TELEMÁTICA
REVISTA DIGITAL de las Tecnologías de la Información y las Telecomunicaciones
Revista Telemática. Vol.20 No.1 , Enero-Marzo, 2021, p.1-11 ISSN 1729-3804

CONTROLES DE SEGURIDAD PARA SISTEMAS DE GESTIÓN DE CONTENIDOS BASADOS EN SOFTWARE LIBRE

Henry Raúl González Brito[1], Raydel Montesino Perurena1[2], Yeleny Zulueta Véliz [3]

[1, 2, 3]Universidad de las Ciencias Informáticas, UCI, Carretera a San Antonio de los Baños, Km 2 ½, reparto Torrens, municipio Boyeros, La Habana, Cuba.
[1]e-mail: henryraul@uci.cu
[2]e-mail: raydelmp@uci.cu
[3]e-mail: yeleny@uci.cu

RESUMEN

Los Sistemas de Gestión de Contenidos representan una tecnología Web muy utilizada en la actualidad, especialmente aquellos basados en Software Libre como WordPress, Drupal y Joomla. Debido a su amplia difusión, estos sistemas son constantemente atacados desde Internet, lo que ha significado también un incremento en los incidentes de ciberseguridad a nivel mundial en este campo, llegándose a cuantificar decenas de miles de portales web comprometidos diariamente cuando se descubren vulnerabilidades críticas de seguridad. Por ello, el propósito de la presente investigación es conceptualizar y estructurar un conjunto de controles de seguridad que puedan ser aplicados de forma sistémica a las instalaciones de portales web basadas en WordPress, Drupal y Joomla y formalizar de este modo a los sistemas de gestión de contenidos como una capa de seguridad en la estrategia de defensa en profundidad de la infraestructura computacional. La investigación conceptualizó un total de 29 controles de seguridad, distribuido en siete grupos diferentes. El 79% de estos controles pueden ser aplicados a otros CMS y el resto pueden ser tomados como base para la determinación de controles específicos. El empleo de estos controles de seguridad permite aumentar los niveles razonables de seguridad en esta tecnología, además de garantizar una mejor gestión de los procesos de seguridad informática en las entidades.

PALABRAS CLAVES: Seguridad Web, CMS, Controles de Seguridad, WordPress, Drupal, Joomla.

SECURITY CONTROLS FOR CONTENT MANAGEMENT SYSTEMS BASED ON FREE SOFTWARE

ABSTRACT

The Content Management Systems represent a Web technology widely used today, especially those based on Free Software such as WordPress, Drupal and Joomla. Due to their wide dissemination, these systems are constantly attacked from the Internet, which has also meant an increase in cybersecurity incidents worldwide in this field, reaching tens of thousands of compromised web portals daily when critical vulnerabilities of security. Therefore, the purpose of the present investigation was to conceptualize and structure a set of security controls that could be applied systemically to the installations of web portals based on WordPress, Drupal and Joomla and thus formalize the CMS as a layer of security in the strategy of defense in depth of the computer infrastructure. The research conceptualized a total of 29 security controls, distributed in seven different groups. 79% of these controls can be applied to other CMS and the rest can be taken as the basis for the determination of specific controls. The use of these security controls allows to increase the reasonable levels of security in this technology, in addition to guaranteeing a better management of the computer security processes in the entities. As future work, studies focused on the design of metrics that can weigh the contribution of each control to security, and the development of automated mechanisms for its audit are proposed.

INDEX TERMS: Web Security, CMS, Security Controls, WordPress, Drupal, Joomla.

Manuscrito recibido: 2020-09-03, aceptado: 2020-11-17
Sitio web:http://revistatelematica.cujae.edu.cu/index.php/tele

1. INTRODUCCIÓN

Las aplicaciones web son la base para la informatización de la sociedad moderna. En la mayoría de los casos, la interrelación de personas y entidades se establece en el ciberespacio a través de ellas. En la actualidad no se concibe una organización, compañía o entidad de cualquier tipo que no tenga presencia en Internet mediante un portal web [1], [2]. Existen diversas tecnologías que facilitan la creación de portales web, dentro de las cuales se encuentran los Sistemas de Gestión de Contenidos o CMS (en inglés Content Management System) [3], [4]. Estos brindan un marco de trabajo basado en arquitecturas y componentes para extender y personalizar sus funcionalidades y comportamiento, lo que evita que las entidades tengan que contratar permanentemente equipos de programadores y puedan centrarse en su objeto social [5]-[7].

En la actualidad, los CMS representan una parte significativa de las tecnologías utilizadas en Internet. Por ejemplo, las estadísticas compiladas por la entidad W3Techs revela que el 56.4% de los primeros diez millones de aplicaciones web más populares del ranking de Alexa usan algún tipo de CMS [8]. La empresa BuiltWith® también refleja que el 74.19% del primer millón de las aplicaciones web más populares del ranking de Alexa son CMS [9]. WordPress, Drupal y Joomla son además los CMS más utilizados según estos reportes como se observa en la Tabla 1. Otras fuentes de estadísticas de Internet corroboran estos datos [10], [11].

Tabla 1: Porcentaje de utilización de CMS respecto a las aplicaciones web que forman parte de los rankings Alexa.

