GUÍA DE REMEDIACIÓN DE VULNERABILIDADES INFORMÁTICAS PARA EL SOFTWARE

YOHAN ESNEIDER HERNÁNDEZ VILLARREAL

FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES
FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS
INGENIERÍA DE SISTEMAS
BOGOTÁ DC.
2017

GUÍA DE REMEDIACIÓN DE VULNERABILIDADES INFORMÁTICAS PARA EL SOFTWARE

YOHAN ESNEIDER HERNÁNDEZ VILLARREAL

TRABAJO DE GRADO

DIRECTOR: INGENIERO LUIS EDUARDO BAQUERO REY

FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES
FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS
INGENIERÍA DE SISTEMAS
BOGOTÁ DC.
2017

PAGINA DE ACEPTACIÓN

_____
_____
_____
_____
_____
_____

_____
Firma presidente del Jurado

_____
Firma del Jurado

_____
Firma del Jurado

Bogotá DC, 16 de junio de 2017

DEDICATORIA

Está investigación la dedico a Dios por permitirme estar en este punto de mi carrera profesional y a mi familia por el tiempo que ocupe en mis estudios.

AGRADECIMIENTOS

Al Ingeniero Jorge Leal por contagiar a los estudiantes en la investigación, el desarrollo y el bien común a través de la ingeniería, al Ingeniero Luis Eduardo Baquero por el direccionamiento y guía en el desarrollo de esta investigación y al Ingeniero Miguel Hernández Bejarano quien a través de sus ejercicios incentiva al desarrollo de retos dentro y fuera de los espacios académicos.

ÍNDICE DE CONTENIDO

ÍNDICE DE ILUSTRACIONES

RESUMEN

La atención de vulnerabilidades informáticas es un proceso técnico diseñado para el mantenimiento y la corrección de errores en el software desde el punto de vista de la seguridad. Aporta un mecanismo para el aseguramiento de sistemas informáticos incluidos los relacionados a la privacidad de las personas y de las empresas. Desconocer la importancia de la atención de vulnerabilidades expone a los sistemas informáticos a incidentes relacionados con la perdida de la propiedad o el acceso a la información almacenada y compartida de personas, de gobiernos o de empresas. (NIST National Institute of Standards and Technology, 2013)

Este documento expone una investigación básicamente exploratoria que a través de una herramienta informática correlaciona datos sobre la importancia del proceso de atención de vulnerabilidades y propone una guía para llevar a cabo este proceso desde el punto de vista de la remediación; proceso de ejecución que consiste en actualizar, desinstalar o configurar el software para corregir vulnerabilidades informáticas.

Palabras clave
Remediación, Seguridad Informática, Software, Vulnerabilidades

INTRODUCCIÓN

El Instituto Nacional de Estándares y Tecnología (NIST), una de las agencias de la administración de tecnología del Departamento de Comercio de los Estados Unidos, publicó en 2003 una actualizada versión de su norma NIST SP800-40 versión 3.0 con el ánimo de proporcionar una ruta a seguir para crear un plan de parcheado y un programa de atención de vulnerabilidades informáticas en las empresas. Está guía de atención incluye entre otras cosas la identificación de las vulnerabilidades informáticas en los sistemas que conforman las plataformas tecnológicas de las empresas.

Las vulnerabilidades informáticas son las fallas o debilidades en el diseño, implementación, operación o manejo de una tecnología dura o blanda como un equipo de cómputo o como el software que el equipo de cómputo emplea. Esa definición es realizada por el Internet Engineering Task Force, organización que produce documentos técnicos relevantes de alta calidad que influyen en la forma en que la gente diseña, usa y administra Internet. (IETF Internet Engineering Task Force, 2000).

En el RFC 2828 publicado por el IETF en mayo del año 2000, al definir las vulnerabilidades informáticas menciona que estas fallas presentes en la tecnología tienen una característica conocida como explotación: mecanismo mediante el cual una vulnerabilidad se usa para afectar un sistema informático. En el RFC también se menciona que la mayoría de la tecnología empleada en los sistemas tiene vulnerabilidades de algún tipo, y que estas se usan para efectuar intentos de evasión de políticas de seguridad de un sistema.

The Mitre Corporation, una organización financiada con recursos de la Dirección de Ciberseguridad Nacional del gobierno de los Estados Unidos, desarrolla y mantiene una base de datos que es reconocida como una lista estandarizada de reporte, seguimiento e identificación de vulnerabilidades informáticas, y además funciona como la lista base del repositorio de los Estados Unidos para la información de vulnerabilidades llamada National Vulnerability Database.

La base de datos que mantiene The Mitre Corporation es reconocida por la abreviatura CVE (Common Vulnerabilities and Exposures); es continuamente actualizada por fabricantes de software en el mundo, por auditores de seguridad, por hackers de sombrero blanco y en general por una comunidad relacionada con el mundo de la seguridad de la información. En el empeño de proporcionar una solución a los registros de las vulnerabilidades constantemente encontradas y reportadas en la lista CVE diferentes compañías de desarrollo de software y desarrolladores en general publican actualizaciones, parches de seguridad, nuevas versiones o configuraciones para corregir estas debilidades en los sistemas. Así mismo, los usuarios que emplean software vulnerable tienen la

responsabilidad de usar estos mecanismos de corrección y en autorizar a los sistemas y a las empresas para efectuar remediaciones sobre sus sistemas de información. (The Mitre Corporation, 2017).

CAPITULO 1. ASPECTOS DE LA INVESTIGACIÓN

1.1. TITULO

GUÍA DE REMEDIACIÓN DE VULNERABILIDADES INFORMÁTICAS PARA EL SOFTWARE.

1.2. DESCRIPCIÓN DEL PROBLEMA

Pedro González Pérez, en su libro Ethical Hacking: Teoría y práctica para la realización de un pentesting, expone que el objetivo final de un proceso de Ethical Hacking es el de evaluar la seguridad de los sistemas, las comunicaciones y las infraestructuras de las que dispone una organización o empresa.

En lo que se refiere a las vulnerabilidades informáticas, un proceso de Ethical Hacking conlleva a diversas pruebas con diversos enfoques que en cualquiera de los casos tiene como objetivo común encontrar esas debilidades que puedan afectar los bienes o los activos de una empresa. En este sentido, el autor determina que un proceso de Ethical Hacking está guiado por la búsqueda de las vulnerabilidades informáticas las cuales provoquen las remediaciones (aplicar contramedidas de corrección), con el fin de detectar y solucionar agujeros de seguridad evitando amenazas de ataque. (Gonzalez, 2015).

En la ilustración 1 se presenta la prueba de concepto de una confección de un ataque informático, la contramedida y las consecuencias de una vulnerabilidad expuesta en el RFC 2828 del IETF publicado en mayo del 2000

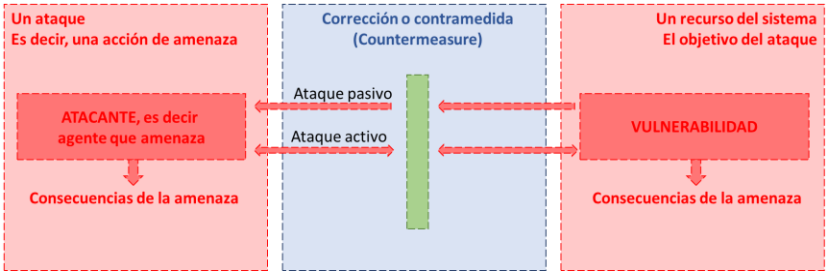


Ilustración 1. Estructura de confección de un ataque, contramedida y consecuencias.

> El desarrollo de un proceso de Hacking es ético en la medida en la que el dueño de los sistemas evaluados es quien contrata un servicio de auditoria que detecte las vulnerabilidades o autoriza expresamente al Hacker, persona que realiza este tipo de procesos. Cuando la contratación la hace un agente distinto al propietario del sistema evaluado o es evaluado sin su consentimiento, el proceso de Hacking deja de ser ético. En casos en los que se busca dañar el sistema evaluado o adquirir datos de él de manera no autorizada es un delito que se considera como un ataque informático, y se conoce como Cracker a la persona que ejecuta este proceso.
>
> En resumen, el Hacking puede tener distintos propósitos, pero en cualquiera de los casos un Hacker o un Cracker usan mecanismos para la detección y explotación de vulnerabilidades informáticas. (IETF Internet Engineering Task Force, 2000).

González en su libro de Ethical Hacking indica que la explotación de vulnerabilidades en un proceso no ético se usa para la obtención de acceso o control de los dispositivos que manipulan la información digital y datos personales o empresariales con un propósito de beneficio personal o económico, cuya solución se puede encontrar en la norma NIST SP800-40 versión 3.0 del 2003 del Instituto Nacional de Estándares y Tecnología de Estados Unidos, la cual da una orientación de buenas prácticas en la elaboración y ejecución de un plan de parcheado de atención de vulnerabilidades informáticas que disminuya el nivel de exposición de un sistema de información y su tecnología.

A pesar de las soluciones expuestas en los anteriores apartados, existe un desconocimiento general acerca de la importancia que tiene la remediación de problemas de seguridad. A continuación se relacionan 3 posibles razones de desconocimiento:

- Las vulnerabilidades informáticas pueden estar presentes en toda la tecnología, tanto duras como en las blandas. Afectan hardware o a las versiones de un software ampliamente usado como sistemas operativos, plugins, herramientas de ofimática o aplicaciones en general, por lo que la identificación de problemas de seguridad en estos elementos puede tomar mucho tiempo e incluso requerirá de un proceso de ingeniería que logre entender las distintas arquitecturas del software que se encuentre vulnerable.

  En la ilustración 2 se muestran registros ordenados alfabéticamente, que se consultan a partir de los datos recolectados de la base de datos de vulnerabilidades CVE que mantiene The Mitre Corporation. Los datos corresponden a 20 aplicaciones de las 2018 que tienen por lo menos una vulnerabilidad informática publicada en Internet al 2 de junio del 2017:

Product Name	Vendor Name	# Of CVE Entries	Product Type
A Better Member-based Asp Photo Gallery	Ontarioabandonedplaces	1	Application
A King Sperm By Dr. Seema Rao	Teknopoint	1	Application
A Very Short History Of Japan	Ireadercity	1	Application
A+	Yunlai	1	Application
A+ Php Scripts News Management System	Marc Melvin	1	Application
A+ Store E-commerce	Web Inhabit	2	Application
A-a-s Application Access Server	A-a-s Application Access Server	1	Application
A-blog	A-blog	6	Application
A-blog Cms	Appleple	2	Application
A-cart	Alan Ward	4	Application
A-cart	Coxco Support	1	Application
A-conman	A-conman	1	Application
A-faq	Alan Ward	2	Application
A-form	Ark-web	1	Application
A-form Bamboo	Ark-web	1	Application
A-form Pc	Ark-web	2	Application
A-form Pc Mobile	Ark-web	2	Application
A-forum	Arnotic	1	Application
A-news	Appleple	1	Application

Ilustración 2. Lista top 20 de aplicaciones con registro de vulnerabilidad.

- El desconocimiento de la responsabilidad de quien detecta, reporta, analiza o soluciona una vulnerabilidad no es claro en algunas empresas e incluso las personas que emplean tecnología para uso personal o domestico desconocen qué es una vulnerabilidad informática.

  Con la correlación realizada a partir de la base de datos CVE y de las remediaciones publicadas para atender las vulnerabilidades usada por InsightVM y Nexpose, herramientas de gestión de vulnerabilidades informáticas, se encuentra que al 27 de abril de 2017 el 8,27% de las vulnerabilidades informáticas de 91.092 evaluadas no tienen una remediación por desconocimiento de la responsabilidad de los fabricantes e instituciones detrás del desarrollo del software. En la ilustración 3 se resumen estos indicadores:

Cantidad total de vulnerabilidades evaluadas	91.092
Cantidad de vulnerabilidades con contramedida publicada	83.554
Cantidad de Vulnerabilidades sin contramedida publicada	7.538
% de vulnerabilidades sin contramedida	8,275%

Ilustración 3. Estadística de vulnerabilidades sin remediación.

- Los conceptos que definen las vulnerabilidades informáticas y los obstáculos y tiempos que se encuentran en la realización de procesos de remediación de vulnerabilidades.

  El Doctor José María Alonso, Chief Data Officer de Telefónica España, el 13 de mayo de 2017 en su explicación de los ataques relacionados con el ransomware conocido como WanaCryptor expuso que la principal razón de la

afectación de la vulnerabilidad CVE-2017-143 se debió a que el software que usan los equipos de Telefónica necesitaba ser probado con anterioridad al proceso de remediación, y que este trabajo no era tan rápido porque el volumen de software suele ser muy grande y de una gran sensibilidad para la continuidad del negocio, por lo que al final el proceso de la remediación de la vulnerabilidad CVE-2017-143 no se realizó en todos los equipos de la compañía al momento del ataque. (Alonso, 2017).

1.3. JUSTIFICACIÓN

Razones Sociales:

Emplear procedimientos para mitigar vulnerabilidades reduce el riesgo de ser víctima de delincuentes, que emplean mecanismos informáticos para realizar intrusiones a infraestructuras computacionales relacionadas al tratamiento de la información digital. Comprobar si los parches recomendados por fabricantes se instalaron correctamente es vital para tener las últimas versiones aseguradas del software que reposa en los equipos de una empresa o en equipos de uso personal. Para equipos con software vulnerable que sean empleados para uso mixto, tanto personal como empresarial, la remediación de problemas de seguridad disminuye la cantidad de mecanismos que buscan afectar la privacidad de las personas al estar en contacto físico con un dispositivo electrónico o que mantiene su información digital en un equipo de cómputo. (CISCO, 2017).

Razones Organizacionales:

De acuerdo a la norma NIST SP800-40 V 3.0 Creating a Patch and Vulnerability Management Program, se establece que los controles de vulnerabilidades informáticas tienen para las empresas beneficios como la reducción en el nivel de amenaza cibernética, hay una reducción del riesgo de compromiso para problemas de seguridad que son de fácil explotación, hay un aseguramiento en sistemas críticos o que contiene datos sensibles. Este argumento es válido incluso para sistemas no críticos si la explotación exitosa conduce a permitir a un atacante la obtención de control total de un sistema.

Razones Legales:

La atención de vulnerabilidades informáticas con su proceso de remediación sirve de mecanismo de protección de los elementos de TI que trabajan en función de los datos personales, los cuales según las leyes Colombianas deben ser protegidos.

En la legislación colombiana, la Ley estatutaria 1581 del 17 de octubre de 2013, reglamentada parcialmente por el Decreto Nacional 1377 de 2013, dictan

disposiciones generales para la protección de datos personales, decretada con el objeto de desarrollar el derecho constitucional que tienen todas las personas a conocer, actualizar y rectificar las informaciones que se hayan recogido sobre ellas en bases de datos o archivos, y los demás derechos, libertades y garantías constitucionales a que se refiere el artículo 15 de la Constitución Política; así como el derecho a la información consagrado en el artículo 20 de la misma, los principios y disposiciones contenidas en la ley aplica una protección de los datos personales registrados en cualquier base de datos que los haga susceptibles de tratamiento por entidades de naturaleza pública o privada.

1.4. IMPACTO

En un mundo favorecido por el uso de la información como elemento activo dentro de actividades empleadas por seres humanos prevalecerán intereses de diversos aspectos, y dependiendo las razones, particularmente las diferentes generaciones estiman esfuerzos para preservar esa información, para transmitirla o evitar su transmisión, para evitar su uso o para usarla. CISCO, empresa pionera en aspectos de Tecnologías de la Información y las Comunicaciones, refiriéndose a la infraestructura computacional relacionada al tratamiento de la información digital, define al esfuerzo que se ejerce para proteger estos datos como Ciberseguridad.

La Ciberseguridad permite ejercer controles técnicos para proteger la información contenida en un host (dispositivo susceptible a tener una conexión de red) o que circule a través de una red de equipos. En los controles técnicos se encuentran los relacionados a la identificación y remediación de vulnerabilidades informáticas. Este es un aspecto que cobra una gran importancia para evitar problemas de ciberseguridad que aprovecha las debilidades de los sistemas para secuestrar o apoderarse de la información de las empresas o para denegar los servicios de las plataformas tecnológicas en el mundo.

La investigación expuesta en este documento representa una propuesta para dejar de desconocer la importancia que tiene la atención de vulnerabilidades informáticas, desde el punto de vista de la remediación: proceso de ejecución que consiste en actualizar, desinstalar o configurar el software para corregir vulnerabilidades informáticas.

1.5. DELIMITACIONES Y ALCANCE

Las vulnerabilidades informáticas pueden estar presentes en toda la tecnología, tanto hardware como software. Para efectos de esta investigación se tratan

aquellas vulnerabilidades informáticas que afectan versiones de un software ampliamente usado en equipos de cómputo como sistemas operativos, plugins o aplicaciones en general. La investigación se centra en la identificación de los problemas relacionados a la no remediación de vulnerabilidades informáticas.

En seguridad informática hay un concepto denominado Defence in depth (Defensa en profundidad), el cual se basa en la premisa de que todo componente de un sistema puede ser vulnerado, y por tanto propone el uso de distintas técnicas que permitan duplicar los elementos de protección para limitar los daños en caso de una intrusión en la primera línea de defensa. Está investigación y sus resultados no hace referencia a la aplicación de este concepto en la atención de vulnerabilidades informáticas, sin embargo, la aplicación de la guía de remediación expuesta se extiende a los equipos que hacen parte de la seguridad perimetral que expone la defensa en profundidad.

Esta exclusión de la defensa en profundidad se hace teniendo en cuenta las apreciaciones que Matt Alderman, Vicepresidente de Global Strategy en Tenable Network Security, expuso en la RSA Conference entre febrero 29 y marzo 4 de 2016 en San Francisco Estados Unidos. Las apreciaciones de Matt Alderman exponen que se debe considerar que siempre hay un intruso en cada componente de un sistema y por lo tanto es vulnerable, independientemente del funcionamiento de los elementos de protección diseñados para limitar daños desde la primera línea de defensa y las líneas siguientes. (Alderman, 2016).

La guía propuesta en los resultados de la investigación se basa en las normas, metodologías y estándares expuestos en el marco referencial de este documento. En ese sentido, la guía representa una interpretación y resumen del marco existente, la cual se construye a partir de la experiencia de los participantes de la investigación en procesos de atención de vulnerabilidades informáticas.

Las metodologías propuestas en este documento buscan dar un lineamiento para la reducción de debilidades de seguridad en Software instalado en equipos de cómputo, pero su uso no representa un mecanismo definitivo para no tener vulnerabilidades informáticas.

Como el software es una forma de propiedad intelectual se deja una constancia que el uso, copia o distribución del código de la herramienta desarrollada como anexo de esta investigación es permitido, es de uso libre, y también lo es la guía expresada en este documento. Sin embargo, es necesario que al mostrar los

resultados se haga una referencia que la información recolectada pertenece a las bases de datos: CVE de The Mitre Corporation, la base de datos empleada y mantenida por Rapid7 en su sistema de gestión de vulnerabilidades Nexpose, y la base de datos de ExploitDB de Offensive Security y Rapid7 Metasploit.

Los datos expuestos en el desarrollo de la investigación y en la herramienta desarrollada están recolectados con fecha de corte al 26 de abril de 2017. En ese sentido, las vulnerabilidades después de esa fecha no están registrados en la base de datos de la herramienta ni en la guía desarrollada.

1.6. OBJETIVOS

1.6.1. OBJETIVO GENERAL:

Diseñar una guía que describa pasos sugeridos para atender vulnerabilidades informáticas exponiendo el impacto que tiene la no remediación de puntos débiles de un software.

1.6.2. OBJETIVOS ESPECÍFICOS:

- Diseñar una guía con el ciclo de vida que tiene la atención de vulnerabilidades informáticas enfocado en la remediación.
- Hacer una prueba de Ethical Hacking que evidencie un mecanismo de explotación de una vulnerabilidad informática no remediada.
- Desarrollar una herramienta informática para consultar una base de datos de vulnerabilidades susceptibles a explotación y que exponga el impacto que tiene la no remediación de puntos débiles de un software.

1.7. PREGUNTA PROBLEMA

¿Es necesaria la remediación de las vulnerabilidades informáticas presentes en software empleado en las empresas y en equipos de uso personal?

1.8. LÍNEA DE INVESTIGACIÓN

La investigación realizada hace parte de las líneas de investigación de Seguridad Informática e Ingeniería de Software, declaradas por el grupo Gridntic de la Fundación Universitaria Los Libertadores en el Departamento Administrativo de Ciencia, Tecnología e Innovación de Colombia, Colciencias.

CAPITULO 2. MARCO REFERENCIAL

2.1. ESTADO DEL ARTE

2.1.1. ANTECEDENTES. VULNERABILIDAD CVE 2017-143

Medios de comunicación en Colombia y el mundo, presentaron en primicia lo que se considera uno de los ataques informáticos con más atención mediática de la última década. Se trató de un ataque que la oficina europea de policía describió sin precedentes y que varios diarios como El Espectador, en Colombia, reportaron como el uso masivo de un programa informático de tipo ransomware que evita el acceso a los usuarios a sus datos digitales de carácter empresarial, personal e incluso de gobierno. (El Espectador, 2017). En la ilustración 4 se presenta un mapa de los países afectados publicado por Intel Security en junio de 2017.

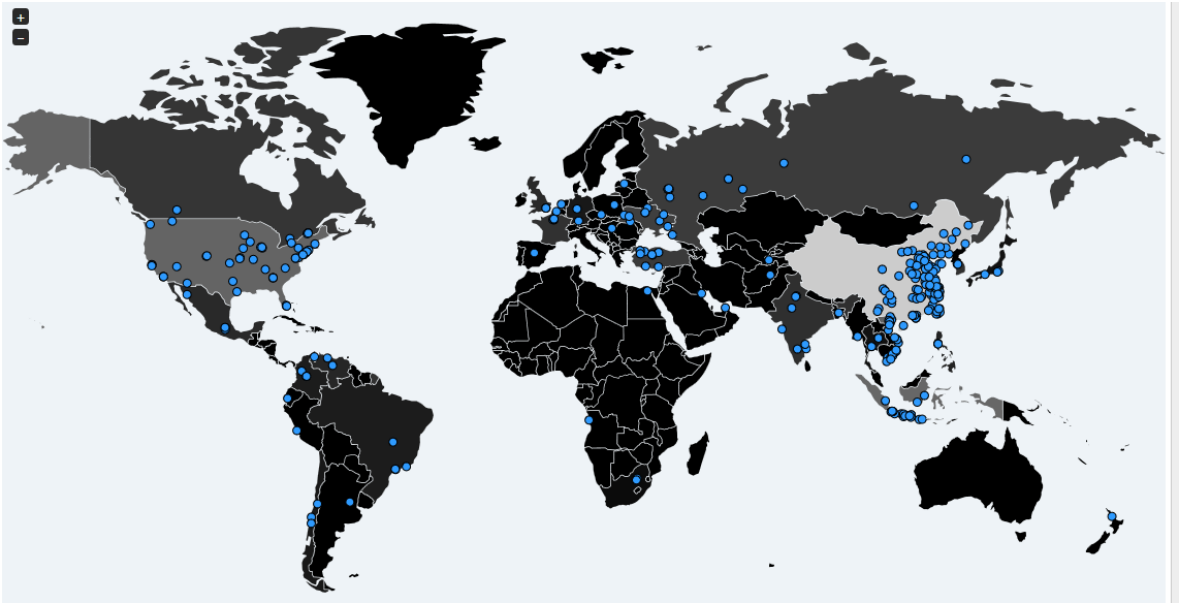


Ilustración 4. Mapa de los países con equipos de cómputo afectados por WanaCryptor.

El funcionamiento exitoso de WanaCryptor se debe a 6 factores principalmente, 4 de ellos relacionados al objeto de esta investigación: el desconocimiento en la atención de vulnerabilidades informáticas:

- WanaCryptor emplea el mecanismo de uso de una de las herramientas que el colectivo Shadow Brokers, también conocidos como Equation Group, robó a la

Agencia Nacional de seguridad de los Estados Unidos y que filtro a través de su repositorio git publicado en internet. (ShadowBroker, 2017).

- Las herramientas robadas a la Agencia Nacional de seguridad de los Estados Unidos estaban diseñas para la explotación de vulnerabilidades informáticas de tipo zero-day, y se desconoce la antigüedad que tiene esas vulnerabilidades. (TheHackerNews, 2016).
- Para el caso especial de WanaCryptor se empleó el mecanismo de una de las herramientas publicadas por Shadow Brokers para explotar 5 vulnerabilidades CVE-2017-143, CVE-2017-144, CVE-2017-145, CVE-2017-146 y CVE-2017-147.
- Las 5 vulnerabilidades estaban presentes en el protocolo SMBv1 de los sistemas Microsoft Windows, el cual representa el 91.72% de equipos conectados a Internet, y la explotación más efectiva se presentaba en las versiones de Windows 7 las cuales representan casi el 50% de esa cifra. (Netmarketshare, 2017).
  En la ilustración 5 se presenta la línea de tendencia de los sistemas operativos para equipos de escritorio usados en Internet según netmarketshare.com.

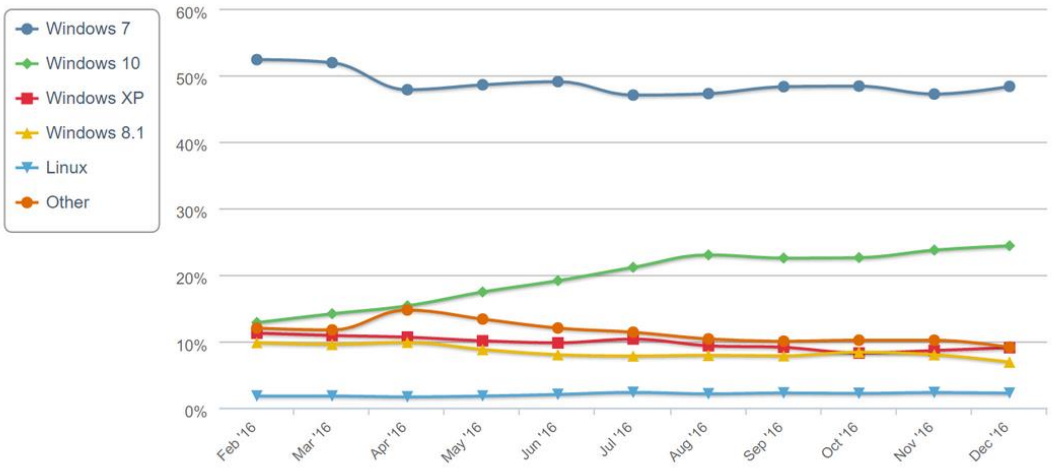


Ilustración 5. Desktop Top Operating System Share Trend.

- El desconocimiento de la responsabilidad. Después de que las vulnerabilidades fueran explotadas infectando equipos con WanaCryptor, Microsoft criticó y responsabilizó a las agencias gubernamentales por acumular la identificación de vulnerabilidades y por mantenerlas en secreto. "Un escenario equivalente con armas convencionales sería el de los militares estadounidenses con algunos de sus misiles Tomahawk robados. Este ataque más reciente representa un vínculo completamente involuntario pero desconcertante entre las dos formas más graves de las amenazas de ciberseguridad en el mundo de

hoy - acción del estado-nación Y la acción criminal organizada ", escribió el presidente y director jurídico de Microsoft, Brad Smith en su blog. (Forbes, 2017).

Por su parte el Gobierno de Estados Unidos publico noticias y recomendaciones relacionadas con WanaCryptor, pero al 7 de junio de 2017, como se muestra en la ilustración 6, en el repositorio de noticias del Gobierno de los Estados Unidos no se encuentran reportes donde se asuma una responsabilidad directa. (Gobierno de los Estados Unidos, 2017).

Ilustración 6. Noticias del gobierno de Estados Unidos

Dentro de las víctimas de los ataques de WanaCryptor se encuentran incluso servidores y equipos con sistemas operativos que Microsoft ya no soporta por el fin de su ciclo de vida. (Microsoft Corporation, s.f.).

Entre julio de 2016 y mayo de 2017 se ha revelado una disminución de uso en equipos con estos sistemas operativos sin soporte, sin embargo, la cifra aún no llega a un cero por ciento como se indica en la ilustración 7. (Netmarketshare, 2017).

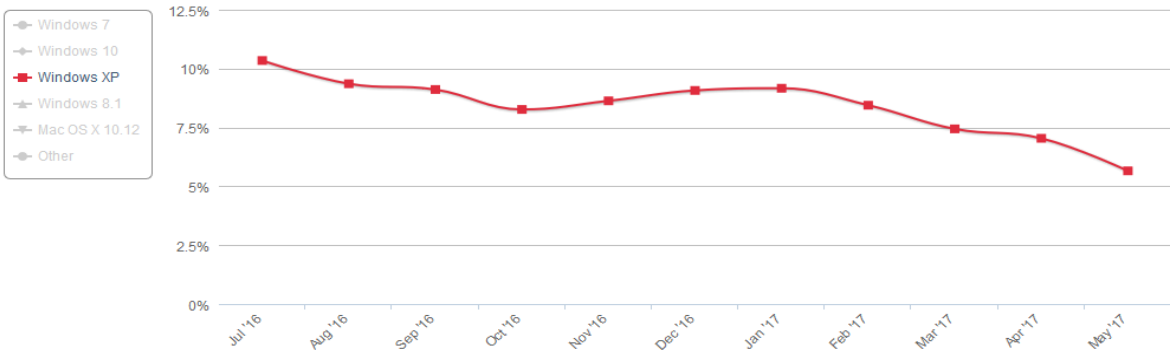


Ilustración 7. Uso del sistema operativo sin soporte, Windows XP, entre julio de 2016 y mayo de 2017

- La no remediación de las vulnerabilidades a pesar de tener las correcciones disponibles. Microsoft el 13 de marzo de 2017, 3 meses después de conocerse las vulnerabilidades CVE-2017-143 a la CVE-2017-148, publico su boletín de seguridad MS17-010.
  El boletín publica el enlace de descarga de las actualizaciones que remedian la vulnerabilidad en cada sistema operativo desde Windows XP y 2003.

