Conference Paper

Brute Force Attack To Exploit Vulnerabilities of Websites with CMS Content Management System

Ataque de fuerza bruta para aprovechar vulnerabilidades de sitios web con CMS sistema de gestión de contenidos

Mario Aquino-Cruz[1], Manuel Ibarra[1], Marleny Peralta-Ascue[1], Edwar Ilasaca-Cahuata[2], and Alejandro Apaza-Tarqui[3]

[1]Escuela Académico Profesional de Ing. Informática y Sistemas, Universidad Nacional Micaela Bastidas de Apurímac, Perú
[2]Departamento Académico de Ciencias Básicas, Universidad Nacional Micaela Bastidas de Apurímac, Perú
[3]Escuela Profesional de Estadística e Informática, Universidad Nacional del Altiplano Puno, Perú

Corresponding Author:
Mario Aquino-Cruz
mario.ac23@gmail.com

Received: 4 December 2018
Accepted: 5 December 2018
Published: 27 December 2018

Publishing services provided by Knowledge E

Selection and Peer-review under the responsibility of the SIIPRIN-CITEGC Conference Committee.

Abstract

The article shows the techniques of brute force attack to exploit vulnerabilities in websites made with CMS content managers taking into account ethical hacking; being that, currently the different public institutions, private and/or natural persons, have a growing demand for their products and/or services are available on the Internet. For example, public institutions in Peru are subject to Law N ° 27806, law on transparency and access to public information, to provide such information to citizens, for what is required. Another example corresponds to companies or private companies, for wanting to have positioning, competitive advantage, and approach to customers and also to web users. This involves hiring the services of people specialized in the development of websites that use CMS content managers, such as wordpress, joomla, drupal, etc., to implement these websites. The managers are prepared to help improve at the time of development and put them into production; however, security is often not taken into account. Among the consequences is the increasing number of computer attacks on these sites; therefore, a way to prevent it is by detecting the vulnerabilities that can be exploited, and thereby reduce the risks to which these websites are exposed to.

Resumen

El artículo muestra técnicas de ataque de fuerza bruta para aprovechar vulnerabilidades a sitios web realizados con gestores de contenido CMS tomando en cuenta el hackeo ético; siendo que, actualmente las diferentes instituciones públicas, privadas y/o personas naturales, tienen una demanda creciente a que sus productos y/o servicios se encuentren disponibles en internet. Por ejemplo, las instituciones públicas en el Perú están obligados según la Ley N° 27806 ley de transparencia y

OPEN ACCESS

acceso a la información pública, a brindar dicha información a los ciudadanos, para lo cual utilizan un sitio web. Otro ejemplo, corresponde a instituciones o empresas privadas, por querer tener posicionamiento, ventaja competitiva y acercamiento a sus clientes o consumidores también utilizan sitios web. Esto conlleva a contratar los servicios de personas especializadas en el desarrollo de sitios web utilizando bastante los gestores de contenido CMS como wordpress, joomla, drupal, etc, para implementar dichos sitios web. Los gestores mencionados son de gran ayuda para optimizar el tiempo al momento de desarrollar y ponerlos en producción; sin embargo, muchas veces no se toma en cuenta el tema seguridad. Entre las consecuencias es el número de ataques informáticos a estos sitios que se incrementa; Por tanto, una manera de prevenirlo es detectando las vulnerabilidades potenciales que pueden ser aprovechadas, y de esta manera disminuir los riesgos a los cuales se exponen estos sitios web.

Keywords: ethical hacking, vulnerabilities, websites, content managers, brute force

Palabras clave: Hackeo ético, vulnerabilidades, sitios web, gestores de contenido, fuerza bruta

1. Introducción

En la actualidad la brecha digital de internet va disminuyendo y cada vez más personas tienen acceso a este servicio, por ejemplo, desde un ordenador, portátil, tablet o algún dispositivo móvil.

