/10/claves-de-seguridad-f%C3%ADsica-en-el-data-center-ii

IV. INVESTIGACIÓN

Teniendo en cuenta lo establecido en el punto anterior, sobre las categorías de controles de seguridad, es posible establecer las capas de seguridad que debe tener una aplicación web, así:

CAPA	CONTROLES
1. Aplicación	SSL / WAF
2. Sistema Operativo	Hardening Web Hardening BD HIDS
3. Red	Firewall
4. Físico	Ubicación / Acceso
5. Políticas	Guía de desarrollo seguro

Tabla 1. – Capas de seguridad en una aplicación web – Fuente: El autor

Para la investigación se tendrán en cuenta las capas 1,2 y 3, donde se aplicarán los controles indicados con herramientas Open Source sobre una aplicación Web que no ha seguido ninguna política de desarrollo seguro, encontrándose totalmente vulnerable, teniendo como objetivo medir la eficacia de la seguridad al aplicar controles en estas capas.

Para el cumplimiento de dicho objetivo se deben tener en cuenta los siguientes ítems de medición:

- Tiempo de respuesta.
- Número de vulnerabilidades detectadas.
- Errores de navegación.
- Controles de seguridad.
- Cifrado de Datos.

V. FASE 1

Se realiza la instalación de un sitio web vulnerable proporcionado por el proyecto OWASP[7]:

https://github.com/adamdoupe/WackoPicko.

La instalación es sencilla y requiere solamente descomprimir los archivos en la ruta de despliegue en un servidor web previamente configurado.

Se realiza la instalación y configuración sobre un sistema operativo Centos 7, con los aplicativos Apache, PHP y MySQL.

Inicialmente no se configura ningún tipo de protección adicional a la configuración por defecto que traen los aplicativos, que generalmente es muy básica y poco confiable en el tema de seguridad.

Al final tenemos un servidor en marcha y un sitio web totalmente eficaz con funciones que generalmente se encuentran en una página pública (login, subir archivos, descargar, buscar, etc.).

A continuación se muestra el sitio montado:

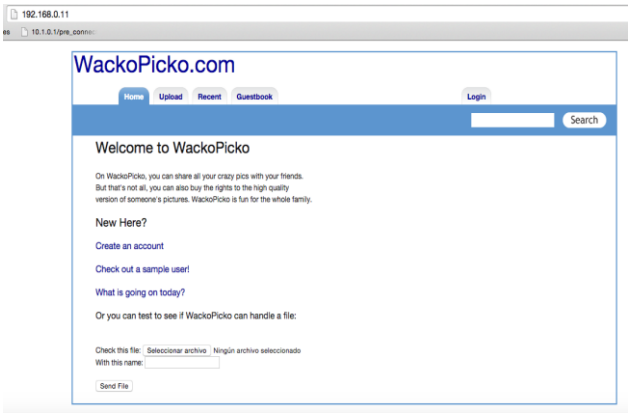

Figura 1. – Sitio Web Montado – Fuente: El autor

Se utiliza la herramienta Vega[8], para ejecutar un análisis de la aplicación a nivel de vulnerabilidades web. Los resultados se muestran a continuación:

[7] https://www.owasp.org/index.php/Main_Page

[8] https://subgraph.com/vega/

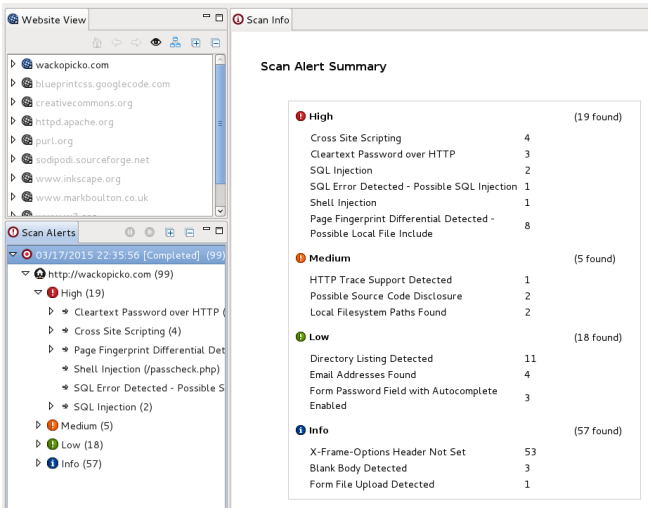

Figura 2. – Resultado Test de Vulnerabilidades en la aplicación Web – Fuente: El autor

A continuación se muestra el número de vulnerabilidades encontradas y su criticidad:

Altas	Medias	Bajas	Informativas
19	5	18	57

Tabla 2. – Resumen Vulnerabilidades por criticidad - Fuente: El autor

Así mismo, se realiza la medición de tiempo de carga de la página:

Prueba	Tiempo (ms)
1	23
2	37
3	28
4	24
5	18
6	30
7	15
8	18
9	15
10	23
Promedio de Carga	**23,1**

Tabla 3. – Resultado promedio de carga página Web - Fuente: El autor

Errores de Navegación: 0

Controles de seguridad implementados: 0

Cifrado del sitio: No

VI. FASE 2

Se realiza la implementación de los controles de seguridad de acuerdo al modelo propuesto en las capas 1,2 y 3 de la tabla 1:

Capa	Controles
1. Aplicación	SSL (openssl) WAF (mod_security)
2. Sistema Operativo	Hardening Web (apache) Hardening BD (MySQL) HIDS (OSSEC)
3. Red	Firewall (iptables)

Tabla 4. – Controles a emplear en la aplicación Web - Fuente: El autor

A continuación se detallan los controles implementados, sin relacionar su instalación y configuración ya que no es el objetivo de este documento.

