Vulnerabilidad de ambientes virtuales de aprendizaje utilizando SQLMap, RIPS, W3AF y Nessus*[1]

[*Vulnerability in Virtual Learning Environments using SQLMap, RIPS, W3AF and Nessus*]

Alexander BARINAS LÓPEZ[2], Andrea Catherine ALARCÓN ALDANA[3], Mauro CALLEJAS CUERVO[4]

Recibo: 20.11.2013 – Aprobación: 15.03.2014

Resumen

Actualmente y cada segundo se realizan ataques masivos a servicios web, los cuales ponen en riesgo la información personal y acceso de aquellos que lo utilizan, atentando contra la autenticidad, integridad y confidencialidad de la información. Los Ambientes Virtuales de Aprendizaje, AVA, no son ajenos a estos ataques, razón por la cual se presenta un estudio que aborda aspectos sobre seguridad de la información en la web, herramientas de seguridad y análisis de vulnerabilidades, así como también se propone algunos criterios a evaluar en cuanto a seguridad de información en AVA y posteriormente se realiza el proceso de

* **Modelo para la citación de este artículo**:
BARINAS LÓPEZ, Alexander; ALARCÓN ALDANA, Andrea Catherine & CALLEJAS CUERVO, Mauro (2014). Vulnerabilidad de Ambientes Virtuales de Aprendizaje utilizando SQLMap, RIPS, W3AF y Nessus. En: Ventana Informática No. 30 (ene-jun). Manizales (Colombia): Facultad de Ciencias e Ingeniería, Universidad de Manizales. p. 247-260. ISSN: 0123-9678

1 Artículo de investigación científica y tecnológica proveniente del proyecto *Modelo de evaluación de seguridad en sistemas de gestión de aprendizaje a partir de las características propuestas en la norma ISO 9126*, ejecutado en el periodo *09.2012 – 09.2013*, e inscrito en el grupo de investigación *GIS* de la *Universidad Pedagógica y Tecnológica de Colombia*..

2 Ingeniero de Sistemas, Especialista en Telemática, Estudiante Maestría en Tecnología Informática. Docente, Fundación Universitaria Juan de Castellanos (Tunja, Boyacá, Colombia), Integrante Grupo de Investigación en Software GIS –UPTC. Correo electrónico: alexbari99@gmail.com

3 Ingeniero de Sistemas y Computación, Especialista en Ingeniería de Software, Magister en Software Libre. Docente – Investigador, Universidad Pedagógica y Tecnológica de Colombia UPTC (Tunja, Boyacá, Colombia), Directora Grupo de Investigación en Software GIS -UPTC. Correo electrónico: acalarcon@gmail.com

4 Ingeniero de Sistemas, Especialista en Ingeniería de Software, Magister en Ciencias Computacionales. Docente – Investigador, Universidad Pedagógica y Tecnológica de Colombia UPTC (Tunja, Boyacá, Colombia), Integrante Grupo de Investigación en Software GIS –UPTC. Correo electrónico: maurocallejas@gmail.com

análisis de seguridad por medio de las herramientas SQLMap, RIPS, W3AF y Nessus.

Palabras Clave: *Seguridad de la información, seguridad Web, análisis de vulnerabilidades, Ambientes Virtuales de Aprendizaje.*

Abstract

Currently and every second massive attacks are made to Web services, which put at risk the personal information and access to those who use it, undermining the authenticity, integrity and confidentiality of information. Virtual Learning Environments are no strangers to these attacks, which currently study that address security aspects of information on the web, security tools and vulnerability analysis and also proposes some criteria to assess as information security in Virtual Learning Environments subsequently is performed process of analysis with the safety tools SQLMap, RIPS, W3AF and Nessus.

Keywords*: Information security, web security, vulnerability analysis, Virtual Learning Environments.*

Introducción

En la actualidad la información es el activo más valioso para el desarrollo de las actividades en las organizaciones, la cual está expuesta a accesos no autorizados, alteración y revelación de información, debido a la no implementación de políticas de seguridad, que conduzcan a un buen funcionamiento de los procesos que desarrollan. Además con los adelantos tecnológicos, se deben crear técnicas y estrategias para mantener la información segura, ya que se aplican métodos, técnicas y herramientas que facilitan el acceso no autorizado a la información, con el propósito de ser manipulada, alterada o con fines delictivos para obtener algún beneficio a nivel monetario o buscando dejar huellas que identifiquen sus acciones criminales. Lo anterior, genera un gran problema para las organizaciones, debido al desconocimiento y subvaloración de la necesidad de invertir en los esquemas de seguridad para la protección de la información.

Es innegable que cada día las organizaciones dependen en mayor medida de la información y la tecnología, y que los sistemas de información actualmente se encuentran soportados por esta. De otro modo, en años anteriores la protección de la información era más sencilla, ya que las grandes plataformas de datos actuaban de forma independiente

sin conectividad alguna, las arquitecturas existentes eran totalmente centralizadas y las terminales tenían capacidades de procesamiento limitadas; en cambio, en la actualidad con la diversidad de redes y plataformas tanto internas como externas, hacen que la información esté cada vez más expuesta a las diferentes amenazas y ataques informáticos, por lo cual, se hace necesario que las entidades administren adecuadamente la seguridad de la información con el fin de minimizar las potenciales vulnerabilidades.