Las tecnologías de la información y de la comunicación (TIC) son las herramientas principales de transformaciones sin precedentes en el mundo contemporáneo. En efecto, ninguna otra tecnología originó tan grandes cambios en la sociedad, en la cultura y en la economía. La humanidad viene alterando significativamente los modos de comunicar, de entretener, de trabajar, de negociar, de gobernar y de socializar, sobre la base de la difusión y uso de las TIC a escala global [1].

La seguridad en aplicaciones y sitios web es un aspecto importante para la protección de los activos. Estos activos pueden ser elementos como un servidor, información almacenada en la base de datos o hasta la reputación de la empresa o gobierno. Una aplicación web debe cumplir tres aspectos importantes para su buen funcionamiento:

integridad, disponibilidad y confiabilidad [2]. Estos tres aspectos hacen necesaria la utilización de herramientas para la detección de vulnerabilidades en el desarrollo de la aplicación y una mejor capacitación por parte de los programadores, para el desarrollo seguro de éstas. Las herramientas más populares para la detección de vulnerabilidades en aplicaciones web son los escáneres automáticos de vulnerabilidades web. Existen tanto comerciales como de software libre. Sin embargo estas herramientas, además de contar con muchas fortalezas, también cuentan con muchas limitaciones, debido principalmente a que las tasas de detección de vulnerabilidades puede variar [3].

Comprender los ataques predominantes, las fallas y los errores humanos que explotan los hackers para atacar los sitios web a las instituciones públicas, privadas o de personas naturales, puede disminuir en gran medida la probabilidad de convertirse en víctimas. Por ello es importante de que estés al tanto de estas amenazas porque incluso el desconocimiento te hace más vulnerable.

Estos ataques pueden tomar una amplia variedad de formas, tales como encontrar y explotar vulnerabilidades encontradas en el software de la víctima, estafas de correo electrónico diseñadas para engañar al usuario para que divulgue información crítica o inicie ataques como virus, ransomware y adquisiciones del sistema.

En este trabajo se pretende obtener datos acerca de la interacción de la herramienta Kali linux, para aprovechar las vulnerabilidades del sitio web y poder tener acceso y control.

Kali linux, y es, casi sin lugar a dudas, la suite más completa en de su clase, habiendo conseguido su objetivo de agrupar las herramientas de este campo en un solo sitio, facilitando su uso.

Kali linux está basada en Debian, y fue diseñada principalmente para la auditoria y seguridad informática en general. Actualmente es mantenida por Offensive Security Ltd. que desarrolló la distribución a partir de la re-escritura de BackTrack (también desarrollada por ellos), una distribución predecesora a Kali, y que gozó de mucho éxito entre las personas que se dedicaban a esta actividad [4].

Kali Linux trae preinstalados una gran cantidad de programas relacionados con el tema de la seguridad informática (más de 600 programas), siendo algunas de las más conocidas Nmap (un escáner de puertos), Wireshark (un sniffer), John the Ripper (Un crackeador de passwords) y la suite Aircrack-ng (Software para pruebas de seguridad en redes inalámbricas), además del inigualable Metasploit, la gran suite de explotación de vulnerabilidades [5].

El resto del trabajo se estructura como sigue: En la sección 2 se realiza una revisión de los principales trabajos relacionados con el análisis de escáneres de vulnerabilidades en aplicaciones web. La sección 3 describe el modelo propuesto en este trabajo, describiendo cada componente utilizado en los experimentos realizados. La sección 4 analiza los resultados obtenidos. Por último, las conclusiones obtenidas de este trabajo se presentan en la sección 5.

