# PRUEBAS DE PENETRACIÓN EN APLICACIONES WEB USANDO HACKEO ÉTICO

***Rina Elizabeth López de Jiménez***
*Ingeniera en Sistemas Informáticos. Docente de la Escuela de Ingeniería en Computación. ITCA-FEPADE Sede Central. E-mail: rina.lopez@itca.edu.sv*

## Resumen

En este artículo se lleva a cabo un análisis de las principales pruebas de intrusión en aplicaciones web, con un breve preámbulo de lo que es el significado de Pentesting, así como de los principales ataques que puedan sufrir las aplicaciones web. Se describen las principales metodologías que utilizan Pentesting, haciendo más amplio el aprendizaje obtenido de cada uno de ellos; a la vez se hace una lista de las herramientas de software más comúnmente utilizadas para las pruebas de penetración, destacando el sistema operativo Kali Linux, una herramienta gratuita capaz de modelar los ataques con el fin de obtener información acerca de las vulnerabilidades que los sitios web pueden tener.

## Palabras clave

Internet, seguridad informática, aplicaciones web, aplicaciones de computadores, vulnerabilidad, redes de computadores, aplicación informática.

## Abstract

In this article an analysis is conducted againts the main intrusion indicators in web applications; With a brief preamble of the term Pentesting, as well as the main attacks that web applications may suffer. It describes the main methodologies used by Pentesting, making the learning obtained from each of those tests more extensive; At the same time a list is made of the most commonly used software tools for penetration testing, highlighting the Kali Linux operating system, a free tool capable of modeling attacks in order to obtain information about the vulnerabilities that websites may have.

## Keywords

Internet, information security, web applications, computer applications, vulnerability, computer networks, computing application.

## Introducción

Sin lugar a dudas, el rápido crecimiento de Internet y el uso de un número ilimitado de aplicaciones web y móviles han llegado a beneficiar a todos y cambiar la forma en que nos comunicamos, así como la forma en que llevamos a cabo diferentes transacciones. Por esto, la importancia de aplicar medidas de seguridad para garantizar la integridad y fiabilidad de la información.

Muchas empresas hoy en día están muy preocupadas porque las aplicaciones web sean las más rápidas o se desarrollen con el mejor software, pero muy pocas se preocupan de que éstas posean la seguridad adecuada. Por esa razón, este artículo describe las diferentes técnicas y pruebas de penetración, utilizando diferentes herramientas basadas en software para establecer las posibles vulnerabilidades que una aplicación web pueda tener.

## Que es Pentesting

El término Pentesting es muy amplio y tiene varias definiciones, tales como:

- "Es el método para la evaluación de un sistema o red mediante la simulación de un ataque de origen hostil". [1]

- "Una prueba de seguridad con un objetivo específico; la prueba se termina cuando el objetivo se logra obtener, o el tiempo disponible termina". (Manual OSSTMM- Open Source Security Testing Methodology Manual).

- Prueba de seguridad donde los evaluadores copian los ataques reales para subvertir las características de seguridad de una aplicación, sistema o red (Instituto Nacional de Estándares y Tecnología, NIST).

**Recibido: 27/03/2017 - Aceptado: 11/06/2017**

- La definición real es que el Pentesting es un conjunto de pruebas objetivas con el fin de detectar las vulnerabilidades de un sistema, teniendo muy claro que ningún sistema es 100% seguro o inviolable [1].

## Seguridad en Aplicaciones Web

El crecimiento y la evolución de Internet ha impactado de manera significativa la forma en que nos comunicamos y realizamos operaciones, lo que genera una gran cantidad de información sensible. Por ejemplo, la incorporación de números privados en sitios de comercio electrónico, servicios web, bancos y redes sociales, entre otros; esto causa que la información pueda ser robada o alterada si estos no tienen las medidas de seguridad necesarias para el manejo de la misma.