A. Aplicación:

Se crea un certificado SSL en el servidor y se redirigirán todas las peticiones al puerto seguro 443, también se agrega y configura el módulo de apache mod_security que cumple la función de WAF.

B. Sistema Operativo:

Se realiza el hardening del servidor quitando los servicios innecesarios, adicionalmente se realiza hardening de los servicios de apache y MySQL.

- Apache:

Se oculta información de versión y plataforma del servidor web instalado.

- Mysql:

Se eliminan bases de datos predeterminadas y usuarios por defecto; se coloca una contraseña para el usuario root y se crea un usuario específico para la aplicación con acceso únicamente local; no se permite acceso remoto al servidor MySql; se instala el detector de intrusos OSSEC[9] analizando todos los logs de acceso generados en la aplicación.

C. Red:

Se configura el firewall para solo aceptar conexiones a través de los puertos 80 y 443.

VII. MEDICIÓN POST-IMPLEMENTACIÓN

Tiempo de carga:

Prueba	Tiempo (ms)
1	46
2	37
3	34
4	44
5	42
6	54
7	38
8	42
9	51
10	70
Promedio de Carga	**45,8**

Tabla 5. – Resultado promedio de carga página Web post-controles - Fuente: El autor

Vulnerabilidades:

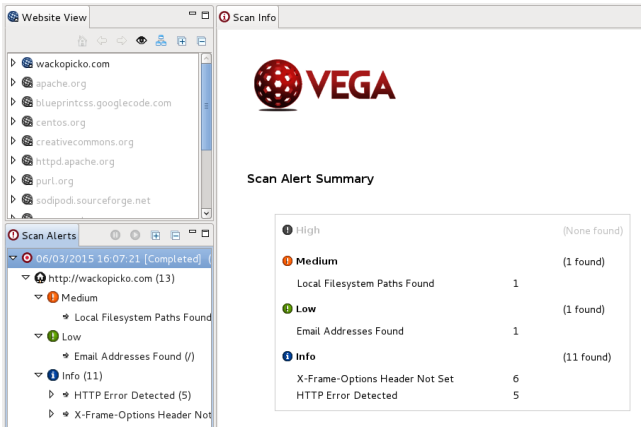

Figura 3. – Resultado Test de Vulnerabilidades post-controles en la aplicación Web – Fuente: El autor

Altas	Medias	Bajas	Informativas
0	1	1	11

Tabla 6. – Resumen Vulnerabilidades por criticidad post-controles - Fuente: El autor

Errores de navegación: 0

Controles implementados:

- Aplicación
- Sistema operativo
- Red

Cifrado de datos: Si

VIII. CONCLUSIONES

Se evidencia que, aunque se aumentó el tiempo de carga de la aplicación web, este no es considerable toda vez que se trata de milisegundos, de igual forma es objeto de compensación adicionando mayor procesamiento y memoria que reponga este timeout.

Las vulnerabilidades encontradas comúnmente en las aplicaciones web, son eliminadas en un 100% a nivel crítico y un 80% de nivel medio, sin necesidad de implementar código de seguridad en el

[9] http://www.ossec.net/

desarrollo, esto indica un control muy eficiente de amenazas.

El cifrado con SSL permite cifrar los datos de ingreso de contraseñas, lo que disminuye el riesgo de captura de las mismas.

Se requiere realizar pruebas rigurosas de carga en la página web con todos los controles, esto teniendo en cuenta que existen reglas de WAF que podrían bloquear algunas categorías o zonas de la aplicación.

Las aplicaciones web, son uno de los servicios con mayor número de ataques, y representa en muchos casos, la puerta de entrada a toda la infraestructura crítica de una organización, la principal arma ante cualquier amenaza es mantener políticas y guías de desarrollo seguro, que se mantengan actualizadas, pero ante un escenario de incertidumbre, ante nuevas amenazas y donde posiblemente no se realicen con disciplina estos procedimientos, la mejor opción es asegurar a nivel de aplicación, sistema operativo y red todos los servicios web, lo que representará al final una barrera que proteja la disponibilidad, confidencialidad e integridad de la información.

Así las cosas, es procedente indicar que la aplicación de controles en las capas indicadas, resulta eficaz en cuanto a nivel de seguridad, bajo la utilización de herramientas Open Source, generando confianza por parte de los usuarios de las organizaciones a un bajo costo.

IX. BIBLIOGRAFÍA

[1] DÍAZ Vicente, FERNÁNDEZ Daniel "Controles técnicos de seguridad para la protección de aplicaciones web".

[2] La seguridad también puede ser 'Open Source', Rosalía Arroyo – Página channelbiz

http://www.channelbiz.es/2014/04/29/la-seguridad-tambien-puede-ser-open-source/

[3] TECHTARGET, "Web application firewall (WAF)".

[4] PÉREZ Carlos, "Seguridad en Sistemas Informáticos (SSI) Laboratorio: OSSEC", Universidad de Valencia.

[5] CARDENAL Gardoki, "Protege tu servidor web con el módulo ModSecurity" HostaliaWhitePapers