2. Teoría del dominio y trabajos previos

En la actualidad existen distintas herramientas de código abierto y otras de pago para poder realizar auditoria, ataques a los distintos servicios que se ofrecen en los sitios web. Se han desarrollado varios trabajos de investigación en donde se intenta evaluar las capacidades y limitaciones de las distintas herramientas de detección

En [6] se hace uso de versiones de aplicaciones como wordpress, drupal y phpBB2, en las cuales se han detectado vulnerabilidades. En estas aplicaciones puede darse el caso de haber otros tipos vulnerabilidades que no hayan sido detectadas o dadas a conocer. En otros trabajos como el realizado en la Universidad de Santa Bárbara [7] se desarrolla una aplicación propia llamada Wackopicko, la cual cuenta con diferentes retos para el rastreo y vulnerabilidades para la prueba de herramientas automatizadas, tanto comerciales como de código libre. En dicho trabajo consideran que una aplicación vulnerable debe contar con los siguientes requisitos: (1) debe tener claramente definidas las vulnerabilidades, (2) debe ser fácilmente personalizable para poder agregar nuevas vulnerabilidades y (3) debe representar a las aplicaciones actuales en términos de funcionalidad y tecnología. En [8] se hace uso de una aplicación con 5 tipos de vulnerabilidades desarrollada en drupal y al igual que en el trabajo mencionado anteriormente, se hacen algunas consideraciones para realizar una mejor evaluación de las herramientas. En [9], consideran el uso de niveles con diferentes mecanismos de defensa para incrementar la seguridad. Con estos niveles se puede evaluar la complejidad de los ataques generados para evadir los mecanismos de defensa que siguen siendo vulnerables. Como puede verse en los distintos trabajos realizados, las vulnerabilidades que han sido consideradas suelen ser muy pocas y las más comunes y conocidas, tales como “SQL injection”, “cross site scripting” (reflejado, almacenado y DOM), inclusión de archivos, gestión de sesiones, “cross site request forguery”, “path transversal”, restricción de acceso insuficientes, protección insuficiente en la capa de transporte, inyección de comandos, contraseñas débiles, manipulación de parámetros e inyección javascript, entre otras. Aunque en cada trabajo se menciona que las

herramientas no son capaces de detectar muchas de las vulnerabilidades implementadas, estos tampoco consideran el evaluar las aplicaciones con una lista de tipos de vulnerabilidades más extensa o bien definida. Considerando lo anterior y a que en cada trabajo se consideran distintos tipos de vulnerabilidades, se hace necesario contar con una lista definida de tipos de vulnerabilidades y con una o varias aplicaciones que contengan a estas [10]. Para realizar una mejor valoración de las herramientas y contar con aplicaciones que consideren la tecnología actual, realistas y con buena documentación.

En [9] se compararon 3 herramientas de análisis dinámico para analizar la precisión en la detección de vulnerabilidades de inyección de SQL. Se utilizaron 3 aplicaciones con vulnerabilidades documentadas y conocidas. Se configuró cada herramienta para realizar un análisis exclusivamente de vulnerabilidades de inyección de SQL a ciegas, almacenadas y reflejadas. Se consideró además la cantidad de tráfico generado por cada herramienta en cada análisis realizado sobre cada aplicación.

Se tomaron en cuenta los siguientes criterios para el trabajo de investigación los cuales se presentan en el siguiente capítulo.

3. Aplicaciones Web Vulnerables

Como primer objetivo del trabajo se ha seleccionado un conjunto lo suficientemente representativo de sitios web que existen actualmente. Para esto, se ha hecho una recopilación de 20 sitios web, entre instituciones públicas, privadas y de personas naturales, de diferentes tipos de dominio, como se muestra en la Tablas 1,2 y 3.

Tabla 1: Cantidad de sitio web con CMS.

Sitios web analizados	Tipo de dominio	Sitio Web con CMS	Sitio Web sin CMS
20	.com,.com.pe,.org,.net, edu.pe,.com.mx	12	8

Los sitios web que utilizan más gestores de contenido se identificaron en instituciones públicas y de personas naturales

Tabla 2: Cantidad de sitio web con CMS por tipo de institución.

Institución pública	Institución privada	Persona natural
4	1	7

Se verifico el gestor de contenido de los sitios web. Teniendo el siguiente resultado

Tabla 3: Cantidad de sitio web por CMS.

Wordpress	Joomla	Drupal
7	4	1

Se ha considerado aplicaciones a las que se le ha encontrado vulnerabilidad como wordpress. En estas aplicaciones han sido detectadas las vulnerabilidades que pueden ser explotadas.