Los ordenadores en todo el mundo son susceptibles al ataque de hackers o crackers capaces de comprometer los sistemas informáticos y robar o eliminar información valiosa. Por esta razón, es esencial saber si estas redes y sistemas informáticos están protegidos de cualquier tipo de intrusiones. [8]

Comúnmente se cree que los fallos o vulnerabilidades se encuentran en servidores web o en el desarrollo de software de la misma aplicación; sin embargo se ha logrado detectar que la mayoría de fallas están dadas por las malas prácticas de los desarrolladores. Por lo tanto, es importante entender que las aplicaciones web no sólo deben ser diseñadas y desarrolladas para cumplir con los objetivos específicos para los que se crean, sino que también deben tener cuidado de cada uno de los datos y la información generada en ellas. [2]


Fig. 1. Robo de información

### A. Bases de Seguridad

***Autenticación:*** se refiere a la pregunta: ¿quién es usted? Es el proceso de identificación única de aplicaciones y servicios de los clientes.
***Autorización:*** se refiere a la pregunta: ¿qué se puede hacer? Es el proceso que rige los recursos y las operaciones a las que el cliente autenticado tiene acceso.
***Revisión de Contabilidad:*** auditoría efectiva y el registro de claves para evitar la no-repudiación. La no-repudiación asegura que un usuario no puede negarse a realizar una operación o el inicio de una transacción.
***Aviso:*** también conocida como la privacidad. Es el proceso de asegurar que los datos se mantienen como privados y confidenciales y no pueden ser vistos por usuarios no autorizados o intrusos que controlan el flujo de tráfico a través de una red. El cifrado se utiliza a menudo para hacer cumplir la confidencialidad.
***Integridad:*** es la garantía de que los datos están protegidos contra posibles modificaciones accidentales o deliberadas (malicioso).
***Disponibilidad:*** Desde una perspectiva de seguridad, disponibilidad significa que los sistemas permanecen disponibles para los usuarios legítimos. [3]

### B. Ataques Comunes

En la actualidad existen números ataques a aplicaciones web; en la siguiente tabla se resumen los más destacados.

| ATAQUE | DESCRIPCIÓN |
|---|---|
| URL de tipo semántico | Este tipo de ataques involucran a un usuario modificando la URL a modo de descubrir acciones a realizar que originalmente no están planeadas para ser manejadas correctamente por el servidor. |
| Cross-Site Scripting | Cross-Site Scripting (XSS) es un tipo de vulnerabilidad de seguridad informática típicamente encontrada en aplicaciones web que permiten la inyección de código por usuarios maliciosos en páginas web. |
| Cross-Site Request Forgery | Este tipo de ataque permite al atacante enviar peticiones HTTP a voluntad desde la máquina de la víctima |
| Peticiones HTTP falsificadas | Un ataque más sofisticado que el anterior es enviar peticiones falsas empleando herramientas especiales para este propósito. |

Tabla I. Ataques más comunes sobre aplicaciones web [2]

### C. Ataques a traves de Bases de Datos

La mayoría de las aplicaciones web hoy en día se encuentran transportando, generando y mostrando una gran cantidad de información proveniente de bases de datos, por lo que mantener la integridad y fiabilidad de la misma se vuelve un tema importante cuando se habla de seguridad. El siguiente listado muestra los principales ataques sobre las bases de datos.

- **Exhibition Access Credentials**
- **SQL Injection**
- **Exposure data**

# Pruebas de Penetración en Aplicaciones web

## A. Ámbitos de Pruebas

• ***Externo:*** ejecutado desde afuera del perímetro de seguridad, por ejemplo Internet.

• ***Interno:*** cuenta con más privilegios en la red, por ejemplo empleados, clientes, proveedores, Internet libre [4].

## B. Tipos de Pruebas de Penetración

Las siguientes son importantes tipos de pruebas de penetración:

• Black Box Penetration Testing
• White Box Penetration Testing
• Grey Box Penetration Testing

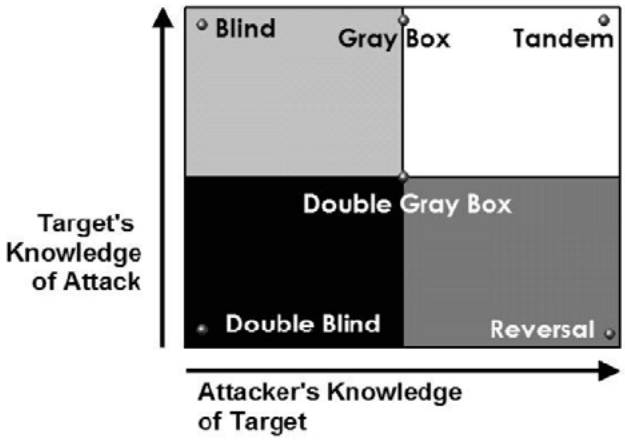


Figura 2. Tipos de Pentesting

• ***Prueba de Penetración Black Box:*** en las pruebas de penetración Black Box, el tester no tiene idea acerca de los sistemas que va a probar. Se interesa por reunir información acerca de la red o el sistema de destino. Por ejemplo, en esta prueba, un tester sólo sabe lo que debe ser el resultado esperado y no sabe cómo llegar al resultado. Él no examina los códigos de programación. Las pruebas de penetración Black Box analizan la cobertura de código y realizan pruebas de flujo de datos, pruebas de ruta, prueba de lazo, etc.