- Cuando un computador no tenía aplicadas las contramedidas publicadas por Microsoft, y era víctima del ataque informático, se le exigía un pago económico para recuperar la información cifrada, tal como se muestra en las ilustraciones 8 y 9:

Ilustración 8. Imagen de WanaCryptor solicitando dinero virtual para el rescate de la información cifrada

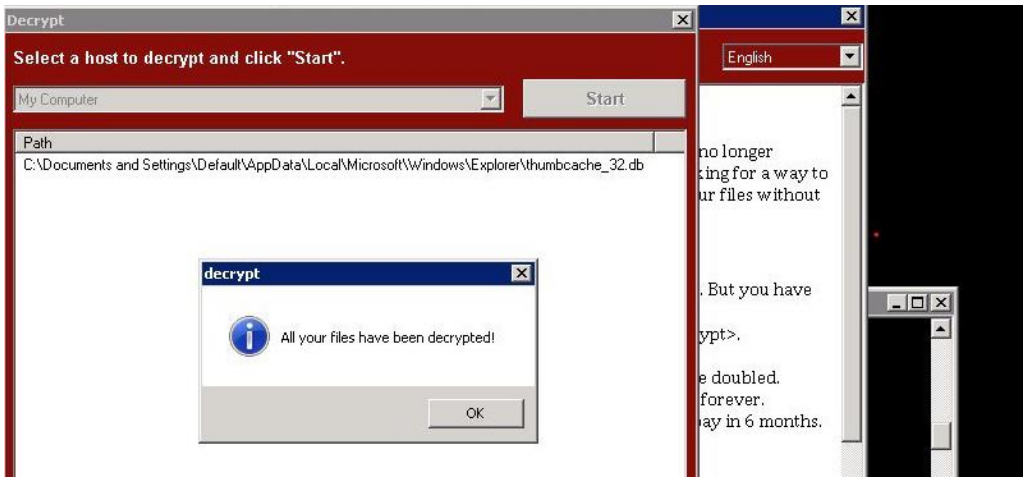


Ilustración 9. Imagen dos de WanaCryptor solicitando rescate de la información

- En la ilustración 10 se muestra la estructura del funcionamiento del ataque de WanaCryptor aprovechándose de las vulnerabilidades informáticas del protocolo SMBv1.

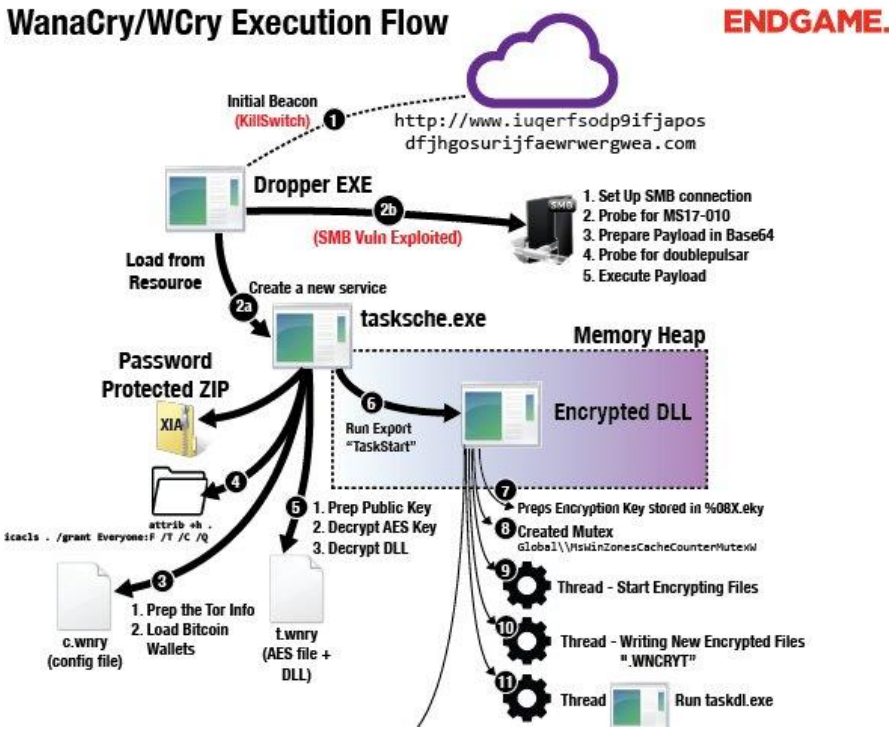


Ilustración 10. Flujo de la ejecución de WanaCryptor

- Existen contramedidas a vulnerabilidades parecidas:

  Aplicar la remediación MS08-067 para servidores y equipos Windows 2000, 2003, XP, Vista y 2008. Descargar el parche correspondiente a los sistemas operativos en la siguiente dirección: https://technet.microsoft.com/en-us/library/security/ms08-067.aspx.

- Las contramedidas a vulnerabilidades CVE-2017-143, CVE-2017-144, CVE-2017-145, CVE-2017-146, CVE-2017-147, CVE-2017-148:

  Aplicar la remediación publicada por Microsoft en su boletín MS17-010. Descargar y aplicar el parche de Microsoft de acuerdo al sistema operativo que se muestra en la ilustración 11. https://technet.microsoft.com/en-us/library/security/ms17-010.aspx.

SISTEMA OPERATIVO	PARCHE A INSTALAR
Windows XP	KB4012598
Windows XP Embedded	KB4012598
Windows XP Embedded	KB4012598
Windows XP x64 Edition	KB4012598
Windows Server 2003, Windows Server 2003, Datacenter Edition	KB4012598
Windows Server 2003, Windows Server 2003, Datacenter Edition	KB4012598
Windows Vista Service Pack 2	KB4012598
Windows Vista x64 Edition Service Pack 2	KB4012598
Windows Vista	KB4012598
Windows Vista	KB4012598
Windows Server 2008	KB4012598
Windows Server 2008	KB4012598
Windows Server 2008	KB4012598
Windows Server 2008 for 32-bit Systems Service Pack 2	KB4012598
Windows Server 2008 for x64-based Systems Service Pack 2	KB4012598
Windows Server 2008 for Itanium-based Systems Service Pack 2	KB4012598
Windows Server 2008 for 32-bit Systems Service Pack 2 (Server Core installation)	KB4012598
Windows Server 2008 for x64-based Systems Service Pack 2 (Server Core installation)	KB4012598
Windows 7 for 32-bit Systems Service Pack 1	KB4012212
Windows 7 for 32-bit Systems Service Pack 1	KB4012215
Windows 7 for x64-based Systems Service Pack 1	KB4012212
Windows 7 for x64-based Systems Service Pack 1	KB4012215
Windows Server 2008 R2 for x64-based Systems Service Pack 1 (Server Core installation)	KB4012212
Windows Server 2008 R2 for x64-based Systems Service Pack 1 (Server Core installation)	KB4012215
Windows Server 2008 R2 for x64-based Systems Service Pack 1	KB4012212
Windows Server 2008 R2 for x64-based Systems Service Pack 1	KB4012215
Windows Server 2008 R2 for Itanium-based Systems Service Pack 1	KB4012212
Windows Server 2008 R2 for Itanium-based Systems Service Pack 1	KB4012215

Ilustración 11. Lista 1 de contramedidas para vulnerabilidades relacionadas con WanaCryptor en sistemas Windows

Para Windows 8, 8.1, 10, 2012, 2012 R2, 2016 aplicar TODAS las actualizaciones pendientes que tenga el sistema operativo y revisar que queden instalados los siguientes parches relacionados en la ilustración 12.

SISTEMA OPERATIVO	PARCHE A INSTALAR
Windows Server 2012	KB4012214
Windows Server 2012	KB4012217
Windows Server 2012 (Server Core installation)	KB4012214
Windows Server 2012 (Server Core installation)	KB4012217
Windows 8	KB4012598
Windows 8	KB4012598
Windows RT 8.1	KB4012216
Windows 8.1 for 32-bit Systems	KB4012213
Windows 8.1 for 32-bit Systems	KB4012216
Windows 8.1 for x64-based Systems	KB4012213
Windows 8.1 for x64-based Systems	KB4012216
Windows Server 2012 R2	KB4012213
Windows Server 2012 R2	KB4012216
Windows Server 2012 R2 (Server Core installation)	KB4012213
Windows Server 2012 R2 (Server Core installation)	KB4012216
Windows 10 for 32-bit Systems	KB4012606
Windows 10 for x64-based Systems	KB4012606
Windows 10 Version 1511 for 32-bit Systems	KB4013198
Windows 10 Version 1511 for x64-based Systems	KB4013198
Windows 10 Version 1607 for 32-bit Systems	KB4013429
Windows 10 Version 1607 for x64-based Systems	KB4013429
Windows Server 2016 for x64-based Systems	KB4013429
Windows Server 2016 for x64-based Systems(Server Core installation)	KB4013429

Ilustración 12. Lista 2 de contramedidas para remediar vulnerabilidades relacionadas con WanaCryptor en sistemas Windows

- Definir reglas en las herramientas de seguridad con las que cuenta la empresa para alertar tráfico o intentos de explotación de alguna de las siguientes vulnerabilidades: CVE-2017-143, CVE-2017-144, CVE-2017-145, CVE-2017-146, CVE-2017-147, CVE-2017-148.

- Deshabilitar el SMB1.0/CIFS File Sharing Support como se expone en la ilustración 13:

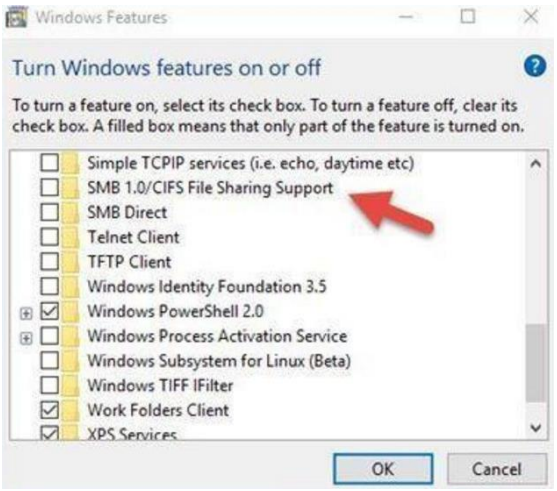


Ilustración 13. Deshabilitar el SMB1.0/CIFS File Sharing Support.

2.1.2. ASPECTOS LEGALES

La atención de vulnerabilidades informáticas con su proceso de remediación sirve de mecanismo de protección de los elementos de TI que trabajan en función de los datos personales, los cuales según las leyes Colombianas deben ser protegidos.

En la legislación colombiana, la Ley estatutaria 1581 del 17 de octubre de 2013, reglamentada parcialmente por el Decreto Nacional 1377 de 2013, dictan disposiciones generales para la protección de datos personales, decretada con el objeto de desarrollar el derecho constitucional que tienen todas las personas a conocer, actualizar y rectificar las informaciones que se hayan recogido sobre ellas en bases de datos o archivos, y los demás derechos, libertades y garantías constitucionales a que se refiere el artículo 15 de la Constitución Política; así como el derecho a la información consagrado en el artículo 20 de la misma, los principios y disposiciones contenidas en la ley aplica una protección de los datos personales registrados en cualquier base de datos que los haga susceptibles de tratamiento por entidades de naturaleza pública o privada.

2.2. MARCO TEÓRICO

En un mundo favorecido por el uso de la información como elemento activo dentro de actividades empleadas por seres humanos prevalecerán intereses de diversos aspectos, y dependiendo las razones, particularmente las diferentes generaciones estimaran esfuerzos para preservar esa información, para transmitirla o evitar su transmisión, para evitar su uso o para usarla.

En términos de sociedad, desde las últimas décadas se innova e implementan mecanismos para obtener un beneficio en el uso de la información que favorezca el desarrollo de actividades de grupos sociales que bajo una razón social e interés económico particular, transmitirá y preservará su información como uno de sus activos más importantes. Agentes externos (personas o grupos sociales) testifican los esfuerzos que personas o grupos sociales (empresas, entidades gubernamentales, entre otros) hacen para la transmisión de su información. (Corletti, Seguridad en redes, 2016).

Dentro de los mecanismos para la transmisión de la información desarrollados en el pasado, distintas sociedades fueron testigos de la implementación de la red Arpanet, la cual se intensifico por intereses militares que buscaban la preservación de información con datos de seguridad nacional de Estados Unidos. Unos de los principales objetivos de la red Arpanet buscaban que mediante la transmisión de copias de la información en distintas ubicaciones geográficas, se redujera el riesgo

de pérdida de información que afectara las operaciones militares de Estados Unidos. (Corletti, Seguridad por Niveles, 2011).

Luego del hecho histórico en el que el desarrollo y la investigación permitieran la transmisión de información digital de un lugar geográfico a otro, diferentes agentes externos crean la necesidad de preservar no solamente información militar de una nación, si no aplicar también este concepto a diferente información situada en diferentes ubicaciones; con esa premisa se lleva a cabo un modelado e implementación de una serie de elementos tecnológicos, procedimentales y humanos.

El modelado e implementación de los elementos que permiten la transmisión de la información, da origen a estándares internacionales, que de acatarse proporcionan mecanismos que permiten la comunicación entre 2 o más elementos que quieran realizar un proceso de transmisión de información.

Para implementar estos estándares a distintos grupos sociales a nivel mundial, se recurre a la necesidad de publicar los protocolos que soportan el modelo, lo que hace que cualquier agente externo tenga un íntimo entendimiento de cómo funcionan las redes de información. Diversos individuos y grupos sociales de estos agentes externos, motivados por algún interés en particular hacen uso del estándar para apropiarse de la información confidencial de cualquier ámbito social, económico, cultural y geográfico en el mundo, ya sea con el propósito de conocer su contenido, alterarlo o indisponer los mecanismos de acceso y transmisión de esa información.

Para evitar que esta última sección de los agentes externos tenga éxito en la apropiación no autorizada de la información, los estándares también protocolizan nuevos mecanismos para la protección de la información, y es en este sentido en el que hace hincapié el concepto de seguridad de la información.

La seguridad de la información incluye una serie de especialidades que también modelan e implementan una serie de elementos tecnológicos, procedimentales y humanos con la intención de procurar la interceptación de canales y redes donde se transmite la información.

El modelo más empleado para las conexiones es el modelo OSI, el cual compromete una serie de elementos procedimentales y técnicos, que para favorecer a su entendimiento e implementación se dividió en 7 niveles:

- Nivel 1 Físico
- Nivel 2 Enlace
- Nivel 3 Red
- Nivel 4 Transporte
- Nivel 5 Sesión
- Nivel 6 Presentación

- Nivel 7 Aplicación

Los procesos técnicos y procedimentales que soportan el modelo OSI son públicos para entender su funcionamiento en las redes y para implementar el modelo. En ese entendimiento del modelo y de cada una de sus capas los profesionales informáticos, de telecomunicaciones, de redes o a fin dan soporte a sus redes corporativas, domesticas, virtuales, privadas o redes conmutadas, entre otras.

Grupos sociales o individuos alrededor del mundo motivados por alguna razón social, ética o filosófica y que tienen el entendimiento del modelo OSI, realizan procesos técnicos y sociales para apropiarse de la información confidencial de cualquier ámbito social, económico, cultural y geográfico en el mundo, ya sea con el propósito de conocer su contenido, alterarlo o indisponer los mecanismos de acceso y transmisión de esa información.

Para evitar esta apropiación se requiere protocolizar nuevos mecanismos tecnológicos, procedimentales y humanos con la intención de evitar la interceptación de redes donde se transmite información.

El modelo de seguridad por niveles es una abstracción de las capas del modelo OSI, llevadas a la implementación de elementos de seguridad en cada uno de los 7 niveles. Este marco toma recursos, definiciones y citas textuales del curso de Ciberseguridad de CISCO y de los libros Seguridad en redes (2016) y Seguridad por niveles (2011) de Alejandro Corletti Estrada:

- ***Nivel 1 Físico:***

  Este es el primer escalón de la seguridad por niveles que consiste en asegurar los elementos físicos que intervienen en las comunicaciones y los espacios geográficos donde se encuentran los medios de transmisión físicos.

  Para aplicar un mecanismo de seguridad necesario para este nivel o capa se identifican los elementos físicos principales lo más detalladamente posible, y luego esta información debe empezar a ser parte de un Sistema de Gestión de la Seguridad de la Información.

  Cuando se tenga incluidos estos datos en el Sistema de Gestión de Seguridad de la Información, es más sencillo detectar las vulnerabilidades informáticas de la capa relacionadas al medio físico donde se establece la conexión. Una vulnerabilidad es un punto débil en la implementación de hardware o software, que de materializarse puede incurrir en una amenaza o riesgo.

  La capa física presenta problemas de seguridad que afectan la confidencialidad y el control de acceso. En este nivel es común encontrar

vulnerabilidades materializados con ataques a líneas P2P mediante el desvío de cables usados para la conexión hacia otros sistemas, también se encuentra la interceptación de las comunicaciones (ataques Man In The Middle), entre otros.

La implementación de seguridad en este nivel se encuentra en el canal de comunicaciones que se emplee, entre lo que destaca:

- Tener medios de transmisión físicos propios la información tendrá un acceso no público, lo cual incrementa la seguridad y evita vulnerabilidades de interceptación.
- En infraestructuras con tecnología de cable de cobre es difícil detectar la interceptación física por lo que en redes convergentes se propone el uso de la fibra óptica, la cual se considera casi imposible de interceptar, ya que la interceptación implica cortar el canal de la fibra.
- En distintas ondas de radio terrestre hay varias posibilidades de implementación como señal distribuida multipunto. Cualquiera de estas implementaciones es interceptable.

Dentro de la implantación de seguridad en el nivel, es aconsejable el uso de auditorías, en las que se preste atención a detalles como:

- Marquillar adecuadamente los patch panel y demás elementos físicos para asegurarse de entender adecuadamente todos los nombres y ubicaciones.
- Control a los gabinetes de comunicaciones, que determine su ubicación y seguridad de acceso a esa ubicación, llaves de acceso, entre otros.
- Planos que identifiquen los conductos que siguen los medios físicos como Zócalos, techos falsos, cable canal, o cualquiera adicional.
- Documentación de la topología de la red.
- Mecanismos establecidos que prevén una expansión de la red.
- Identificación y depuración de las puertas de acceso a la red.

- ***Nivel 2 Enlace***

En este nivel o capa, igual que al anterior (nivel físico) las vulnerabilidades informáticas se encontraran en el medio sobre el que se hace la transmisión de datos.

Este nivel tiene gran importancia por que encapsula los datos de todos los niveles anteriores, por lo que es susceptible a una interceptación para realizar procesos de desencapsulamiento, donde básicamente están todos los datos que se transmiten en una red.

Para asegurar esta capa se emplean herramientas que funcionan como Analizadores de protocolos.

Dentro de la implantación de seguridad en el nivel, es aconsejable el uso de auditorías, en las que se preste atención a detalles como:

- Tener la lista completa del direccionamiento MAC de las tarjetas de red.
- Como el trabajo físico de esta capa consta de aprender por qué puerto se conecta cada dirección MAC para conmutar el tráfico es necesario hacer que los dispositivos involucrados sean administrables de forma remota o local para efectuar las configuraciones necesarias de acuerdo a las necesidades de la red.
- Analizar el tráfico para evitar problemas relacionados al performance empleado por el Broadcast.
- Analizar las colisiones.
- Detectar que en la red no existan Sniffers ni analizadores de protocolos.

- ***Nivel 3 Red***

En esta capa prevalece la función de manejar las rutas. Usa el protocolo IP.

A nivel del modelo se tiene la Capa de Internet, en esta capa se puede realizar cualquier ataque que afecte el protocolo IP. Se incluyen como ataques contra esta capa las técnicas de sniffing, la suplantación de mensajes, la modificación de datos, los retrasos de mensajes y la denegación de mensajes.

Cualquier atacante puede suplantar un paquete si indica que proviene de otro sistema. La suplantación de un mensaje se puede realizar, por ejemplo, dando una respuesta a otro mensaje antes de que lo haga el suplantado. En esta capa, la autenticación de los paquetes se realiza a nivel de máquina (por dirección IP) y no a nivel de usuario. Si un sistema suministra una dirección de máquina errónea, el receptor no detectará la suplantación. Para conseguir su objetivo, este tipo de ataques suele utilizar otras técnicas como la predicción de números de secuencia TCP, el envenenamiento de tablas caché, entre otros. Por otro lado, los paquetes se pueden manipular si se alteran sus datos y se reconstruyen de forma adecuada los controles de las cabeceras. Si esto es posible, el receptor será incapaz de detectar el cambio.

Dentro de la implantación de seguridad en el nivel, es aconsejable el uso de auditorías, en las que se preste atención a detalles como:

- Controlar las contraseñas de los router.
- Inhabilitar las configuraciones que muchas veces en forma innecesaria quedan habilitados y no se emplean (Broadcast Subnetting, local loop, puertos, rutas, entre otros).
- Tener una copia de seguridad de las configuraciones de los routers.
- Tener preparadas las listas de control de acceso.

- Auditorías de tráfico ICMP.
- Auditoría ARP: El ataque ARP es uno de los más difíciles de detectar pues se refiere a una asociación incorrecta de direcciones MAC e IP, por lo tanto se debe analizar todas las tramas que circulan por la red y comparar permanentemente las mismas con un patrón de referencia válido.

- ***Nivel 4 Transporte***

  La capa de transporte garantiza la calidad del servicio cuando la aplicación lo requiera.

  En cuanto a los mecanismos de seguridad incorporados existe una serie de ataques que aprovechan ciertos defectos en su diseño.

  Una de las vulnerabilidades más graves contra estos mecanismos de control puede permitir la posibilidad de interceptación de sesiones TCP establecidas, con el objetivo de secuestrarlas y dirigirlas a otros equipos.

  Dentro de la implantación de seguridad en el nivel, es aconsejable el uso de auditorías, en las que se preste atención a detalles como:

  - Auditorías de establecimientos y cierres de sesión:
    - Ataques LAND.
    - Inundación de SYN.
    - Auditoría en Puertos UDP y TCP: Dentro del encabezado de TCP o UDP se encuentran los campos Puerto Origen y Puerto Destino, los cuales son uno de los detalles más importantes a auditar dentro de una red pues a través de ellos, se puede ingresar a un Host y operar dentro de este.
      - Auditoría de Troyanos: Se deberá prestar especial atención a este tipo de actividades, lo cual como se acaba de mencionar en el punto anterior, implica procesos muy similares.

- ***Nivel 5 Sesión, Nivel 6 Presentación, Nivel 7 Aplicación***

  Una vez superado el nivel cuatro (transporte), todas las funciones y/o servicios se orientan en la capa de aplicación del modelo en las capas 5, 6, y 7.

  En la capa de aplicación se encuentra la mayor cantidad de protocolos existentes, por lo que para aplicar seguridad en esta capa es necesario implementar mecanismos de seguridad en todos ellos.

  - Auditoría de servidores de correo, Web, TFP y TFP, Proxy.
  - Limitar el acceso a áreas específicas de esos servidores.

- Especificar las listas o grupos de usuarios con sus permisos correspondientes. Prestar especial atención a la cuenta “Anónimos” y a toda aquella que presente nombres de fácil aprovechamiento.
- Requerir contraseñas seguras.
- Siempre controlar los archivos.log.
- Deshabilitar índices de directorios.
- Deshabilitar todos los servicios de red que no sean empleados por el servidor.
- Auditorías de accesos remotos.
- Auditoria a ráfagas de e-mail.
- Auditoria a bombardeos de SYSLOG y SNMP.
- Configurar credenciales de acceso al FTP (Puerto TCP 20 y 21)
- Prestar especial atención a la configuración de los DNS, en especial al tráfico TCO sobre el puerto 53.

En su academia virtual de aprendizaje, CISCO, empresa dedicada a la industria de Tecnologías de la información y la comunicación, definiendo la ciberseguridad expone que la red de la información electrónica conectada en Internet se ha convertido en una parte integral de nuestra vida cotidiana, que todos los tipos de organizaciones, públicas y privadas como instituciones médicas, financieras o de gobierno utilizan esta red para funcionar de manera eficaz avocándose a un uso necesario de elementos de las tecnologías de la información y las comunicaciones.

Se utiliza la red para recopilar, normalizar, tratar, procesar, almacenar y compartir grandes cantidades de información digital, que en ocasiones requiere incluso una protección ante perdida de datos, perdida de acceso o acceso no autorizado a estos datos. En esa apreciación CISCO define la ciberseguridad como el esfuerzo que se ejerce para proteger estos datos e información digital.

Este esfuerzo que se ejerce para proteger la información es una responsabilidad individual y colectiva, porque hay datos que son de carácter personal y colectivo. El ingeniero español Pablo González, en los primeros capítulos en su obra Ethical Hacking – Teoría y práctica para la realización de un pentesting, expone porqué para las empresas es necesario pensar en el esfuerzo de protección que busca la ciberseguridad. González, expone que la Internet es una herramienta pero también una amenaza que acecha a empresas en el mundo. Aunque el principal objetivo de una empresa es generar beneficios mediante el ofrecimiento de productos y servicios, es ineludible el hecho en que en el ejercicio de su actividad económica las empresas usan la red y comparten su información digital para la realización de compras, de negocios, de venta de activos, de consulta de información, modificación de nóminas, entre otros.

CISCO por su parte también sensibiliza la protección de los datos de carácter personal. A medida que las personas pasan más tiempo en línea a través de la red, su identidad es pública en el ciberespacio y puede ser visible para sus amigos

y familiares o incluso con personas con las que no interactúan. La difusión de la información personal con o sin consentimiento y aprobación del titular es lo que en el ciberespacio se conoce como privacidad, pero puede existir un vacío en la definición de quien es el responsable de la privacidad.

En la legislación colombiana, la Ley estatutaria 1581 del 17 de octubre de 2013, reglamentada parcialmente por el Decreto Nacional 1377 de 2013, y por la cual se dictan disposiciones generales para la protección de datos personales, decreta que con el objeto de desarrollar el derecho constitucional que tienen todas las personas a conocer, actualizar y rectificar las informaciones que se hayan recogido sobre ellas en bases de datos o archivos, y los demás derechos, libertades y garantías constitucionales a que se refiere el artículo 15 de la Constitución Política; así como el derecho a la información consagrado en el artículo 20 de la misma, los principios y disposiciones contenidas en la ley aplica una protección de los datos personales registrados en cualquier base de datos que los haga susceptibles de tratamiento por entidades de naturaleza pública o privada.

La ley incluso aplica al tratamiento de datos personales efectuado en territorio colombiano o cuando al Responsable del Tratamiento o Encargado del Tratamiento no establecido en territorio colombiano le sea aplicable la legislación colombiana en virtud de normas y tratados internacionales.

En resumen, está ley estatutaria 1581 pretende la protección de los datos personales, CISCO sensibiliza a las personas sobre el tiempo e información que se publica mientras se está en línea en internet y Pedro González expone a las empresas porqué es necesaria la protección de la información empresarial.

La Real academia de la lengua española define a las credenciales como un documento que acredita a una persona para desempeñar una determinada función. En la ciberseguridad una credencial es el mecanismo para proteger la información digital mediante la acreditación de una persona en un sistema. CISCO expone en su curso de fundamentos de ciberseguridad que las credenciales en la red son valiosas. Por ejemplo, aprovechando el hecho de las credenciales otorgan el acceso a las cuentas de una persona en American Airlines, un grupo de delincuentes informáticos robaron aproximadamente 10.000 cuentas de esa aerolínea para reservar ilegalmente vuelos gratuitos. (CISCO, 2017).

El robo de credenciales no representa la única amenaza en el mundo de la ciberseguridad. En el libro Ethical Hacking, Pedro González expone diferentes actos en el mundo de la ciberseguridad como los ataques cibernéticos a través de la explotación de vulnerabilidades informáticas que repercuten en denegación de servicio de sistemas, en la infección de amenazas persistentes avanzadas, en la perdida y fuga de información, ente otras.

En colaboración con los expertos German Sánchez, José Soriano, Jhonattan Fiestas, Umberto Schiavo y Chema Alonso, Pablo González en el libro Pentesting con Kali 2.0 incluso expone la técnica detrás de la explotación de las vulnerabilidades informáticas mediante la obtención de información digital mediante la auditoria de protocolos informáticos configurados de manera insegura, análisis de vulnerabilidades y ataques de contraseñas, uso de exploits, payloads, auxiliares, uso de herramientas como meterpreter o mimikatz.

Para evitar la explotación de vulnerabilidades informáticas se hace un proceso de identificación, atención y remediación de vulnerabilidades. Este es un aspecto que cobra una gran importancia para evitar problemas de ciberseguridad de carácter internacional que aprovecha las debilidades de los sistemas para secuestrar o apoderarse de la información de las empresas o para denegar los servicios de las plataformas tecnológicas. Sin embargo, a pesar de este esfuerzo, se pueden encontrar empresas que emplean software vulnerable y que no hacen una atención periódica ni oportuna a las vulnerabilidades que tienen; incluso en algunos casos se ignora la afectación que tiene la no atención de vulnerabilidades informáticas en la privacidad de los datos personales y privacidad de las personas.

La investigación expuesta en este documento representa una propuesta para dejar de desconocer la importancia que tiene la atención de vulnerabilidades informáticas, desde el punto de vista de la remediación –proceso de ejecución que consiste en actualizar, desinstalar o configurar el software para corregir vulnerabilidades informáticas.

El FIRST es la principal organización en el mundo que dicta los estándares para la respuesta a incidentes de seguridad de la información. La afiliación a FIRST permite a los equipos de respuesta a incidentes responder con mayor eficacia a incidentes tanto reactivos como proactivos. FIRST reúne una variedad de equipos de respuesta para organizaciones gubernamentales, comerciales y educativas. Tiene como objetivo fomentar la cooperación y promover el intercambio de información entre los miembros y la comunidad en general. (First, 2017)

El Common Vulnerability Scoring System (CVSS) es un estándar para evaluar la gravedad de las vulnerabilidades de seguridad de un sistema informático. Intenta asignar las puntuaciones de gravedad a las vulnerabilidades, permitiendo priorizar las remediaciones y los recursos de acuerdo a la amenaza. Las puntuaciones se calculan sobre la base de una fórmula que depende de varias métricas que se aproximan a la facilidad de aprovechamiento y al impacto que tiene la materialización de esa debilidad. Las puntuaciones oscilan entre 0 y 10, siendo 10 las más graves. Si bien muchos utilizan sólo el puntaje CVSS para determinar la gravedad, también existen puntajes temporales y ambientales, para tener en cuenta la disponibilidad de mitigaciones y el grado de extensión de los sistemas vulnerables dentro de una organización. La versión actual de CVSS (CVSSv3.0) fue lanzada en junio de 2015. (First, 2017).

La investigación realizada por el Consejo Asesor Nacional de Infraestructura (NIAC) en 2003/2004 llevó al lanzamiento de CVSS versión 1 en febrero de 2005, con el objetivo de "ser diseñado para proporcionar clasificaciones abiertas y universalmente estándar de vulnerabilidades de software". Este borrador inicial no había sido objeto de revisión por parte de otros organismos. En abril de 2005, la NIAC seleccionó el Foro de Equipos de Respuesta y Seguridad de Incidentes (FIRST) para convertirse en el custodio de CVSS para su desarrollo futuro. (First, 2017).