4. Experimentos y Resultados

Se ha realizado un análisis de los sitios web con herramientas automatizadas de código abierto para la búsqueda de vulnerabilidades web muy conocidas. Éstas se encuentran preinstaladas en la distribución de Kali Linux, utilizada en este trabajo. Se ha realizado fases para un hackeo ético, 1 reconocimiento, 2 escaneo, 3 obtener el acceso, 4 mantener el acceso y 5 borrar huellas.

Se han utilizados estas herramientas para su análisis. Este análisis se hace con el fin de examinar y probar sus capacidades de detección de vulnerabilidades.

4.1. Procedimiento

Para poder realizar el análisis con las herramientas se realizó lo siguiente:

1. Se instalaron, configuraron y actualizaron las herramientas a utilizar.
2. Se utilizó nmap y zenmap con los siguientes parámetros -T4 -A -v -Pn seguido del dominio por ejemplo www.dominiodeprueba.com, de los 20 sitios web de la Tabla I, y se utilizó un intense scan, no ping.
3. Se obtuvieron el puerto que utilizan, el estado de los puertos si están abiertos o cerrados y los servicios que están corriendo. En la tabla 4, se muestra el resumen de los 7 sitios web en wordpress, donde S = SI y N = NO
4. Se verificó que existen xploits disponibles para varios servicios en el metasploit framework de kali linux, asi mismo en https://www.exploit-db.com, pero en este trabajo de investigación nos centramos en utilizar la herramienta wpscan, para obtener el listado de usuarios y luego realizar el ataque de fuerza bruta mediante el uso de diccionario de claves.
5. una de las técnicas que se utilizó, fue mediante el navegador o browser, agregando al dominio del sitio web lo siguiente wp-admin, por ejemplo

www.dominio.com/wp-admin. Dando como resultado el login de wordpress, de los sitios que utilizaban este gestor de contenidos, como se muestra en le Figura 1. También se podía verificar de manera gráfica viendo el código fuente del sitio web.

Tabla 4: Resumen de los puertos, estado y servicios de sitios en wordpress.

Puerto	estado	servicios	1	2	3	4	5	6	7
21	Open	ftp	S	S	S	N	S	N	S
22	Open	Ssh	S	N	S	S	S	S	S
25	Open	Smtp	S	S	S	S	S	S	S
26	Open	Rsftp	S	N	N	N	N	N	N
80	Open	http	S	S	S	S	S	S	S
110	Open	Pop3	S	S	S	S	S	S	S
113	Open	Ident	S	N	N	N	N	N	N
143	Open	Imap	S	S	N	S	N	S	N
443	Open	https	S	S	S	N	S	N	N
465	Open	Smtps	S	S	S	S	S	S	S
587	Open	Submission	S	N	N	N	N	N	N
993	Open	Imaps	S	N	N	S	S	N	N
995	Open	Pop3s	S	S	N	N	S	N	N
8080	Open	http-proxy	S	N	N	N	N	N	N
8443	Open	https-alt	S	N	N	N	N	N	N
3914	Open	Listcrt	S	N	N	N	N	N	N
16992	Open	Amt-soap-http	S	N	N	N	N	N	N
50000	Open	Ibm-db2	S	S	N	N	S	N	N
3336	Open	mysql	S	S	S	N	S	S	S

6. Luego de identificar los sitios que están realizados en wordpress, y que muestra la interfaz de logue, se procedió a realizar el escaneo y listado de usuarios de esos sitios web. Mediante la herramienta de kali linux, wpscan, utilizando el siguiente *codigo wpscan –url www.dominio.com –enumerate u.*

   Este proceso demora entre 20 segundos a 3 minutos dependiendo del sitio web, analiza el index.php, los templates, el Nameserver, la version de wordpress que utiliza, muestra el Enumerating plugins from passive detection, y finalmente enumera los nombres de usuario a través de Enumerating usernames, como se muestra en la Figura 2 y 3