• ***Prueba de Penetración White Box:*** esta es una prueba completa; el tester ha sido dotado de toda la gama de información sobre los sistemas y / o en la red, tales como esquemas, código fuente, detalles del sistema operativo, dirección IP, etc. Normalmente se considera como una simulación de un ataque de una fuente interna. También se conoce como caja estructural de vidrio, caja clara y las pruebas de caja abierta.

• ***Pruebas de Penetración Grey Box:*** En este tipo de pruebas, un tester generalmente proporciona información parcial o limitada con los detalles internos de un sistema. Puede ser considerado como un ataque de un hacker externo que habían tenido acceso ilegítimo a los documentos de infraestructura de red de una organización [5].

| Types | Advantages | Disadvantages |
|---|---|---|
| Black Box | • Tester need not necessarily be an expert as it dices not demand specific languaje<br>• Texter verifies contradictions in the actual system and especifications<br>• Test is generaly conducted with the perspective of a user, not the designer | • Particularly, these kinds of test ases area difficult to design<br>• Possibly, it is not worth, incase designer has already conducted a test case<br>• It does not conduct everything |
| White Box | • It ensures that all independent paths of module have been exercised<br>• It ensures that all logical decisions have been verified along with their true and false value<br>• It discovers the typographical errors and does syntax checking<br>• Its finds the design errors that may have occurred because of the difference between logical flow of the program and the actual execution | Does not have |
| Grey Box | • As the tester does not require the Access of source code, it is non-intrusive and unbiased<br>• As there is clear difference between a developerand a tester, so there is leaut risk of personal conflict<br>• You don't need to provide the internal information about the program funcions and other operations | Does not have |

Tabla 2. Ventajas y desventajas de los tipos de pruebas de penetración

## c. Principales Etapas del Pentesting

Las pruebas de penetración son una combinación de técnicas que consideran varios temas relacionados a los sistemas; los prueba, analiza y da soluciones. Se basa en un procedimiento estructurado que lleva a cabo las pruebas de penetración paso a paso.


Figura 3. Principales etapas del Pentesting

• ***Planeación y Preparación***

La planificación y preparación comienza con la definición de las metas y objetivos de las pruebas de penetración. El cliente y el tester deben definir conjuntamente los objetivos de modo que ambas partes tengan los mismos objetivos y comprensión. Los objetivos comunes de las pruebas de penetración son:

a. Identificar la vulnerabilidad y mejorar la seguridad de los sistemas.
b. Tener la seguridad de TI confirmada por un agente externo.
c. Aumentar la seguridad de la infraestructura de la organización / personal.

• ***Reconocimiento***

El reconocimiento incluye un análisis de la información preliminar. Muchas veces un tester no tiene mucha información que no sea la información preliminar, una dirección IP o bloque de direcciones IP. El tester inicia mediante el análisis de la información disponible y, si es necesario, recibe mayor información como descripciones de sistemas, planes de la red, etc. desde el cliente. Este paso es una especie de prueba de penetración pasiva. El único objetivo es obtener información completa y detallada de los sistemas.

• ***Descubrimiento***

En este paso, un pentester probablemente usará las herramientas automatizadas para analizar activos de destino para descubrir vulnerabilidades. Estas herramientas normalmente tienen sus propias bases de datos que dan los detalles de las últimas vulnerabilidades. El tester es capaz de realizar:

**a. Descubrimiento de red:** sistemas, servidores y otros dispositivos adicionales.

**b. Host Discovery:** determina los puertos abiertos en estos dispositivos.

**c. Servicio de Interrogación:** interroga a los puertos para descubrir los servicios reales que se ejecutan en ellos.

• ***Análisis de Información y Riesgos***

En este paso, el tester analiza y evalúa la información reunida antes en los pasos de prueba para penetrar dinámicamente el sistema. Según el número de sistemas y el tamaño de la infraestructura, así es el tiempo que se tarda. Al analizar, el tester considera los siguientes elementos:

a. Los objetivos definidos de la prueba de penetración.
b. Los riesgos potenciales para el sistema.
c. El tiempo estimado necesario para evaluar posibles fallos de seguridad para las pruebas de penetración activa posteriores.