CMS	Reporte W3Techs (Top 10 millones Alexa)	Reporte Built With® (Top 1 millón Alexa)
WordPress	34.7 %	35.99 %
Drupal	1.7 %	2.91 %
Joomla	2.7 %	1.83 %
Otros CMS	17.3 %	33.45 %
Total	56.4 %	74.9 %

El crecimiento del uso de los CMS ha significado también un incremento en los incidentes de ciberseguridad a nivel mundial que involucran esta tecnología [4], [12]-[16], con el agravante que cuando se descubre una vulnerabilidad de seguridad [17], [18]. Estos son masivamente explotados a través de las botnets controladas por grupos de ciberdelincuentes [19]-[21], llegándose a cuantificar decenas de miles de portales web comprometidos diariamente. Ejemplo de ellos son los casos que tuvieron lugar con la vulnerabilidad de WordPress CVE-2017-5487, las vulnerabilidades de Joomla CVE-2015-8562 y CVE-2017-8917 y las vulnerabilidades de Drupal CVE-2014-3704, CVE-2018-7600, CVE-2018-7600 y CVE-2018-7602.

La presencia de vulnerabilidades en el software no constituye el único reto de seguridad en los CMS. Un número significativo de incidentes [22]-[24] tienen lugar debido a las deficiencias en el establecimiento de controles de seguridad efectivos que garanticen una configuración y gestión adecuada de los portales web [13], [25]-[28]. Incluso actividades básicas como la actualización periódica de componentes son comúnmente ignoradas [29]-[32], lo que facilita las campañas de explotación de vulnerabilidades anteriormente mencionadas.

A diferencia de las vulnerabilidades de software, la solución de las vulnerabilidades de configuración y mantenimiento está en manos de las entidades y personal encargado de la gestión de los portales web basados en los CMS [33]. Aunque diversos investigadores han estudiado un grupo de cuestiones relacionadas con la seguridad de los CMS [34]-[40], hasta el momento no se ha podido encontrar, en la bibliografía consultada, una formalización de los controles de seguridad en estas tecnologías. Por ello, el propósito de la presente investigación fue conceptualizar y estructurar un conjunto de controles de seguridad que pudieran ser aplicados de forma sistémica a las instalaciones de portales web basadas en WordPress, Drupal y Joomla y formalizar de este modo al CMS como una capa de seguridad en la estrategia de defensa en profundidad de la infraestructura computacional.

2. CONTROLES DE SEGURIDAD PARA LOS SISTEMAS DE GESTIÓN DE CONTENIDOS

Materiales y Métodos

Para la realización de la investigación se emplearon los siguientes métodos de investigación:

- **Histórico-lógico**: Se utilizó para el estudio de la evolución de las medidas de seguridad en los CMS.
- **Análisis-Síntesis**: Se empleó para extraer las características principales y comparar las diferentes medidas internas para el fortalecimiento de la seguridad en instalaciones de CMS.

TELEMÁTICA
REVISTA DIGITAL de las Tecnologías de la Información y las Telecomunicaciones
Revista Telemática. Vol.20 No.1 , Enero- Marzo, 2021, p.1-11 ISSN 1729-3804

- **Experimentación**: Se utilizó para la realización de despliegues de pruebas y puesta a punto de los controles propuestos como parte de la investigación, tanto en entornos simulados como reales.

Estructuración de los controles de seguridad para los Sistemas de Gestión de Contenidos

Para realizar la propuesta de controles en los CMS WordPress, Drupal y Joomla se tuvieron en cuenta tres grupos de clasificación: Proceso, Servidor y Configuraciones del CMS. El primer grupo (Proceso [P]) está centrado en aquellos controles que tienen un peso mayor en proceso. Por ejemplo, todos los del grupo para mitigar la presencia de vulnerabilidades conocidas tienen esta característica. Su mayor dificultad radica en la dependencia casi absoluta del establecimiento de un sistema de trabajo continuo, lo que las hace vulnerables a situaciones asociadas a la naturaleza humana o su ausencia en los procedimientos de la administración de los servicios telemáticos.

El segundo grupo (Servidor [S]) está enfocado en los controles que tienen una fuerte dependencia de los servidores web, de bases de datos, así como del sistema de archivos del sistema operativo. Son controles que dependen de los permisos que tengan los administradores web, cuestión que normalmente es bastante restringida cuando se trata de servicios de hosting contratados a terceros.

En el tercer y último grupo (Configuraciones del CMS [C]) se encuentran los controles que representan opciones de configuración según las funcionalidades que brindan los CMS a través de los paneles de administración y archivos de configuración propios. Su aplicación depende de dos factores principales, el primero está relacionado con la necesidad o no de su utilización, según la disponibilidad de componentes que lo requieran y la segunda cuestión está en función de si el CMS, según su diseño técnico, contienen o no debilidades que pueden estar presente en otros CMS.

Existen controles comunes que pueden ser aplicados a todos los CMS y otros de tipo más específico debido a las diferencias en las funcionalidades de configuración y administración de estos como se muestra en la Fig. 1. En total se conceptualizaron un total de 29 controles de seguridad distribuidos en siete grupos diferentes. A continuación, se presentará una breve explicación de cada grupo, junto con los controles que lo componen y el objetivo que persigue su aplicación.