Debido a las necesidades en relación a la seguridad de la información han surgido comunidades, grupos u organizaciones cuya finalidad es propender por la disminución de los riesgos en términos de seguridad; de esta manera, nace el proyecto OWASP[5] (OWASP, 2013a), proyecto de código abierto dedicado a determinar y combatir las causas que hacen que el software sea inseguro; al igual se describen normas y estándares como ISO 9126, ISO 27001 y 27002, las cuales especifican los requisitos necesarios para establecer, implantar, mantener y mejorar un Sistema de Gestión de la Seguridad de la Información-SGSI, de acuerdo a la Calder & Watkins (2008, 36-39); además se citan artículos relacionados con el análisis y explotación de entornos Web, gestión de riesgos y vulnerabilidades en entornos Web.

El presente artículo en su primer parte da a conocer los aspectos sobre seguridad de la información, seguridad Web, herramientas de seguridad y análisis de vulnerabilidades enfocados a AVA (Ambientes Virtuales de Aprendizaje). En la segunda parte se establecen los criterios de seguridad a evaluar en los AVA, se realiza el proceso de análisis de seguridad por medio de las herramientas *SQLMap*, RIPS, W3AF y *Nessus*; y por último se presentan los resultados del respectivo análisis y las conclusiones de la investigación.

1. Fundamento teórico

1.1 Seguridad de la información en la web

Aguilera López (2010, 9) adopta el término *seguro* como el estado de libertad y exención de todo peligro, daño o riesgo, es así que la seguridad, según Lehtinen & Gangemi (2006, 9-11), significa mantener en secreto aquello que no se quiere revelar y que pueda caer en manos del enemigo o mantener estrategias para evitar el riesgo de vulneración; mientras que al hablar de seguridad de la información, esta se entiende como «*proceso en el que se da*

5 Acrónimo de *Open Web Application Security Project* = Proyecto abierto de seguridad de aplicaciones web.

cabida a un creciente número de elementos: aspectos tecnológicos, de gestión-organizacionales, de recursos humanos, de índole económica, de negocios, de tipo legal, de cumplimiento, etc.; abarcando no solo aspectos informáticos y de telecomunicaciones sino también aspectos físicos, medioambientales, humanos, etc.» (Areitio, 2008, xv).

Para hacer el estudio de riesgos a los que puede estar expuesta la seguridad de la información, Feng et al., (2013, 57-58) consideran que, es necesario tener en cuenta los elementos que integran el sistema; se logra a través de diálogos directos con quienes tienen la responsabilidad de la organización; de igual manera, se pueden tomar muestras y practicar pruebas que permitan prever los peligros que afectan el sistema de información para adoptar medidas que controlen las posibles amenazas en la seguridad.

De acuerdo con Areitio (2008, 2-4), los principales objetivos inherentes a la seguridad de la información son:

- Disponibilidad y accesibilidad de los sistemas y datos, sólo para su uso autorizado. Es importante asegurar el funcionamiento oportuno y eficiente del sistema de tal forma que los usuarios puedan acceder sin dificultad.
- Integridad. Permite que la información no sea modificada o alterada por intrusos.
- Confidencialidad de datos y de la información del sistema. Mediante este mecanismo se respeta la privacidad de la información y se evita su revelación a quienes no están autorizados.
- Responsabilidad a nivel individual (registros de auditoría). Es tomada como una política en la organización, ya que este requisito permite detectar todas las acciones realizadas por los usuarios previniendo la manipulación indebida por parte de personas ajenas.
- Confiabilidad (aseguramiento). Muestra el cumplimiento efectivo de los anteriores objetivos que reflejan la confianza en la seguridad del sistema a nivel técnico y operacional.

Conociendo los principales objetivos de la seguridad de la información, es necesario tener en cuenta que en la actualidad la oferta de bienes y servicios de la mayoría de las empresas está enfocándose en el comercio electrónico a través de la web, porque ven en ella muchas facilidades para darse a conocer, como aseguran Daltabuit et al. (2007, 705-706), pero hay que contemplar que debido a la demanda en su uso, afirma Tanenbaum (2012, 729), la información se ve expuesta

a un sinnúmero de amenazas, las cuales son divulgadas a través de diferentes fuentes de información.

Un servidor Web puede ser atacado para extraer, capturar, manipular y en ocasiones alterar información poniendo en riesgo la disponibilidad, integridad y la confidencialidad de la misma, consideran Ofuonye & Miller (2013, 1689), teniendo en cuenta que una amenaza es todo aquello que pueda afectar cualquiera de los tres pilares de la seguridad: disponibilidad, integridad y confidencialidad. Pensando en la protección del valioso recurso que es la información, surgen herramientas para el análisis de seguridad en aplicaciones web, señala Holm (2012, 1), las cuales son software especializado en la tarea de recolección, análisis y presentación de información concerniente a seguridad que facilitan los procesos de auditoría y test penetración, indica Mansfield-Devine (2013, 6-9).