Sin embargo, de la lista de sistemas identificados, el tester puede optar por probar sólo aquéllos que contienen vulnerabilidades potenciales.

• ***Intentos de Intrusión Activos***

Este es el paso más importante que debe realizarse con el debido cuidado e indica en qué medida las vulnerabilidades potenciales que se identificaron en el paso de descubrimiento poseen los riesgos reales. Este paso se debe realizar cuando se necesita una verificación de vulnerabilidades potenciales. Para aquellos sistemas que tienen requisitos de alta integridad, la vulnerabilidad potencial y el riesgo deben considerarse cuidadosamente antes de realizar procedimientos críticos de limpieza.

• ***Análisis Final***

Este paso considera principalmente todos los pasos realizados hasta ese momento y una evaluación de las vulnerabilidades presentes en forma de riesgos potenciales. Además, el tester recomienda eliminar las vulnerabilidades y los riesgos. Por encima de todo, el tester debe asegurar la transparencia de las pruebas y las vulnerabilidades que reveló.

• ***Preparación del Informe***

La preparación del informe debe comenzar con los procedimientos generales de las pruebas, seguido de un análisis de las vulnerabilidades y los riesgos. Los altos riesgos y las vulnerabilidades críticas deben tener prioridad y seguidas por las de orden inferior. Sin embargo, al documentar el informe final, es necesario considerar los siguientes puntos:

a. Resumen general de pruebas de penetración.
b. Detalles de cada paso y la información recolectada durante la prueba de penetración.
c. Detalles de todas las vulnerabilidades y riesgos descubiertos.
d. Detalles de limpieza y fijación de los sistemas.
e. Sugerencias para la seguridad futura. [5]

## D. Metodologías

• ***OWASP (Open Web Application Security Project)***

Es un proyecto de código abierto dedicado a identificar y combatir las causas que hacen que el software sea inseguro. La Fundación OWASP es una organización sin fines de lucro que apoya y administra los proyectos e infraestructura de OWASP. Los documentos más exitosos

incluyen OWASP Guide y el documento de autoevaluación ampliamente adoptado OWASP Top 10.

Este es un método de prueba para aplicaciones web basadas en dos fases: pasiva y activa. Su enfoque es "caja negra", preferiblemente poca o ninguna información conocida incluso en el contexto que se harán las pruebas.

***Fase Pasiva.***
Está fase consiste en probar para entender la lógica de la aplicación que está bajo testing y así poder comprobar si arroja cualquier elemento que podría significar una puerta abierta para el análisis detallado.

***Fase Activa.***
El "tester" comienza a probar todo lo recomendado en el proceso de esta metodología. Esta fase se centra específicamente en 9 subcategorías de 66 procesos:

a. Pruebas de Gestión de Configuración (recopilación de información + gestión de configuración).
b. Pruebas de Autenticación
c. Pruebas de Autorización
d. Pruebas de Gestión de Sesiones
e. Pruebas de Lógica Empresarial
f. Pruebas de Validación de Datos
g. Pruebas de Servicio.
H. Prueba de Servicios Web
i. Pruebas de Ajax [6]

***• OSSTMM (Open Source Security Testing Methodology Manual)***

Es uno de los estándares profesionales más completo y comúnmente utilizado en auditorías de seguridad para revisar la Seguridad de Sistemas que se encuentran en Internet. Incluye un marco que describe los pasos que se harían para la implementación de la auditoría.

***• ISSAF ((Information Systems Security Assessment Framework)***

Los Sistemas de Información del Marco de Evaluación de Seguridad son una metodología estructurada para el análisis de seguridad en múltiples dominios y detalles específicos de la prueba o de las pruebas para cada uno de ellos. Su objetivo es proporcionar procedimientos muy detallados para la comprobación de sistemas de información que reflejen situaciones reales.