El trabajo sobre CVSS versión 2 comenzó en abril de 2005 con la especificación final que se lanzó en junio de 2007. La retroalimentación posterior dio lugar a trabajos que comenzaron en CVSS versión 3 en 2012, terminando con CVSSv3.0 lanzado en junio de 2015.

Terminología: La evaluación CVSS mide tres áreas de preocupación:

- Métricas de base para las cualidades intrínsecas a una vulnerabilidad
- Métricas temporales para características que evolucionan durante la vida útil de la vulnerabilidad
- Métricas ambientales para vulnerabilidades que dependen de una implementación o entorno particular

Se genera una puntuación numérica para cada uno de estos grupos métricos. Una cadena vectorial (o simplemente "vector" en CVSSv2), representa los valores de todas las métricas como un bloque de texto.

El modelo cuantitativo de Common Vulnerability Scoring System (CVSS) permite a los usuarios ver las características de vulnerabilidad subyacentes que se utilizaron para generar las puntuaciones. Por lo tanto, CVSS es muy adecuado como un sistema de medición estándar para industrias, organizaciones y gobiernos que necesitan puntajes de impacto de vulnerabilidad preciso y consistente. La National Vulnerability Database (NVD) proporciona puntuaciones CVSS para casi todas las vulnerabilidades conocidas. (National Vulnerability Database, 2017).

En particular, la NVD admite el estándar CVSS (Common Vulnerability Scoring System) versión 2 para todas las vulnerabilidades CVE. NVD proporciona CVSS 'puntuaciones de base' que representan las características innatas de cada vulnerabilidad. En la actualidad no proporciona "puntuaciones temporales" (puntuaciones que cambian con el tiempo debido a eventos externos a la vulnerabilidad). Sin embargo, NVD proporciona una calculadora de puntuación CVSS que le permite agregar datos temporales e incluso calcular calificaciones ambientales (calificaciones personalizadas para reflejar el impacto de la vulnerabilidad en su organización). Esta calculadora contiene soporte para que las agencias gubernamentales de los Estados Unidos personalicen las puntuaciones de impacto de vulnerabilidad basadas en las clasificaciones del Sistema FIPS 199.

(National Vulnerability Database, 2017), a continuación en la ilustración 14 se presentan las entidades mencionadas en estos apartados.

Ilustración 14. Bases de datos de vulnerabilidades

La detección de vulnerabilidades sobre infraestructuras tecnológicas se puede realizar a través de un scanner de vulnerabilidades informáticas como Nexpose, Nessus, Acunetix o McAfee Vulnerability Manager –anteriormente conocido como Foundstone y que está próximo a desaparecer. (McAfee, a Intel Company, 2015)

McAfee Vulnerability Manager realiza evaluaciones en el nivel de aplicaciones y sistemas, que incluyen la identificación de puertos abiertos y sus configuraciones, bases de datos (DB2, MySQL, Oracle, Microsoft SQL Server y Sybase), configuraciones de directivas, claves de registro, permisos de archivos y unidades, y servicios en ejecución.

El repositorio de McAfee Vulnerability Manager cuenta además con la identificación y clasificación de las vulnerabilidades provista por la *NVD (National Vulnerability Database*), un repositorio de los Estados Unidos que a partir de la lista estandarizada *CVE (Common Vulnerabilities and Exposures)* y otros organismos de atención cuantifican y miden el riesgo de cada vulnerabilidad conocida ubicándola en una de las siguientes categorías:

- ✓ **Critical:** Vulnerabilidades de más alto nivel de calificación.
- ✓ **Severe:** Vulnerabilidades de medio nivel de clasificación.
- ✓ **Moderate:** Vulnerabilidades de bajo nivel de calcificación.

Estas categorías son calculadas a partir de la medición de algunas métricas:

- ✓ Nivel de complejidad que tiene el fabricante del activo vulnerado para corregir una vulnerabilidad.
- ✓ Nivel de intrusión (capacidad que tendría un atacante o un Exploit de acceder al activo vulnerado).
- ✓ Nivel de permisos que puede tener la materialización de la vulnerabilidad, la cual permitiría la ejecución arbitraría de código.

La periodicidad con la que se actualiza el catálogo de vulnerabilidades puede variar de acuerdo a los tiempos usados para la detección, documentación, creación de scripts de identificación y creación de contramedidas que debe tener cada vulnerabilidad. Esto implica que la atención de vulnerabilidades sea un procedimiento periódico que se realiza de manera repetida sobre las infraestructuras tecnológicas.

El ICASI Common Vulnerability Reporting Framework (CVRF) es un lenguaje basado en XML que permite a diferentes partes interesadas de distintas organizaciones compartir información crítica relacionada con la seguridad en un solo formato, acelerando el intercambio de información. Funciona como un marco común y consistente para el intercambio no sólo de la información de vulnerabilidad, sino cualquier documentación relacionada con la seguridad. CVRF fue creado para llenar una brecha importante en la estandarización de la vulnerabilidad: la carencia de un marco estándar para la creación de la documentación del informe de la vulnerabilidad. Originalmente derivado del proyecto IETF (Internet Engineering Task Force), el CVRF reemplaza los muchos formatos de informes no estándar utilizados anteriormente, acelerando así el intercambio de información y el procesamiento. (ICASI, 2016).

En lo que se refiere a la aplicación de procesos de remediación, se puede usar metodologías como las que expone la norma NIST SP 800-40 revisión 3. A continuación se expone un marco teórico que toma recursos, definiciones y citas textuales de la norma NIST SP 800-40 revisión 3 publicada por el Instituto Nacional de Estándares del gobierno de Estados Unidos:

Esta norma está diseñada para ayudar a las organizaciones a comprender los conceptos básicos de las tecnologías de administración de parches de segundad ofrecido por los fabricantes de software y hardware para la remediación de vulnerabilidades informáticas. Se basa en el supuesto de que la organización tiene una capacidad de gestión de parches madura y se centra en aumentar su nivel de automatización.

Las organizaciones que buscan una guía más básica sobre el establecimiento de programas de gestión de parches que no pueden ser satisfechas con las actuales tecnologías de gestión de parches pueden usar la versión anterior de la norma, la NIST SP 800-40 Versión 2.

Esta versión de la norma fue creada para administradores de seguridad, ingenieros y otros responsables de adquirir, probar, priorizar, implementar y verificar parches de seguridad. Los auditores y otras personas que necesitan evaluar la seguridad de los sistemas también pueden encontrar esta publicación útil.

La administración de revisiones es el proceso para identificar, adquirir, instalar y verificar parches de productos y sistemas que son vulnerables. Los parches

corrigen problemas de seguridad y funcionalidad en software y firmware. Desde una perspectiva de seguridad, los parches son más a menudo de interés porque están mitigando las vulnerabilidades de fallas de software; La aplicación de parches para eliminar estas vulnerabilidades reduce significativamente las oportunidades de explotación. Además, los parches suelen ser la forma más eficaz de mitigar las vulnerabilidades de fallas de software, y son la única solución totalmente efectiva.

Los parches de seguridad sirven para otros propósitos que no sean sólo la fijación de fallas de software; También pueden agregar nuevas funciones al software y al firmware, incluidas las capacidades de seguridad. Las nuevas características también se pueden agregar a través de actualizaciones, que llevan el software o el firmware a una versión más reciente

Las actualizaciones también pueden solucionar problemas de seguridad y funcionalidad en versiones anteriores de software y firmware. Además, los proveedores suelen dejar de soportar versiones anteriores de sus productos, lo que incluye no liberar parches como contramedida a nuevas vulnerabilidades, haciendo así que las versiones no soportadas más antiguas sean menos seguras con el tiempo.

Como se ha explicado en las citas textuales de este documento, hay varios desafíos que complican la administración de parches de seguridad. Las organizaciones que no superen estos desafíos no reparan los sistemas de manera eficiente, lo que compromete los sistemas de información.

La priorización y las pruebas son pasos que hacen parte de un plan de parcheado. Lo ideal en la atención de vulnerabilidades es que una organización instale cada nuevo parche en la menor unidad de tiempo posible, sin embargo, en realidad esto no es posible porque las organizaciones tienen recursos limitados, lo que hace necesario priorizar qué parches deben instalarse antes que otros parches.

Estas complicaciones se suman al hecho de no poder a instalar parches sin probarlos primero, lo que podría causar interrupciones operacionales, donde la afectación puede dar lugar a que las organizaciones tengan miedo del proceso de remediación y aplicación de parches.

Los fabricantes de software ampliamente usado, han respondido a este conflicto mejorando la calidad de sus parches y agregando parches para sus productos. En

lugar de liberar una gran cantidad de contramedidas y parches se están liberando parches en un solo paquete en un periodo de tiempo. Esto permite a una organización realizar pruebas una vez y desplegar parches una vez, lo cual es mucho más eficiente que probar y desplegar todos los parches por separado.

Otro reto importante que menciona la norma, en la administración y despliegue de parches es que no hay un único mecanismo para aplicar parches:

• Un software puede ser capaz de actualizarse automáticamente.
• Una herramienta de administración centralizada del sistema operativo puede ser capaz de iniciar el parcheado.
• Las aplicaciones de administración de parches de terceros pueden iniciar el parcheado.
• El control de acceso a la red, las tecnologías de comprobación de la integridad y tecnologías similares pueden iniciar el parcheado.
• Un usuario puede ser capaz de dirigir manualmente el software para actualizarse.
• Un usuario puede instalar manualmente un parche o una nueva versión del software.

Está multiplicidad pueden tratar de parchear el mismo software, lo cual es problemático cuando la organización no quiere que se apliquen ciertos parches debido a problemas con esos parches, el proceso de prueba o el requerimiento que tienen los parches para reiniciar un servidor/equipo. También puede suceder que una herramienta o administrador puede asumir que otro ya está aplicando un parche o grupo de parches. Las organizaciones deben identificar todas las maneras en que los parches pueden aplicarse y actuar para resolver cualquier conflicto entre los métodos de aplicación del parche.

Otro problema implico que se encuentra en el proceso de remediación de vulnerabilidades mediante la aplicación de parches, está relacionado con los permisos de los usuarios en sesión. Si los usuarios pueden realizar cambios en el software de su equipo, como habilitar actualizaciones automáticas, deshabilitar el software de administración de parches como Windows Update en el caso de Microsoft, instalar versiones antiguas de software o desinstalar parches, pueden afectar el proceso de administración de parches.

La administración de parches de la empresa es relativamente sencilla cuando todos los equipos están completamente administrados y ejecutan aplicaciones y

sistemas operativos regulares o de una misma versión. Pero, cuando se emplean arquitecturas de host distintas, la gestión de parches es considerablemente más difícil. Por ejemplo:

• Equipos no administrados. Es mucho más difícil controlar el parche cuando los equipos no se administran de forma centralizada.

• Equipos fuera de la oficina

• Equipos de otras redes no protegidas por controles de seguridad de red de la empresa

• Dispositivos móviles. Los smartphones, tabletas y otros dispositivos móviles ejecutan sistemas operativos diferentes a los de los equipos, por lo que su parcheo es diferente. Incluso en ocasiones se hace necesario conectar el dispositivo móvil a un equipo para la descarga actualizaciones a través de ese equipo.

• Virtualización de sistemas operativos. Los parches se deben mantener por cada imagen del sistema operativo y la instantánea empleada en la virtualización.

• Las actualizaciones de firmware, como la actualización del BIOS del sistema, generalmente requieren privilegios especiales e implican procedimientos diferentes a los de otros tipos de actualizaciones como los tratados en la norma NIST SP 800-147.

Otros desafíos que se encuentran:

- Gestión de inventario de software

La administración de parches empresariales depende de tener un inventario actual y completo del software que se puede modificar, las aplicaciones y sistemas operativos instalados en cada equipo. Este inventario debe incluir también qué versión de cada pieza de software está instalada. Sin esta información, los parches correctos no pueden ser identificados, adquiridos e instalados.

- Sobrecarga de recursos

La administración de parches puede provocar que los equipos puede consumir ancho de banda de red excesivo, o si los parches proceden de un servidor

centralizado de despliegue de parches, se pueden abrumar los recursos de ese servidor.

- Efectos secundarios de la instalación

La instalación de un parche puede causar efectos secundarios. Por ejemplo alterar inadvertidamente los ajustes de configuración de seguridad existentes al agregar nuevas configuraciones. Esto puede crear un nuevo problema de seguridad en el proceso de arreglar la vulnerabilidad original a través del parcheo.

- Verificación de la implementación del parche

Un parche instalado puede no tener el efecto deseado hasta que no se reinicie el software afectado o el sistema operativo. Este hecho hace especialmente difícil examinar un equipo y determinar si un parche concreto se instaló o si está aplicado.

- La lista blanca de aplicaciones

La tecnología del software que hace parte de las listas blancas de una organización puede entrar en conflicto con las tecnologías de administración de parches, debido a que la lista blanca de aplicaciones funcionan basándose en las características conocidas de los ejecutables y otros componentes de la aplicación que pueden modificarse mediante parches o cambios de versión.

Identificar los parches que hacen faltan en los equipos se puede realizar con tres técnicas:

- El análisis basado en agentes:
  Requiere que un agente esté ejecutándose en cada equipo, con uno o más servidores que administren el proceso de revisión usando los agentes. Cada agente es responsable de determinar qué software vulnerable está instalado en el equipo, y reportarlo a los servidores para determinar qué parches nuevos están disponibles para ese equipo.

  Otras de las funciones del agente son realizar cualquier cambio de estado necesario para que los parches surtan efecto, reiniciar la aplicación parcheada, reiniciar el sistema operativo.

Cada agente se ejecuta con privilegios de administrador para que pueda realizar estas acciones. En comparación con la exploración sin agentes y la supervisión pasiva de la red, las tecnologías de administración de parches basadas en agentes se recomienda para equipos que no están en la red local todo el tiempo, como computadoras portátiles y teléfonos inteligentes.

- El análisis sin agentes.

  Tiene uno o más servidores que realizan el escaneo de red de cada host que se va a parchear y determinar qué parches necesitan cada equipo. La exploración sin agente requiere que los servidores tengan privilegios administrativos en cada equipo, de modo que puedan devolver resultados de escaneo más precisos y tengan la capacidad de instalar parches e implementar cambios de estado en los hosts.
  La principal ventaja del análisis sin agente es que no requiere la instalación y ejecución de un agente en cada host.

- El monitoreo de redes pasivas.

  Supervisan el tráfico de la red local para identificar las aplicaciones y en algunos casos los sistemas operativos vulnerables. Estas tecnologías pueden ser eficaces para identificar hosts que no están siendo mantenidos por otras soluciones de administración
  La principal desventaja de la supervisión de red pasiva es que sólo funciona con software donde se puede identificar la versión basada en su tráfico de red suponiendo que no está cifrado.

2.3. MARCO CONCEPTUAL

La atención de vulnerabilidades produce:

- Contramedidas: Actualizaciones, parches o nuevas configuraciones que proveen una solución completa o parcial de una o más vulnerabilidades.

- Exploits: Conjunto de programas que buscan materializar una vulnerabilidad.

- Zero day: Es una vulnerabilidad que a pesar de no ser publica si tiene un Exploit que se aprovecha de ella.

Se usan para materializar vulnerabilidades que no están catalogadas, o que su reparación no está adicionada en los repositorios de la lista CVE (Common Vulnerabilities and Exposures) ni en el repositorio NVD (National Vulnerability Database) por lo que consideran amenazas tecnológicas de alta criticidad.

Otros conceptos relacionados con la seguridad

- Ciberespacio: Es el ambiente físico y virtual compuesto por host (equipos susceptibles a tener una conexión a una red) y el software que emplea en redes de telecomunicaciones o de datos.

- Ciberseguridad: Es el esfuerzo realizado para ejercer controles técnicos que protejan la información contenida en un host (dispositivo susceptible a tener una conexión de red) o que circule a través de una red de equipos.

- Ciberdefensa: Es la capacidad de un Estado de gobierno para prevenir, detectar y neutralizar amenazas o un acto hostil de naturaleza cibernética que afecte la soberanía del estado.

- Vulnerability Scanner: Los scanner de vulnerabilidades son programas informáticos diseñados para acceder a los activos informáticos con el propósito de escanearlos.

- Programas intrusos (insiders): Son técnicas que aprovechan las vulnerabilidades y se emplean para evadir la seguridad de un equipo o una red.

- Filtración de datos: Acción que compromete un sistema exponiendo la información a un entorno no confiable.

- Firewall: Es tecnología (hardware y software) diseñado para bloquear o permitir el flujo de un tráfico. En capas de transporte controla los sockets, IP y puertos, y en capas superiores controla tráfico por protocolos.

- Sistema de detección de intrusiones (IDS): Monitorea y analiza los eventos de un equipo o red para encontrar y proporcionar casi en tiempo real advertencias de intentos de acceso no autorizado.

- Sistema de prevención de intrusiones (IPS): Monitorea y analiza los eventos de un equipo o red para encontrar y proporcionar casi en tiempo real advertencias de intentos de acceso no autorizado, y además tomar acciones de bloqueo ante esas advertencias.

- Vector de ataque: Método que utiliza una amenaza para atacar un sistema.

A continuación se expone un marco conceptual de la Ingeniería del Software que involucra los ciclos del desarrollo de software seguro y del desarrollo de software en general para hacer hincapié en los elementos y conceptos relacionados a la gestión del mantenimiento.

La gestión del mantenimiento del software permite cerrar las vulnerabilidades contenidas en el código fuente de un Software y también permite actualizar los elementos impactados en la remediación de vulnerabilidades informáticas en tecnologías que hacen parte del Software o que un código fuente emplea para su funcionamiento.

Secure Software Development Life Cycle (S-SDLC):

> El ciclo de vida de desarrollo de software seguro establece una ruta de buenas prácticas y recomendaciones que al aplicarse al proceso de Ingeniería del Software fortalece la seguridad de las transacciones y entorno del Software sin comprometer el cumplimento de los requerimientos funcionales y no funcionales establecidos.

Objetivos del Secure Software Development Life Cycle (S-SDLC):

> En la obra publicada en 2006, Building Secure Software, John Viega y Gary McGraw enlistan una serie de objetivos que conforma un ciclo de vida del desarrollo de software seguro. A continuación se exponen los objetivos con citas textuales traducidas de la obra originalmente escrita en inglés:

- Prevenir: La prevención es una de las actividades con menos énfasis en el proceso de proyectos, por lo cual, muchos de los ataques y perjuicios a los desarrollos, se presentan debido a las grietas ocasionadas por las apariciones de vulnerabilidades que no fueron atendidas en su momento o con anterioridad a su presencia.

- Auditar: Un objetivo primordial de la seguridad del software es llevar la trazabilidad de los procesos con el fin de conocer de qué manera, cuándo, quién, y cómo ha sido vulnerado el software.

- Autenticar: En la actualidad, el mecanismo establecido de asignación de contraseña a cada usuario, establece el nivel de autenticación que les permite, ingresar, visualizar, modificar, ya sea desde las vistas o directamente en las bases de almacenamiento. La asignación de perfiles permite controlar los accesos y de esta manera blinda los datos de la manipulación indeseada por usuarios sin autorización.

- Asegurar la integridad: Hace referencia al hecho de identificar si algo fue modificado desde su creación.

Objetivos de los proyectos de software:

Así como se definen los objetivos principales en el ciclo de vida para el desarrollo de software seguro, no se desconocen los objetivos primordiales de la Ingeniera de Software en las que además del cumplimento de los requerimientos funcionales y no funcionales establecidos, se tienen los siguientes objetivos en los proyectos de software:

- Funcionalidad: Hace referencia a la capacidad que debe tener el desarrollo de cumplir a cabalidad con los requerimientos funcionales planteados por los usuarios.

- Usabilidad: Define la facilidad y practicidad con la cual pueda ser manipulado el software.

- Eficiencia: Suministrar al usuario un rendimiento que espera encontrar en un Software.

- Time to market: Este objetivo hace referencia al tiempo en que el producto alcanza su venta o entrega teniendo en cuenta el momento en que fue iniciado.

- Simplicidad: El desarrollo de un sistema simple, pero a la vez seguro, es una de las principales metas en cualquier proyecto de desarrollo.

Metodologías y controles en el ciclo de desarrollo de Software seguro:

- Correstness by Construction (CbyC):

  Busca que la codificación del código sea de manera correcta desde su inicio, procurando la corrección y eliminación de errores desde su ingreso, para ellos se apalanca en procesos rigurosos de seguridad, donde sus requerimientos presentan un alto detalle.

  La secuencia establecida para la construcción segura del software según CbyC se expone en la ilustración 15 obtenida de las Metodologías para desarrollar software seguro publicadas por Carlos Joaquín Brito Abundis en el sitio web http://recibe.cucei.udg.mx/revista/es/vol2-no3/computacion05.html consultado el 6 de junio de 2017.

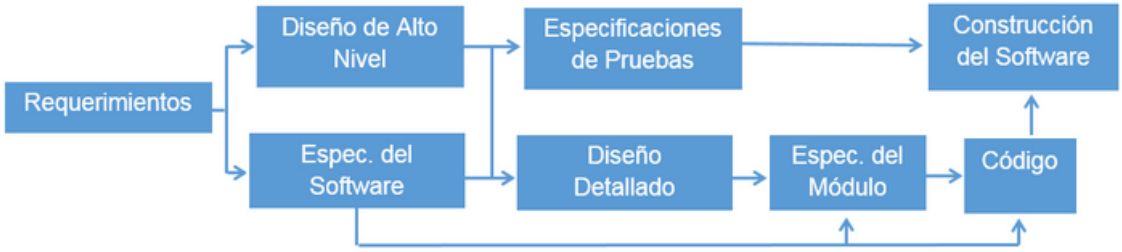


Ilustración 15. Proceso de desarrollo propuesto por el CbyC (Amey, 2006)

Fases de la metodología:

- Fase de requerimientos:

Esta fase define los requerimientos de usuario, especificando las funciones y propósito, así mismo, tiene en cuenta la definición de requerimientos no funcionales expresados en los diagramas de clases.

- Fase de diseño de alto nivel:

Describe la composición a gran escala del desarrollo incluyendo las bases de datos y funcionalidad, teniendo como punto principal los requerimientos no funcionales en cuanto a su ámbito de seguridad en cada punto álgido.

- Fase de especificación del software:

Tiene como finalidad especificar la interfaz de usuario.

- Fase de diseño detallado

Define el conjunto de módulos, procesos y funcionalidades.

- Fase de especificación de módulos:

Se define el estado y comportamiento de los módulos para definir y garantizar el flujo de información.

- Fase codificación:

En esta fase se induce el desarrollo de pruebas con el fin de disminuir y eliminar los errores presentes en el código.

- Fase de especificaciones de pruebas.

  Como particularidad en la metodología CbyC, no es necesario ejecutar proceso de pruebas de unidad ni caja blanca, puesto que su foco central es realizar estas validaciones a nivel de sistema.

- Fase de construcción de software.

  La metodología presenta enfoque técnico en el cual, trata de disminuir los defectos, aumentando considerablemente su capacidad para minimizar fallas.

- Security Development Lifecycle (SDL)

  Metodología encaminada al mejoramiento del desarrollo de software. Fue propuesta en 2004 por Microsoft e incluye un proceso de modelado de amenazas en el cual se busca identificar vulnerabilidades a nivel de código (Microsoft, 2016).

  SDL cuenta con dos versiones de ejecución con el fin de tener un mejor acoplamiento dependiendo del tipo de desarrollo, ellas son:

  SDL – Versión rígida: Enfocada en equipos de proyectos y desarrollo de productos de gran envergadura donde los cambios son mínimos.
  SDL – Versión Ágil: Sus desarrollos son incrementales con aumento en la frecuencia de ejecución y seguimiento de actividades haciendo énfasis en su seguridad.

  Fases de la metodología:

  - Fase de Entrenamiento:

    Contempla el proceso de capacitación al equipo técnico conformado por los desarrolladores y grupos de pruebas con el fin de estar al tanto en las últimas actualizaciones en materia de seguridad.

  - Fase de Requerimientos:

    En conjunto con un consultor de seguridad, debe ser revisado y planteado un procedimiento que incorpore las actividades de seguridad, para que se cumplan metas que se propondrán para iniciar con el proyecto y continuar con una siguiente tapa de diseño acorde a lo solicitado.

- Fase de Diseño :

  Desarrolla el proceso de modelado de amenazas en cada componente, esto con el fin de identificar la posible aparición de riesgos y así mismo, implantar planes de identificación y mitigación.

- Fase Implementación

  A partir del resultado de la fase de diseño el grupo desarrollador construirá y dirigirá su código de manera que puedan proteger el proceso y disminuir la afectación que puedan tener estas amenazas o ataques, para ello, se sugiere la codificación mediante el uso de estándares.

- Fase de Verificación:

  En esta fase, en la que el software ya cuenta con una versión inicial y en la cual se desarrollan pruebas de validación de seguridad, se ejecutan revisiones íntegras y profundas en las secciones que se han identificado como blanco de ataques.

- Fase de Lanzamiento:

  Como parte del proceso, esta fase debe ser ejecutada de dos a seis meses antes de la entrega del producto final al cliente.

- Fase de Respuesta:

  Como valor agregado a la metodología, se pretende ampliar la cobertura en cuanto a conocimiento y respuesta contra incidentes de seguridad.

Esta metodología ha sido implementada por Microsoft como abanderada en los procesos de desarrollo de los diferentes Sistemas Operativos.

Estándares involucrados en el desarrollo de software seguro:

- Common Criteria (ISO/IEC 15408)

Estándar que permite a los desarrolladores definir las propiedades de seguridad y así mismo, validar que estas especificaciones se cumplan, para ello, se identifica Objetivo de Evaluación o sus siglas en ingles TOE - Target of Evaluation (Common Criteria, 2012).

Existen tres tipos de stakeholders involucrados en los procesos ejecutados en TOE:

- Consumidores o clientes: Son la razón de ejecutar procesos de evaluación de la seguridad, y se ejecutan con el propósito de validar que los requerimientos iniciales se estén cumpliendo de manera segura.

- Desarrolladores: Da una ruta a los desarrolladores en la preparación de Objetivos de Evaluación basado en los requerimientos de seguridad contenidos en una Declaración de Seguridad.

- Evaluadores: Common Criteria estructura las actividades que serán ejecutadas en una fase de evaluación para validar los objetivos identificados.

- Systems Security Engineering Capability Maturity Model - SSE-CMM (ISO/IEC 21827)

  Este estándar se caracteriza por ser una métrica no basada en seguimiento de procesos, se encuentra plasmada en ISO/IEC 21827 Information technology -- Security techniques -- Systems Security Engineering -- Capability Maturity Model® (SSE-CMM®) (ISO, 2008), que tiene como alcance:

  - Ciclo de vida de desarrollo general, incluyendo desarrollo, operación, mantenimiento etc.
  - Funcionalidad integra de relación entre ámbitos del proyecto, teniendo en cuenta hardware, software, recurso humano, ingeniería de pruebas.
  - Comunicación con otras entidades organizativas, gestiones del sistema, certificaciones, acreditaciones, entre otros.

- Comprehensive, Lightweight Application Security Process (CLASP)

  Dirige a los desarrolladores en el proceso de revisión y validación desde las primeras etapas del ciclo de vida del desarrollo de software de manera que este proceso se convierta en estructurado (OWASP, 2016).

- ista CLASP: La metodología plantea 104 fallas de seguridad agrupadas en 5 niveles de vistas, como se muestran en la ilustración 16:

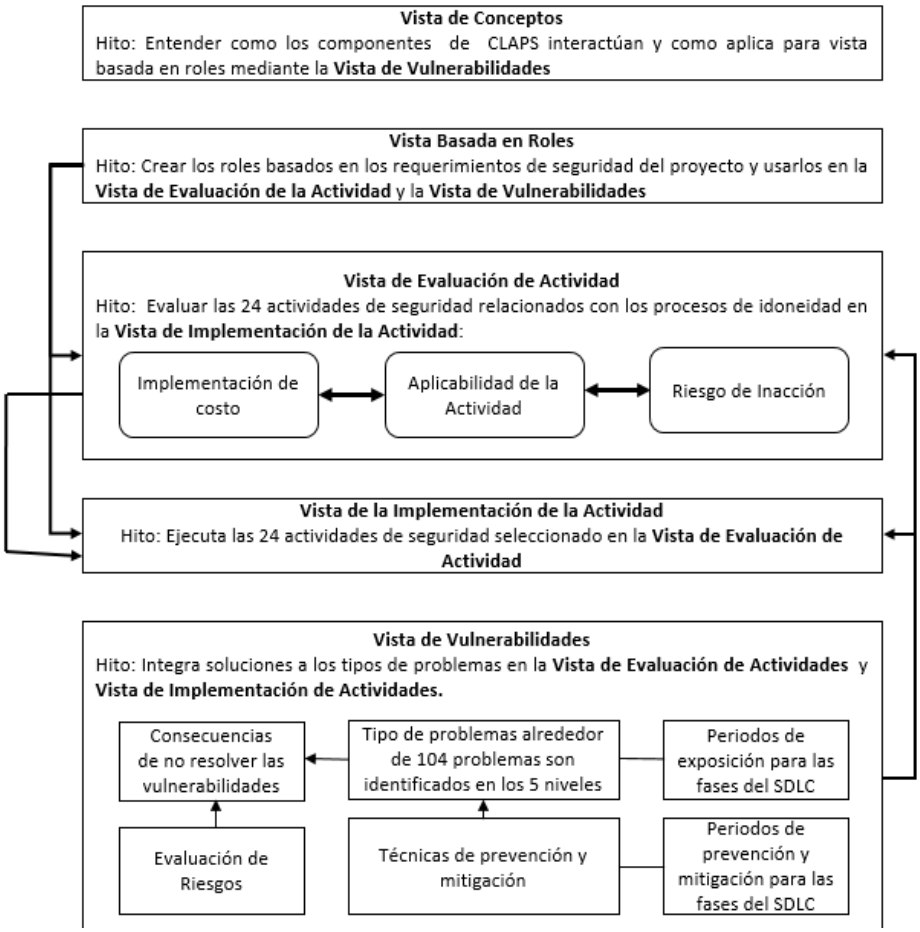


Ilustración 16. Relación de los 5 niveles de vistas CLASP (Owas.org)

- Recursos CLASP: Como apoyo a los procesos de planificación y ejecución de actividades, CLASP presenta una serie de recursos o herramientas que permiten el acceso a los diferentes artefactos.

- Caso de uso de Vulnerabilidad: Aquí se describe los puntos en los cuales se presenta una vulnerabilidad en las aplicaciones de software, con estos casos de uso, se pretende dar al usuario una vista fácil respecto evidenciando su causa y efecto.