Los *pentest* (test de penetración), los cuales son parte de las estrategias para realizar hacking ético, son una evaluación activa de las medidas de seguridad de la información, que permiten conocer el nivel de vulnerabilidad de una aplicación, para de esta manera tomar las medidas y estrategias necesarias para reducir dicho nivel, puesto que, según OWASP (2008), una vulnerabilidad es un agujero o una debilidad que puede ser un defecto o bug en la aplicación, que permite a un atacante causar daños y/o controlar el flujo normal de la misma, ocasionando que la organización no presente el rendimiento esperado, por ende deteriorando su imagen y nivel competitivo.

Con el propósito de reducir las debilidades en el acceso y manejo de la información, surgen los análisis de vulnerabilidades, conocidos como los procesos por los cuales se pretende conocer el nivel de seguridad en el que se encuentra un ente (organización, sistema, software, etc.), destacando que los análisis llevados a cabo por profesionales, según Petreska et al. (2010, 1008-1009) y Salini & Kanmani (2012, 1-6), se rigen a través de metodologías que describen exhaustivamente el proceso que ha de llevar el *pentester*, desde la recolección y análisis de la información hasta la presentación de reportes; ejemplo de estos modelos es la *Guía de Pruebas OWASP*, que al momento de escribir este artículo, se encuentra en su versión 3.0.

Actualmente las aplicaciones web tienen un sinnúmero de propósitos, finalidades y servicios, haciendo que sus niveles de seguridad de la información sean diversos por sus características técnicas y funcionales, por esta razón es necesario analizar las vulnerabilidades teniendo en cuenta características propias del tipo de aplicación que se va a estudiar. Esta investigación se orienta al análisis de vulnerabilidades

en los AVA, que según Carmona & Rodríguez (2009, 10-16) y Capacho (2011, 253), son un espacio de aprendizaje, el cuál valiéndose de la tecnología de hardware, software, Internet, multimedia entre otros, constituyen una nueva manera de tecnología en la educación, facilitando así a estudiantes y tutores, el mejor manejo de los procesos de asesoría, conocimiento y colaboración. Para UNESCO (2013), la educación para todos a través de los AVA, permite reducir la brecha entre los menos favorecidos y de esta manera lograr mejorar la calidad de vida de las personas.

Hoy día, en el mercado existen diversos AVA que han permitido cambiar y de cierta manera mejorar las estrategias utilizadas en el proceso enseñanza aprendizaje, es así que las instituciones, partiendo de los factores, recursos y características propias de su comunidad académica han optado por implementar y usar algunos de los AVA, entre los cuales se puede mencionar las plataformas *Moodle* y *Dokeos*.

Al respecto, se han desarrollado varios trabajos, puesto que la tecnología y la información han traído consigo la aparición de diversos actores (personas, aplicaciones, técnicas, procedimientos y herramientas) que hacen que las organizaciones estén cada vez más susceptibles de sufrir ataques informáticos, lo cual se evidencia en el top de vulnerabilidades más representativas, según OWASP (2013b): inyección de código, mal manejo de las sesiones en aquellas aplicaciones que utilizan autenticación, validación mínima de la información ingresada por el atacante, acceso no autorizado a información crítica debido a errores en el diseño o desarrollo, protección incorrecta de datos sensibles, escasos controles en el servidor permitiendo que un atacante acceda a funciones que no debe, secuestro de sesiones a través de las credenciales o identificador de sesión (*cookies*) o lograr suplantar la sesión del usuario.

Como se puede ver, la presencia de vulnerabilidades relacionados con la seguridad en el software, aseguran Chowdhury & Zulkernine (2011, 294-295), se ha convertido en una preocupación importante en la industria del software, aunque, de acuerdo con Arora et al. (2010, 164), Ofuonye & Miller (2013, 1689-1690) y Shahmehri et al. (2012, 997-999), se están haciendo grandes esfuerzos para reducir las vulnerabilidades en el software, aunque el número de incidentes que se derivan de su explotación sigue siendo alto. Actualmente, señalan Stevanovic, An & Vlajic (2012, 8707-8708), uno de los ataques más perjudiciales para la seguridad en Internet es el de Denegación de Servicio Distribuida (DDoS), a través del cual los intrusos ejecutan ataques DDoS automatizados por medio de robots Web con el fin de capturar sesiones y mezclarse entre los usuarios normales sin dejar huella.

2. Metodología

Para analizar y evaluar el nivel de seguridad en que se encuentran los AVA como caso de estudio, se tienen en cuenta los siguientes criterios.