7. al obtener los nombres de usuarios para el sitio web. Se procedió a hacer el ataque de fuerza bruta a través de un diccionario de claves. Este diccionario es una combinación de números y letras, Dicho diccionario es una archivo.txt, el cual se tiene

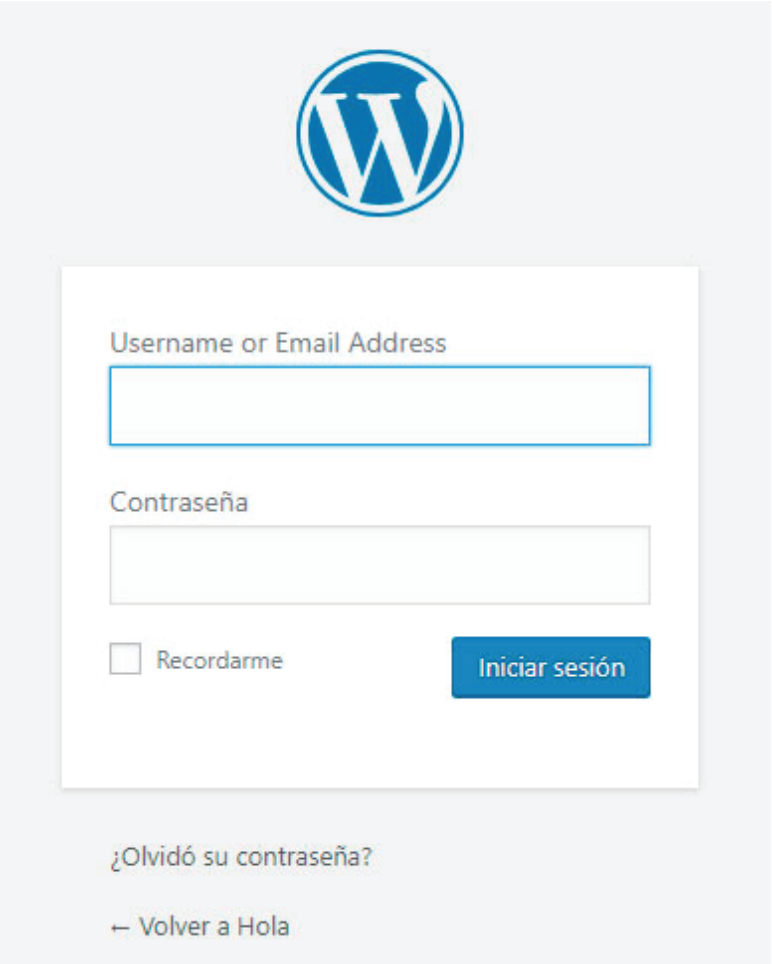

Figura 1: Login Wordpress.

que ubicar en el siguiente directorio para poder utilizarlo /usr/share/wordlists/. tal como se muestra en la Figura 4.

Para realizar el ataque de fuerza bruta, utilizando un diccionario, seguiremos utilizando wpscan y el codigo es el siguiente:

wpscan –url www.dominio.com –wordlists /usr/share/wordlists/diccionario.txt –username nombredeusuario –threads 10.

Donde resaltaremos 4 aspectos importantes del código 1 en –url debe ir el dominio del sitio web del que ya tenemos el usuario, 2 en – wordlists va la dirección donde se encuentra nuestro archivo de diccionario de claves a utilizar, 3 –username aquí se coloca el nombre de usuario que ya obtuvimos en el paso 6 ver Figura 2,4 y 5 en –threads, va la intensidad con la que se realizara el ataque de fuerza bruta para obtener la clave de acceso. El código utilizado es el siguiente pero por seguridad del sitio ponemos un ejemplo de dominio, y si utilizamos el usuario obtenido en los pasos