- Touchpoints: Conformada por 7 puntos de control establecidos en el 2004 por la IEEE conocido como el modelo de GaryMCgraw y Cigital, se consideran uno de los tres pilares de la seguridad de

software siendo tomada como la unión entre la parte técnica y práctica del desarrollo haciendo énfasis de esa manera en el empleo de buenas prácticas (Mc Graw & Jhon, 2006).

- Está compuesta por dos tipos de actividades, unas destructivas (sombrero negro) y unas constructivas (sombrero blanco), en su orden de efectividad se pueden enumerar de la siguiente manera:

  - Fase Revisión de código

    Dirigida a la revisión del código fuente con el fin de identificar las posibles vulnerabilidades que puedan afectar el desarrollo del producto.

  - Análisis de riesgos de arquitectura

    Con el fin de identificar los riesgos posibles en el desarrollo, esta fase enfoca su actividad en el trabajo en conjunto que se presente entre los arquitectos, diseñadores y analistas, quedando documentados todos los puntos de vista.

  - Pruebas de penetración:

    Se realizan intrusiones relacionadas a hacking ético y controlado, con el fin de identificar que posibles inconvenientes se pudieran presentar a futuro:

    - Con objetivo: Se buscan vulnerabilidades en componentes específicos que son de mayor importancia dentro de una red.

    - Sin objetivo: Examinar la totalidad de los elementos que componen la infraestructura evaluada.

    - Black box: Examina sin conocer el objetivo evaluado.

    - Gray Box: Examina con algún tipo de información del objetivo evaluado.

    - White box: Examina conociendo políticas e información detallada del objetivo evaluado.

- Pruebas de seguridad basada en riesgos

  Las pruebas de seguridad se deben contemplar también en la identificación, análisis y respuesta de la gestión del riesgo.

- Casos de abuso

  Según Gary MacGraw, este punto es la forma más directa para entender qué hace un atacante. Se definen como los casos de uso pero buscan afectar el Software con la finalidad de identificar qué se debe revisar, proteger y fortalecer, de qué o quién y por cuánto tiempo.

- Requerimientos de seguridad

  En la definición de los requerimientos funcionales y no funcionales se deben construir procesos enfocados al proceso de seguridad.

- Operaciones de seguridad

  De manera recurrente y continua debe ser monitoreado el comportamiento de seguridad del software, este proceso se convierte en la principal forma de defensa de un sistema de información.

En la ilustración 17 se exponen las fases del ciclo de desarrollo de software seguro según la obra publicada en 2006, Building Secure Software, de John Viega y Gary McGraw:

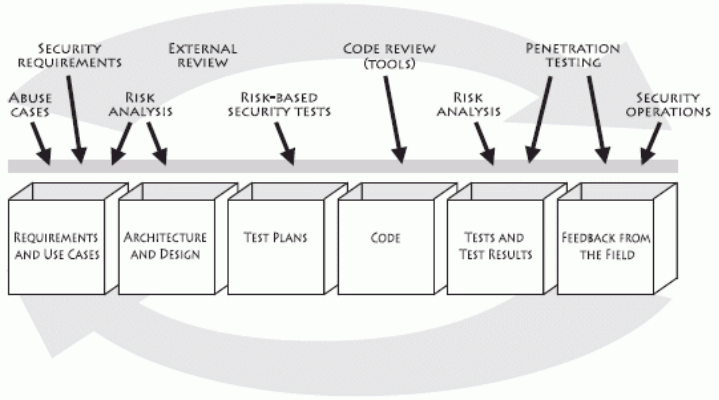


Ilustración 17. Fases del ciclo de desarrollo de software seguro según MacGraw

- OWASP: Open Web Application Security Project

  Es un proyecto de seguridad al principio dedicado a la seguridad de aplicaciones WEB, y que a partir de su versión de 2017 también asume un enfoque para aplicaciones móviles. Suministra abiertamente procedimientos de seguridad aplicado a este tipo de aplicaciones que presentan formas de ataques.

  OWASP ha producido una serie de guías para facilitar la adquisición de una base de datos de conocimiento:

  - Referencia de Escritorio en Seguridad de Aplicaciones de OWASP: Contiene las descripciones y conceptos básicos de la seguridad como: agentes de amenazas, vulnerabilidades, ataques, impactos.

  - Guías de desarrollo de OWASP: abarca todo el control de seguridad desde la perspectiva del desarrollador de un producto de software, proporcionando un conjunto de controles de seguridad que fortalecerá el sistema desarrollado.

  - Guía de revisión de Código OWASP: Incorpora principios relacionados a la gente y los procesos relacionados en un ciclo de vida de desarrollo de software

    Su análisis respecto a las vulnerabilidades que afectan las aplicaciones, se divide en tres puntos descritos en la ilustración 18:

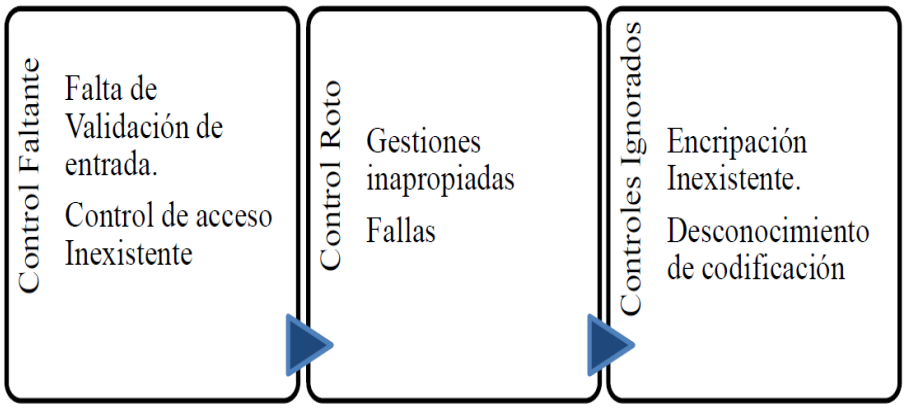


Ilustración 18. Teoría del Análisis de vulnerabilidades según OWASP

- OTP (OWASP Testitng Project):

  Es una metologia de pruebas de software para validar los controles de seguridad efectuados sobre el desarrollo de una aplicación web. La metodología se divide en 2 partes, en la primera se abarcan los siguientes puntos:

  - Principios del testeo
  - Explicación de las técnicas de testeo.
  - Explicación general acerca del framework de testeo de OWASP.

  En la segunda parte, se planifican todas las técnicas necesarias para testear cada fase y entregable del ciclo de vida del desarrollo de software.

Para la trazabilidad de cambios realizados en el código fuente de un Software que se le hace una gestión de mantenimiento, se propone dentro del ciclo de su desarrollo emplear un sistema de control de versiones:

Un Sistema Control de Versiones es una herramienta encargada de llevar el control de las diferentes versiones o modificaciones realizadas en el desarrollo de un software. Generalmente este tipo de software se usa cuando el proceso de codificación es realizado por varios desarrolladores en un mismo producto

Usar un sistema de control de versiones permite reconstruir un Software a partir de las versiones anteriores y llevar una bitácora de qué modificaciones y cambios se han realizado, en qué fecha y quien los ha realizado.

Hay dos tipos de sistemas de controles de versiones.

- Centralizados

  Se almacena la información de la codificación de un Software en un servidor centralizado. Los desarrolladores se conectan a ese servidor para realizar los cambios que requiera y subir esa nueva versión al servidor. En este tipo de sistemas de control de versiones se encuentran:

  - Concurrent Version Systems (CVS): Permite el proceso de actualización generando una nueva versión del Software para controlar la actualización paralela de versiones por más de un desarrollador al mismo tiempo.

  - Subvesion: Permite también el control de versiones a los documentos, la estructura y diferentes versiones de directorios permitiendo su repositorio.

- Distribuidos

  Permite que el desarrollador realice cambios localmente y al finalizar hace una clonación del repositorio del Software en el servidor. Permite generar versiones por desarrollador, y cuando sea revisada la versión aprobada se puede sincronizar con el código fuente original que se aloja en el servidor. En este tipo de sistemas de control de versiones se encuentran:

  - GIT: Desarrollado por Linus Torvalds, quien es la persona responsable del inicio y mantenimiento del kernel Linux.

    En este sistema de control de versiones cuando el desarrollador descarga una versión local del Software y realiza las modificaciones, Git genera un listado de los elementos modificados y los selecciona para identificarlos al crear la nueva versión.

    Cuando son varias versiones, GIT realiza la función Merge que se emplea para compactar las versiones creadas por cada desarrollador.

  - Mercurial: Tiene un funcionamiento similar a Git, guarda una copia de los archivos del proyecto de software de manera local, pero no se conecta directamente con el repositorio origen.

CAPITULO 3. MARCO METODOLÓGICO

3.1. ASPECTOS METODOLÓGICOS DE LA INVESTIGACIÓN.

3.1.1. TIPO DE INVESTIGACIÓN.

Como la investigación expuesta en este documento se trata de un proyecto de ingeniería, se deja constancia que el tipo de investigación desarrollada es básicamente exploratoria, porque a pesar de tener una base científica y metodológica que exponen los procesos de atención de vulnerabilidades, las dificultades en los procesos de la remediación no ha sido lo suficientemente estudiado. La investigación también tiene características de tipo explicativo, porque busca describir la problemática de la no remediación de vulnerabilidades e intenta encontrar las causas que tiene.

3.1.2. METAS A ALCANZAR.

Con los conocimientos adquiridos en las etapas de la investigación, se identifican metas que se desean conseguir a corto, mediano y largo plazo, las cuales buscan exponer a la comunidad académica de la Fundación Universitaria los Libertadores y personas interesadas en la lectura de este documento, la necesidad que tiene la remediación de vulnerabilidades informáticas:

- Exponer datos relacionados a las vulnerabilidades informáticas
- Exponer el tiempo requerido en los mecanismos que permiten explotar vulnerabilidades informáticas
- Desarrollar una herramienta que permita consultar con dos criterios, las distintas vulnerabilidades que tiene un software específico y los mecanismos de explotación y contramedida existentes.

3.1.3. PRODUCTOS A ENTREGAR.

- Documento con explicación de qué es una vulnerabilidad informática exponiendo antecedentes de la no remediación de vulnerabilidades.
- Video exponiendo el tiempo requerido para explotar una vulnerabilidad informática.
- Una guía que exponga los pasos sugeridos para la atención de vulnerabilidades informáticas desde la perspectiva de la remediación.

- Una base de datos y un software para consultar las distintas vulnerabilidades que tiene un software específico y los mecanismos de explotación y contramedida existentes.

3.1.3.1. DISEÑO DE LA HERRAMIENTA A DESARROLLAR:

3.1.3.1.1. SELECCIÓN DE HERRAMIENTA DE DESARROLLO

Para el desarrollo de la guía y la herramienta anexa se usa el sistema gestor de base de datos Workbench, el servidor de base de datos MySQL Server, el lenguaje de programación Java 8 y su IDE de desarrollo Oracle Netbeans 8.2.

3.2. ESTRUCTURA DE LA UNIDAD DE ANÁLISIS

3.2.1. PREGUNTAS A RESOLVER

- ¿Es sencillo atacar un equipo vulnerable a CVE-2017-143?
- ¿Es sencillo estimar equipos y servidores que se encuentren en Internet y que tengan vulnerabilidades de Java al momento del estudio?

3.2.2. VARIABLES E INDICADORES

- Cantidad de vulnerabilidades publicadas en CVE.
- Cantidad de versiones versus la cantidad de vulnerabilidades publicadas.
- Tiempo requerido para la explotación de una vulnerabilidad.

3.2.3. POBLACIÓN DE ESTUDIO

- Equipos y servidores que en Internet se encuentren vulnerables a CVE-2017-143 al momento del estudio.
- Equipos y servidores que se encuentren en Internet y que tengan vulnerabilidades de Java 1.7.0 al momento del estudio.

3.2.4. MUESTRA

- Un equipo vulnerable a CVE-2017-143 para realizar una explotación de la vulnerabilidad.

- Registro en Internet de vulnerabilidades y versiones de java junto a una cifra encontrada de equipos y servidores que en Internet tienen instalada una versión de java vulnerable.

3.2.5. INSTRUMENTOS

Para el estudio se requieren las bases de datos mencionadas en el marco referencial y teórico de este documento, un equipo vulnerable y un equipo para explotar la vulnerabilidad en el equipo vulnerable.

3.2.6. PARTICIPANTES

- Investigador y desarrollador: YOHAN ESNEIDER HERNÁNDEZ VILLARREAL.
- Asesores Expertos: ING. LUIS EDUARDO BAQUERO.

3.2.7. ASPECTOS ECONÓMICOS

El material para el análisis, para el diseño de la guía y la información correlacionada en la herramienta desarrollada son de libre distribución y de fuentes certificadas y mencionadas en la introducción de este documento. Esto garantiza el correcto funcionamiento del desarrollo y lo convierte en un proyecto factible económicamente.

3.2.8. ASPECTOS TÉCNICOS Y OPERACIONES DE LA GUÍA Y SUS ANEXOS

La función de la herramienta desarrollada consiste en recolectar, normalizar y exponer los datos en un desarrollo hecho en Java Standard Edition 8 por lo que puede ser ejecutado desde un servidor o computador de sistema operativo de la familia Windows, Linux o Mac que tengan instalado y que soporte Java Runtime Environment 8.

La base de datos creada a partir de los listados coleccionados se importa desde un servidor o computador de sistema operativo de la familia Windows, Linux o Mac que tengan instalado y que soporten MySQL 5.7.

Al tratarse de una correlación de información basado en consultas, las personas que deseen usar el desarrollo solo necesitan buscar los nombres y versiones de sistemas operativos, plugins, herramientas de ofimática, frameworks y en general software relacionado a las capas de presentación y aplicación.

3.3. MARCO LEGAL

En el análisis y en los resultados se muestra una serie de imágenes de un ejercicio real de explotación de vulnerabilidades ejecutado en el proceso de esta investigación con el único objetivo de mostrar la facilidad que tiene el mecanismo de explotación. Estas imágenes no representan una guía de explotación ni tampoco pretende abordar procesos propios del Ethical Hacking, debido a que estos no hacen parte del contenido de esta investigación. Se deja constancia, para exponer el cumplimiento de la legislación Colombiana, en especial, a la Ley 1273 de enero 2009 (Ley de delitos informáticos en Colombia) la cual establece nuevos códigos penales referentes a la protección de la información y las penas en las que se puede incurrir relacionado con delitos informáticos.

3.4. PROPIEDAD INTELECTUAL

Como el software es una forma de propiedad intelectual se deja una constancia que el uso, copia o distribución del código de la herramienta desarrollada como anexo de esta investigación es permitido, es de uso libre, y también lo es la guía expresada en este documento. Sin embargo, es necesario que al mostrar los resultados se haga una referencia que la información recolectada pertenece a las bases de datos: CVE de The Mitre Corporation, la base de datos empleada y mantenida por Rapid7 en su sistema de gestión de vulnerabilidades Nexpose, y la base de datos de ExploitDB de Offensive Security y Rapid7 Metasploit.

CAPITULO 4. RESULTADOS DE LA INVESTIGACIÓN

4.2. ATAQUE SENCILLO A UN EQUIPO VULNERABLE A CVE-2017-143

En el análisis y en los resultados se logró explotar exitosamente la vulnerabilidad CVE-2017-143 en un equipo muestra.

A continuación se expone en las ilustraciones 19 a la 25, un ejercicio real de explotación de vulnerabilidades ejecutado en el proceso de esta investigación con el único objetivo de mostrar la facilidad que tiene el mecanismo de explotación en relación al tiempo que demora su ejecución.

- Obtención de datos:

  Se toman dos equipos de acuerdo al marco metodológico y se procede a iniciar la prueba de explotación tomando el tiempo requerido, como se expone la ilustración 19.

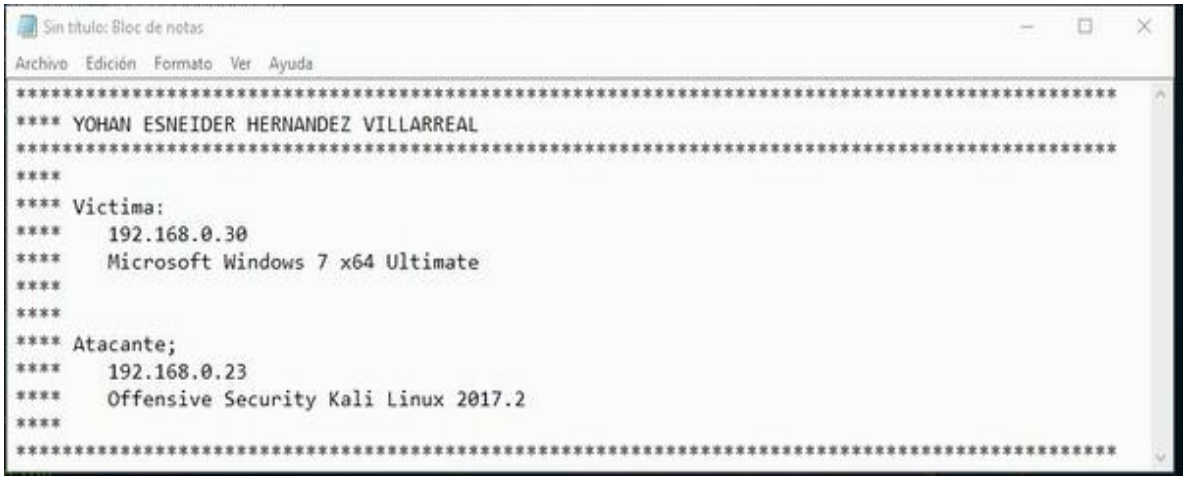


Ilustración 19. Datos del objetivo a atacar

- Búsqueda de exploits publicados para una vulnerabilidad

  En Metasploit Framework se buscan los exploits que existan para las vulnerabilidades:

  CVE-2017-143

  CVE-2017-144

CVE-2017-145

CVE-2017-146

CVE-2017-147

CVE-2017-148

Cuando se encuentran los exploits se seleccionan como se muestra en la ilustración 20.

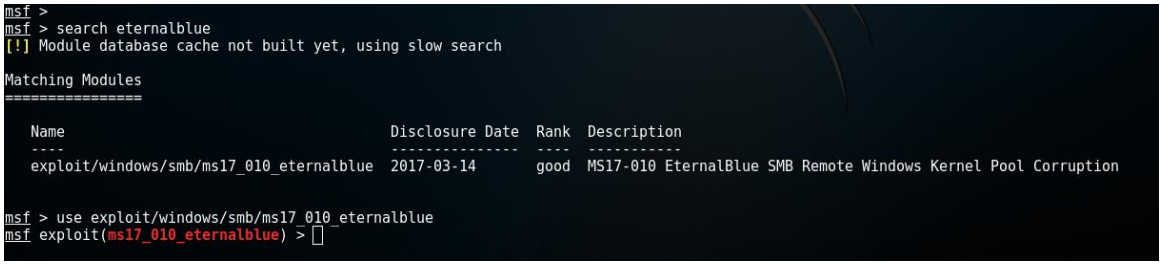


Ilustración 20. Búsqueda y selección del exploit de la vulnerabilidad CVE-2017-143 en Metasploit.

- Como se muestra en la ilustración 21 se buscan y seleccionan los mecanismos de postexplotacion. En este ejercicio se selecciona un mecanismo que permita generar una sesión de control de la víctima mediante Meterpreter, pidiendo que sea el atacante quien busque al equipo victimario para abrir la sesión. Esto permitiría que en una red con equipos de protección perimetral se evadan porque el victimario no ingresa a la red, sino que es la victima quien sale a través de los equipos de protección en la red.

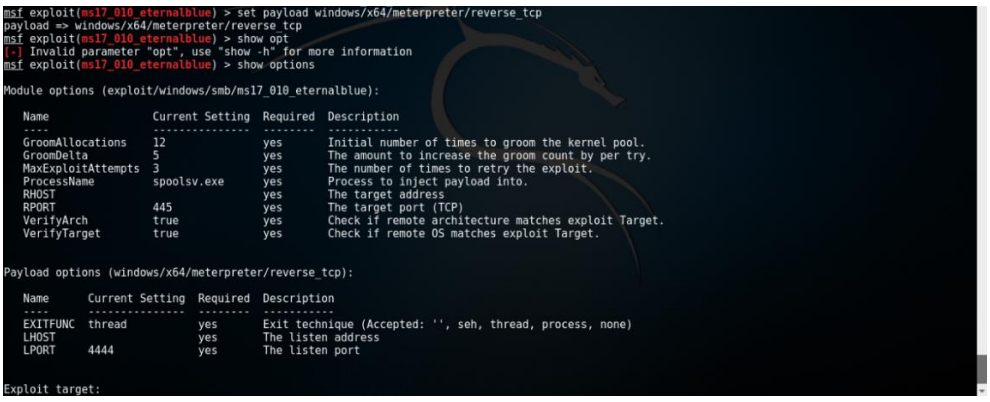


Ilustración 21. Entrega del Payload para ejecutar un meterpreter sobre el target como post explotación.

- Para este ejercicio solo es necesario colocar las ip del equipo víctima y el equipo victimario, como se muestra en la ilustración 22:

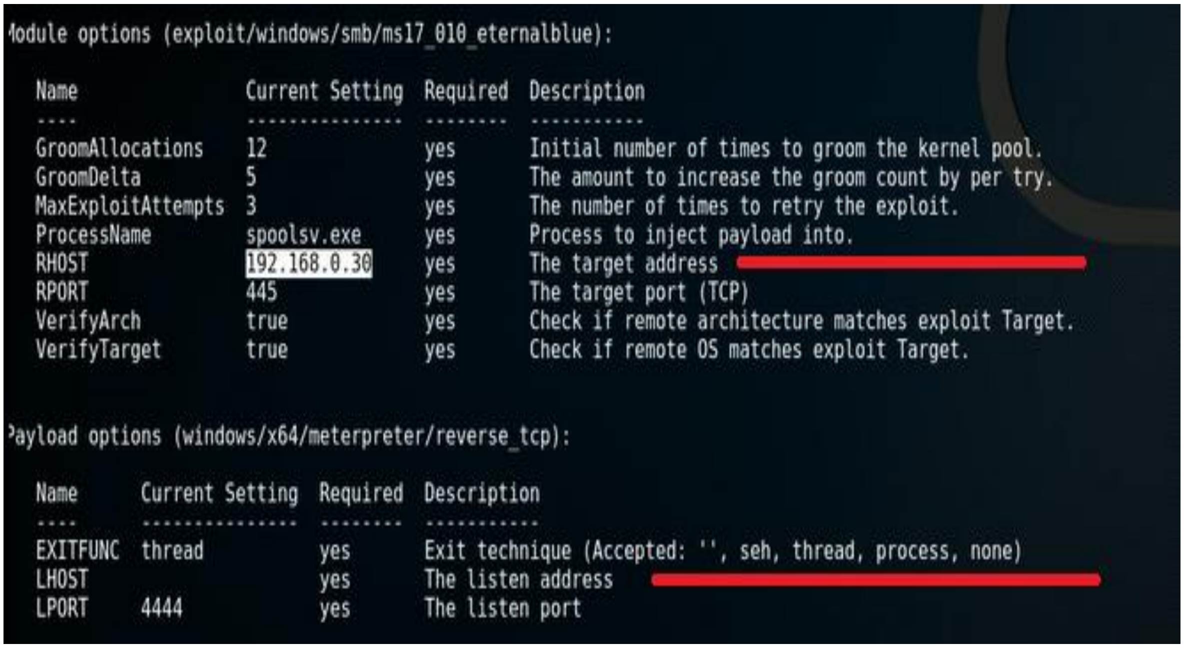


Ilustración 22. Datos de equipos víctima y victimario en Metasploit

- Con los pasos anterior se explota la vulnerabilidad y se establece la sesión, como en la ilustración 23:

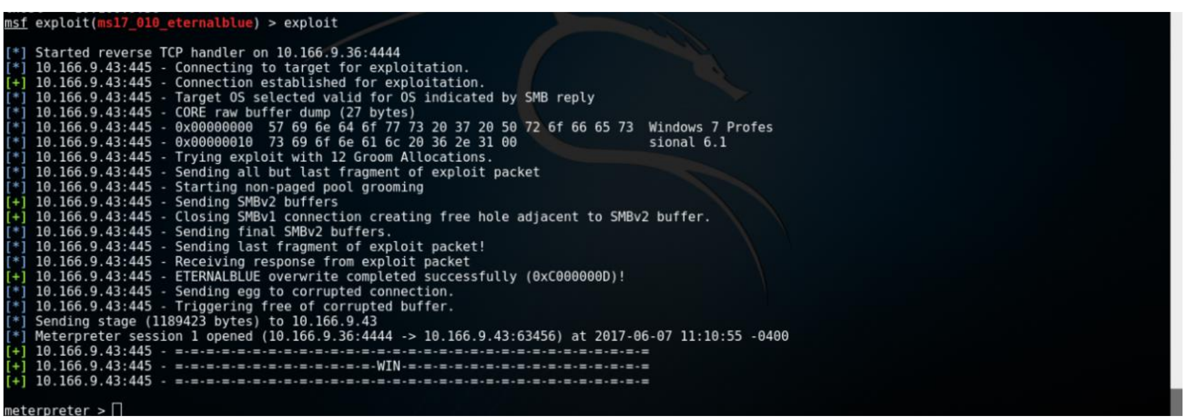


Ilustración 23. Explotación exitosa y ejecución de meterpreter

- Con la sesión establecida se tienen permisos de SYSTEM sobre el equipo victima por lo que se pueden ejecutar distintos procesos para obtener usuarios, documentos, el control de la cámara web o el control del micrófono del equipo. Esto se muestra en la ilustración 24:

Ilustración 24. Inclusión de Mimikatz en la victima, robo de contraseñas de la víctima, acceso al Shell de la víctima.

- Evidencia de sistema atacado, en la ilustración 25:

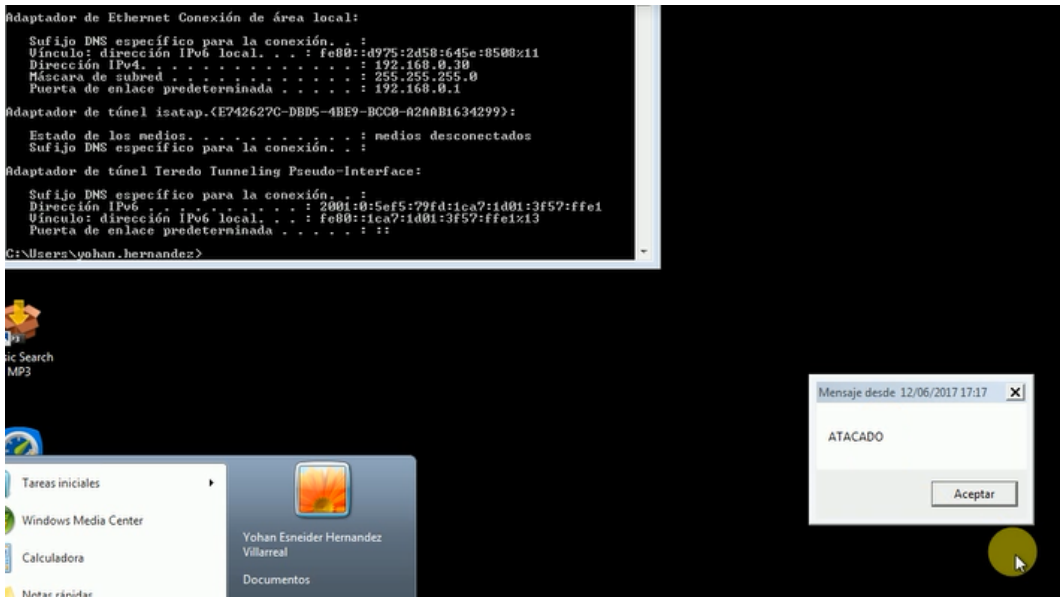


Ilustración 25. Ataque exitoso.

El ataque realizado en este ejercicio se anexa a este documento en el video Video_Explotacion_CVE-2017-143.mp4

4.3. ENCONTRANDO EQUIPOS VULNERABLES CON MECANISMO DE EXPLOTACIÓN. CASO JAVA.

Como se expresa en el planteamiento del problema, las vulnerabilidades informáticas no solamente están presentes en sistemas operativos, como el caso expuesto por WanaCryptor. Además hay vulnerabilidades más críticas que la CVE-2017-143 al 148 en el sentido de que su explotación no siempre genera una atención mediática porque su principal objetivo es atacar pasando desapercibido; de hecho hay más herramientas robadas a la Agencia Nacional de los Estados Unidos que se pueden emplear en switches, en sistemas operativos Solaris y otras tecnologías distintas a Microsoft. (ShadowBroker, 2017).

Para representar vulnerabilidades críticas además de las expuestas con WanaCryptor se pueden tomar las aplicaciones o software de uso masivo con grandes problemas de seguridad publicados en Internet. Para el siguiente registro de hechos se usa el plugin de Java como referente, porque es el lenguaje de programación más usado con un rating de 14.49% según el estudio de TIOBE para junio de 2017, el cual se expone en la ilustración 26.

TIOBE Index for June 2017

June Headline: Programming language Kotlin jumps into the top 50

Jun 2017	Jun 2016	Change	Programming Language	Ratings	Change
1	1		Java	14.493%	-6.30%
2	2		C	6.848%	-5.53%
3	3		C++	5.723%	-0.48%
4	4		Python	4.333%	+0.43%
5	5		C#	3.530%	-0.26%
6	9	▲	Visual Basic .NET	3.111%	+0.76%
7	7		JavaScript	3.025%	+0.44%
8	6	▼	PHP	2.774%	-0.45%
9	8	▼	Perl	2.309%	-0.09%
10	12	▲	Assembly language	2.252%	+0.13%

Ilustración 26. Lista de lenguajes de programación más empleados.

Java a junio de 2017 cuenta con un registro 4.354 vulnerabilidades publicadas en la CVE desde 1999 a mayo de 2017, como se expone en las ilustraciones 27 y 28.

Vulnerability Trends Over Time

Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	Http Response Splitting	Bypass something	Gain Information	Gain Privileges	CSRF	File Inclusion	# of exploits
1999	4		1							1		1			
2000	8	1	3	2		1						3			
2001	28	8	10	8				2				3			
2002	47	10	12	8		1	4			4	5	4			
2003	12	1	8	7		1				1	2				
2004	26	4	10	6		3	1	1		1	2	5			
2005	61	2	7	1		5	8	3		3	3	3			
2006	148	1	8	5		30	3	2	2	3					
2007	125	5	12	17		12	7	2		1	1	2	1		3
2008	116	2	4	5		6	3			3		1			2
2009	156	1	3	3		4	3			2	1	4			2
2010	207	1	5	1			1						1		1
2011	224	3	5	2		1						1			3
2012	380	7	10	6	1	1		1		4	3	2			7
2013	505	2	14	10	4	2	1			32	5				3
2014	471	22	17	9	6		1	2		8	8	2			7
2015	605	88	32	39	10	1	5		1	5	10	5			
2016	800	81	41	53	12		8	15		9	34	11			
2017	431	35		1							22				
Total	4354	274	202	183	33	68	45	28	3	77	96	47	2		28
% Of All		6.3	4.6	4.2	0.8	1.6	1.0	0.6	0.1	1.8	2.2	1.1	0.0	0.0	

Ilustración 27. Vulnerability Trends Over Time.