- Integridad: Es la propiedad que busca mantener la información exactamente cómo fue generada, protegiéndola de modificaciones accidentales y/o intencionales.
- Disponibilidad: Es la cualidad, característica o condición que posee la información de encontrase disponible de quienes quieren acceder a ella.
- Confidencialidad: Es la capacidad para prevenir la divulgación parcial o completa de la información sensible a terceros.
- Los criterios mencionados son analizados utilizando las cuatro herramientas de seguridad, presentadas en la Tabla 1.

Tabla 1. Descripción de las herramientas de seguridad

Herramienta	Descripción	Licencia
RIPS	Herramienta para análisis de vulnerabilidades en código estático PHP	GPL 3.0
SQLMap	Herramienta de pruebas de penetración de código abierto, que automatiza el proceso de detectar y explotar vulnerabilidades de inyección SQL	GNU
Nessus	Programa de escaneo de vulnerabilidades en diversos sistemas operativos	GNU
W3AF	Proyecto cuyo objetivo es desarrollar un Framework para ayudar a proteger las aplicaciones web mediante la búsqueda y explotación de vulnerabilidades	GPL 2.0

Estas herramientas fueron evaluadas mediante un proceso de investigación y métricas de viabilidad, y posteriormente el proceso de análisis de seguridad se lleva a cabo mediante análisis de código estático (utilizando RIPS), donde se analiza el código fuente en busca de posibles fallos en la validación de las entradas del aplicativo y pruebas de caja negra (usando las herramientas *Nessus*, W3AF y *SQLMap*), en el cuál se evalúan las entradas, procesos y los resultados así como el comportamiento de la aplicación.

El proceso de análisis inicia con la recolección de la información general de los aplicativos *Dokeos* y *Moodle* seguido por las pruebas de seguridad y análisis de resultados.

2.1 Recolección de información. La búsqueda de información inició con la consulta de vulnerabilidades públicas en bases de datos de sus desarrolladores o de terceros, como *exploit-db* (*Exploit Database*), *Common Vulnerabilities and Exposures, CVE* (Mitre Co., 2014) y *Open Sourced Vulnerability Database* (OSVDB); y después se obtienen las

características técnicas de los AVA (*Moodle* y *Dokeos*) en las versiones de análisis, las cuales se muestran en la Tabla 2.

Tabla 2. Características técnicas de los AVA Dokeos y Moodle (Construida a partir de las características de los AVA Dokeos y Moodle)

	Dokeos	Moodle
Versión	2.0	2.4.5
Tamaño	104 MB	129 MB
Número de ficheros	13.955	14.074
Arquitectura	Modelo vista controlador MVC	Modelo vista controlador MVC
Lenguaje	PHP	PHP
Base de datos	MySQL	MySQL, PostgrSQL, MSMSQL, Oracle, SQLite
Permisos exigidos	Lectura/escritura	Lectura/escritura

2.2 Descripción de las pruebas. A continuación se presenta la descripción de las pruebas de caja blanca y negra aplicadas para determinar el nivel presentado por los AVA, frente a algunas vulnerabilidades descritas por la comunidad libre y abierta sobre seguridad en aplicaciones (OWASP, 2013b), las cuales son presentadas en la Tabla 3.

Tabla 3. Vulnerabilidades en aplicaciones Web (OWASP, 2013b)

Vulnerabilidad	Descripción
A1 – Injection	Corresponde a las inyección de código, siendo las inyecciones SQL una de las más comunes.
A2 – Broken Authentication and Session Management	Corresponde al mal manejo de las sesiones en aquellas aplicaciones que utilizan autenticación.
A3 – Cross – Site Scripting (XSS)	Ocurre cuando existe validación pobre de la información ingresada por el atacante.
A4 – Insecure Direct Object References	Puede derivar en un acceso no autorizado a información crítica debido a errores en el diseño o desarrollo.
A5 – Security Misconfiguration	Corresponde a configuraciones no adecuadas que pueden impactar en la seguridad de la propia aplicación.
A6 – Sensitive Data Exposure	Se refiere a la protección incorrecta de datos críticos tales como, por ejemplo, números de tarjetas de crédito, contraseñas, entre otros.
A7 – Missing Function Level access Control	Corresponde a la falta de controles desde el servidor, permitiendo a un posible atacante acceder a funciones a las que no debería.
A8 – Cross – Site Request Forgery (CSRF)	Permite a un atacante generar peticiones sobre una aplicación vulnerable a partir de la sesión de la víctima.
A9 – Using Components with Known Vulnerabilities	Corresponde a la explotación de librerías, frameworks y otros componentes vulnerables por parte de un atacante con el fin de obtener acceso o combinar con otros ataques.
A10 – Unvalidated Redirects and Forwards	Los atacantes aprovechan el uso de redirecciones de sitios web a otros sitios utilizando información no confiable (untrusted) para redirigir a las víctimas a sitios de phishing o que contienen malware.

2.2.1 Pruebas de caja blanca. Las pruebas inician con la búsqueda de vulnerabilidades en los AVA mediante la herramienta RIPS cuya configuración era encontrar todos los fallos posibles en el código fuente como la ejecución de código, ejecución de comandos, inyección de cabeceras, inclusión, revelación y manipulación de ficheros, inyección de SQL, LDAP y XPATH.