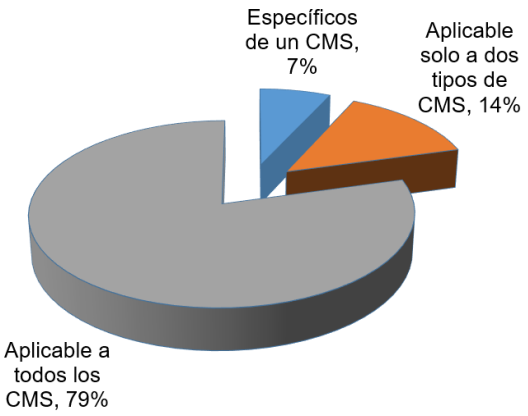

Figura 1: Distribución de los controles aplicables a los CMS estudiados.

Controles para mitigar la presencia de vulnerabilidades conocidas

Como se mencionaba anteriormente, la aplicación tardía de parches de seguridad representa el principal problema que afecta a los CMS en la actualidad. Un alto porciento de los portales web son comprometido anualmente por la presencia de vulnerabilidades en el núcleo, complementos y temas que podían haber sido resueltas si se hubieran aplicado los parches y actualizaciones recomendadas. Por tanto, los controles que se proponen tienen como objetivo mitigar la presencia de vulnerabilidades en la instalación y mantenimiento del CMS:

1) **Mantener el núcleo, complementos y temas actualizados**: Eliminar las vulnerabilidades conocidas en el código fuente mediante la aplicación de los parches de seguridad recomendados.
2) **Obtener componentes de fuentes confiables**: Evitar la instalación de componentes que incrementen los riesgos de seguridad del portal, mediante el análisis de las fuentes de obtención y reputación de los mismos.
3) **Instalar solamente complementos y temas imprescindibles**: Disminuir la superficie de ataque o exposición del portal mediante la instalación de un menor número de componentes.
4) **Desarrollar un programa de vigilancia tecnológica**: Aplicar de forma oportuna medidas de seguridad mediante el seguimiento de alertas y recomendaciones periódicas emitidas por entidades especializadas.

Controles para dificultar la identificación de la tecnología base

Si se les dificulta a los adversarios la identificación de la tecnología base utilizada en el CMS, será más difícil que a través de pocas peticiones HTTP obtengan datos suficientes para seleccionar los códigos dañinos que pudieran tener mayor efectividad en un intento de intrusión. Esto los obligará a realizar un proceso de interacción con el portal más intenso, lo que aumentará las posibilidades de que se disparen las alertas y acciones de los mecanismos de seguridad (ej. sistemas de detección y prevención de intrusiones y cortafuegos) si estos están debidamente configurados y son monitoreados por personal especializado.

Por tanto, una capa de protección básica de cualquier aplicación web consiste en suprimir, tanto como sea posible, los datos y patrones que delaten la tecnología instalada. Para ello se proponen los siguientes controles:

5) **Gestionar el archivo robots.txt**: Garantizar que los datos del archivo robots.txt no comprometan la seguridad, mediante la aplicación de configuraciones efectivas alineadas a las características del portal.
6) **Eliminar los archivos residuales del proceso de instalación**: Impedir la exposición de información sobre la tecnología base mediante la eliminación de archivos residuales del proceso de instalación.
7) **Suprimir los metadatos que exponen la tecnología base:** Impedir la exposición de información sobre la tecnología base mediante la eliminación de los metadatos correspondientes en los encabezados HTTP y recursos HTML.
8) **Borrar los datos del tema empleado**: Evitar el conocimiento de la versión del tema empleado mediante la eliminación de los datos de la firma de autoría.
9) **Limitar la publicación de mensajes de errores**: Impedir la exposición de información sensible mediante la gestión de los mensajes de error del portal.
10) **Evitar la navegación de directorios**: Impedir la exposición de información interna del portal mediante el bloqueo de la indexación de los directorios.

Controles para la gestión del sistema de archivos

El sistema de archivos juega un papel esencial en la protección de un CMS porque en él se despliegan los diversos recursos y código fuente que conforman el portal web. Los adversarios conocen la estructura de directorios, su función y ubicación de los archivos de interés y por ello tratan de manipularlos a través de la explotación de vulnerabilidades de tipo directorio transversal, inclusión local de archivos, inyección de shellcodes o comandos del sistema operativo, por solo citar algunos ejemplos.

Si los permisos del sistema de archivos en su interrelación con el portal y el servidor web se gestionan adecuadamente, puede evitarse incluso el escalamiento de una intrusión sí, por ejemplo, un malware fuera inyectado en un directorio sin permisos para la ejecución de script, lo que interrumpiría el proceso del ataque en ese punto. Es necesario tener en cuenta que los CMS explícitamente no traen opciones de configuración para gestionar el sistema de archivos, por tanto, esto tiene que ejecutarse a nivel del sistema operativo, sin embargo, por la importancia que esto reviste, se proponen dos controles imprescindibles que deben aplicarse en esta capa de defensa:

11) **Cambiar los permisos de directorios y archivos**: Dificultar la manipulación de directorios y recursos, mediante la asignación de permisos en el sistema de archivos ajustados a las funciones que desempeñan.
12) **Cambiar la ubicación de archivos de configuración**: Impedir el acceso a los archivos de configuración, mediante su traslado a directorios distintos a los asignados por defecto.