ISSAF se utiliza principalmente para cumplir con los requisitos de evaluación de las organizaciones y también puede utilizarse como referencia para nuevas implementaciones relacionadas con la seguridad de la información. [6]

## Herramientas para hacer Pentesting

Hay muchas herramientas de software para pentesting basado en software libre, cada una capaz de hacer diferentes tipos de pruebas de penetración. A continuación, se muestra una lista de las herramientas más utilizadas:

***• Burp Suite***
Burp Suite es una excelente plataforma para PenTest y sitios web de seguridad. Esta herramienta tiene muy buenas características como:

a. Intercepción de proxy
b. Spider (a "rastreo")
c. Detección automática de vulnerabilidades
d. Herramienta de repetición
e. Habilidad para escribir plugins propios

***• Acunetix – Scanner for web vulnerabilities***
Potente herramienta para MS Windows que detecta un gran número de vulnerabilidades en la forma de carga de archivos, incluyendo Cross-Scripting, Inyección de SQL, Inyección de CRLF, Ejecución de Código, Transversal de Directorio, Inclusión de Archivos.

También tiene un escáner integrado de puertos propios, que aunque no reemplazan nmap (nmap.org). Cuenta con varias herramientas independientes para realizar tareas específicas como Fuzzer HTTP, HTTP Editor, Authentication Tester, etc.

***• SQLmap***
Desarrollado por Bernardo Damele y Miroslav Stampar. SQLmap es una herramienta que ninguna aplicación web de tipo auditor debe prescindir. Se trata de una herramienta basada en la web y de código abierto que automatiza la detección y explotación de vulnerabilidades de inyección de SQL y extrae información de bases de datos.

Actualmente soporta:

a. MySQL
b. Oracle
c. PostgreSQL
d. Microsoft SQL Server

También parcialmente soporta otras bases de datos como:

a. Microsoft Access
b. DB2
c. Informix
d. Sybase
e. Interbase

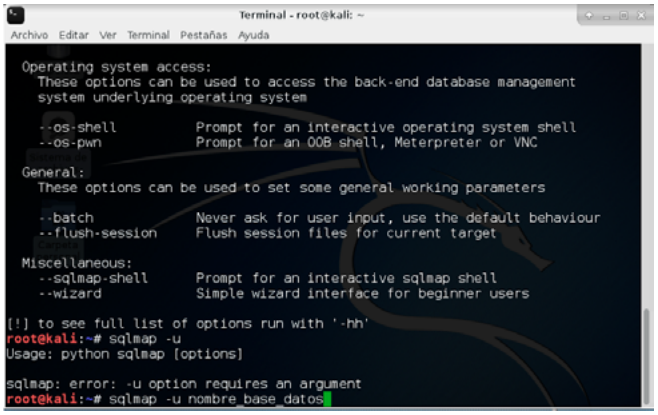


Figura 4. SqlMap on Kali Linux

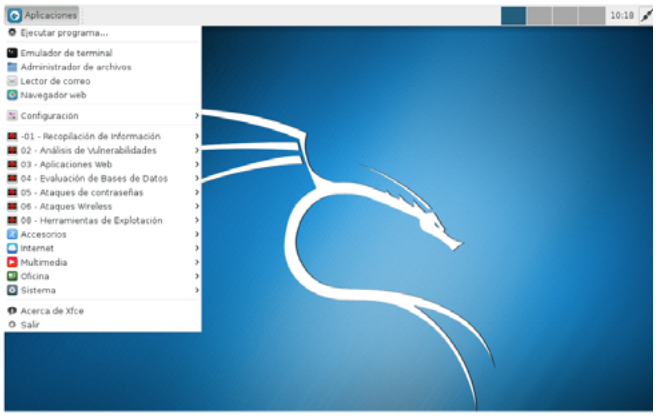


Figura 5. Kali Linux Menu

• ***WhatWeb***

Esta herramienta es útil para uno de los primeros pasos en una auditoría o Pentesting; recopila información e identifica si el "target" utiliza cualquier plataforma de Content Manager (CMS) Blog, Servidores, Javascripts, etc.