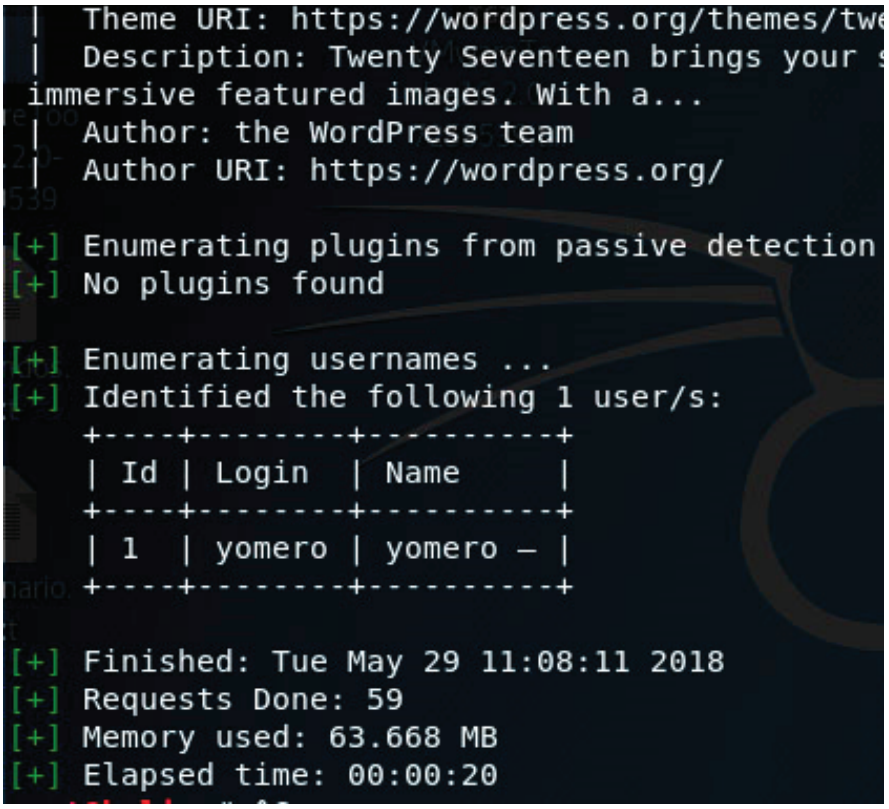

FIGURA 2: usernames del sitio web, con tipo de dominio.com.

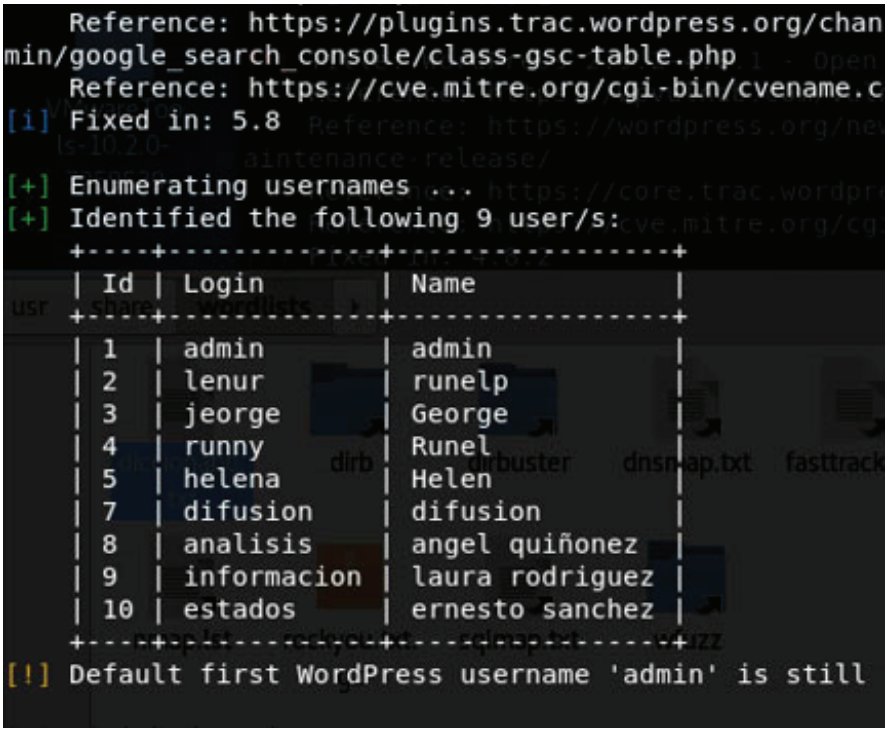

FIGURA 3: usernames de sitio web, con tipo de dominio.org.