Controles para la gestión de credenciales de usuarios

Las credenciales de usuario (comúnmente conocidas también por cuentas de usuarios) permiten hacer uso de las funcionalidades del portal. Por esta razón, los adversarios tratan de obtener continuamente acceso a credenciales de usuarios válidas que les faciliten aprovecharse de las opciones habilitadas con mayores niveles de privilegios. Las

Revista Telemática. Vol.20 No.1 , Enero- Marzo, 2021, p.1-11 ISSN 1729-3804

credenciales que tengan asignadas roles de administración pudieran parecer más importantes y valiosas que otras, sin embargo, es necesario destacar que un número de vulnerabilidades en los CMS necesitan credenciales válidas, sin importar el rol asignado, para que puedan ser explotadas.

Se hace necesario, por tanto, prestar especial atención al modo en que se gestionan las credenciales de usuarios para conformar una sólida defensa antes los intentos de ataques y vulnerabilidades asociadas. Para ello se proponen los siguientes controles:

13) **Evitar la enumeración de credenciales de usuarios**: Disminuir las posibilidades de éxito de los ataques de diccionario, mediante el bloqueo de la automatización del proceso de recolección de credenciales de usuarios.
14) **Cambiar las credenciales del administrador principal por defecto**: Dificultar los ataques de diccionario contra las credenciales de administración mediante la sustitución por otras de difícil predicción.
15) **Utilizar contraseñas robustas en las credenciales de administración**: Dificultar los ataques de diccionario contra las credenciales de administración mediante el uso de contraseñas robustas.
16) **Deshabilitar el registro de usuarios si no va ser utilizado**: Disminuir riesgos injustificados de explotación de vulnerabilidades que requieran credenciales de usuarios mediante el bloqueo del registro de usuarios en portales que no lo necesiten.
17) **Restringir el acceso a funcionalidades de autenticación**: Reducir los riesgos de accesos no autorizados a las funcionalidades de autenticación mediante la restricción de su exposición solamente para el conjunto de usuarios que lo necesiten.
18) **Modificar la clave sal de las funciones criptográficas**: Impedir que ataques criptográficos contra las credenciales de usuario tenga éxito mediante la modificación de las claves sal por defecto.

Controles para la gestión de peticiones HTTP

Los CMS, al igual que otras aplicaciones web, pueden ser afectados por la manipulación de operaciones del protocolo HTTP. Diferentes tipos de ataques, como los secuestros de clic (clickjacking), intercepción de tráfico a través de conexiones sin cifrar (sniffer), black hat SEO, explotación de métodos HTTP habilitados como TRACE o sin protección como PUT y DELETE o los clásicos ataques CSRF (Cross-Site Request Forgery o falsificación de petición en sitios cruzados) y XSS (Cross-Site Scripting o secuencia de comandos en sitios cruzados), por solo citar algunos ejemplos, se aprovechan de implementaciones y gestiones inadecuadas de las peticiones HTTP en el servidor web [26]-[38].

Aunque la mayoría de las configuraciones de seguridad deben hacerse a nivel del servidor web, involucrando no solamente al CMS en cuestión que se pretenda proteger, sino a todas las aplicaciones web del servicio de hosting, es importante establecer controles elementales que contribuirán a mejorar la posición de defensa en esta área. Para ello se proponen los siguientes controles:

19) **Incorporar encabezados de respuesta HTTP de seguridad**: Reducir el riesgo de manipulación y mal uso de transacciones HTTP mediante la inclusión de campos de encabezado de seguridad en las respuestas del servidor web.
20) **Procesar los caracteres especiales**: Evitar la inyección de código dañino ofuscado mediante el filtrado de secuencias de escape y codificaciones especiales.
21) **Supresión lógica de funciones XML-RPC**: Evitar la explotación del API XML-RPC mediante la supresión de funciones vulnerables.

Controles para la interacción con bases de datos

Los CMS almacenan los contenidos del portal en Sistemas de Gestión de Bases de Datos (SGBD) o las referencias a estos cuando se trata de recursos de videos, sonido, archivos u otros de mayor tamaño. Aunque los SGBD pueden estar en el mismo sistema operativo que el servidor web, es una práctica generalizada situarlos en un servidor independiente para que el procesamiento y carga de trabajo se distribuya mejor a lo largo de la infraestructura computacional.

La modificación del contenido de tablas específicas de la base de datos puede significar, por ejemplo, la adición de nuevos usuarios con permisos elevados de administración o simplemente la suplantación de la información

previamente publicada. Por ello deben establecerse controles de seguridad independientes y específicos para los servidores de SGBD, evitando que una intrusión directa en estos, afecte al portal.

Para garantizar la interacción segura entre el servidor web y el servidor del SGBD deben aplicarse mecanismos de protección robustos que van más allá del ámbito de las configuraciones que proveen los CMS, sin embargo, durante el proceso de instalación, sí es necesario tener en cuenta dos controles básicos en esta área:

22) **Modificar el prefijo de las tablas de la base de datos**: Evadir los intentos de ataques de inyección de código SQL mediante la modificación de los prefijos de las tablas de la base de datos.
23) **Utilizar contraseñas robustas para el acceso a las bases de datos**: Dificultar los intentos de ataque de diccionario contra el servidor del SGBD mediante el uso de contraseñas robustas de difícil predicción.