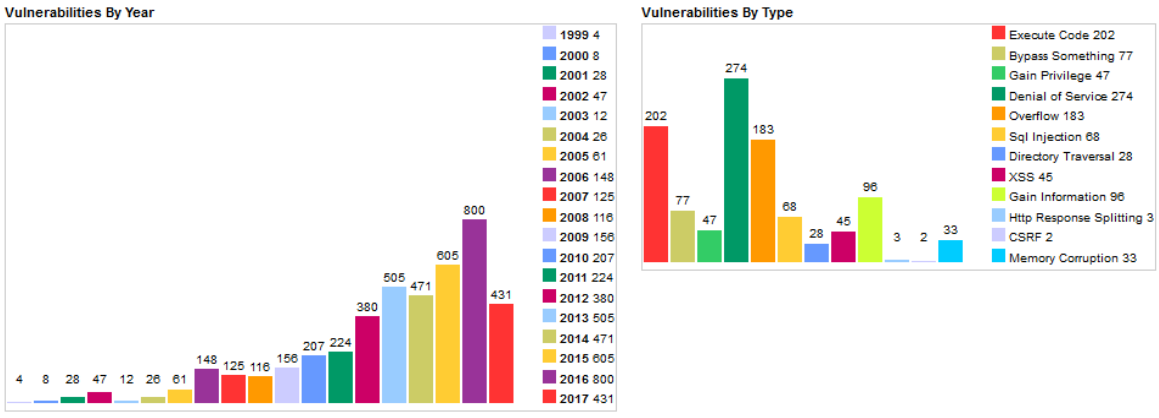


Ilustración 28. Estadística por año y por tipo de vulnerabilidad publicada de Java en CVE.

Para los clientes con contrato de Oracle Corporation, se definieron las Políticas de Soporte Técnico de Software de Oracle que entro vigencia desde el 26 de septiembre de 2016. En los términos de soporte Oracle define que el soporte técnico entra en vigencia a partir de la fecha de entrada en vigencia de su orden, a menos que se establezca lo contrario en dicha orden. Si la orden fue presentada a través de la Tienda de Oracle (Oracle Store), la fecha de entrada en vigencia es la fecha en la que Oracle aceptó la orden. A menos que se establezca lo contrario en la orden, los términos del soporte técnico de Oracle, incluidos los precios, abarcan un período de soporte de 12 meses (el “período de soporte”). (Oracle Corporation, s.f.).

Para el caso especial de usuarios que acceden a recursos de instalación de Java públicos aceptan un contrato de uso, como se muestra en la ilustración 29 y 30:

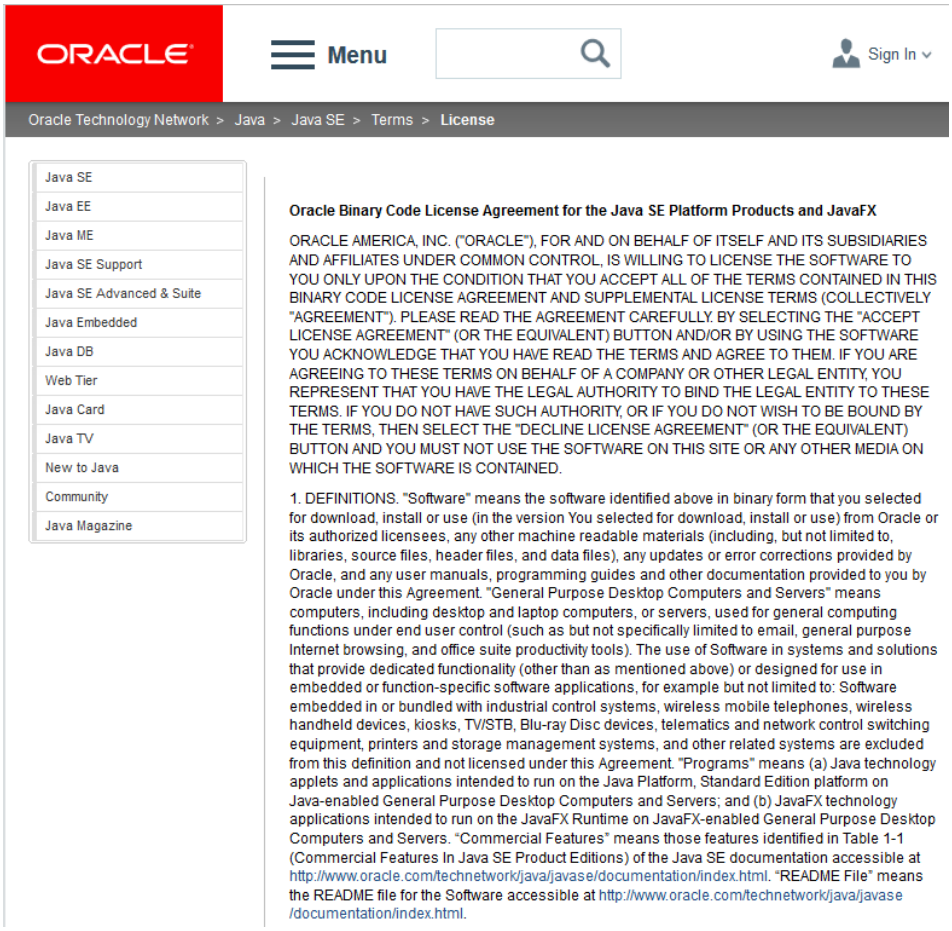
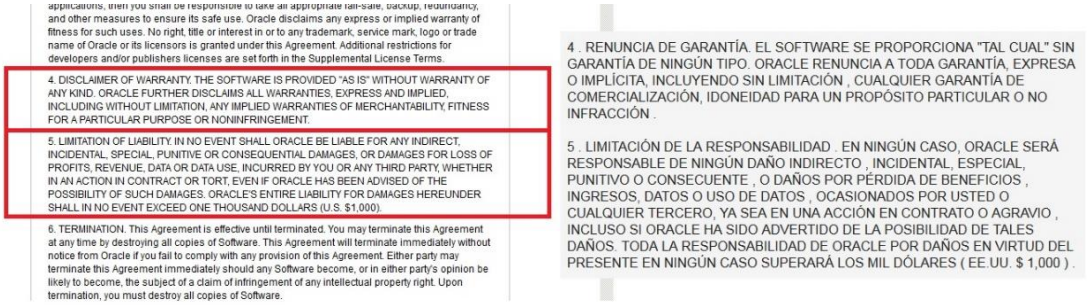


Ilustración 29. Términos de uso de Java Standard Edition.

Dentro de estos términos, Oracle deja constancia y renuncia a toda garantía y responsabilidad sobre el software para uso en comercialización e idoneidad para un propósito particular o infracción de una ley.



applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle disclaims any express or implied warranty of fitness for such uses. No right, title or interest in or to any trademark, service mark, logo or trade name of Oracle or its licensors is granted under this Agreement. Additional restrictions for developers and/or publishers licenses are set forth in the Supplemental License Terms.

4. DISCLAIMER OF WARRANTY. THE SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. ORACLE FURTHER DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT.

5. LIMITATION OF LIABILITY. IN NO EVENT SHALL ORACLE BE LIABLE FOR ANY INDIRECT, INCIDENTAL, SPECIAL, PUNITIVE OR CONSEQUENTIAL DAMAGES, OR DAMAGES FOR LOSS OF PROFITS, REVENUE, DATA OR DATA USE, INCURRED BY YOU OR ANY THIRD PARTY, WHETHER IN AN ACTION IN CONTRACT OR TORT, EVEN IF ORACLE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. ORACLE'S ENTIRE LIABILITY FOR DAMAGES HEREUNDER SHALL IN NO EVENT EXCEED ONE THOUSAND DOLLARS (U.S. $1,000).

6. TERMINATION. This Agreement is effective until terminated. You may terminate this Agreement at any time by destroying all copies of Software. This Agreement will terminate immediately without notice from Oracle if you fail to comply with any provision of this Agreement. Either party may terminate this Agreement immediately should any Software become, or in either party's opinion be likely to become, the subject of a claim of infringement of any intellectual property right. Upon termination, you must destroy all copies of Software.

4 . RENUNCIA DE GARANTÍA. EL SOFTWARE SE PROPORCIONA "TAL CUAL" SIN GARANTÍA DE NINGÚN TIPO. ORACLE RENUNCIA A TODA GARANTÍA, EXPRESA O IMPLÍCITA, INCLUYENDO SIN LIMITACIÓN , CUALQUIER GARANTÍA DE COMERCIALIZACIÓN, IDONEIDAD PARA UN PROPÓSITO PARTICULAR O NO INFRACCIÓN .

5 . LIMITACIÓN DE LA RESPONSABILIDAD . EN NINGÚN CASO, ORACLE SERÁ RESPONSABLE DE NINGÚN DAÑO INDIRECTO , INCIDENTAL, ESPECIAL, PUNITIVO O CONSECUENTE , O DAÑOS POR PÉRDIDA DE BENEFICIOS , INGRESOS, DATOS O USO DE DATOS , OCASIONADOS POR USTED O CUALQUIER TERCERO, YA SEA EN UNA ACCIÓN EN CONTRATO O AGRAVIO , INCLUSO SI ORACLE HA SIDO ADVERTIDO DE LA POSIBILIDAD DE TALES DAÑOS. TODA LA RESPONSABILIDAD DE ORACLE POR DAÑOS EN VIRTUD DEL PRESENTE EN NINGÚN CASO SUPERARÁ LOS MIL DÓLARES (EE.UU. $ 1,000) .

Ilustración 30. Fragmento donde Oracle Corporation renuncia a la aceptación de garantía y responsabilidad

En lo que respecta al soporte de actualizaciones, Oracle publica actualizaciones de sus versiones de java con el propósito de presentar una contramedida para remediar vulnerabilidades sobre java e instrucciones y funcionalidades nuevas sobre java. Al 9 de junio de 2017, existen 148 versiones de java en 8 familias del producto de las cuales solamente la versión SE 8 tiene soporte con actualizaciones de seguridad vigentes, es decir, que las versiones anteriores a la versión SE 8 no tiene contramedidas de remediación de vulnerabilidad si se publican en la CVE. En la ilustración 31 se muestran las versiones por familia:

	Año publicación			Año fin de soporte	
Release	**Inicial**	**Final**	**N. de versiones**	**Publico**	**Para clientes con contrato**
JDK Alpha	1995	-	1*	-	-
JDK Beta	1995	-	1*	-	-
Java 1.0	1996	2002	5*	1.4 (octubre 2008)	1.4 (febrero 2013)
Java J2SE 5.0	2004	2009	23	Abril de 2008 - Descarga hasta noviembre de 2009	Mayo de 2015
Java SE 6	2006	2017	55	Abril de 2013	Enero de 217 solamente para Solaris
Java SE 7	2011	2017	38	Abril de 2015	Enero de 217 solamente para Solaris
Java SE 8	2014	Actualmente	25	Octubre de 2014 para Windows XP y 2003. Aun con soporte publico	
Java SE 9	Desde julio 2017	-	-	-	-
Java SE 10	-	-	-	-	-
			148		
* Cantidad estimada de versiones					

Ilustración 31. Versiones publicadas de java.

Como se evidencia en estos antecedes, se han publicado en los últimos 21 años 148 versiones de Java con el propósito de que sus clientes, tanto públicos como con contrato, actualicen la tecnología de java y remedien las 4.354 vulnerabilidades que están publicadas en la CVE a junio de 2017. De hecho, se

espera que en julio de 2017 se publique una nueva familia de Java (The Register, 2016) y se especula que se está construyendo la familia 10 de acuerdo a unas mejoras a la estructura del lenguaje que están publicadas en el sitio web de java en octubre de 2012 y actualizada en mayo de 2015. (Oracle Corporation OpenJDK Java, 2015). Con la publicación de las nuevas versiones se espera que las versiones SE 8 queden sin soporte público en los próximos años.

En las ilustraciones 32 a la 37 se exponen las versiones publicadas de Java.

		Java Alpha y Beta	
Consecutivos		***Release***	***Release date***
General	***De versión***		
1	1	Java Alpha	1995
2	2	Java Beta	1995

Ilustración 32. Tabla de versiones de java alpha y beta publicados

		Java 1.X	
Consecutivos		***Release***	***Release date***
General	***De versión***		
3	1	JDK 1.0	23/01/1996
4	2	JDK 1.1	19/02/1997
5	3	J2SE 1.2	8/12/1998
6	4	J2SE 1.3	8/05/2000
7	5	J2SE 1.4	6/02/2002

Ilustración 33. Tabla de versiones de java 1.x publicados.

		Java 5.X	
Consecutivos		Release	Release date
General	De versión		
8	1	Java SE 5	4/10/2004
9	2	Java SE 5 Update 1	25/12/2004
10	3	Java SE 5 Update 2	16/03/2005
11	4	Java SE 5 Update 3	3/05/2005
12	5	Java SE 5 Update 4	4/07/2005
13	6	Java SE 5 Update 5	18/09/2005
14	7	Java SE 5 Update 6	7/12/2005
15	8	Java SE 5 Update 7	29/05/2006
16	9	Java SE 5 Update 8	13/08/2006
17	10	Java SE 5 Update 9	12/11/2006
18	11	Java SE 5 Update 10	22/12/2006
19	12	Java SE 5 Update 11	8/03/2007
20	13	Java SE 5 Update 12	11/06/2007
21	14	Java SE 5 Update 13	5/10/2007
22	15	Java SE 5 Update 14	5/10/2007
23	16	Java SE 5 Update 15	6/03/2008
24	17	Java SE 5 Update 16	23/07/2008
25	18	Java SE 5 Update 17	3/12/2008
26	19	Java SE 5 Update 18	25/03/2009
27	20	Java SE 5 Update 19	29/05/2009
28	21	Java SE 5 Update 20	6/08/2009
29	22	Java SE 5 Update 21	9/09/2009
30	23	Java SE 5 Update 22	4/11/2009

Ilustración 34. Tabla de versiones de java 5.x publicados.

		Java 6.X	
Consecutivos		Release	Release date
General	De versión		
31	1	Java SE 6	23/12/2006
32	2	Java SE 6 Update 1	7/05/2007
33	3	Java SE 6 Update 2	3/07/2007
34	4	Java SE 6 Update 3	3/10/2007
35	5	Java SE 6 Update 4	14/01/2008
36	6	Java SE 6 Update 5	5/03/2008
37	7	Java SE 6 Update 6	16/04/2008
38	8	Java SE 6 Update 7	16/04/2008
39	9	Java SE 6 Update 10	15/10/2008
40	10	Java SE 6 Update 11	3/12/2008
41	11	Java SE 6 Update 12	12/12/2008
42	12	Java SE 6 Update 13	24/03/2009
43	13	Java SE 6 Update 14	28/05/2009
44	14	Java SE 6 Update 15	4/08/2009
45	15	Java SE 6 Update 16	11/08/2009
46	16	Java SE 6 Update 17	4/11/2009
47	17	Java SE 6 Update 18	13/01/2010
48	18	Java SE 6 Update 19	30/03/2010
49	19	Java SE 6 Update 20	15/04/2010
50	20	Java SE 6 Update 21	7/07/2010
51	21	Java SE 6 Update 22	12/10/2010
52	22	Java SE 6 Update 23	8/12/2010
53	23	Java SE 6 Update 24	15/02/2011
54	24	Java SE 6 Update 25	21/03/2011
55	25	Java SE 6 Update 26	7/06/2011
56	26	Java SE 6 Update 27	16/08/2011
57	27	Java SE 6 Update 29	18/10/2011
58	28	Java SE 6 Update 30	12/12/2011

		Java 6.X	
Consecutivos		Release	Release date
General	De versión		
59	29	Java SE 6 Update 31	14/02/2012
60	30	Java SE 6 Update 32	26/04/2012
61	31	Java SE 6 Update 33	12/06/2012
62	32	Java SE 6 Update 34	14/08/2012
63	33	Java SE 6 Update 35	30/08/2012
64	34	Java SE 6 Update 37	16/10/2012
65	35	Java SE 6 Update 38	11/12/2012
66	36	Java SE 6 Update 39	1/02/2013
67	37	Java SE 6 Update 41	19/02/2013
68	38	Java SE 6 Update 43	4/03/2013
69	39	Java SE 6 Update 45	16/04/2013
70	40	Java SE 6 Update 51	18/06/2013
71	41	Java SE 6 Update 65	15/10/2013
72	42	Java SE 6 Update 71	14/01/2014
73	43	Java SE 6 Update 75	15/04/2014
74	44	Java SE 6 Update 81	15/07/2014
75	45	Java SE 6 Update 85	16/10/2014
76	46	Java SE 6 Update 91	21/01/2015
77	47	Java SE 6 Update 95	14/04/2015
78	48	Java SE 6 Update 101	15/07/2015
79	49	Java SE 6 Update 105	20/10/2015
80	50	Java SE 6 Update 111	20/01/2016
81	51	Java SE 6 Update 113	5/02/2016
82	52	Java SE 6 Update 115	21/04/2016
83	53	Java SE 6 Update 121	19/07/2016
84	54	Java SE 6 Update 131	18/10/2016
85	55	Java SE 6 Update 141	17/01/2017

Ilustración 35. Tabla de versiones de java 6.x publicados.

		Java 7.X				Java 7.X	
Consecutivos		Release	Release date	Consecutivos		Release	Release date
General	De versión			General	De versión		
86	1	Java SE 7	28/07/2011	105	20	Java SE 7 Update 55	15/04/2014
87	2	Java SE 7 Update 1	18/10/2011	106	21	Java SE 7 Update 60	28/05/2014
88	3	Java SE 7 Update 2	12/12/2011	107	22	Java SE 7 Update 65	15/07/2014
89	4	Java SE 7 Update 3	14/02/2012	108	23	Java SE 7 Update 67	4/08/2014
90	5	Java SE 7 Update 4	26/04/2012	109	24	Java SE 7 Update 71	14/10/2014
91	6	Java SE 7 Update 5	12/06/2012	110	25	Java SE 7 Update 72	14/10/2014
92	7	Java SE 7 Update 6	14/08/2012	111	26	Java SE 7 Update 75	20/01/2015
93	8	Java SE 7 Update 7	30/08/2012	112	27	Java SE 7 Update 76	20/01/2015
94	9	Java SE 7 Update 9	16/10/2012	113	28	Java SE 7 Update 79	14/04/2015
95	10	Java SE 7 Update 10	11/12/2012	114	29	Java SE 7 Update 80	14/04/2015
96	11	Java SE 7 Update 11	13/01/2013	115	30	Java SE 7 Update 85	15/07/2015
97	12	Java SE 7 Update 13	1/02/2013	116	31	Java SE 7 Update 91	20/10/2015
98	13	Java SE 7 Update 15	19/02/2013	117	32	Java SE 7 Update 95	19/01/2016
99	14	Java SE 7 Update 17	4/03/2013	118	33	Java SE 7 Update 97	5/02/2016
100	15	Java SE 7 Update 21	16/04/2013	119	34	Java SE 7 Update 99	23/03/2016
101	16	Java SE 7 Update 25	18/06/2013	120	35	Java SE 7 Update 101	18/04/2016
102	17	Java SE 7 Update 40	10/09/2013	121	36	Java SE 7 Update 111	19/07/2016
103	18	Java SE 7 Update 45	15/10/2013	122	37	Java SE 7 Update 121	18/10/2016
104	19	Java SE 7 Update 51	14/01/2014	123	38	Java SE 7 Update 131	17/01/2017

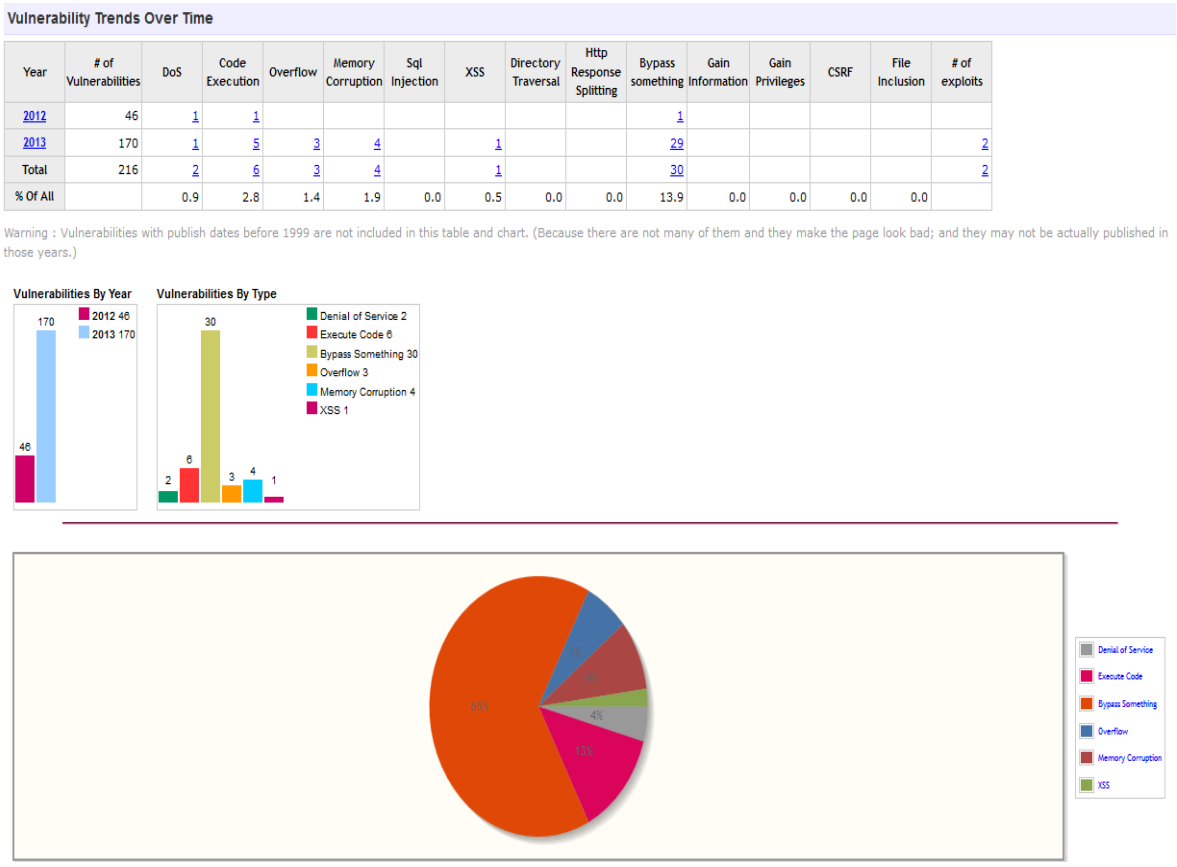
Ilustración 36. Tabla de versiones de java 7.x publicados.

		Java 8.X	
Consecutivos		Release	Release date
General	De versión		
124	1	Java SE 8	18/03/2014
125	2	Java SE 8 Update 5	15/04/2014
126	3	Java SE 8 Update 11	15/07/2014
127	4	Java SE 8 Update 20	19/08/2014
128	5	Java SE 8 Update 25	14/10/2014
129	6	Java SE 8 Update 31	19/01/2015
130	7	Java SE 8 Update 40	3/03/2015
131	8	Java SE 8 Update 45	14/04/2015
132	9	Java SE 8 Update 51	14/07/2015
133	10	Java SE 8 Update 60	18/08/2015
134	11	Java SE 8 Update 65	20/10/2015
135	12	Java SE 8 Update 66	16/11/2015
136	13	Java SE 8 Update 71	19/01/2016
137	14	Java SE 8 Update 72	19/01/2016
138	15	Java SE 8 Update 73	3/02/2016
139	16	Java SE 8 Update 74	3/02/2016
140	17	Java SE 8 Update 77	23/03/2016
141	18	Java SE 8 Update 91	19/04/2016
142	19	Java SE 8 Update 92	19/04/2016
143	20	Java SE 8 Update 101	19/07/2016
144	21	Java SE 8 Update 102	19/07/2016
145	22	Java SE 8 Update 111	18/10/2016
146	23	Java SE 8 Update 112	18/10/2016
147	24	Java SE 8 Update 121	17/01/2017
148	25	Java SE 8 Update 131	18/04/2017

Ilustración 37. Tabla de versiones de java 8.x publicados

A partir de las 148 versiones de Java publicadas se requiere que los clientes públicos y con contrato actualicen la tecnología de java y remedien las vulnerabilidades, sin embargo, en Internet se pueden encontrar equipos que aún tienen versiones de java sin soporte y con vulnerabilidades. Para evidenciar este caso se toma como ejemplo las versiones SE 7 anteriores a 1.7.0 update 4 que están sin soporte:

- En la ilustración 38 se muestra la evidencia de 216 vulnerabilidades publicadas entre 2012 y 2013, de denegación de servicio, ejecución de código, bypass, desbordamiento de memoria, corrupción de memoria y cross site scripting publicadas para java versiones SE anteriores a la 1.7.0 update 4 que están sin soporte



Vulnerability Trends Over Time

Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	Http Response Splitting	Bypass something	Gain Information	Gain Privileges	CSRF	File Inclusion	# of exploits
2012	46	1	1							1					
2013	170	1	5	3	4		1			29					2
Total	216	2	6	3	4		1			30					2
% Of All		0.9	2.8	1.4	1.9	0.0	0.5	0.0	0.0	13.9	0.0	0.0	0.0	0.0	

Warning : Vulnerabilities with publish dates before 1999 are not included in this table and chart. (Because there are not many of them and they make the page look bad; and they may not be actually published in those years.)

Ilustración 38. Registros de vulnerabilidades JAVA 1.7

- En Shodan, un motor de búsqueda que permite encontrar equipos (routers, servidores, entre otros.) conectados a Internet con un criterio de banners de servicios y recogiendo datos sobre servidores web (HTTP puerto 80, 8080, HTTPS puerto 443, 8443), datos de FTP (21), SSH (22) Telnet (23), SNMP (161) y SIP (5060), entre otros, se consultan los equipos que en Internet tienen java 1.7.0.

  En la ilustración 39 se evidencia como al 9 de junio de 2017 se encuentran 18.059 equipos vulnerables en países como Estados Unidos, Brasil, Alemania, Irlanda, China, entre otros. En los equipos vulnerables se encuentran varios sistemas operativos y productos que funcionan en conjunto con java 1.7.

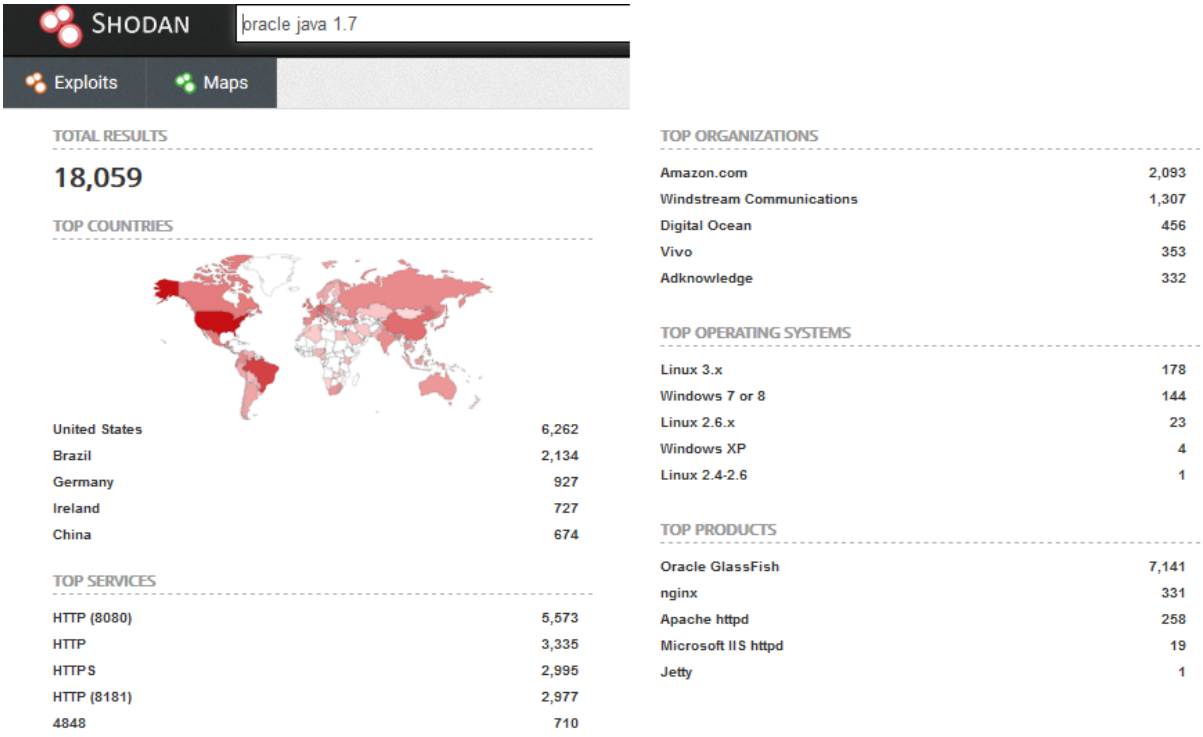


Ilustración 39. 18.059 equipos vulnerables en Internet por usar java 1.7.

Con la evidencia en los datos mostrados hasta ahora, se hace una investigación básicamente exploratoria para revisar porqué hay equipos con versiones vulnerables en equipos en Colombia. A continuación se muestran los resultados encontrados:

- En ocasiones las actualizaciones que al instalar remedian las vulnerabilidades, pueden requerir la intervención del usuario. Se muestran avisos en el sistema que los usuarios deben aceptar, como se muestra en la ilustración 40:

Ilustración 40. Remediaciones con intervención de usuario en Windows 7.

- En la ilustración 41 se expone como en ocasiones las actualizaciones que se instalan no reemplazan la versión del software vulnerable:
  - Al instalar una versión de java de la misma familia pero con otra arquitectura del sistema (32 o 64 bits) no se elimina la versión vulnerable
  - Al instalar una versión de otra familia no elimina la versión vulnerable.