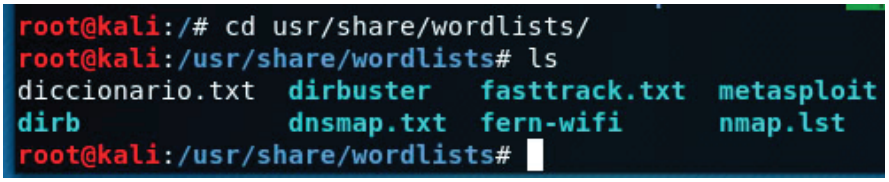

Figura 4: Ruta de ubicación de diccionario.txt.

anteriores, siendo el resultado la obtención de la clave de dicho usuario, el cual ya se puede acceder al administrador del sitio web. Ver Figura 5.

wpscan –url http://dominio.com--wordlist/usr/share/wordlists/diccionario.txt –username yomero –threads 10

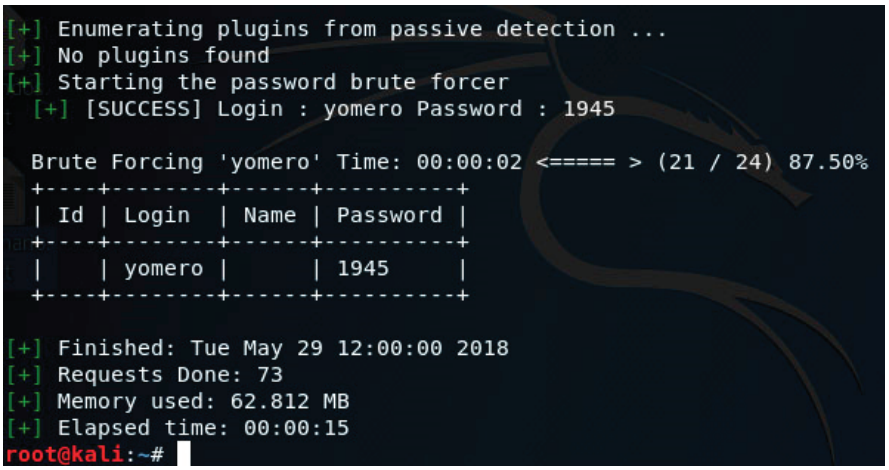

Figura 5: obtención de la clave del usuario.

9. De esta manera se realizaron los mismos procedimientos para todos los sitio webs, realizado con el gestor de contenido en wordpress, obteniendo los resultados esperados, ya que los usuarios o administradores, suelen utilizar contraseñas no seguras, por ejemplo la mayoría de los sitios utilizaron solo numero entre 4 a 6 cifras como contraseña y algunas utilizaron letras, o letras con números. En la siguiente tabla se muestra un resumen de resultados respecto a los 7 sitios web analizados.

Tabla 5: Cantidad de sitio web vulnerados con ataque de fuerza bruta.

Sitios web analizados	Sitio Web vulnerados	Sitio Web no vulnerados
7	5	2

De la Tabla 5, el resultado nos muestra de los 7 sitios web en wordpress analizados, 5 fueron vulnerados que vendría a ser el 71.4%, y los 2 sitios web no vulnerados que son el 28.6%.

Del total de 20 sitios web, que se tomaron en cuenta para este trabajo de investigación, 5 sitios web fueron vulnerados, aprovechando las vulnerabilidades con las que cuenta, que vendría a ser el 25% del total de sitios web analizados.