Controles para asegurar las operaciones de administración

Las aplicaciones web y sobre todo los CMS no son sistemas estáticos, continuamente son actualizados con nuevos elementos mediante la acción humana. Este proceso también presenta debilidades que si no son adecuadamente gestionadas pueden llevar al traste toda la actividad de configuración de seguridad y fortificación que se haya alcanzado en la infraestructura computacional. Es por ello que siempre deben analizarse los procesos organizacionales que inciden sobre el portal, así como los posibles riesgos de seguridad que pueden estar presente en la interacción y administración de este. Posteriormente esto debe convertirse en acciones que eleven el nivel de concienciación del personal, minimizando de este modo la ocurrencia de incidentes de seguridad informática por causa de una formación deficiente o el desconocimiento de medidas básicas de seguridad.

.

Para evitarlo se proponen los siguientes controles:

24) **Restringir el acceso al panel de administración**: Dificultar a los atacantes la explotación de las funcionalidades de administración del portal mediante la restricción de su acceso mediante cortafuegos.
25) **Cifrar las conexiones de acceso a los paneles de administración**: Evitar la intercepción de las operaciones de administración mediante el uso de canales cifrados.
26) **Proteger las estaciones de trabajos y mecanismos de control de versiones de los administradores**: Mitigar los riesgos de seguridad en el proceso de gestión de contenidos del portal mediante la fortificación de la infraestructura de desarrollo.
27) **Utilizar navegadores independientes para la administración**: Evitar que ataques del lado del navegador afecten al portal mediante el uso de un navegador independiente para ejecutar actividades de gestión de contenidos.
28) **Desarrollar un programa de copias de seguridad**: Evitar que la ocurrencia de un incidente de seguridad pueda causar la pérdida parcial o total del portal mediante la realización de copias de seguridad periódicas.
29) **Almacenar periódicamente los registros de acceso del portal en una ubicación segura**: Contribuir a la solución rápida de incidentes de seguridad mediante el análisis de los registros de acceso del portal.

3. ANÁLISIS DE APLICACIÓN DE LOS CONTROLES PROPUESTOS

Los controles propuestos son representados en la Tabla 2, utilizando la codificación utilizada para su enumeración [1]-[29], junto con las siglas [W: WordPress, D: Drupal, J: Joomla] de los CMS. En los casos que el control no se aplica se utilizó el símbolo [-] en la celda de intercepción. Un análisis cuantitativo inicial permite comprobar que pueden aplicarse todos los controles propuestos en WordPress, hasta un 90% en Drupal y hasta un 83% en Joomla (Tabla 2).

Tabla 2: Clasificación de los controles propuestos.

TELEMÁTICA
REVISTA DIGITAL de las Tecnologías de la Información y las Telecomunicaciones
Revista Telemática. Vol.20 No.1 , Enero- Marzo, 2021, p.1-11 ISSN 1729-3804

Grupo	Control	W	D	J
Controles para mitigar la presencia de vulnerabilidades conocidas	1	P	P	P
	2	P	P	P
	3	P	P	P
	4	P	P	P
Controles para dificultar el conocimiento de la tecnología base	5	C	C	C
	6	C	C	C
	7	C	C	C
	8	C	-	C
	9	C	C	C
	10	S	S	S
Controles para la gestión del sistema de archivos	11	S	S	S
	12	S	S	-
Controles para la gestión de credenciales de usuarios	13	C	-	-
	14	C	C	-
	15	P	P	P
	16	C	C	C
	17	S	S	-
	18	C	C	-
Controles para la gestión de peticiones HTTP	19	S	S	S
	20	C	C	C
	21	C	-	-
Controles para la interacción con bases de datos	22	S	S	-
	23	P	P	P
Controles para asegurar las operaciones de administración	24	S	S	S
	25	S	S	S
	26	P	P	P
	27	P	P	P
	28	P	P	P
	29	P	P	P

Aunque indudablemente actúan diversos factores que provocan la existencia de las vulnerabilidades de seguridad a nivel de código, resulta importante señalar como el orden de completitud mostrado en la Fig. 2. es proporcional con los reportes anteriormente referenciados de Sucuri sobre las causas que provocan los incidentes de ciberseguridad en estas tecnologías. Por tanto, ante un mayor número de controles de seguridad aplicables, menor es la probabilidad de que un intento de ataque tenga éxito contra el CMS.

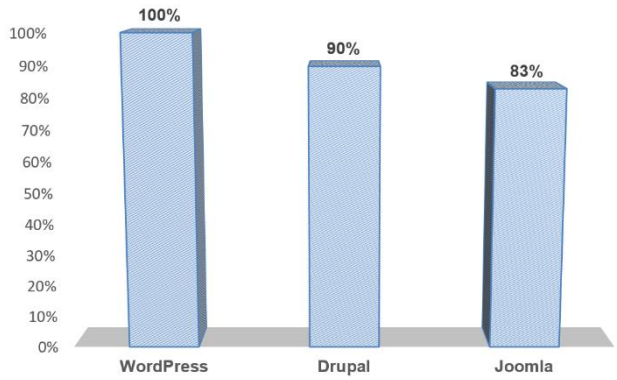

Figura 2: Cuantificación de la aplicación de los controles a los CMS.