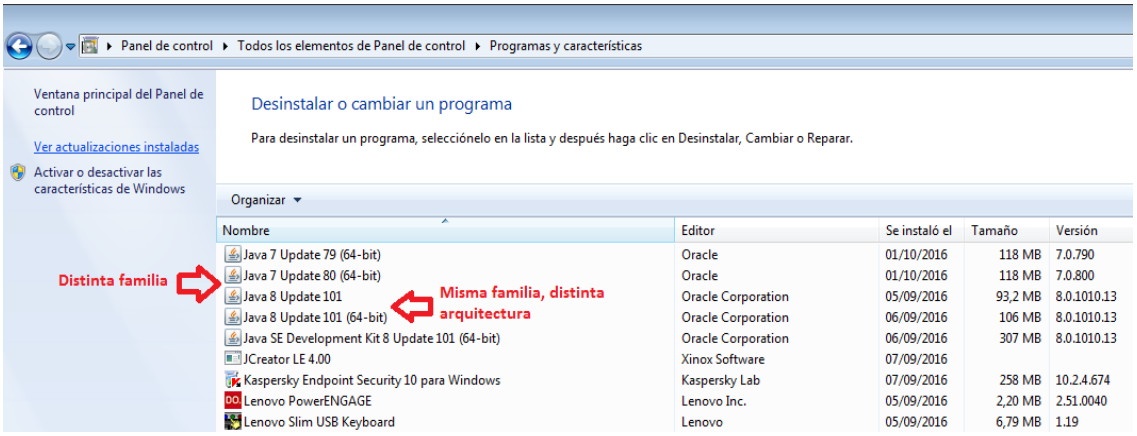


Ilustración 41. Instalación de java de distintas familias y con distintas arquitecturas.

  - En ocasiones aplicar una actualización afecta el funcionamiento de una aplicación. En este sentido la disponibilidad puede afectar a las organizaciones. La ilustración 42 hace referencia a un aviso de cierre de un centro de atención por problemas de disponibilidad en su sistema.

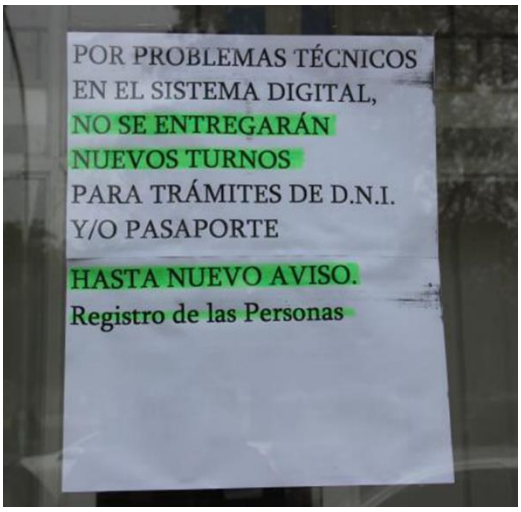


Ilustración 42. Imagen tomada de Google de un aviso de problemas técnicos

Este problema se ha visto en varias aplicaciones de entidades públicas y empresariales en Colombia y en el mundo. En las ilustraciones 43 a la 51 se evidencian aplicaciones afectadas por la actualización de java.

- Mal funcionamiento del Sistema de Información para el Control de Sustancias y Productos Químicos de la Policía Nacional de Colombia

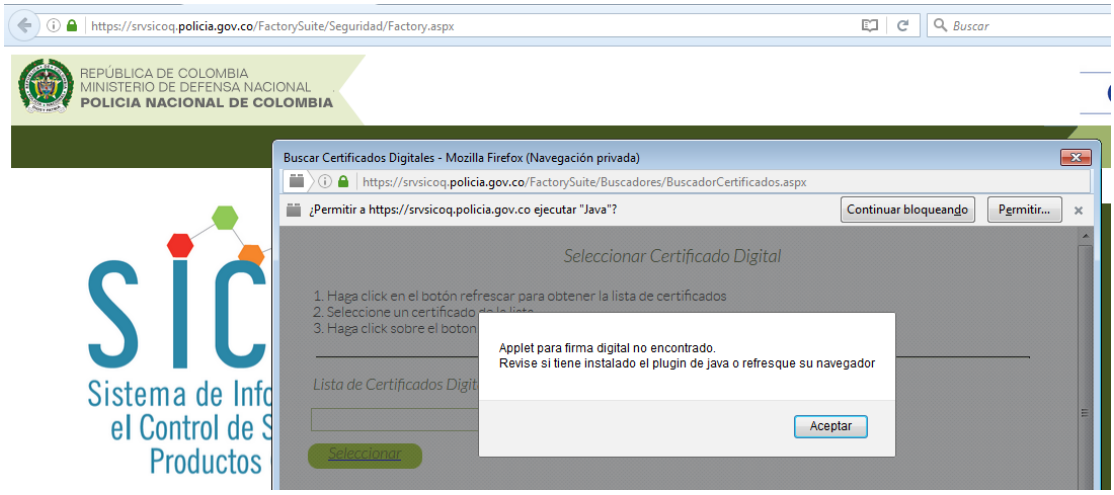


Ilustración 43. Mal funcionamiento del Sistema de Información para el Control de Sustancias y Productos Químicos de la Policía Nacional de la Republica de Colombia

- No funcionamiento del Sistema de Información Geotécnica de la empresa de Acueducto y Aseo de Bogotá

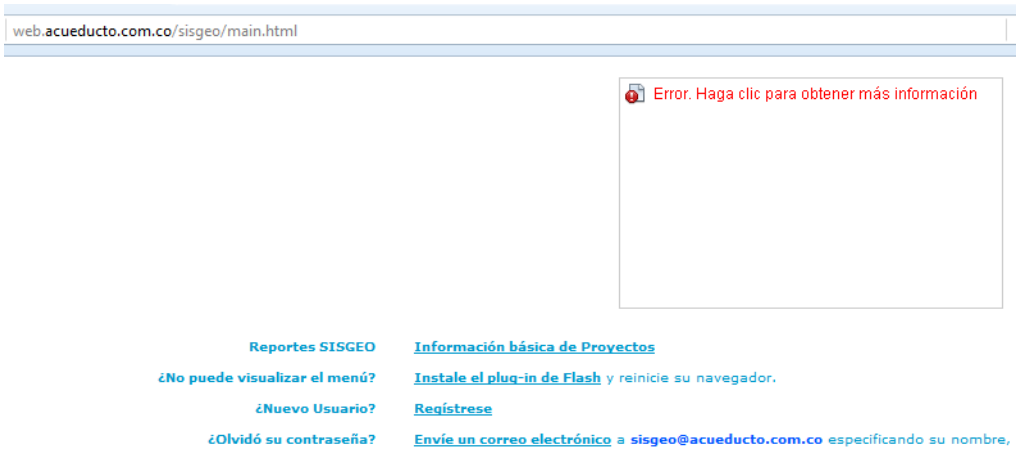


Ilustración 44. Error por usar una versión de java actualiza en el Sistema de Información Geotécnica de la Empresa de Acueducto y Aseo de Bogotá.

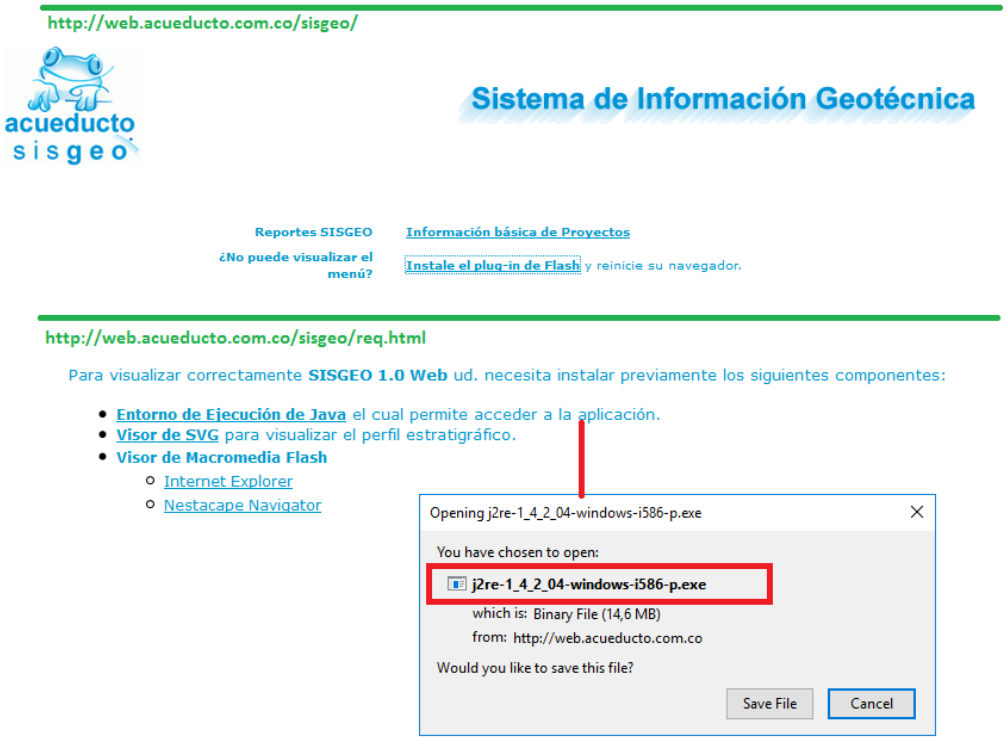


Ilustración 45. Versión requerida por el Sistema de Información Geotécnica de la Empresa de Acueducto y Aseo de Bogotá.

- El no funcionamiento de la aplicación MUISCA de la Dirección de Impuestos y Aduanas Nacionales de la Republica de Colombia

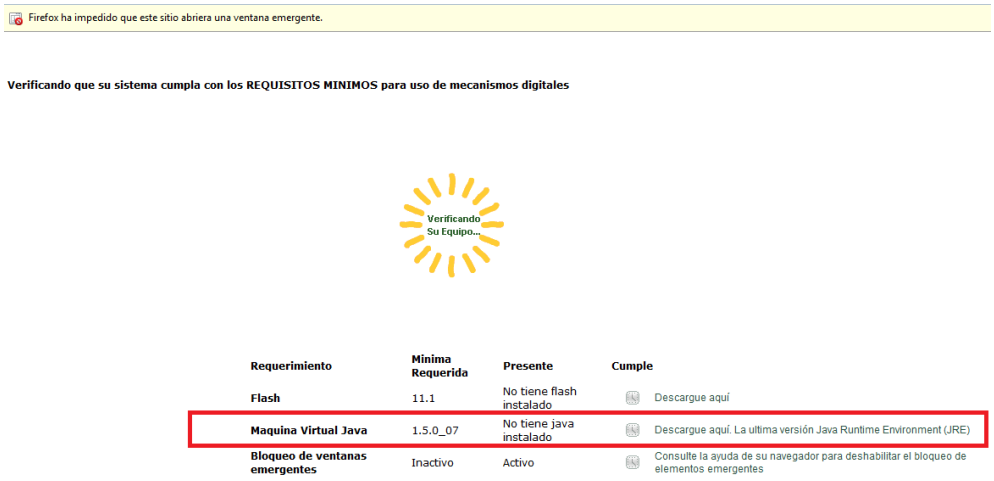


Ilustración 46. Exigencia de una versión de java vulnerable en MUISCA Dian

El funcionamiento de la aplicación presentaba tantos problemas por el cambio de versión de java que la DIAN tenía una lista de errores conocidos, que en caso de no tener conocimiento técnicos, los conceptos no eran muy claros:

Ilustración 47. Lista de errores publicados por DIAN para los usuarios que tenían problemas con la aplicación MUISCA

El funcionamiento de la aplicación afectaba a Contadores, Auxiliares contables, y en general cualquier profesional que requería acceder a la plataforma para cargar los datos financieros de las empresas

Ilustración 48. Noticia de funcionamiento lento de la infraestructura de la DIAN.

Cuando en las empresas o los profesionales que usaban la plataforma MUISCA en sus equipos de uso doméstico, y no encontraban la versión de Java requerida, se conseguía de foros en Internet a pesar de no estar publicadas en la página oficina de Oracle Corporation:

Ilustración 49. Foro personal con la versión de java 5 update 07 publicada para descarga

- Exigencia de la web de certicamara para usar una versión desactualizada y vulnerable de java.

Ilustración 50. Exigencia de la versión SE 7 de Java en la web

Incluso este sitio tiene versiones de software para sistemas operativos sin soporte:

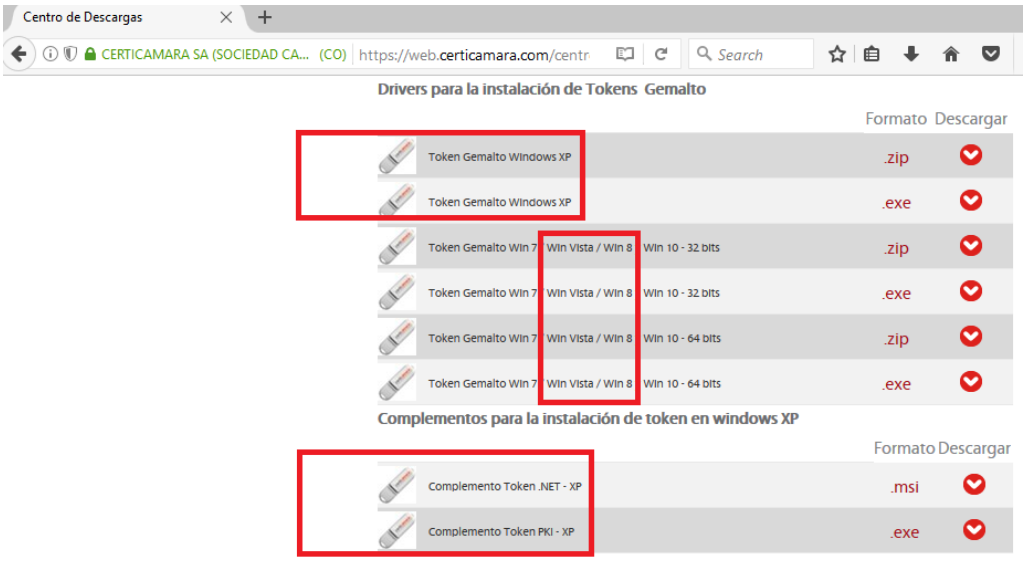


Ilustración 51. Catálogo de descarga de software de la web de la certicamara

Mientras exista una vulnerabilidad puede existir un mecanismo para aprovecharse de las vulnerabilidades. En las ilustraciones 54 a la 55 muestran como se pueden encontrar videos en YouTube con explicaciones sobre cómo usar mecanismo de explotación de vulnerabilidades java

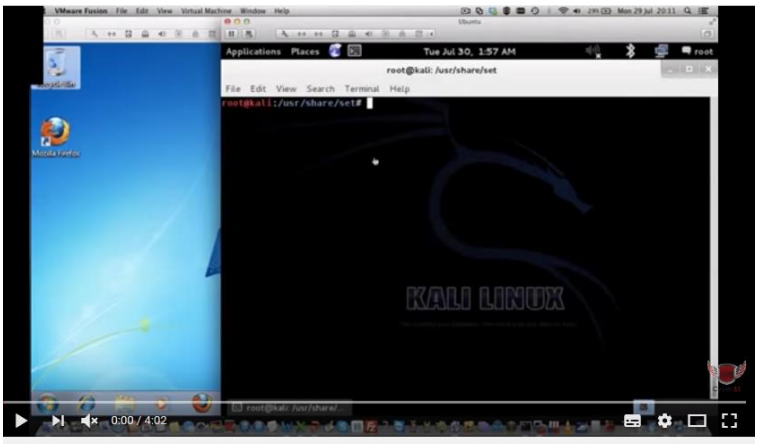


Ilustración 52. How to hack Windows 7 Java Applet Exploit Client side Attack

Ilustración 53. Atacking Windows 8 with Java Exploit and Metasploit (Antivirus Vypass/Evasion)

Ilustración 54. Java Every-Days:Exploiting Software Running on 3 Billion Devices.

Los impedimentos para remediar y las consecuencias por no hacerlo a pesar de que existen mecanismos para actualizar versiones de Java vulnerables, cobraron consecuencias graves para el uso de Java en Internet. Google retiro el uso de java en su navegador Google Chrome. En las ilustraciones 55 y 56 se evidencia como Oracle y Google anunciaban que el plugin de java deja de funcionar para Chrome.

Ilustración 55. Aviso de Oracle para el no funcionamiento de java en Google Chrome

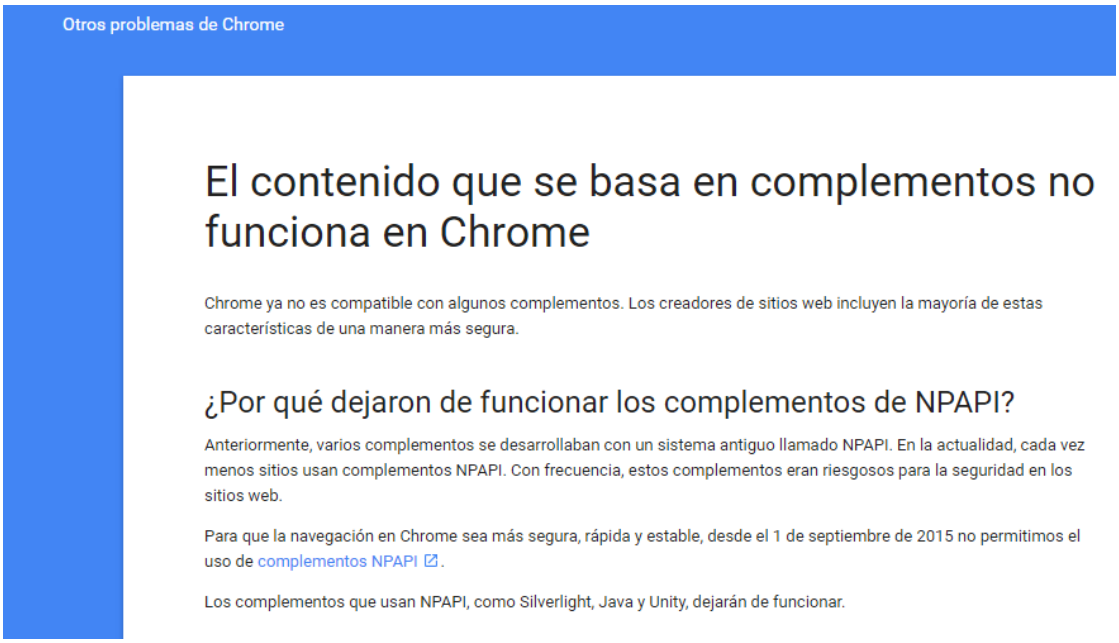
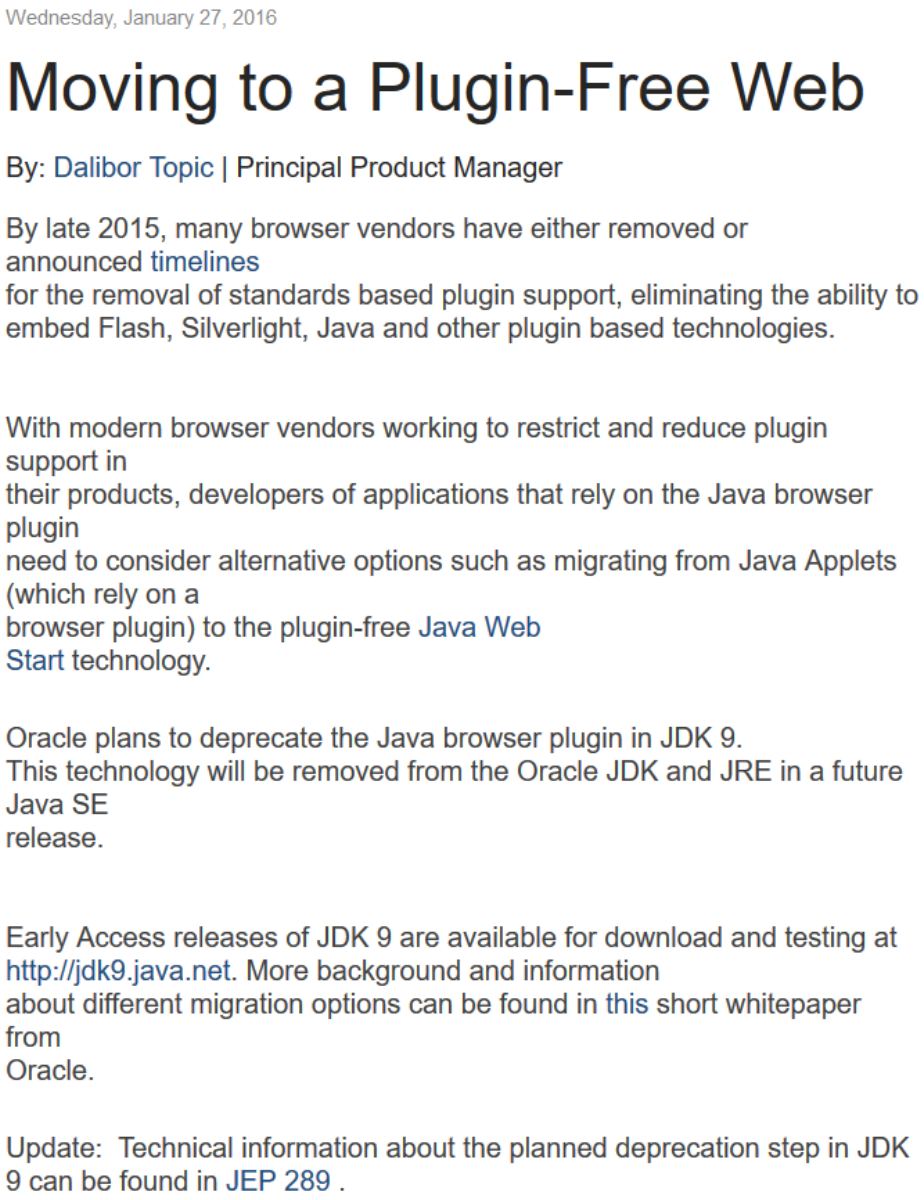


Ilustración 56. Anuncio de Google donde expresa que java no funciona en Google Chrome.

Por estas acciones, la vida del plugin de Java terminó el 22 de septiembre de 2016, cuando se publicó el JDK 9. Sin embargo, no es claro cuánto tiempo se

seguirá viendo por culpa de aplicaciones que se dañen si se remedia o desinstala java. (Oracle Corporation, 2016)

La ilustración 57 muestra el blog donde se anuncia el fin del plugin de java.



Wednesday, January 27, 2016

Moving to a Plugin-Free Web

By: Dalibor Topic | Principal Product Manager

By late 2015, many browser vendors have either removed or announced timelines for the removal of standards based plugin support, eliminating the ability to embed Flash, Silverlight, Java and other plugin based technologies.

With modern browser vendors working to restrict and reduce plugin support in their products, developers of applications that rely on the Java browser plugin need to consider alternative options such as migrating from Java Applets (which rely on a browser plugin) to the plugin-free Java Web Start technology.

Oracle plans to deprecate the Java browser plugin in JDK 9. This technology will be removed from the Oracle JDK and JRE in a future Java SE release.

Early Access releases of JDK 9 are available for download and testing at http://jdk9.java.net. More background and information about different migration options can be found in this short whitepaper from Oracle.

Update: Technical information about the planned deprecation step in JDK 9 can be found in JEP 289 .

Ilustración 57. Blog de Oracle donde expone el final del plugin de java.

CAPITULO 5. LA GUÍA Y SUS ANEXOS

Este documento tiene el propósito de presentar una guía de mantenimiento para plataformas TI que presenta un panorama de la atención de vulnerabilidades informáticas. Se centra en la remediación de vulnerabilidades en escenarios donde hay Software desarrollado o adquirido por las empresas y que a pesar de la falta de mantenimiento se usa en el ejercicio de una actividad económica.

Presenta el desarrollo de una herramienta informática que expone los problemas asociados al uso de un software vulnerable y su impacto en las personas que no están involucradas en la técnica detrás de una aplicación o software.

La investigación adelantada durante el desarrollo de este documento busca expresar desde la definición de la ciberseguridad, todos aquellas fallas técnicas a las que no se les presta atención y que tienen un impacto en la ciberseguridad de las empresas y en la privacidad de las personas.

La aplicación técnica de guías como estas o la exposición de datos mediante herramientas informáticas busca favorecer y mejorar los procesos involucrados en la atención de vulnerabilidades informáticas, permitiendo que las empresas de desarrollo y empresas consumidoras de software mitiguen los problemas de seguridad en las plataformas TI.

A continuación se muestran los resultados de la investigación.

5.1. GUÍA DE REMEDIACIÓN DE VULNERABILIDADES INFORMÁTICAS EN SOFTWARE

Esta guía revisa en detalle el proceso de evaluación de vulnerabilidades de plataformas tecnológicas, suministrando información de la ejecución paso a paso para ejecutar las actividades requeridas.

En el propósito del documento y en el alcance no se hace mención a la duración y los periodos en los que se realiza la gestión de vulnerabilidades. Es importante conocer que la atención de las vulnerabilidades requiere de la realización del ciclo metodológico durante un periodo no mayor a 45 días. (45 días porque cada 4 semanas salen nuevas correcciones de software Microsoft y software diferente a Microsoft publican actualizaciones cada 4 o 6 semanas).

Atendiendo una alerta mundial: (Ejemplo CVE-2017-143 al 148)

En este escenario el proceso de atención de vulnerabilidades consiste en 7 pasos mencionados en la ilustración 58, que se ejecutan por cada servidor/equipo. Se contempla un paso adicional para casos en los que aplique restauración:

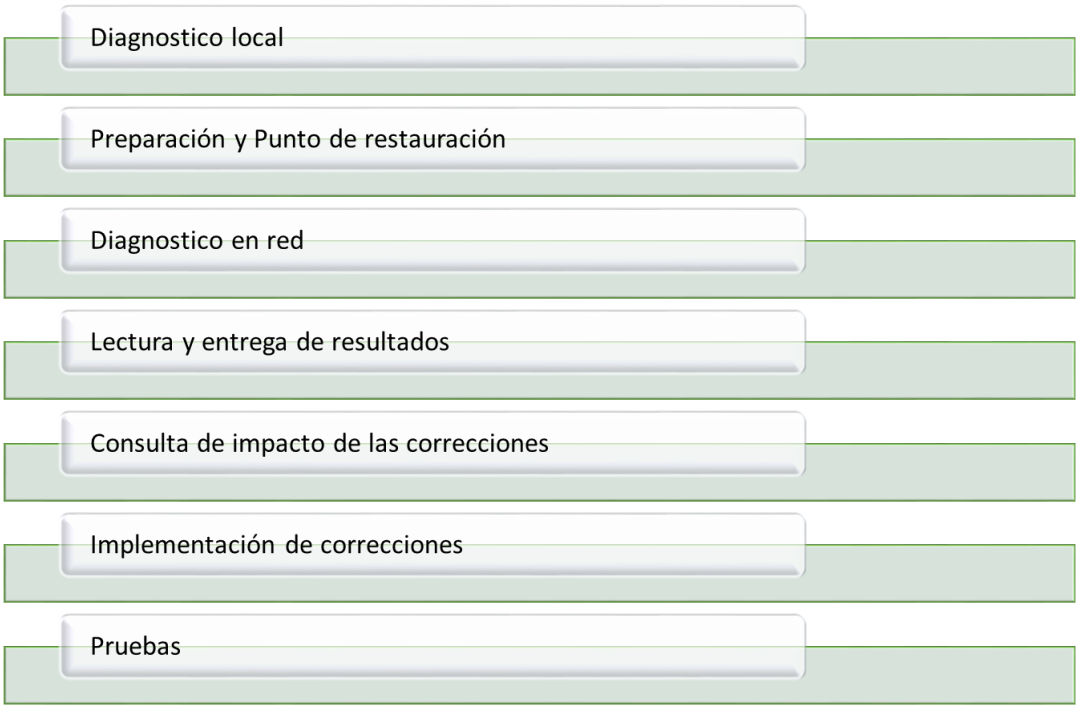


Ilustración 58. Metodología de remediación de vulnerabilidades informáticas extraordinarias

- Paso 1. Diagnostico local:

  Consiste en 4 tareas por servidor que buscan detectar la amenaza que tiene alerta mundial. Ejemplo CVE-2017-143 a 148

  - Se ejecuta un Nmap con un script diseñado para identificar la ausencia de la remediación o la presencia de una vulnerabilidad.

    Para este ejemplo, se muestra este instructivo para verificar en red qué equipos Windows les falta el boletín de seguridad de Microsoft MS17-010.

    1. En el navegador entrar a https://nmap.org/dist/nmap-7.40-setup.exe y descargar el contenido.
    2. Instalar el medio descargado como se describe en los pasos de la ilustración 59.

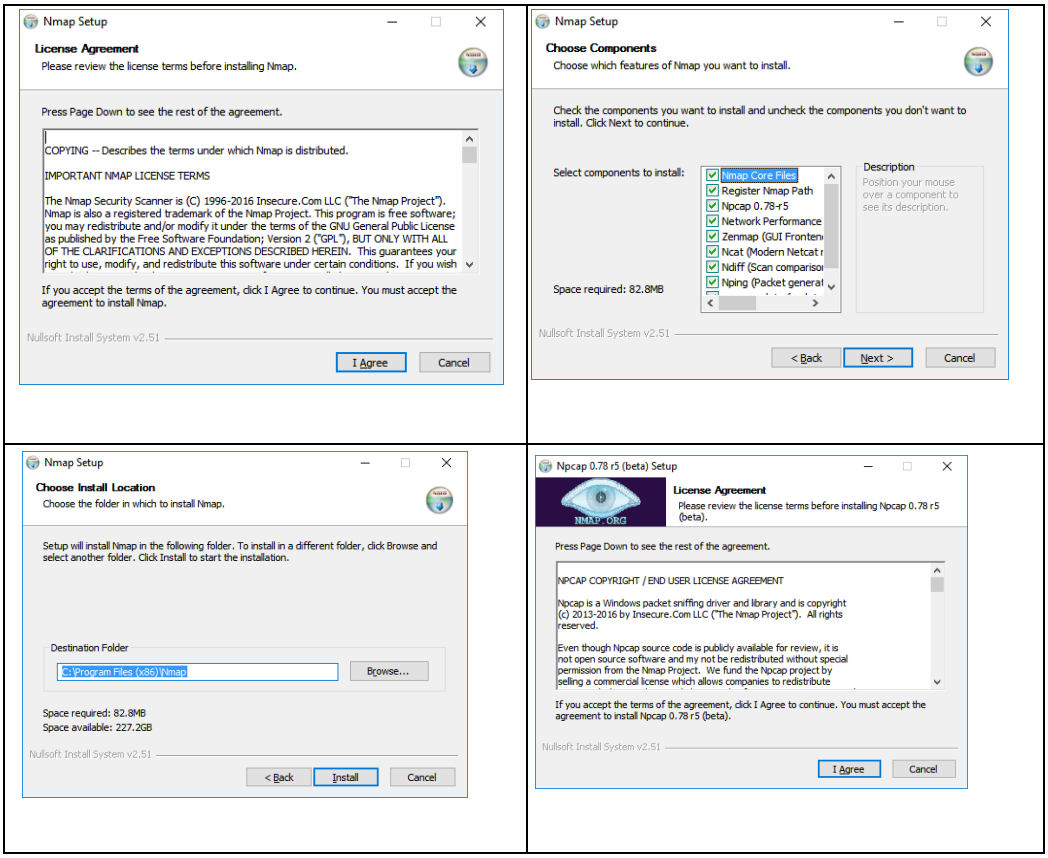


Ilustración 59. Imágenes del procedimiento de instalar Nmap en un Windows.