5. Conclusiones y trabajos futuros

En el presente trabajo de investigación se ha realizado el análisis y comprobación de las capacidades de la herramienta kali linux, utilizando el componente wpscan, siguiente todos los procedimientos para un hackeo ético, y se pudo aprovechar de las distintas vulnerabilidades se los sitios web que fueron desarrollados con gestores de contenido CMS, particularmente en Wordpress. Con respecto a las contraseñas se ha podido observar que se siguen manteniendo el uso tradicional de años atrás como por ejemplo admin, 123456; confirmando que no existe o hay muy poca concientización del usos y manejo de contraseñas.

Mediante este trabajo de investigación se pudo concluir que con un poco de tiempo dedicado a ataques de fuerza bruta con diccionario, se puede vulnerar sitios web, que utilicen contraseñas débiles, como un medio de autenticación. Logrando verificar que estos sitios son muy vulnerables, sean por desconocimiento en la configuración del sitio web por parte del proveedor que les brinda el servicio, o por no tomar en cuenta políticas de utilización de contraseñar seguras.

Las actividades de concientización y capacitación a los empleados y/o funcionarios de las instituciones, son factores claves para mitigar los riesgos asociados a la seguridad; un empleado capacitado que sabe cómo crear contraseñas robustas y cambiarlas periódicamente, por lo tanto, es menos propenso a que sufra ataques de fuerza bruta y por ende colaborará con la seguridad de la empresa y de su hogar.

Finalmente las recomendaciones que se hacen luego de este trabajo de investigación es que estos sitios web, sería que le den importancia al tema de seguridad informática, pudiendo hacer un análisis de vulnerabilidades de sus sitios web, antes de subirlos a la nube, y también realizarlos periódicamente. Realizar actividades de concientización y capacitación a los responsables directos o indirectos que administran estos servicios, de esta manera minimizar los riesgos asociados a seguridad informática, cuando estén capacitados podrán crear contraseñas robustas y podrán cambiarlas periódicamente. De esta manera ser menos propenso a estos tipos de ataques

Los trabajos futuros a desarrollar luego de esta investigación, serian analizar sitios web con otros gestores de contenidos o realizados con otras herramientas, aprovechar

vulnerabilidades de otros servicios que se detectaron, utilizar las diferentes herramientas de código abierto que se encuentran disponibles para realizar este tipo de ataques.

Referencias

[1] Carneiro Roberto, Toscano Juan Carlos, Diaz Tamara, Los desafíos de las TIC para el cambio, Madrid: Fundación Santillana, 2009.

[2] Fernando Román Muñoz, Iván Israel Sabido Cortes, Luis Javier García Villalba, «Aplicaciones web vulnerables a propósito,» de VIII Congreso Internacional de Computación y Telecomunicaciones, Lima, 2016.

[3] Y. Martirosyan, «Security Evaluation of Web Application Vulnerability Scanners Strengths and Limitations Using Custom Web Application,» East Bay, California State University, 2012.

[4] J. Muniz y A. Lakhani, Web Penetration Testing with Kali Linux, Packt Publishing Ltd, 2013.

[5] R. Singh Patel, Kali Linux Social Engineering, editorial Packt Publishing Ltd, 2013.

[6] E. B. D. J. M. J. Bau, «Automated Black-Box Web Application Vulnerability Testing,» de IEEE Symposium on Security and Privacy, Berkeley, 2010.

[7] M. C. a. G. V. Adam Doupé, «Why Johnny Can't Pentest: An Analysis of Blackbox Web Vulnerability Scanners,» de 7th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA'10), 2010.

[8] A. M. Ferreira y H. Kleppe, «Effectiveness of Automated Application Penetration Testing Tools,» 2011.

[9] E. Fong, R. Gaucher, V. Okun y P. E. Black, «Building a Test Suite for Web Application Scanners,» de Hawaii International Conference on System Sciences, Proceedings of the 41st Annual, Waikoloa, HI, USA, 2008.

[10] F. Román, I. I. Sabido y L. J. García, «Capacidades de detección de las herramientas de análisis de vulnerabilidades en aplicaciones Web,» de XIII Reunión Española sobre Criptología y Seguridad de la Información, 2014.