El alcance de los controles presenta mayor uniformidad en el grupo de procesos. Lo que demuestra su aplicabilidad en otros tipos de CMS, como puede apreciarse en la Fig. 3. Resulta interesante además apreciar como existe mayor diferencia en el grupo de controles a nivel de servidor, lo que ilustra la interrelación operacional y dependencia de estos componentes desde el punto de vista de la protección proactiva. Estos controles fueron aplicados en diferentes entornos de evaluación, usando tanto los paneles de administración propios del CMS como la modificación de archivos de configuración en el servidor web a través de editores preinstalados como nano en los sistemas operativos GNU/Linux.

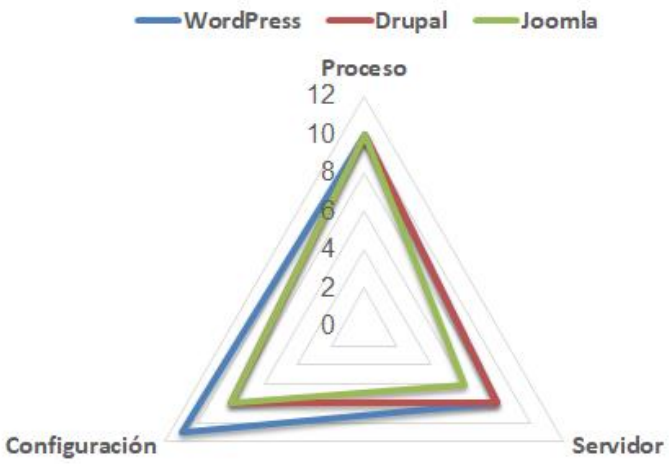

Figura 3: Controles vs grupos y CMS.

4. CONCLUSIONES

La investigación conceptualizó un total de 29 controles de seguridad para los CMS WordPress, Drupal y Joomla, distribuido en siete grupos diferente. El 79% de estos controles pueden ser aplicados a otros CMS y el resto pueden ser tomados como base para la determinación de controles específicos. El empleo de los controles en los CMS permite aumentar los niveles razonables de seguridad en estas tecnologías, además de garantizar una mejor gestión de los procesos de seguridad informática en las entidades. Como trabajo futuro se proponen estudios enfocados en el diseño de un conjunto de métricas que puedan ponderar la contribución de cada control a la seguridad del CMS, así como el desarrollo de mecanismos automatizados para su auditoría.

RECONOCIMIENTOS

Esta investigación ha sido llevada a cabo en el marco del proyecto “Metodología Ágil para pruebas de penetración en aplicaciones web (MAPPAW)” el cual forma parte del Programa de Prioridad Nacional de Ciencia, Tecnología e Innovación “Informatización de la Sociedad”. Los autores también desean agradecer a los ingenieros Leonardo Aguilera Blanco y Leobel Rodríguez Chang por la aplicación práctica de estos controles.

REFERENCIAS

[1] F. L. Almeida, "Concept and dimensions of web 4.0," *International Journal Of Computers Technology,* vol. 16, no. 7, pp. 7040-7046, 2017, doi: 10.24297/ijct.v16i7.6446.

[2] A. Mendoza, P. Chinprutthiwong, and G. Gu, "Uncovering HTTP Header Inconsistencies and the Impact on Desktop/Mobile Websites," in *Proceedings of the 2018 World Wide Web Conference*, Lyon, France, April 2018: International World Wide Web Conferences Steering Committee, pp. 247-256, doi: 10.1145/3178876.3186091.

[3] T. Canavan, *CMS Security Handbook: The Comprehensive Guide for WordPress, Joomla, Drupal, and Plone*. Indianapolis, Indiana: John Wiley and Sons, 2011.

[4] J.-M. Martinez-Caro, A.-J. Aledo-Hernandez, A. Guillen-Perez, R. Sanchez-Iborra, and M.-D. Cano, "A Comparative Study of Web Content Management Systems," *Information*, vol. 9, no. 2, p. 27, 2018, doi: 10.3390/info9020027.

[5] S. Barnes, S. Goodwin, and R. Vidgen, "Web content management," in *14th Bled Electronic Commerce Conference*, Bled, Slovenia, June 25 - 26 2001, p. 47.

[6] F. Trias, V. de Castro, M. Lopez-Sanz, and E. Marcos, "Migrating traditional Web applications to CMS-based Web applications," *Electronic Notes in Theoretical Computer Science*, vol. 314, pp. 23-44, 2015, doi: 10.1016/j.entcs.2015.05.003.

[7] K. Vlaanderen, F. Valverde, and O. Pastor, "Model-Driven Web Engineering in the CMS Domain: A Preliminary Research Applying SME," Berlin, Heidelberg, June 12-16 2009: Springer Berlin Heidelberg, in Enterprise Information Systems, pp. 226-237, doi: 10.1007/978-3-642-00670-8_17.

[8] W3Techs. "Usage of Content Management Systems for Websites." https://w3techs.com/technologies/overview/content_management/all (accessed 21/10/2019, 2019).

[9] BuiltWith®. "CMS Usage Distribution in the Top 1 Million Sites." https://trends.builtwith.com/cms (accessed 21/10/2019, 2019).

[10] WhatCMS.org. "Technology Reports." https://whatcms.org/Tech_Reports (accessed 21/10/2019, 2019).

[11] Wappalyzer. "CMS Market leaders." https://www.wappalyzer.com/categories/cms (accessed 21/10/2019, 2019).