3. Mover el anexo smb-vuln-ms17-010.nse en la siguiente ruta del equipo, tal como se muestra en la ilustración 60:

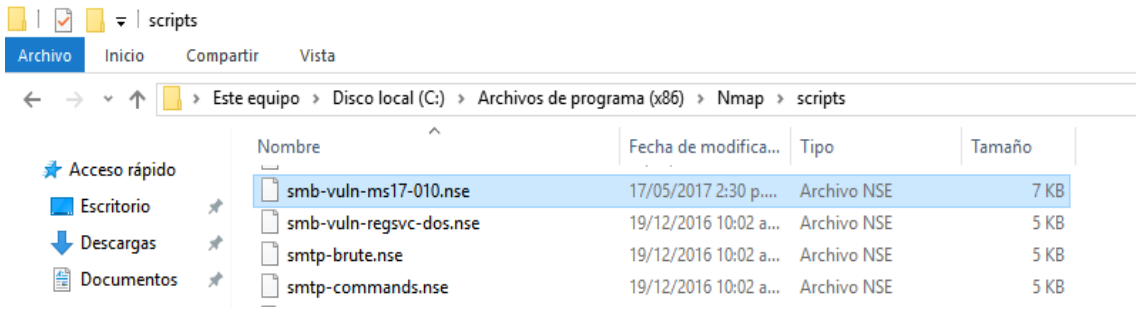


Ilustración 60. Ruta de ubicación de los scripts de Nmap instalado en un Windows.

4. Ejecutar Zenmap como se muestra en la ilustración 61.

Ilustración 61. Icono de ejecutable de Zenmap en Windows

5. En el campo Comando ingresar la siguiente sintaxis cambiando X.X.X.X/X por la dirección de red y sufijo de red (Ejemplo 192.168.0.0/24)

```
nmap -sC -p445 --open --max-hostgroup 3 --script smb-vuln-ms17-010.nse X.X.X.X/X
```

6. Hacer click en Escaneo, como se muestra en la ilustración 62.

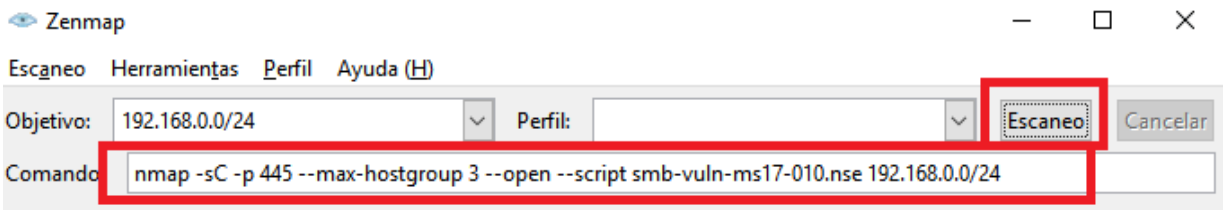


Ilustración 62. Ejecución de script en Nmap en Windows a un segmento de red

El resultado del diagnóstico indica que el equipo/servidor no es vulnerable como aparece en la ilustración 63

```
Starting Nmap 7.40 ( https://nmap.org ) at 2017-05-17 14:45 Hora est. Pacífico, Sudamérica
Nmap scan report for 192.168.0.8
Host is up (0.00038s latency).
PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: 00:25:AB:70:46:96 (AIO LCD PC BU / TPV)

Host script results:
|_smb-vuln-ms17-010: Could not connect to 'IPC$'
```

Ilustración 63. Resultado que aparece cuando NO es vulnerable a SMB-MS17-010

El resultado del diagnóstico indica que el equipo/servidor es vulnerable como aparece en la ilustración 64

```
Starting Nmap 7.40 ( https://nmap.org ) at 2017-05-17 14:49 Hora est. Pacífico, Sudamérica
Nmap scan report for 192.168.140.128
Host is up (0.0014s latency).
PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: 00:0C:29:90:A3:F6 (VMware)

Host script results:
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|        servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|_      https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
```

Ilustración 64. Resultado que aparece cuando es VULNERABLE a SMB-MS17-010

- Se ejecuta un analizador local de vulnerabilidades (ejemplo Windows con el Microsoft Baseline Security Analyser) para tener un escenario inicial de correcciones ausentes en el sistema operativo y reportarlas para iniciar con un proceso de consulta de posible afectación de las aplicaciones que residen en el servidor/equipo.
- Ejecutar una revisión de integridad de las llaves de registro del sistema operativo.
- Ejecutar una revisión de lo hashes del registro del sistema operativo.

- Paso 2. Preparación y punto de restauración.

  Este paso incluye dos macro tareas:

- Se ejecutan las siguientes configuraciones locales en cada servidor/equipo:

  Se crea un usuario administrador local para realizar el análisis. (Al finalizar el proceso se elimina el usuario).

  Se crea un punto de restauración del sistema operativo en caso de que al finalizar el proceso se requiera un rollback, y se hace una copia de seguridad de las bases de datos y archivos que residan en el servidor/equipo.

- Dependiendo del estado actual de los servidores al momento de realizar la preparación se pueden requerir dos reinicios.

  Reinicio al bajar el nivel de seguridad del control de cuentas de Usuario.

  Reinicio para crear punto de restauración.

- Paso 3. Diagnostico en red

  A partir de las configuraciones realizadas en el paso 2, a través de herramientas de escaneo de vulnerabilidades se hace un scan en red o punto a punto a cada servidor con el propósito de identificar vulnerabilidades asociadas a configuraciones y correcciones de seguridad de productos o software instalado en el servidor/equipo, sea de Microsoft o de otro fabricante.

- Paso 4. Lectura y entrega de resultados:

  Con la información recolectada de los escaneos de vulnerabilidades del diagnóstico local del paso 1 y del diagnóstico detallado del paso 3, se hace una relación de las configuraciones, actualizaciones, cambios de versión y desinstalación de software que se requieran para asegurar cada servidor.

- Paso 5. Consulta del impacto al aplicar correcciones:

  Si en el paso 4 no hay una información clara del impacto que pueda tener una corrección dentro del proceso de remediación, se procederá a informar al fabricante o consultar las bases de información de las aplicaciones para decidir si se remedia algún componente del servidor o no.

- Paso 6. Implementación de correcciones:

  Con los pasos anteriores ya definidos se inicia el proceso técnico de aplicar las correcciones requeridas para el aseguramiento de cada servidor.

- Paso 7. Pruebas:

Se ejecutan pruebas cada vez que se implementen correcciones con el propósito de identificar alguna falla que puedan presentar las funciones o aplicaciones de cada servidor. En caso de una denegación de servicio se procede a restaurar el servidor al punto creado en el inicio del aseguramiento.

Haciendo un control periódico

El control de vulnerabilidades informáticas que se ejecuta sobre la plataforma se divide en los 6 pasos que se muestran en la ilustración 65, y se ejecutan de manera mensual o cada 45 días:

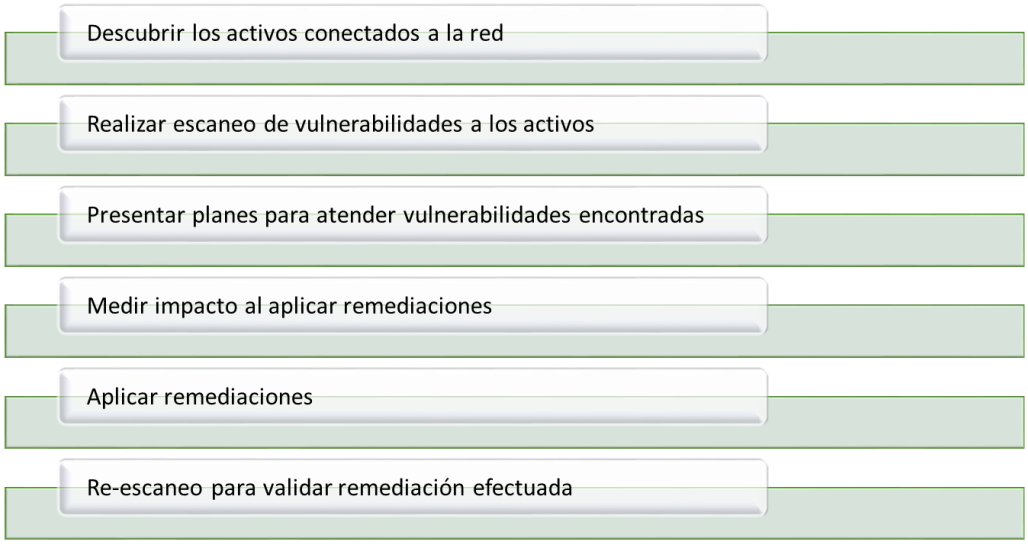


Ilustración 65. Imagen 7. Metodología de remediación de vulnerabilidades informáticas periódico

- Paso 1. Descubrir los activos conectados a la red:

  Aquí a través de herramientas de escaneo de vulnerabilidades se hace un scan para descubrir qué equipos (PC, servidores, portátiles, equipos de red) están conectados en la red, con el propósito de saber la totalidad de elementos que conforman la infraestructura tecnológica.

  Conocer la totalidad de los equipos conectados en la red es indispensable para gestionar la seguridad de toda la infraestructura, de esa manera se hace un aseguramiento a los activos de información de la empresa.

- Paso 2. Realizar escaneo de vulnerabilidades a los activos:

  Este paso incluye dos tareas:

- Para un resultado más efectivo, se ejecutan configuraciones locales en los equipos que permiten una certeza mayor en los resultados del scan de vulnerabilidades.
- En este paso se busca identificar y analizar los datos de una exploración de las vulnerabilidades (puntos débiles en el software instalado en activos) encontradas en Sistemas Operativos, Bases de Datos, aplicaciones, archivos y detecta cuales de esas vulnerabilidades representan un mayor riesgo para los activos informáticos.

- Paso 3. Presentar planes para atender vulnerabilidades encontradas:

  Con la información recolectada de los escaneos de vulnerabilidades del Paso 2, se elaboran y presentan planes aseguramiento, mitigación y remediación de vulnerabilidades informáticas.

- Paso 4. Medir impacto al aplicar remediaciones sobres los activos:

  Se ejecutan las actividades presentadas en los planes de aseguramiento, mitigación y remediación de vulnerabilidades en un ambiente de pruebas presentados en el Paso 3, analizado su impacto con documentación de fabricantes y en casos especiales se recurre a soporte con el fabricante para saber si se puede o no ejecutar en los demás equipos.

- Paso 5. Aplicar remediaciones en cada equipo:

  Se ejecutan las actividades presentadas en los planes de aseguramiento, mitigación y remediación de vulnerabilidades del Paso 3 en todos los equipos previa revisión y correcto funcionamiento analizado en el paso 4.

- Paso 6. Re-escaneo para validar remediación efectuada

  En este paso se busca identificar las vulnerabilidades que fueron atendidas para medir la disminución de las vulnerabilidades encontradas.

Documentando y haciendo planes compensatorios para lo que no se puede remediar.

Si en la ejecución de alguno de los procesos de la guía, no se pueden remediar una o varias vulnerabilidades, se sugiere hacer una documentación para escalar a las directivas o personas responsables. Si no se hace una remediación por culpa de la gestión del mantenimiento del software es necesario aplicar una de las metodologías de desarrollo y estándares expuestos en el marco referencia de este documento. Para cualquiera de los casos por lo que no sea posible la remediación

se sugiere documentar en un sistema de información las vulnerabilidades que no se pueden atender, de acuerdo a los siguientes escenarios:

- Vulnerabilidades sin contramedida por parte del fabricante.
    - Se acepta el riesgo.
    - Se formulan controles compensatorios como:
        - Reportar y Revisar permanente con el proveedor o Internet para identificar remediación o evitar (Bypass) temporalmente la falla.
        - Ubicación del dispositivo en un segmento de red con control de acceso.
        - Vigilancia especial a los registros o log.
        - Adquisición de herramientas especiales.
        - Adicionar evidencias de los controles compensatorios aplicados.
        - Mínimo se revisa y actualiza mensualmente su vigencia y se adiciona evidencia de la revisión.
    - Si se encuentra la solución definitiva cambiar el riesgo a plan de tratamiento y manejarla como tipo II.

- Vulnerabilidades que el administrador del sistema no remedia por miedo o desconocimiento:
    - Se trata el riesgo.
    - Se establece el plan de remediación con las siguientes actividades.
        - Revisar y preparar conjunto de parches que remedian la vulnerabilidad.
        - Analizar impacto en la plataforma completa y aplicativa que corren.
        - Actualizar y ampliar conjunto de parches que afectan la plataforma y aplicativos.
        - En caso de no poder aplicar el arreglo cambiar el tratamiento del riesgo a aceptar y tratar el hallazgo como tipo I.
        - Programar RFC tomando las medidas necesarias para devolverlo acorde con el impacto del cambio.
        - Ejecutar y probar el cambio.
        - Poner en producción y divulgar el cambio a los impactados e interesados.

- Vulnerabilidades que se conoce que afectan las aplicaciones del negocio
    - Se acepta el riesgo.
    - Se formulan controles compensatorios como:
        - Analizar alternativas y tomar acciones para remediación temporal de la falla como:
            - Ubicación del dispositivo en un segmento de red con control de acceso.
            - Vigilancia especial a los registros o log.
            - Adquisición de herramientas especiales.

- Adicionar evidencias de los controles compensatorios aplicados.
- Mínimo se revisa y actualiza mensualmente su vigencia y se adiciona evidencia de la revisión.
- Si se encuentra la solución definitiva cambiar el riesgo a plan de tratamiento y manejarla como tipo II.

- Vulnerabilidades que se conoce pero que no aparecen en una herramienta de gestión de vulnerabilidades:

    - Se trata el riesgo.
    - Se establece el plan de remediación con las siguientes actividades.
        - Revisar y preparar conjunto de parches y actividades que remedian la vulnerabilidad
        - Analizar impacto en la plataforma completa y aplicativa que corren.
        - Actualizar y ampliar conjunto de parches y actividades que afectan la plataforma y aplicativos.
        - En caso de no poder aplicar el arreglo cambiar el tratamiento del riesgo a aceptar y tratar el hallazgo como tipo I.
        - Programar RFC tomando las medidas necesarias para devolverlo acorde con el impacto del cambio.
        - Ejecutar y probar el cambio.
        - Poner en producción y divulgar el cambio a los impactados e interesados.

Ejecutando otros controles

En casos en los que no se quiera depender de los recursos de identificación de un escáner de vulnerabilidades, se sugieren otros planes compensatorios que aunque no corrigen vulnerabilidades si pueden cerrar algunos problemas de seguridad:

- Administradores locales: Los servidores/equipos que están conectados a una misma red deben tener usuarios diferentes al usuario nativo.

- Controladores de dominio propios y de terceros: Revisión de políticas que se aplican a través del controlador de dominio a los equipos vistos en la red, diligencia cuestionarios de hoja de vida del servidor como aparece en la ilustración 66 y 67:

SERVIDORES CON ROL DE DIRECTORIO ACTIVO	
ESTRUCTURA Y ROLES	
¿Hay claridad en la cantidad de servidores que son controlador de dominio?	Se debe saber cuantos servidores hay como controlador de dominio
¿Cada controlador de dominio tiene solo un rol/servicio asignado?	Un controlador de dominio que tiene varios roles asignados son mas dificiles de mantener. Se hace mas compleja la seguridad dado a que si un rol esta mal configurado otro se puede ver afectado
¿En el controlador de dominio se tiene rol DHCP? De ser así ¿están definidas las reglas y scope de DHCP por Vlan?	Se debe tener scope de DHCP por Vlan, para que no se reserven IP de una Vlan equivocada
¿El nivel funcional del dominio y del bosque es igual al de la versión del sistema operativo?	Se recomienda usar el nivel funcional del sistema operativo.
CUENTAS DE USUARIO	
¿Todos las personas que usan los recursos de la infraestructura TI usan un Usuario de Dominio?	Todos las personas que usan recursos deben tener usuario de dominio y no local
¿Solo existe un usuario administrador de dominio?	No deben existir mas de un administrador de dominio a menos que mas de una persona administre el controlador de dominio
¿Las personas que no administran recursos usan usuarios normales y no administradores de dominio?	El control de cuentas de usuario solicita autorización a un administrador local para ejecutar cualquier cambio dentro del sistema
¿Cada usuario utiliza una única identificación (usuario y contraseña) asignada específicamente a él?	Cada usuario que tiene autorizado el acceso debe tener una cuenta unica; no debe compartir la contraseña con otra persona.

Ilustración 66. Primera parte del cuestionario para revisar controladores de dominio

SERVIDORES CON ROL DE DIRECTORIO ACTIVO	
POLITICAS	
¿La gestión de contraseñas garantiza el cumplimiento de una contraseña segura?	Las contraseñas de deben tener prerequisitos de complejidad o usar segundos factores de autenticación.
¿Se obliga a los usuarios a cambiar las contraseñas al iniciar sesión por primera vez?	Las contraseñas deben cambiarse en el primer inicio de sesión de los usuarios.
¿Los usuarios que no ingresan por noventa días o pertenecen a personas que ya no están en la compañía son inactivados?	Deben deshabilitarse, no eliminarse para que quede registro de logs de cada usuario
¿Usuarios de personas que ya no están en la compañía no han sido eliminados?	No debe existir usuarios eliminados
¿Esta documentado cuales son las GPO de seguridad y cuales son de mapeos o de impresión?	Debe existir claridad entre las politicas que se emplean para permisos de red o fileserver frente a las politicas de seguridad
¿Hay politicas de seguridad aplicadas a equipos?	Debe existir claridad entre las politicas que se emplean para permisos en equipos
¿Hay politicas de seguridad aplicadas a usuarios?	Debe existir claridad entre las politicas que se emplean para permisos en usuarios
¿Están documentadas todas las políticas de Dominio no aplicadas?	Debe existir claridad entre las politicas que están creadas pero no se emplean
AMBITOS Y ALCANCE	
¿Se sabe cuantos y cuales son los usuarios que existen en el dominio?	Se debe saber la cantidad de usuarios para mejorar el control de auditoria de cambios y modificaciones
¿Se sabe cuantos y cuales son los grupos que existen en el directorio activo?	Se debe saber la cantidad de grupos para mejorar el control de auditoria de cambios y modificaciones
¿De los grupos que existen, ninguno se usa para excluir la aplicación de alguna GPO?	Grupos que se usen para evitar una politica o conceder un permiso especial
¿Si hay algun grupo que permite la exclusión de alguna GPO, se sabe qué usuarios están en este grupo?	Usuarios que pertenecen a grupos de excepcuón
¿Todos los dispositivos finales y servidores de la compañía están en el dominio?	Todos los equipos deben pasar por el dominio para aplicar politicas de seguridad
MANTENIMIENTO Y ADMINISTRACIÓN	
¿Se han depurado los DNS en los últimos 2 meses?	Las actualizaciones de seguridad deben ser aplicadas con una periodicidad no mayor a 60 días
¿Hay un plan de auditoria que revise la ampliación o modificación de usuarios, grupos, unidades organizativas, sites?	Debe existir mecanismos de control para verificar los cambios que realiza el administrador del controlador de dominio
¿Están documentadas las actividades de operación del controlador de dominio?	Deben estar definidas las tareas que realiza cada usuario sobre el servidor
¿Los responsables de la administración están capacitados técnicamente para realizar sus tareas?	Las personas que administran u operan los servidores deben estar capacitadas para hacerlo
¿Los responsables de la administración, soporte y operación han sido capacitados y han firmado un acuerdo de uso y privacidad de la información?	Las personas que administran u operan los servidores deben conocer su responsabilidad en el uso de estos recursos, y deben firmar un acuerdo de no divulgación

Ilustración 67. Segunda parte del cuestionario para revisar controladores de dominio

- Control de conexión y desconexión de servidores/equipos en la red: Se debe autorizar de manera expresa si un servidor o un pc externo a la red pueden ingresar a la red de datos. En este paso se sugiere también

diligenciar un cuestionario como hoja de vida del servidor/equipo como aparece en las ilustraciones 68, 69 y 70:

SERVIDORES	
CUENTAS DE USUARIO	
¿Está limitado el número de cuentas de usuario que están creadas localmente?	Cuántas y cuales son las personas que están autorizadas para acceder al servidor
¿Está restringido el acceso al equipo para que solo la persona autorizada ingrese en él?	Se revisan los usuarios creados localmente en el servidor para ver que solo las personas autorizadas para acceder al servidor tienen usuario y contraseña asignado.
¿En sistemas operativos Windows el control de cuentas de usuario (UAC) está en el nivel más alto de notificación?	El control de cuentas de usuario solicita autorización a un administrador local para ejecutar cualquier cambio dentro del sistema
¿Cada usuario utiliza una única identificación (usuario y contraseña) asignada específicamente a él?	Cada usuario que tiene autorizado el acceso al servidor debe tener una cuenta unica; no debe compartir la contraseña con otra persona.
PRIVILEGIOS EN CUENTAS DE USUARIO	
¿En sistemas Windows, está controlado el uso de Microsoft PowerShell?	Si no se controla o se usa mal, Powershell es permitir alterar tareas administrativas en el equipo
¿En sistemas Linux, los usuarios especiales y root son usados por personas autorizadas?	Un usuario normal no debe tener privilegios de administrador o especiales.
¿El acceso privilegiado a desarrolladores o administradores al servidor es controlado?, ¿Se sabe qué cambios realiza?	Deben estar documentadas las actividades y los cambios que realizan desarrolladores y administradores sobre el servidor
ACCESO REMOTO	
¿Están instalados o se emplean servicios de mensajería instantánea en los servidores o aplicaciones?, si se usan ¿Está restringida la federación (conectividad publica) que permiten conexiones con usuarios de otras organizaciones o personas?	Verifica que no se pueden establecer conexiones del servidor desde otra organización o una persona ajena a la empresa
¿Están instalados o se emplean servicios de acceso o escritorio remoto en los servidores o aplicaciones?, si se usan ¿Está restringida la federación (conectividad publica) que permiten conexiones con usuarios de otras organizaciones o personas?	Verifica que no se pueden establecer conexiones del servidor desde otra organización o una persona ajena a la empresa
¿Están deshabilitados los servicios asociados a conexiones remotas? De requerirse, ¿Esta establecida una lista de cuentas para que el ingreso sea autorizado únicamente a ellas?	Debe controlarse quién se puede conectar al servidor por terminal server
SOFTWARE NO AUTORIZADO	
¿Si están instalados o se emplean herramientas de auditoría o ethical hacking, el servidor ubicado en una DMZ?	Si el servidor emplea o tiene instalado este software debe estar una DMZ, ya que puede llamar la atención de atacantes u organizaciones de seguridad informática
¿El servidor no tiene instalado ni usa software que emplea protocolos p2p, rsh, ftp?	El rendimiento y seguridad del servidor que emplea este tipo de software va a desmejorar si no se tienen configurados o vigilados
¿El servidor solamente tiene instalado el software que realmente requiere o se utiliza?	No debe estar instalado software ni aplicaciones que no son esenciales o que nunca se usan. De esta manera se puede reducir vulnerabilidades innecesarias en el sistema.

Ilustración 68. Primera parte del cuestionario para revisar servidores

SERVIDORES	
PROTECCION DE ARCHIVOS	
¿Están protegidos contra lectura y escritura archivos importantes almacenados en los sistemas operativos de los servidores o en sus aplicaciones o bases de datos?	Debe verificarse que ningun usuario pueda eliminar archivos en las carpetas del servidor.
CONFIGURACIONES Y CONFIGURACIONES DE SEGURIDAD	
¿Está activado el firewall del sistema operativo y de las soluciones endpoint disponibles en el servidor?	Deben estar activadas las disposiciones de seguridad ofrecidas por fabricantes
¿Está configurado el firewall del sistema operativo con reglas de entrada y salida de puertos necesarios?	Las reglas exceptuadas de entrada y salida del firewall deben estar documentadas para que no existan reglas mal exceptuadas
¿Los servidores donde residan aplicaciones publicadas hacia Internet, están ubicados en DMZ?	Si no esta en una DMZ, la VLAN de servidores se ve expuesta ante un atacante u organización
¿La gestión de contraseñas de las aplicaciones garantiza el cumplimiento de una contraseña segura?	Las contraseñas de las aplicaciones deben tener prerequisitos de complejidad o usar segundos factores de autenticación.
¿Se obliga a los usuarios a cambiar las contraseñas al iniciar sesión por primera vez?	Las contraseñas de las aplicaciones deben cambiarse en el primer inicio de sesión de los usuarios.
¿Están documentadas las relaciones de confianza y sesiones que tiene cada servidor?	Debe existir claridad de que servidores tienen relación de confianza entre ellos y las sesiones y recursos compartidos entre usuarios.
¿Los servicios no utilizados e innecesarios están deshabilitados en el sistema operativo de los servidores?	No debe estar automaticos, ni iniciados los servicios que no son esenciales o que nunca se usan. De esta manera se puede reducir vulnerabilidades innecesarias en el sistema.
¿Las contraseñas de las cuentas privilegiadas (root, administrador, etc.) están deshabilitadas o se cambian cada treinta días?	Estas cuentas si no están deshabilitadas después de su uso pueden ser usadas para efectuar cambios inapropiados o no autorizados sobre el servidor.
¿Los usuarios que no ingresan por noventa días o pertenecen a personas que ya no están en la compañía son inactivados?	Estas cuentas si no están deshabilitadas después de su uso pueden ser usadas para efectuar cambios inapropiados o no autorizados sobre el servidor.
ROLES	
¿Cada servidor tiene solo un rol/servicio asignado?	Un servidor que tiene varios roles asignados son mas dificiles de mantener. Se hace mas compleja la seguridad dado a que si un rol esta mal configurado otro se puede ver afectado
¿Las bases de datos están replicadas en otros servidores mediante la creación de espejo?	Si no hay una replica espejo de la base de datos y está falla, habrá indisponibilidad.
¿La aplicación tiene habilitado o usa protocolos de comunicación OpenSSL o SSL?	OpenSSL y SSL en sus distintas versiones han sido remplazadas por TLS porque estos protocolos han sido descontinuados por inseguros
¿Para roles de DA, están documentadas todas las políticas de Dominio aplicadas y no aplicadas?	Debe existir claridad en las politicas de seguridad que se aplican y no se aplican en el directorio activo

Ilustración 69. Segunda parte del cuestionario para revisar servidores

SERVIDORES	
MANTENIMIENTO Y RESPALDOS	
¿Las actualizaciones automáticas están desactivadas para los servidores donde residan aplicaciones y bases de datos que se afectan con la aplicación de actualizaciones de seguridad?	Cuando reside en el servidor una aplicación o base de datos critica, que puede afectarse cuando una actualización de seguridad se instala, las actualizaciones automáticas deben estar desactivadas
¿Las actualizaciones automáticas están activadas para los servidores donde no residan aplicaciones ni bases de datos que se afectan con la aplicación de actualizaciones de seguridad?	Cuando no reside en el servidor una aplicación o base de datos critica, que puede afectarse cuando una actualización de seguridad se instala, las actualizaciones deben ser automáticas.
¿Se han realizado actualizaciones en los últimos 2 meses?	Las actualizaciones de seguridad deben ser aplicadas con una periodicidad no mayor a 60 días
¿El servidor se encuentra con la última versión de firmware y actualizaciones publicadas por el fabricante?	Los servidores deben estar actualizados a su ultima versión y debe tener aplicadas las actualizaciones de seguridad publicadas por fabricantes.
¿Existe un cronograma de mantenimiento preventivo para los servidores? Si existe, ¿Se cumple los tiempos establecidos?	Para todos los servidores (con actualizaciones automáticas habilitadas y no habilitadas) requieren un mantenimiento preventivo con actividades de actualización, reinicio, depuración, entre otros.
¿Los medios de instalación y actualizaciones de seguridad que instalan se adquieren de una fuente conocida y de confianza?	Los medios de instalación usados en los servidores deben ser entregadas por el fabricante o en su defecto deben ser descargadas de fuentes oficiales.
¿Está documentada la lista de actualizaciones, parches y correcciones que se le han instalado al servidor?	Debe existir una lista de las actualizaciones realizadas a cada servidor.
¿Existen copias de backup que permiten reestablecer el estado de los servidores?	Deben tener politicas de backup para tener copias de respaldo por si un servidor presenta algún daño fisico o logico
¿Tiene repuestos para los servidores, para hacer mantenimientos de emergencia disminuyendo tiempos de disponibilidad?	Deben tener repuestos (discos duros, memorias ram, fuentes de energía, tarjetas de red) por si se presenta daño fisico en los servidores
¿Todos los servidores están ubicados en el datacenter, con controles que garantizan los requisitos ambientales, de energía, comunicaciones, protección contra incendios y acceso físico?	Ningún servidor fisico debe estar en un lugar distinto al centro de computo
¿Existen proyecciones, para necesidades de capacidad futuras, (espacio en discos duros)?	Deben tener un plan para atender una posible ampliación en el almacenamiento o procesamiento en los servidores
¿Las licencias de activación de los distintos componentes de aplicación están protegidos ante robo o pérdida?	Deben estar protegidas para que no se hagan copias o instalaciones no autorizadas. Puede producir multas o perdidad de los medios
¿Existe un servidor con un rol y servicio de WSUS, u otra herramienta que haga sus funciones?	Cuando no reside en el servidor una aplicación o base de datos critica, que puede afectarse cuando una actualización de seguridad se instala, las actualizaciones deben ser automáticas.
ADMINISTRACIÓN	
¿Están documentadas las actividades de operación y de soporte a los servidores?	Deben estar definidas las tareas que realiza cada usuario sobre el servidor
¿Se monitorean las actividades de red que consumen los servidores?	Se deben revisar las actividades de red que genera cada servidor. Esto previene que un servidor tenga indisponibilidad por capacidad en sus tarjetas de red
¿Los responsables de la administración, soporte y operación están capacitados en las distintos sistemas operativos?	Las personas que administran u operan los servidores deben estar capacitadas para hacerlo
¿Los responsables de la administración, soporte y operación han sido capacitados y han firmado un acuerdo de uso y privacidad de la información?	Las personas que administran u operan los servidores deben conocer su responsabilidad en el uso de estos recursos, y deben firmar un acuerdo de no divulgación

Ilustración 70. Tercera parte del cuestionario para revisar servidores

5.2. MECANISMO DE MEDICIÓN

Se proponen dos indicadores de medición para las metodologías expresadas en el anterior capítulo:

- Vulnerabilidades de cualquier nivel de criticidad con mecanismos de explotación.