• ***Nessus***

Nessus apunta a un uso más amplio en el campo de pruebas, es decir, un gran número de redes de dispositivos, etc. También es muy útil para aplicaciones web Pentesting para habilitar y configurar los módulos correctos. [7]

• ***Kali Linux***

Kali Linux es una distribución Linux basada en Debian destinada a pruebas avanzadas de penetración y auditoría de seguridad. Kali contiene varios centenares de herramientas destinadas a diversas tareas de seguridad de la información, tales como pruebas de penetración, forense e ingeniería inversa. Kali Linux es desarrollado, financiado y mantenido por Offensive Security, una compañía líder en capacitación en seguridad de la información. Kali Linux fue lanzado el 13 de marzo de 2013 como una reconstrucción completa de arriba a abajo de Backtrack Linux, adhiriéndose completamente a los estándares de desarrollo de Debian.

a. Incluye más de 600 herramientas de prueba de penetración.
b. Kali Linux siempre será libre y gratuito.
c. Posee un árbol de desarrollo de código abierto.
d. Kali se adhiere al estándar de jerarquía del sistema de archivos.
e. Soporta dispositivos inalámbricos de gran alcance. [9]

• ***Parrot Security OS***

Se trata de un sistema operativo basado en Debian, desarrollado por el equipo de Frozenbox. Este sistema operativo está diseñado para realizar pruebas forenses de informática, piratería ética, criptografía, etc. Parrot Security OS promete ser un sistema operativo ligero y altamente eficiente.

• ***Backbox Linux***

Se trata de un sistema operativo basado en Ubuntu, que se centra en la evaluación de seguridad y pruebas de penetración. Cuenta con una amplia gama de herramientas de análisis de seguridad para aplicaciones web, análisis de redes, etc.

• ***Samurai Web Testing Framework***

Es básicamente un entorno real de Linux que viene pre-configurado para funcionar como una plataforma de prueba de web-test. Contiene varias herramientas de hacking gratuitas y abiertas para detectar vulnerabilidades en el código de sitios web.

## Conclusiones

Después de realizado el análisis de la información obtenida se puede confirmar que ninguna aplicación web es perfectamente segura y libre de ataques, pero con el uso de técnicas o test de intrusión, Pentesting, como herramientas de Hackeo Ético, todas esas vulnerabilidades pueden ser superadas, evitando los ataques que socavan la integridad y fiabilidad de los datos que se manejan.

## Referencias

[1] R.Guirado, "Penetration Testing : conceptos generales y situación actual [en línea]".
Montevideo, 2009. Disponible en: https://www.isaca.org/chapters8/Montevideo/Events/Documents/pene-

tration%20testing%20-%20conceptos%20generales%20y%20situacin%20actual.pdf. [Accedido: 22 -mar-2016]

[2] "Aspectos Básicos de la Seguridad en Aplicaciones Web Documentos - CSI -", [En línea]. Disponible en: https://www.seguridad.unam.mx/historico/documento/index.html-id=17 [Accedido: 12 -ene-2016]

[3] "Web Application Security Fundamentals", 2003. [Online]. Available: https://msdn.microsoft.com/en-us/library/ff648636.aspx. [Accessed: 12-Jan- 2016]

[4] J. Terceiro. "ESEI Coding Dojo, pentesting ~ El mundo en bits", [En línea]. Disponible en: http://www.elmundoenbits.com/2012/02/esei-coding-dojo-pentesting.html#.WSNV5dyeW1t. [Accedido: 12-feb- 2016]

[5] "Types of Penetration Testing", [Online]. Available: https://www.tutorialspoint.com/penetration_testing/types_of_penetration_testing.htm
[Accessed: 12-Jan- 2016]

[6] "Metodologías más usadas en pentesting. Estudio comparativo", [En línea]. Disponible en: https://es.scribd.com/doc/98081446/Metodologias-mas-usadas-en-pentesting-Estudio-comparativo. [Accedido: 19 -feb- 2016]

[7] R. Caire, "5 Herramientas útiles en Penetration Testing para Aplicaciones Web", seguridad y ética, 2012. [En línea]. Disponible en https://seguridadetica.wordpress.com/2012/04/11/5-heramientas-utiles-en-penetration-testing-para-aplicaciones-web/. [Accedido: 22-feb- 2016]

[8] A.R. Plata, "Ethical Hacking | Documentos - CSI -", [En línea]. Disponible en: https://www.seguridad.unam.mx/historico/documento/index.html-id=7.[Accedido: 23 -may- 2017]

[9] "What is Kali Linux ?. Kali Linux", [Online]. Available: http://docs.kali.org/introduction/what-is-kali-linux. [Accessed: 21-Feb- 2016]