[12] O. Ojagbule, H. Wimmer, and R. J. Haddad, "Vulnerability Analysis of Content Management Systems to SQL Injection Using SQLMAP," in *SoutheastCon 2018*, 19-22 April 2018: IEEE, pp. 1-7, doi: 10.1109/SECON.2018.8479130.

[13] M. Laverdière and E. Merlo, "Detection of protection-impacting changes during software evolution," in *2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 20-23 March 2018: IEEE, pp. 434-444, doi: 10.1109/SANER.2018.8330230.

[14] G. Petrică, S. Axinte, I. C. Bacivarov, M. Firoiu, and I. Mihai, "Studying cyber security threats to web platforms using attack tree diagrams," in *2017 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 29 June-1 July 2017: IEEE, pp. 1-6, doi: 10.1109/ECAI.2017.8166456.

[15] H. u. Rehman, M. Nazir, and K. Mustafa, "Security of Web Application: State of the Art," in *Information, Communication and Computing Technology. ICICCT 2017*, Singapore, May 13 2017: Springer Singapore, in Information, Communication and Computing Technology, pp. 168-180, doi: 10.1007/978-981-10-6544-6_17.

[16] H. R. González Brito and R. Montesino Perurena, "Capacidades de las metodologías de pruebas de penetración para detectar vulnerabilidades frecuentes en aplicaciones web," *Revista Cubana de Ciencias Informáticas*, vol. 12, no. 4, pp. 52-65, 2018.

[17] G. R. Perez, G. Robles, A. Serebrenik, A. Zaidman, D. German, and J. M. González-Barahona, "How bugs are born: a model to identify how bugs are introduced in software components," *Empirical Software Engineering*, pp. 1294–1340, 2019, doi: 10.1007/s10664-019-09781-y.

[18] R. G. Kula, D. M. German, A. Ouni, T. Ishio, and K. Inoue, "Do developers update their library dependencies?," *Empirical Software Engineering*, vol. 23, no. 1, pp. 384-417, 2018, doi: 10.1007/s10664-017-9521-5.

[19] T. Guarda, S. Bustos, W. Torres, and F. Villao, "Botnets the Cat-Mouse Hunting," in *Digital Science. DSIC18 2018*, Cham, 19-21 October 2019: Springer International Publishing, in Digital Science, pp. 408-416, doi: 10.1007/978-3-030-02351-5_46.

[20] A. K. Sood, S. Zeadally, and R. Bansal, "Cybercrime at a Scale: A Practical Study of Deployments of HTTP-Based Botnet Command and Control Panels," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 22-28, 2017, doi: 10.1109/MCOM.2017.1600969.

[21] H. R. González Brito, "Estudio de patrones de intentos de ciberataques asociados a las vulnerabilidades del complemento RevSlider," *Revista Cubana de Ciencias Informáticas*, vol. 12, no. 1, pp. 43-57, 2018.

[22] Sucuri, "Hacked Website Report 2018," Sucuri Inc, California, Estados Unidos, 2019.

[23] A. Lemay, J. Calvet, F. Menet, and J. M. Fernandez, "Survey of publicly available reports on advanced persistent threat actors," *Computers & Security*, vol. 72, pp. 26-59, 2018, doi: 10.1016/j.cose.2017.08.005.

[24] Sucuri, "Website Hacked Trend Report 2016 - Q3," Sucuri Inc, California, Estados Unidos, 2016.

[25] U. Lapteva and O. Kuzyakov, "Rationale for Principles of Developing Control and Protection of Web Content Using CMS Drupal," in *2018 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)*, October 2018: IEEE, pp. 1-6, doi: 10.1109/FarEastCon.2018.8602487.

[26] J. J. Singh, H. Samuel, and P. Zavarsky, "Impact of Paranoia Levels on the Effectiveness of the ModSecurity Web Application Firewall," in *2018 1st International Conference on Data Intelligence and Security (ICDIS)*, April 2018: IEEE, pp. 141-144, doi: 10.1109/ICDIS.2018.00030.

[27] I.-C. Cernica, N. Popescu, and B. Tiganoaia, "Security Evaluation of Wordpress Backup Plugins," in *2019 22nd International Conference on Control Systems and Computer Science (CSCS)*, 2019: IEEE, pp. 312-316, doi: 10.1109/CSCS.2019.00056.

[28] E. S. Sagatov, D. A. Shkirdov, and A. M. Sukhov, "Analysis of Network Threats Based on Data from Server-Traps," in *2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 24-26 June 2019: IEEE, pp. 1-5, doi: 10.1109/NTMS.2019.8763847.

[29] B. Carlson, K. Leach, D. Marinov, M. Nagappan, and A. Prakash, "Open Source Vulnerability Notification," in *IFIP International Conference on Open Source Systems*, 2019: Springer, pp. 12-23, doi: 10.1007/978-3-030-20883-7_2.

[30] F. Li and V. Paxson, "A Large-Scale Empirical Study of Security Patches," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, Dallas, Texas, USA, October 2017, 3134072: ACM, pp. 2201-2215, doi: 10.1145/3133956.3134072.

[31] G. Betarte, E. Giménez, R. Martínez, and Á. Pardo, "Machine learning-assisted virtual patching of web applications," *arXiv preprint arXiv:1803.05529*, 2018.