Ecuación 1. Calculo de remediación por criterio de vulnerabilidades explotables remediadas

$$\frac{\text{Vulnerabilidades explotables remediadas}}{\text{Vulnerabilidades explotables encontradas}} \times 100$$

0-40 %	Inaceptable
41-79 %	Tolerable
80-100 %	Aceptable

Este indicador refleja la forma en que se evalúa la presencia de las vulnerabilidades con mecanismo de explotación dentro de las plataformas.

- Vulnerabilidades de cualquier nivel de criticidad

Ecuación 2. Calculo de remediación de vulnerabilidades remediadas

$$\frac{\text{Vulnerabilidades remediadas}}{\text{Vulnerabilidades encontradas}} \times 100$$

0-40 %	Inaceptable
41-79 %	Tolerable
80-100 %	Aceptable

Este indicador refleja la forma en que se evalúa la presencia de las vulnerabilidades en general dentro de las plataformas.

Ejemplos mostrados en las ilustraciones 71 y 72

Cantidad de vulnerabilidades remediadas	**83.554**	**92%**
Cantidad total de vulnerabilidades encontradas	**91.092**	

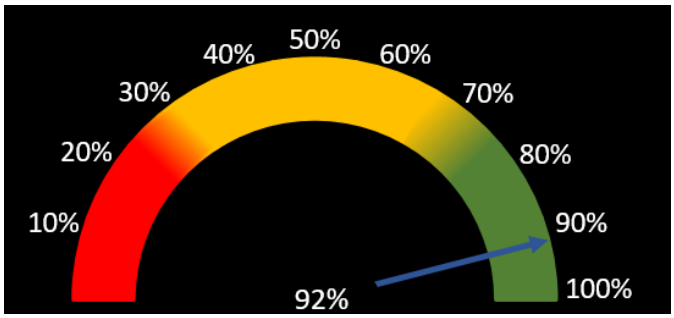


Ilustración 71. Ejemplo para mostrar resultados basados en porcentaje y color 1. Velocímetro

Ilustración 72. Ejemplo para mostrar resultados basados en porcentaje y color 2. Número

5.3. MECANISMO DE DIVULGACIÓN DE INFORMACIÓN

Es importante para los directivos o cualquier interesado en la atención de vulnerabilidades mostrar los resultados marcando la evidencia crítica y el software más afectado, de la manera en la que aparece en las ilustraciones 73 y 74.

VULNERABILIDADES			
CRITICAS	SEVERAS	MODERADAS	TOTAL
30	15	4	49

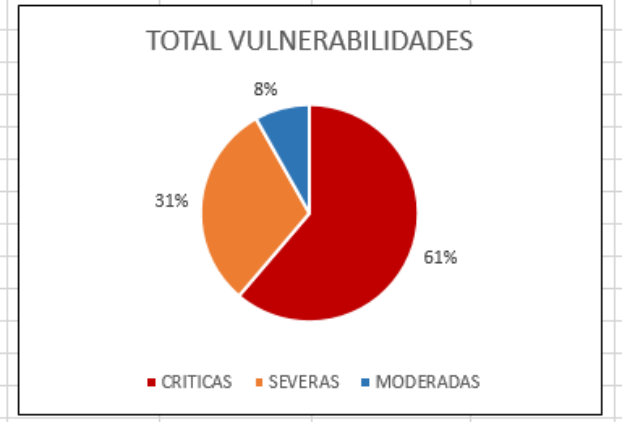


Ilustración 73. Ejemplo como mostrar vulnerabilidades por riesgo

VULNERABILIDADES EXCEPTUADAS			
PHP	APACHE	MYSQL	TOTAL
89	20	111	220

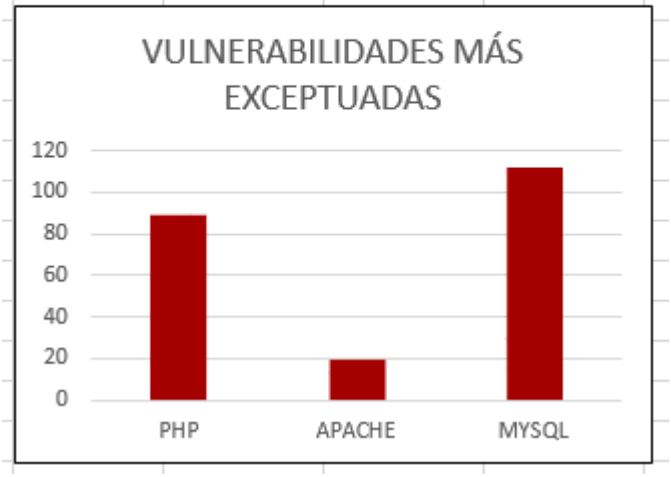


Ilustración 74. Ejemplo como mostrar vulnerabilidades que no se puede remediar

5.4. HERRAMIENTA DESARROLLADA PARA CONSULTA RÁPIDA:

A continuación se muestra la herramienta desarrollada que expone los problemas asociados al uso de un software vulnerable y su impacto mediante una simple consulta con dos criterios. Se muestra en las ilustraciones 75 a 78.

Ilustración 76. Acceso a la aplicación

Ilustración 75. Consulta de vulnerabilidades

Ilustración 78. Lista de vulnerabilidades

Ilustración 77. Estadística de vulnerabilidades encontradas

A continuación, en la ilustración 79, se muestra el modelo entidad relación de la base de datos de la herramienta desarrollada y en la ilustración 80 se muestra la información de las tablas de la base de datos.

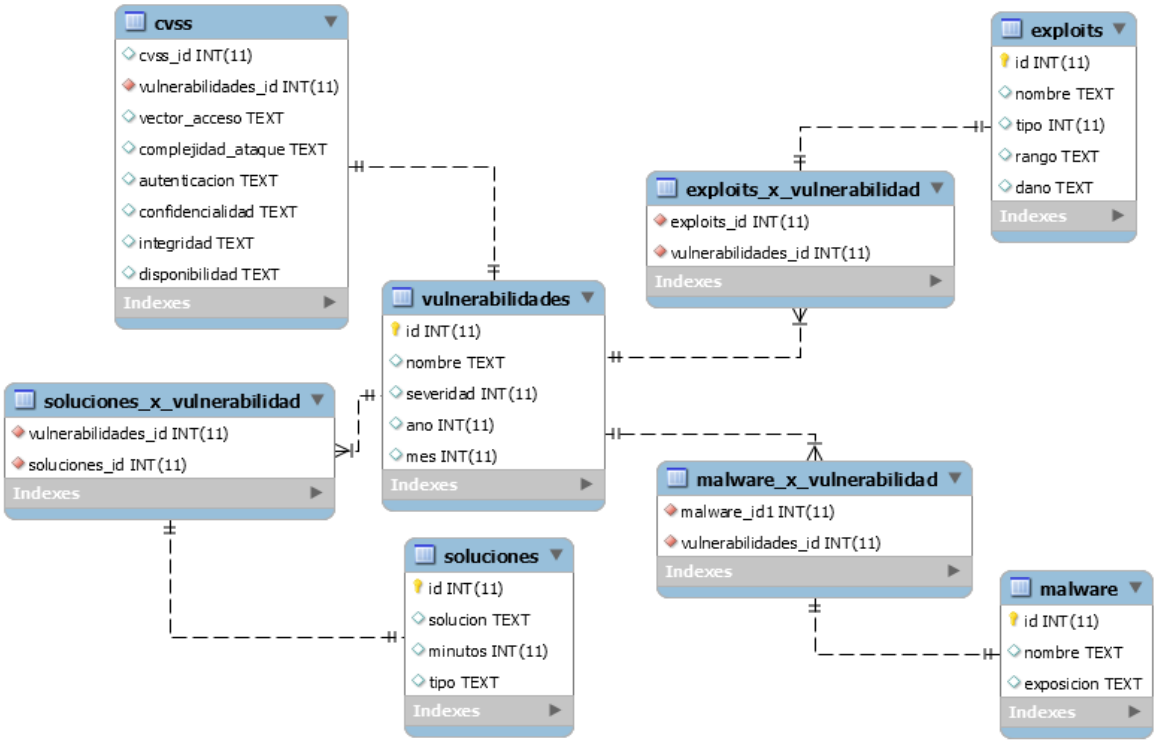


Ilustración 79. MER base de datos de la herramienta desarrollada

Name	Rows	Avg Row Length	Data Length	Data Free	Create Time
cvss	84249	56	4.5 MiB	0.0 bytes	2017-05-18 10:39:30
exploits	4496	353	1.5 MiB	4.0 MiB	2017-05-16 10:08:16
exploits_x_vulnerabilidad	22082	71	1.5 MiB	4.0 MiB	2017-05-17 10:30:11
malware	149	109	16.0 KiB	0.0 bytes	2017-05-16 10:18:03
malware_x_vulnerabilidad	7101	41	288.0 KiB	0.0 bytes	2017-05-17 10:11:28
soluciones	158289	129	19.5 MiB	0.0 bytes	2017-05-16 10:19:25
soluciones_x_vulnerabilidad	909624	36	31.6 MiB	0.0 bytes	2017-05-18 14:17:22
vulnerabilidades	106700	172	17.5 MiB	4.0 MiB	2017-05-16 15:14:39

Ilustración 80. Información de las tablas de la base de datos tomada del motor empleado: MySQL Server

A continuación, en la ilustración 81, se muestra la estructura del proyecto en Netbeans con las clases desarrolladas para la herramienta, en la ilustración 80 se muestra la clase con la que se hace la conexión a la base de datos con el propósito de evidenciar la contraseña y hostname donde se aloja el servidor de MySQL, la ilustración 82 muestra una parte del código desarrollado y de la ilustración 83 a la 87 los códigos SQL para las consultas.

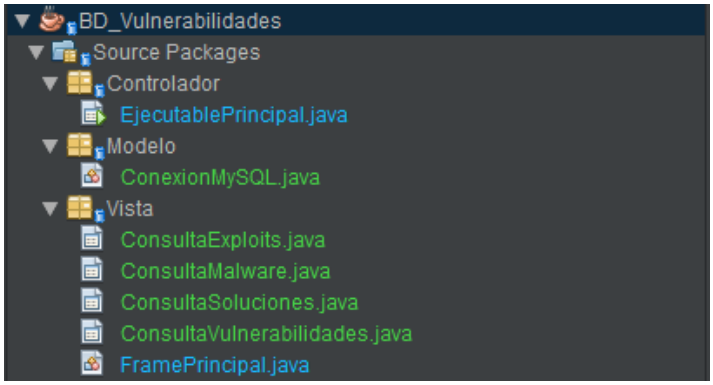


Ilustración 81. Estructura del proyecto en Netbeans con las capas y las clases de la aplicación.

```java
package Modelo;

import java.sql.Connection;
import java.sql.DriverManager;

public class ConexionMySQL {

    public Connection conecction = null;

    public void Conectarse() {

        if (conecction != null) {
            return;
        }

        String url = "jdbc:mysql://localhost/correlacion";
        String password = "1234";

        try {
            Class.forName("com.mysql.jdbc.Driver");

            conecction = DriverManager.getConnection(url, "root", password);

            if (conecction != null) {

                System.out.println("Conectado a la base de datos");
            }

        } catch (Exception e) {
            //JOptionPane.showMessageDialog(null, "PROBLEMAS EN LA CONEXION");
            System.out.println("Problemas de conexion a la base de datos");

        }
    }
}
```

Ilustración 82. Clase en java para la conexión de la aplicación con la base de datos, incluida la contraseña del motor.

```java
package Vista;

import java.awt.*;
import javax.swing.JFrame;
import static javax.swing.JFrame.EXIT_ON_CLOSE;
import javax.swing.JTabbedPane;
import java.sql.SQLException;

public class FramePrincipal extends JFrame {

    /**
     * Yohan Esneider Hernandez
     */
    public static ConsultaVulnerabilidades objVulnerabilidades = new ConsultaVulnerabilidades();
    public static ConsultaExploits objExploits = new ConsultaExploits();
    public static ConsultaMalware objMalware = new ConsultaMalware();
    public static ConsultaSoluciones objSoluciones = new ConsultaSoluciones();
    public static JTabbedPane objJTabbedPane = new JTabbedPane();

    public FramePrincipal() throws SQLException {

        objJTabbedPane.addTab("Vulnerabilidades", objVulnerabilidades);
        objJTabbedPane.addTab("Exploits", objExploits);
        objJTabbedPane.addTab("Malware", objMalware);
        objJTabbedPane.addTab("Soluciones", objSoluciones);

        this.setLayout(new GridLayout(1, 2));
        this.getContentPane().add(objJTabbedPane);

        this.setSize(1250, 650);
        this.setLocationRelativeTo(null);
        this.setVisible(true);
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
        this.setResizable(true);
    }
}
```

Ilustración 81. Sistema de paneles empleado en la aplicación.

```sql
select
    vulnerabilidades.nombre,
    vulnerabilidades.severidad,
    vulnerabilidades.ano,
        cvss.vector_acceso,
        cvss.complejidad_ataque,
        cvss.autenticacion,
        cvss.confidencialidad,
        cvss.integridad,
        cvss.disponibilidad
from
    vulnerabilidades vulnerabilidades
inner join
    cvss cvss
    on
        cvss.vuln_id = vulnerabilidades.id
where upper(vulnerabilidades.nombre) like '%CRITERIO%' and upper(vulnerabilidades.nombre) like '%CRITERIO%'
```

Ilustración 824. Script en sql para consulta de vulnerabilidades y cvss empleado por la aplicación.

```sql
select
    exploits.nombre,
    exploits.tipo,
    exploits.rango,
    exploits.dano
from
    vulnerabilidades vulnerabilidades
left join
    exploits_x_vulnerabilidad exploits_x_vulnerabilidad
    on
        vulnerabilidades.id = exploits_x_vulnerabilidad.vuln_id
left join
    exploits exploits
    on
        exploits_x_vulnerabilidad.exploit_id = exploits.id
where
    upper(vulnerabilidades.nombre) like '%CITERIO%' and upper(vulnerabilidades.nombre) like '%CITERIO%'
```

Ilustración 85. Script en sql para consulta de exploits empleado por la aplicación.

```sql
select
    malware.nombre, malware.exposicion
from
    vulnerabilidades vulnerabilidades
left join
    malware_x_vulnerabilidad malware_x_vulnerabilidad
    on
        malware_x_vulnerabilidad.vuln_id = vulnerabilidades.id
left join
    malware malware
    on
        malware_x_vulnerabilidad.malware_id = malware.id
where upper(vulnerabilidades.nombre) like '%CRITERIO%' and upper(vulnerabilidades.nombre) like '%CRITERIO%'
```

Ilustración 83. Script en sql para consulta de malware que afecta vulnerabilidades empleado por la aplicación.

```sql
select
    vulnerabilidades.nombre,
    soluciones.solucion,
    soluciones.minutos,
    soluciones.tipo
from
    vulnerabilidades vulnerabilidades
left join
    soluciones_x_vulnerabilidad soluciones_x_vulnerabilidad
    on
        soluciones_x_vulnerabilidad.vuln_id = vulnerabilidades.id
left join
    soluciones soluciones
    on
        soluciones.id = soluciones_x_vulnerabilidad.solucion_id
where upper(vulnerabilidades.nombre) like '%CRITERIO%' and upper(vulnerabilidades.nombre) like '%CRITERIO%'
```

Ilustración 847. Script en sql para consulta de remediaciones empleado por la aplicación.

DISCUSIÓN DE CONCLUSIONES

Las necesidades que dieron origen a la investigación y desarrollo de esta guía, se plasman con la intención de exponer los problemas que tiene la remediación de vulnerabilidades informáticas para el beneficio de los sistemas de información y para asegurar la privacidad de los datos personales y la privacidad de las personas. Sin embargo, a pesar de basarse en normas establecidas para buenas prácticas y a la muestra tangible de eventos de seguridad que ocurren u ocurrieron en el pasado, desde la investigación se entiende que no basta con una guía para asegurar sistemas críticos, ni contempla un procedimiento de cómo se debe hacer un proceso de mantenimiento del código fuente de un software ni una hoja de pruebas para los sistemas, por lo que solo se presenta desde el punto de vista metodológico una de las tantas maneras que se pueden atender las vulnerabilidades informáticas en un proceso de remediación periódico.

Las vulnerabilidades que salen en un sistema de gestión de vulnerabilidades como Nessus, Nexpose, Acunetix, o cualquiera que use la base CVE no necesariamente representa resultados con veracidad, podrán existir registro de vulnerabilidades que solo se tratan de un falso positivo –el sistema reconoce una vulnerabilidad que en realidad no existe en el equipo o software evaluado. De igual manera, el hecho de que no aparezca en un sistema de gestión de vulnerabilidades no significa que un sistema es invulnerable, como lo es en el caso de las vulnerabilidades de tipo zero-day.

Es difícil establecer un promedio de las vulnerabilidades que existe en un equipo y en una red de equipos, dependerá en todos los casos del software empleado, los sistema operativos y las versiones qué tan antiguas son y cuántas han sido remediadas, por lo que el proceso de atención de vulnerabilidades puede ser complejo en cualquiera de sus etapas. Sin embargo, a pesar de no tener una claridad de los datos o un estado completo, siempre será bueno atender las vulnerabilidades que se conozcan, al final, cerrar una vulnerabilidad no implica estar al cien por ciento seguro pero si significa una puerta de entrada menos para un atacante.

En casos en los que el desconocimiento de la atención de vulnerabilidades informáticas no permite la remediación en una red de computadores administradas por falta de compromiso o evangelización en una empresa o institución, se pueden emplear los antecedentes expuestos en este documento y la herramienta para mostrar a las vulnerabilidades a las que se expone un sistema, a las explotaciones que tiene, al malware que existe que puede hacer daño con las vulnerabilidades y al esfuerzo requerido para remediar las vulnerabilidades.

BIBLIOGRAFÍA

Alfonzo, P., & Mariño, S. (25 de Febrero de 2015). *Ciencia y Técnica Administrativa*. Obtenido de http://www.cyta.com.ar/ta1202/v12n2a3.htm

Alonso, J. M. (05 de 2017). *Blog un informatico en el lado del mal*. Obtenido de www.elaladodelmal.com

Barrero, G. (04 de 2012). *Blog de Gustavo Barrero*. Obtenido de Aqui la version de java que la dian requiere: https://www.gustavobarrero.com/2012/04/aqui-la-version-de-java-que-la-dian-requiere/

Barrientos, P. A. (25 de Abril de 2014). *Respositorio Institucional de la UNLP*. Obtenido de Universidad Nacional de la Plata: http://sedici.unlp.edu.ar/handle/10915/34969

Barrientos, P. A. (25 de 04 de 2015). *Repositorio Institucional de la UNLP*. Obtenido de Universidad Nacional de la Plata: http://sedici.unlp.edu.ar/handle/10915/34969

Certicamara. (09 de 06 de 2017). *Centro de descargas de certicamara*. Obtenido de https://web.certicamara.com/centro-de-descargas/

CISCO. (10 de 05 de 2017). Curso Introducción a la ciberseguridad.

Cmmi Institute. (24 de Abril de 2015). *Cmmi Institute*. Obtenido de http://cmmiinstitute.com/#hire-suppliers

Codenomicon. (29 de Abril de 2014). *Heartbleed*. Obtenido de Codenomicon: http://heartbleed.com/

Corletti, A. (2011). *Seguridad por Niveles.* Madrid: DarFe Learning Consulting. S.L.

Corletti, A. (2013). *IP versión 6 (Parte 01) – Encabezado.* Madrid: DarFe Learning Consulting. S.L.

Corletti, A. (2013). *IP versión 6 (Parte 01) – Sus componentes,.* Madrid: DarFe Learning Consulting. S.L.

Corletti, A. (2013). *IP versión 6 (Parte 02) – Direccionamiento.* Madrid: DarFe Learning Consulting. S.L.

Corletti, A. (2016). *Seguridad en redes.* Madrid: DarFe Learning Consulting.

DataIFX. (17 de 01 de 2017). *Noticia de funcionamiento lento de la infraestructura de la DIAN.* . Obtenido de http://www.dataifx.com/noticias/hasta-ahora-dian-no-ha-declarado-contingencia-por-problemas-en-su-plataforma

EC-Council. (08 de 05 de 2017). Certified Ethical Hacker v9.

EC-Council. (02 de 05 de 2017). Certified Network Defender v1.

El Espectador. (13 de 05 de 2017). *Elespectador.com*. Obtenido de http://www.elespectador.com/tecnologia/el-virus-wannacry-ya-afecta-unos-100-paises-articulo-693699

Endgame. (30 de 05 de 2017). *Endgame dontwcry youve got*. Obtenido de https://www.endgame.com/blog/technical-blog/dont-wcry-youve-got-endgame

Fiestas, J. (7 de Noviembre de 2014). *Telefonica Eleven Paths*. Obtenido de http://blog.elevenpaths.com/2014/11/qa-pruebas-para-asegurar-la-calidad-del.html

First. (07 de 06 de 2017). *First*. Obtenido de Common Vulnerability Scoring System SIG: https://www.first.org/cvss

Forbes. (14 de 05 de 2017). *Respnsabilidades por WanaCryptor*. Obtenido de https://www.forbes.com/sites/thomasbrewster/2017/05/14/microsoft-just-took-a-swipe-at-nsa-over-wannacry-ransomware-nightmare/#78bbe3ac3585

Foundation, Free Software. (14 de Febrero de 2015). *GNU*. Obtenido de Free Software Foundation: http://www.gnu.org/home.es.html

Fusario, R. J., Crotti, P. S., Bursztyn, A., & Civale, O. (2012). *Teoría De Control Para Informáticos.* MEXICO: ALFAOMEGA.

Garcia Ramírez, F. (25 de Abril de 2015). *Tecnologico Comfenalco*. Obtenido de Academia.com: http://www.academia.edu/2332917/Testing_%C3%81gil_de_Software_con_Herramientas_Libres_y_Abiertas

Garcia, I., Garcia, F., Piattini, M., & Pino, F. (2011). *Calidad De Sistemas De Información - 2. ed. actualizada.* MEXICO: ALFAOMEGA.

Gasteiz, V. (14 de Febrero de 2015). *Eeuskadi*. Obtenido de https://www.euskadi.eus/y79-03/es/contenidos/informacion/herramientas_ejie/es_0213/adjuntos/JUnit.%20Manual%20de%20Usuario%20v1.0.pdf

Gobierno de los Estados Unidos. (06 de 2017). *Gobierno de los Estados Unidos*. Obtenido de https://search.usa.gov/search?affiliate=gobiernousa&query=wanacry

Gozalez, P. (2015). *Ethical Hacking Teoria y practica para la realizacion de un pentesting.* Madrid: 0xword.

ICASI. (10 de 12 de 2016). *ICASI CVRF*. Obtenido de http://www.icasi.org/cvrf/

IEEE, Computer Society;. (10 de Febrero de 2015). *University of Alaska Anchorage*. Obtenido de http://www.math.uaa.alaska.edu/~afkjm/cs401/IEEE830.pdf

IETF Internet Engineering Task Force. (05 de 2000). *Internet Security Glossary.* Obtenido de https://www.ietf.org/rfc/rfc2828.txt

Intel Security. (07 de 06 de 2017). *Intel Malwaretech*. Obtenido de https://intel.malwaretech.com/botnet/wcrypt

Iso. (25 de Abril de 2015). *Iso 25000*. Obtenido de http://iso25000.com/

Kaner, C., & Bach, J. (24 de Abril de 2014). *Context Driven Ttesting*. Obtenido de http://context-driven-testing.com/

McAfee, a Intel Company. (10 de 2015). *McAfee Vulnerability Manager End of Life*. Obtenido de https://www.mcafee.com/us/products/vulnerability-manager-end-of-life.aspx

Microsoft Corporation. (s.f.). *Fin de soporte a Windows XP y 2003*. Obtenido de https://www.microsoft.com/es-es/windowsforbusiness/end-of-xp-support

Microsoft Inc. (10 de 12 de 2016). *Catalog Update Microsoft*. Obtenido de catalog.update.microsoft.com

Montoya, D. E. (2016). *Deep Web: TOR, FreeNet & I2P.* Madrid: 0xword.

Nasa. (14 de Febrero de 2015). *Nasa*. Obtenido de http://space.jpl.nasa.gov/msl/Programs/mariner.html

National Vulnerability Database. (07 de 06 de 2017). *Computer Security Resource Center*. Obtenido de https://nvd.nist.gov/

Netmarketshare. (05 de 2017). *Estadistica de navegadores y sistemas operativos empleados en internet*. Obtenido de https://netmarketshare.com/

NIST National Institute of Standards and Technology. (07 de 2013). *National Institute of Standards and Technology website.* Obtenido de http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-40r3.pdf

Oracle Corporation. (27 de 01 de 2016). *Blog de Oracle* . Obtenido de https://blogs.oracle.com/java-platform-group/entry/moving_to_a_plugin_free

Oracle Corporation. (09 de 06 de 2017). *Terminos de licencia*. Obtenido de http://www.oracle.com/technetwork/java/javase/terms/license/index.html

Oracle Corporation OpenJDK Java. (29 de 05 de 2015). *Oracle Corporation OpenJDK Java*. Obtenido de http://openjdk.java.net/jeps/169

Oracle Corporation. (s.f.). *Políticas de Soporte Técnico de Software de Oracle*. Obtenido de https://www.oracle.com/lad/assets/software-policy-lad.pdf

Pablo Gonzalez Perez, G. S. (2013). *Pentesting con Kali.* Madrid: 0xword.

Pérez , B. (25 de Abril de 2014). *Universidad de la Republica Uruguay*. Obtenido de Facultad de Ingenieria: http://www.fing.edu.uy/~bperez/public/ProTestJIISIC.pdf

Rambla, J. L. (2014). *Seguridad en redes de datos IPv4 e IPv6.* Madrid: 0xword.

Rapid7. (27 de 04 de 2017). Base de datos de vulnerabilidades, exploits, malware.

Rapid7 Metasploit. (05 de 2017). Exploit/windows/smb/ms17_010_eternalblue.

Rengifo, J. I. (25 de Abril de 2014). *Universidad Eafit*. Obtenido de https://repository.eafit.edu.co/bitstream/handle/10784/412/JoseIgnacio_Rengifo_2010.pdf?sequence=1

Sanchez Alonso, S., Sicilia Urban, M. A., & Rordriguez Garcia, D. (2012). *Ingeniería Del Software - Un Enfoque Desde La Guía SWEBOK.* MEXICO: ALFAOMEGA Y GARCETA.

Schmuller, J. (2000). *Aprendiendo UML en 24 horas.* Florida: Pearson Educación.

ShadowBroker. (27 de 04 de 2017). *GitHub de Shadow Broker*. Obtenido de https://github.com/misterch0c/shadowbroker

Shodan. (09 de 06 de 2017). *Browser vulnerability on Internet of the Things*. Obtenido de https://www.shodan.io/search?query=oracle+java+1.7

The Mitre Corporation. (27 de 04 de 2017). *CVE - Requirements and Recommendations for CVE Compatibility - Information-technology Promotion Agency, Japan (IPA) - Vulnerability Countermeasure Information Database (JVN iPedia)*. Obtenido de https://cve.mitre.org/compatible/questionnaires/106.html

The Register. (14 de 09 de 2016). *Fecha de lanzamiento de java 9*. Obtenido de http://www.theregister.co.uk/2016/09/14/jdk_9_release_delay/

TheHackerNews. (14 de 12 de 2016). *TheHackerNews.com*. Obtenido de http://thehackernews.com/2016/12/nsa-hack-shadow-brokers.html

TIOBE. (07 de 06 de 2017). *TIOBE*. Obtenido de Top vendors trend: https://www.tiobe.com/tiobe-index/

Wikipedi. (09 de 06 de 2017). *Historia de versiones de java*. Obtenido de https://en.wikipedia.org/wiki/Java_version_history#cite_note-246

ANEXOS

- Base de datos desarrollada para la herramienta y consultas de la investigación.
    - Dump de la base de datos ‘correlacion’ con las siguientes tablas:
        - correlacion_cvss.sql
        - correlacion_exploits.sql
        - correlacion_exploits_x_vulnerabilidad
        - correlacion_malware
        - correlacion_malware_x_vulnerabilidad
        - correlacion_soluciones
        - correlacion_soluciones_x_vulnerabilidad
        - correlacion_vulnerabilidades
    - Libros en Excel con la misma información de las tablas de la base de datos:
        - Base de datos vulnerabilidades.
        - Segunda Base de datos vulnerabilidades.
    - Scripts de consultas de la base de datos que usa la aplicación
        - consulta listados.sql
        - consultas correlacion.sql
        - exploits.sql
        - malware.sql
        - soluciones.sql
        - vulnerabilidades.sql
- Carpeta del desarrollo de la herramienta informática para consultas
    - La aplicación se llama BD_Vulnerabilidades
    - Las clases que se hicieron en tres capas:
        - Controlador
            - EjecutablePrincipal.java
        - Modelo
            - ConexionMySQL.java
        - Vista

        - ConsultaExploits.java
        - ConsultaMalware.java
        - ConsultaSoluciones.java
        - ConsultaVulnerabilidades.java
        - FramePrincipal.java
    - Imágenes con instrucciones de la aplicación

- Script smb-vuln-ms17-010_nse para usar con Nmap en la detección de la vulenrabilidad CVE-2017-143

- Video Explotación CVE-2017-143
    - Video_exp_cve_17_143

- Certificaciones personales de estudios realizados, y que influyeron en el desarrollo de este trabajo de grado:
    - Certificación de asistencia al curso de EC-Council Certified Network Defender, Mayo 2017
    - Certificación de asistencia al curso de EC-Council Certified Ethical Hacker, Mayo 2017
    - Certificación de curso completado de Introducción a la Ciberseguridad de CISCO, Mayo 2017
    - Certificado de la Dirección de Investigaciones de la Fundación Universitaria los Libertadores del curso virtual de Investigación 1, Junio 2016
    - Certificado de la Dirección de Investigaciones de la Fundación Universitaria los Libertadores del curso virtual de Investigación 2, Diciembre 2016