2.2.2 Pruebas de caja negra. Estas pruebas se realizaron mediante las herramientas *SQLMap*, W3AF y *Nessus*, las cuales se describen a continuación:

- SQLMap: Esta herramienta se configuró de tal manera que identificara todos los puntos posibles de inyección SQL para su explotación y el posterior análisis de comportamientos y salidas de los aplicativos.
- W3AF: En esta herramienta se usaron *pluggins* de análisis y explotación de vulnerabilidades Web.
- Nessus: Fue configurado para encontrar vulnerabilidades generales en las aplicaciones Web.

3. Resultados y discusión

3.1 Análisis de pruebas en el AVA – Moodle

A continuación se presentan algunas imágenes (pantallazos) que muestran los resultados de las pruebas sobre el AVA *Moodle*, utilizando las herramientas mencionadas, permitiendo así establecer un análisis sobre el estado actual del nivel de vulnerabilidad sobre la información en esta plataforma. El ejercicio con RIPS en *Moodle* (Figura 1, superior derecha), permitió analizar la totalidad de los archivos públicos buscando en su código posibles vulnerabilidades. Cabe destacar que este tipo de análisis es denotativo, ya que muestra qué tan estructurada y desarrollada está una aplicación web.

Uno de los aspectos llamativos en la investigación, es la carencia de vulnerabilidades y falsos positivos en *Moodle* (Tabla 3), dado que su código bien estructurado le permitió a RIPS analizarlo totalmente, pero las precauciones tomadas a la hora de validar las entradas en todas sus funciones, fue simplemente eficaz.

A continuación se procede con los análisis de caja negra, utilizando *SQLMap* (Figura 1, superior izquierda), W3AF y NESSUS (Figura 1, inferior), se evidenció de nuevo una característica fundamental de esta plataforma, que fue su alto nivel de seguridad sobre la información, puesto que utilizando estas herramientas no se detectó ninguna vulnerabilidad.

3.2 Análisis de pruebas en el AVA – Dokeos

Las siguientes figuras permiten conocer los reportes de las pruebas sobre el AVA *Dokeos*, utilizando las herramientas mencionadas generando los siguientes resultados:

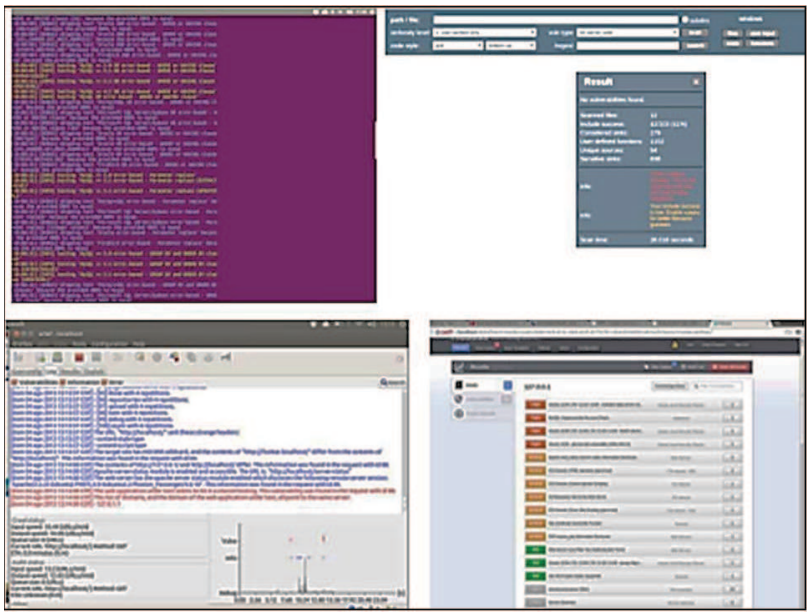

Figura 1. Análisis con SQLMap y RIPS, W3AF y Nessus en Moodle

El análisis de Dokeos demostró la falta de rigurosidad a la hora de validar las entradas; con la herramienta RIPS (Figura 2, superior derecha) se hallaron 79 posibles vulnerabilidades (Tabla 3), 71 de ellas correspondientes a Inclusión de archivos, 3 de ejecución de código, 4 revelación de ficheros, y un posible control de flujo; las vulnerabilidades encontradas permiten la ejecución remota de código, ya sea por inserción de archivos externos, o con inyección de código envenenado, con lo cual se puede llegar a tomar el control interno del servidor.

En las pruebas de caja negra, con W3AF, *Nessus* (Figura 2 inferior) y SQLMap (Figura 2, superior izquierda), denotaron tres nuevas vulnerabilidades XSS, logrando de esta manera ejecutar con URL malformadas, la ejecución de código javascript en el navegador de la víctima, de esta manera lograr robar los identificadores de sesión para su posterior secuestro, este tipo de ataque se conoce como *hijacking*, con el cual sin necesidad de conocer las credenciales de usuario y contraseña se puede lograr tomar el control de la sesión del usuario.