[32] D. Borbor, L. Wang, S. Jajodia, and A. Singhal, "Securing Networks Against Unpatchable and Unknown Vulnerabilities Using Heterogeneous Hardening Options," in *Data and Applications Security and Privacy XXXI. DBSec 2017*, Cham, July 2017: Springer International Publishing, in Data and Applications Security and Privacy XXXI, pp. 509-528, doi: 10.1007/978-3-319-61176-1_28.

[33] H. R. González Brito, "Configuraciones internas para el fortalecimiento de la seguridad en WordPress," in *VIII Congreso Internacional de Tecnologías y Contenidos Multimedia. INFORMÁTICA 2018*, Havana, March 2018, pp. 1-9.

[34] P. So, "Authenticating Requests in Drupal 8," in *Decoupled Drupal in Practice: Architect and Implement Decoupled Drupal Architectures Across the Stack*. Berkeley, CA: Apress, 2018, pp. 113-140.

[35] M. Laverdière and E. Merlo, "Classification and Distribution of RBAC Privilege Protection Changes in Wordpress Evolution (Short Paper)," in *2017 15th Annual Conference on Privacy, Security and Trust (PST)*, 28-30 August 2017: IEEE, pp. 349-3495, doi: 10.1109/PST.2017.00048.

[36] N. Setiani and T. Dirgahayu, "Clustering technique for information requirement prioritization in specific CMSs," in *2016 International Conference on Data and Software Engineering (ICoDSE)*, 26-27 October 2016: IEEE, pp. 1-6, doi: 10.1109/ICODSE.2016.7936107.

[37] A. K. Kyaw, F. Sioquim, and J. Joseph, "Dictionary attack on Wordpress: Security and forensic analysis," in *2015 Second International Conference on Information Security and Cyber Forensics (InfoSec)*, 15-17 November 2015: IEEE, pp. 158-164, doi: 10.1109/InfoSec.2015.7435522.

[38] S. Kratov, "On providing the fault-tolerant operation of information systems based on open content management systems," in *Journal of Physics: Conference Series*, May 2018, vol. 944, no. 1: IOP Publishing, p. 012067, doi: 10.1088/1742-6596/944/1/012067.

[39] T. Koskinen, P. Ihantola, and V. Karavirta, "Quality of WordPress plug-ins: an overview of security and user ratings," in *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, 3-5 September 2012: IEEE, pp. 834-837, doi: 10.1109/SocialCom-PASSAT.2012.31.

[40] Y. Fei, J. Ning, and W. Jiang, "A quantifiable Attack-Defense Trees model for APT attack," in *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, October 2018: IEEE, pp. 2303-2306, doi: 10.1109/IAEAC.2018.8577817.

SOBRE LOS AUTORES

M.Sc. Henry Raúl González Brito: Graduado de Ingeniero Informático por la Universidad de Camagüey y la CUJAE en el año 2005 y Máster en Gestión de Proyectos Informáticos en el 2012 por la Universidad de Ciencias Informáticas. Integra el claustro de varias maestrías impartiendo posgrados en la temática de Seguridad Informática. Actualmente es subdirector del Centro de Telemática (TLM) de la UCI y coordinador de la Especialidad de Posgrado en Seguridad Informática. Sus áreas de investigación están relacionadas con la seguridad en aplicaciones web y metodologías de pentesting. ORCID: 0000-0002-3226-9210

Dr.C. Raydel Montesino Perurena: Graduado de Ingeniero en Telecomunicaciones y Electrónica en el año 2003 en la Universidad Tecnológica de La Habana (CUJAE). Ocupó el cargo de Director de Seguridad Informática de la Universidad de las Ciencias Informáticas (UCI) en el período 2005 - 2012. Obtuvo el título de Doctor en Ciencias Técnicas en el año 2013. Actualmente es profesor, investigador y Vicerrector Primero de la UCI. Sus áreas de investigación están relacionadas con la gestión de la seguridad informática, específicamente en lo referente a estándares, métricas, automatización de controles y sistemas de gestión de información y eventos de seguridad (SIEM). ORCID: 0000-0003-4747-3166

Dr.C. Yeleny Zulueta Véliz: Graduada de Ingeniería Informática en el año 2004 en la Universidad de Camagüey y Máster en Gestión de Proyectos Informáticos en el año 2007 por la Universidad de Ciencias Informáticas. Obtuvo el título de Doctora en Tecnologías de la Información y las Comunicaciones por la Universidad de Granada, España, desde el 2014. Actualmente es Profesora Titular en la Facultad de Ciencias y Tecnologías Computacionales de la UCI. Sus intereses de investigación están relacionados con la toma de decisión difusa, modelos computacionales lingüísticos, computación con palabras y operadores de agregación. ORCID: 0000-0003-0253-528X

CONFLICTO DE INTERESES

No existe conflicto de intereses de los autores o de las instituciones a las cuales pertenece en relación al contenido del artículo aquí reflejado.

CONTRIBUCIONES DE LOS AUTORES

- **Henry Raúl González Brito:** Conceptualización, preparación, creación y desarrollo del artículo.
- **Raydel Montesino Perurena:** Revisión crítica de cada una de las versiones del borrador del artículo y aprobación de la versión final a publicar.
- **Yeleny Zulueta Véliz:** Sugerencias acertadas para la conformación de la versión final.