Las pruebas mencionadas denotan el estado en el que la plataforma *Dokeos* se encuentra, mostrando su poca madurez en el desarrollo de medidas de seguridad efectivas, evidenciando que la carencia de modelos eficaces para la validación de entradas es un problema que afecta directamente a la entidad que la utilice.

Tabla 3. Resultados comparativos de los análisis

AVA	Total	Code execution	SQL injection	XSS	Bypass something	Gain information	CSRF	File inclusion	Flow control
Moodle	0	0	0	0	0	0	0	0	0
Dokeos	83	3	0	3	0	4	1	71	1

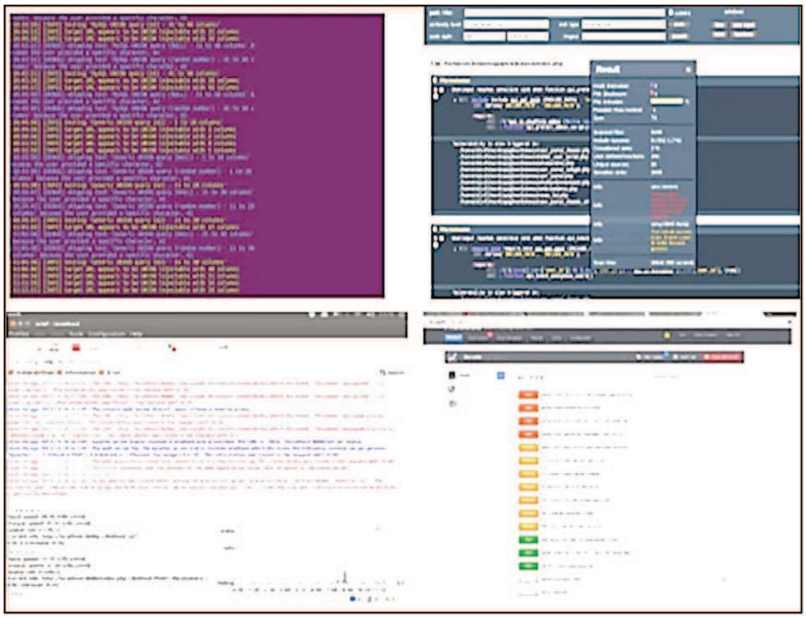

Figura 2. Análisis con SQLMap y RIPS, W3AF y Nessus en Dokeos

4. Conclusiones

En el proceso de investigación se evidencia que gran parte de la responsabilidad y eficacia de las medidas para salvaguardar la seguridad de la información, radica en las tareas de administración del sistema, las configuraciones que se toman a la hora de instalar la plataforma, como la utilización de privilegios adecuados tanto en el sistema de ficheros y la base de datos; estos son factores relevantes a la hora de determinar el nivel de escalabilidad que puede llegar a tener un determinado ata-

que. Otro pilar fundamental, es la educación y capacitación en términos de seguridad a los usuarios de las plataformas; todas estas medidas juntas, permiten reducir el riesgo de sufrir ataques efectivos, y disminuir potencialmente los daños que pueden llegar a provocar alguno de ellos.

El nivel de vulnerabilidad que posee un AVA puede ser disminuido potencialmente con medidas de seguridad, sencillas y efectivas como las mencionadas en el párrafo anterior, por ello se ve la necesidad de diseñar un modelo para la evaluación de seguridad de aplicaciones web y la creación de políticas de seguridad sencillas y efectivas en su implementación.

El desarrollo de modelos e implementaciones en el ámbito de seguridad, se ve maduro en *Moodle*, debido a su trascendencia a lo largo de los años, demostrando en este estudio que no es vulnerable ante posibles ataques conocidos, lo cual no quiere decir que sea 100% seguro, ya que dicha afirmación depende de otros factores, como son la gestión del administrador, la gestión de los usuarios, y el sistema donde haya sido implementado; en contraste con la poca experiencia que tiene *Dokeos* en este campo, aspectos como las estrategias para mejorar el nivel de seguridad de la información, deben ser tomados en cuenta para mejorar la calidad en general de esta aplicación.

La falta de chequeos eficaces por parte de las plataformas para comprobar la seguridad de las mismas, es una característica que se debería implementar en futuras versiones en estas plataformas. Adicionalmente los equipos de soporte y mantenimiento de *Moodle* y *Dokeos* deberían fortalecer sus herramientas de colaboración para poder conocer más rápidamente sus actualizaciones, buscando reducir las potenciales intrusiones o afectaciones de estas aplicaciones.

Referencias bibliográficas

AGUILERA, LÓPEZ Purificación (2010). Seguridad informática [en línea]. Madrid (España): Editorial Editex. 240 p. ISBN: 978-84-9771-657-4.

AREITIO, Javier (2008). Seguridad de la información: redes, informática y sistemas de información [en línea]. Madrid (España): Learning Paraninfo. 566 p. ISBN: 978-84-9732-502-8.

ARORA, Ashish; FORMAN, Chris; NANDKUMAR, Anand & TELANG, Rahul (2010). Competition and patching of security vulnerabilities: An empirical analysis [online]. In: Information Economics and Policy, Vol. 22, No. 2, (May). Philadelphia (PA, USA): Elsevier Inc. p. 164–177. ISSN: 0167-6245 <doi:10.1016/j.infoecopol.2009.10.002> <http://ac.els-cdn.com/S0167624509000651/1-s2.0-S0167624509000651-main.pdf?_tid=09eb72fc-a0b1-11e3-af37-00000aab0f6b&acdnat=1393616978_7d1cdf22adb92aa9a1f102b9dfdb8588> [consulta: 28/02/2014]

CALDER, Alan & WATKINS, Steve (2012). IT Governance: An international guide to data security and ISO27001/ISO27002. 5 ed. [online]. London (UK): Kogan Page. 376 p. ISBN: 978-0-7494-6486-1-8. <http://books.google.com.co/books?id=s7q8kNkjNhgC&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false> [consult: 28/02/2014].

CAPACHO PORTILLA, José Rafael (2011). Evaluación del aprendizaje en espacios virtuales – TIC. Barranquilla (Colombia): Editorial Universidad del Norte. 344 p. ISBN: 978-958-741-122-5

CARMONA SUÁREZ, Édgar Javier & RODRÍGUEZ SALINAS, Elizabeth (2009). Tecnologías de la Información y la Comunicación: Ambientes Web para la Calidad Educativa. Armenia (Colombia): Ediciones Elizcom. 182 p. ISBN: 978-958-99325-0-6

CHOWDHURY, Istehad & ZULKERNINE, Mohammad (2011). Using complexity, coupling, and cohesion metrics as early indicators of vulnerabilities [online]. In: Journal of Systems Architecture, Vol. 57, No. 3, (Mar.) New York (NY, USA): Elseiver Inc. p. 294–313. ISSN: 1383-7621 <doi:10.1016/j.sysarc.2010.06.003> <http://ac.els-cdn.com/S1383762110000615/1-s2.0-S1383762110000615-main.pdf?_tid=9a1831f8-a5a2-11e3-8097-00000aacb35f&acdnat=1394160534_3a04f1175944141bf291222a6d945c5c> [consult: 28/02/2014]

DALTABUIT GODAS, Enrique; HERNÁNDEZ AUDELO, Leobardo; MALLEN FULLERTON, Guillermo & VÁZQUEZ GÓMEZ, José de Jesús (2007). La seguridad de la información. Ciudad de México (México): Limusa. 771 p. ISBN: 978-968-18-6935-9.

EXPLOIT DATABASE (2013). The Exploit Database [online]. Reno (NV, USA): Exploit Database. <http://www.exploit-db.com> [consult: 07/08/2013].

FENG, Nan; WANG, Harry & LI, Minqiang (2013). A security risk analysis model for information systems: Causal relationships of risk factors and vulnerability propagation analysis [online]. In: Information Sciences, Vol. 256, (Jan). New York (NY, USA): Elsevier Inc. p. 57-73. ISSN: 0020-0255 <doi:10.1016/j.ins.2013.02.036> <http://ac.els-cdn.com/S0020025513001539/1-s2.0-S0020025513001539-main.pdf?_tid=d0ebabac-a5b5-11e3-85f6-00000aacb35f&acdnat=1394168786_0cc353abcdfb34a5898750b7e2ef6b74> [consult: 28/09/2013]

HOLM, Hannes (2012). Performance of automated network vulnerability scanning at remediating security issues [online]. In: Computers & Security, Vol. 31, No. 2, (Mar). Cambridge (MA, USA): Elsevier Inc. p. 164–175. ISSN: 0167-4048. <doi:10.1016/j.cose.2011.12.014> <http://ac.els-cdn.com/S0167404811001696/1-s2.0-S0167404811001696-main.pdf?_tid=ef8752b2-a5c1-11e3-b060-00000aab0f02&acdnat=1394173991_ad590e8bcbafe8c06ffb44358873000d> [consult: 28/09/2013]

LEHTINEN, Rick; RUSELL, Deborah & GANGEMI SR., G. T. (2006). Computer Security Basics [online]. Sebastopol (CA, USA): O'Reilly Media. 2 ed. 312 p. ISBN: 978-0-596-00669-3 <http://books.google.es/books?id=DyrLV0kZEd8C&printsec=frontcover&dq=Computer+Security+Basics&hl=en&sa=X&ei=kGgZU7-XJIu7kQeo7YD4Dw&ved=0CC4Q6wEwAA#v=onepage&q=Computer%20Security%20Basics&f=false> [consult: 28/09/2013].

MANSFIELD-DEVINE, S. (ed.) (2013). Security review: the past year. In: Computer Fraud & Security, Vol. 2013, No. 1 (Jan). Cambridge (MA, USA): Elsevier Inc. p. 5–11. ISSN: 1361-3723.

MITRE CORPORATION (2014). Technical Guidance for Handling the New CVE-ID Syntax [online]. Common Vulnerabilities and Exposures: The Standard for Information Security Vulnerability Names (21/02/2014). Bedford (MA, USA): Mitre Corporation. <http://cve.mitre.org/cve/identifiers/tech-guidance.html> [consult: 17/03/2014]

OFUONYE, Ejike & MILLER, James. (2013). Securing web-clients with instrumented code and dynamic runtime monitoring [online]. In: Journal of Systems and Software, Vol. 86, No. 6, (Jun). New York (NY, USA): Elsevier Inc. p. 1689–1711. ISSN: 0164-1212 <doi:10.1016/j.jss.2013.02.047> <http://ac.els-cdn.com/S0164121213000514/1-s2.0-S0164121213000514-main.pdf?_tid=5d2474fc-a604-11e3-88b4-00000aab0f02&acdnat=1394202522_ad4081caa5c2a9c0915a07c738517b1f> [consult: 30/09/2013]

ORGANIZACIÓN DE LAS NACIONES UNIDAS PARA LA EDUCACIÓN, LA CIENCIA Y LA CULTURA, UNESCO (2013). ICT in Education [online]. Paris (France): Unesco. <http://en.unesco.org/themes/ict-education> [consult: 29/09/2013]

OWASP (2008). Category: Vulnerability/es: ¿Qué es una vulnerabilidad? [en línea]. Bel Air (MD, USA): OWASP Foundation. (09/10/2008). <https://www.owasp.org/index.php/Category:Vulnerability/es> [consulta: 04/08/2013].

OWASP (2013a). About The Open Web Application Security Project [online]. Bel Air (MD, USA): OWASP Foundation. <https://www.owasp.org/index.php/About_OWASP> [consult: 30/09/2013]

OWASP (2013b). Category: OWASP Top Ten Project [online]. Bel Air (MD, USA): OWASP Foundation. <https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project> [consult: 30/09/2013].

PETRESKA, Irina; TOMOVSKI, Igor; GUTIERREZ, Eugenio; KOCAREV, Ljupčo; BONO, Flavio & POLJANSEK, Karmen (2010). Application of modal analysis in assessing attack vulnerability of complex

networks [online]. In: Communications in Nonlinear Science and Numerical Simulation, Vol. 15, No. 4, (Apr). Cambridge (MA, USA): Elsevier Inc. p. 1008–1018. ISSN: 1007-5704 <doi:10.1016/j.cnsns.2009.05.002> <http://ac.els-cdn.com/S1007570409002639/1-s2.0-S1007570409002639-main.pdf?_tid=5b7f02c2-a607-11e3-ae05-00000aab0f27&acdnat=1394203808_284cc16c722f5c889da4c0b5671e19f1> [consult: 30/09/2013]

SALINI, P. & KANMANI, S. (2012). Security Requirements Engineering Process for Web Applications [online]. In: Procedia Engineering, Vol. 38, Cambridge (MA, USA): Elsevier. p. 2799–2807. ISSN: 1877-7058 <doi:10.1016/j.proeng.2012.06.328> <http://ac.els-cdn.com/S1877705812022412/1-s2.0-S1877705812022412-main.pdf?_tid=79e54dc4-a608-11e3-9ad2-00000aacb360&acdnat=1394204288_0bd438d942dbd3e68d5579e0431362e1> [consult: 28/09/2013].

SHAHMEHRI, Nahid; MAMMAR, Amel; MONTES DE OCA, Edgardo; BYERS, David; CAVALLI, Ana; ARDI, Shanai & JIMENEZ, Willy (2012). An advanced approach for modeling and detecting software vulnerabilities [online]. In: Information and Software Technology, Vol. 54, No. 9, (Sep). Newton (MA, USA): Elseiver Inc. p. 997–1013. ISSN: 0950-5849 <doi:10.1016/j.infsof.2012.03.004> <http://ac.els-cdn.com/S0950584912000535/1-s2.0-S0950584912000535-main.pdf?_tid=5ed4abe6-a609-11e3-8be3-00000aab0f6c&acdnat=1394204694_f87dd69d3ef0833cc01a41e49348a29b> [consult: 28/09/2013]

STEVANOVIC, Dusan.; AN, Aijun. & VLAJIC, Natalija. (2012). Feature evaluation for web crawler detection with data mining techniques [online]. En Expert Systems with Applications, Vol. 39, No. 10, (Aug). Tarrytown (NY, USA): Elseiver Inc. p. 8707–8717. ISSN: 0957-4174 <doi:10.1016/j.eswa.2012.01.210> <http://ac.els-cdn.com/S0957417412002382/1-s2.0-S0957417412002382-main.pdf?_tid=4aefb190-a60c-11e3-bfbb-00000aab0f6b&acdnat=1394205927_69720957887edc43b24b8ec13fa5c4f7> [consult: 30/09/2013]

TANENBAUM, Andrew & WETHERALL, David (2012). Redes de Computadoras. 5 ed. Naucalpan (Juárez, México): Pearson Educación. 729 p. ISBN: 978-607-32-0817-8