Método de inclusión de Hacking ético en el proceso de Testing de software

Ariel Giannone, Hernán Amatriain, Dario Rodriguez y Hernán Merlino.

Programa de Maestría en Sistemas de Información Universidad Tecnológica Nacional – Facultad Regional Buenos Aires. Laboratorio de Investigación y Desarrollo en Ingeniería de Explotación de Información
Grupo de Ingeniería de Explotación de Información y Grupo Investigación en Sistemas de Información
giannonearIel@gmail.com, hamatriain@gmail.com, dariorodriguez1977@gmail.com, hmerlino@gmail.com

Resumen. La infraestructura digital se hace cada vez más compleja e interconectada, la dificultad de lograr la seguridad en aplicaciones aumenta exponencialmente; y se plantea la necesidad de una actualización en las herramientas de testeo para la seguridad informática. Se propone un modelo de inclusión de hacking ético en el proceso de testeo de software.

Palabras clave. Seguridad Informática, Hacking ético, OWASP, Testeo de software, intrusión, pen testing.

1. Introducción

El crecimiento explosivo de Internet ha traído muchas cosas buenas tales como el comercio electrónico, facilidad en el acceso a grandes cantidades de almacenamiento de material de referencia, computación colaborativa, e-mail, nuevas vías para la publicidad, información distribuida, por nombrar unos pocos. A medida que la infraestructura digital se hace cada vez más compleja e interconectada, la dificultad de lograr la seguridad en aplicaciones aumenta exponencialmente. [OWASP, 2013] La información para la organización es un activo que debe ser protegido, la piratería es muy común en Internet y tiene afectado a la organización en términos de dinero, la pérdida de recursos y pérdida de imagen. [Sheoran & Singh , 2014]
Cuando el número de estas intrusiones informáticas destructivas se hicieron “famosas”, debido a la visibilidad del sistema o la magnitud del daño infligido, se convirtieron en noticias y los medios de comunicación recogieron estas historias. En lugar de usar el término preciso de "criminal informático", los medios de comunicación comenzaron a usar el término "Hacker" [Raymond, 1991] para referirse a las personas que irrumpen en los ordenadores para la diversión, la venganza, o su propia ganancia.
En su búsqueda de una manera de abordar el problema, las organizaciones informatizadas se dieron cuenta de que una de las mejores formas de evaluar la amenaza de intrusión a sus intereses sería tener profesionales independientes de

seguridad informática intentando entrar en sus sistemas. Este esquema es similar a tener auditores independientes entrando en una organización para verificar sus registros de contabilidad. En el caso de seguridad informática, estos "hackers éticos" emplean las mismas herramientas y técnicas que los intrusos, pero sin dañar el sistema de destino ni robar información.
Ahora bien, estas etapas deben realizarse en un marco de control, gestión y supervisión constante la cual otorgue tranquilidad y seguridad tanto al profesional que se "coloca" en los pies del criminal como a la organización en su totalidad. Es allí donde apunta este trabajo, poder incluir de manera segura y metódica la fase de revisión por hacking ético dentro del proceso de Testing de software.

2. Descripción del Problema

En general, las políticas de seguridad de la información o los controles por sí solos no garantizan la protección total de la información, ni de los sistemas de información, servicios o redes. Después de los controles que habitualmente se implementan, vulnerabilidades residuales probablemente permanezcan haciendo ineficaz la seguridad de la información y por lo tanto los incidentes son aun más probables. Una preparación insuficiente por una organización para hacer frente a este tipo de incidentes hará cualquier respuesta menos efectiva, y aumentar así el grado de impacto comercial potencial adverso. [ISO/IEC 27035:2011].Por otro lado para lograr éxito en las pruebas, este proceso debe ser planificado con antelación. Todos los aspectos técnicos, de gestión y estratégicos deben estar sumamente cuidados. La planificación es importante para todas las pruebas, ya sea desde un simple análisis de contraseña a una prueba de penetración completa en una aplicación web [Mayorga Jácome et al, 2015; Santos Castañeda, 2016; Onofa Calvopiña et al, 2016; López Vallejo, 2017].
Claro que el profesional de seguridad, al llevar a cabo un test de penetración como parte de su trabajo de hacking ético, necesita contar con ese tipo de lógica y tiene que aplicarla, más allá de utilizar las técnicas y herramientas open source [Comunidad Linux,2014; OISSG, 2012; GNU, 2014], comerciales o privadas [Tenable Network Security, 2014], dado que necesita imitar un ataque de la mejor manera y con el máximo nivel posible [Coronel Suarez, 2016; Hurtado Sandoval et al, 2016; López Alvarez, 2016; López Vallejo, 2017]. Para eso, tendrá que emplear todos los recursos que tenga a su alcance, utilizar al extremo sus conocimientos, poder de deducción y análisis mediante el razonamiento y así determinar qué es lo mejor que puede intentar, cómo, dónde y con qué. Por ejemplo, saber si un pequeño dato, por más chico o insignificante que parezca, le será útil y cómo proseguir gracias a él [Tori, 2008].
Tal como exponen diversos autores y organizaciones, se aconsejan herramientas, tareas y como planificar, pero no existe, hasta el momento, ninguna metodología o protocolo que establezca formalmente los pasos a seguir por un profesional dentro del proceso mismo de testing de software, que ayude a lograr un sistema con la menor cantidad de vulnerabilidades expuestas.

3. Solución Propuesta

Se propone desarrollar e incorporar un método de hacking ético para la evaluación de vulnerabilidades dentro del procedimiento mismo de Testeo de un sistema. Suministrando, de esta manera, a los encargados de testing en sectores de Seguridad Informática un grupo de actividades y herramientas que les brinde el soporte necesario para poder prevenir los problemas que en la actualidad son de creciente interés por las pérdidas económicas que conllevan.
Esta metodología se incluye en el proceso mismo de testeo de software, tal como detalla la figura 1, siendo aplicable a cualquier modelo de desarrollo de sistemas.

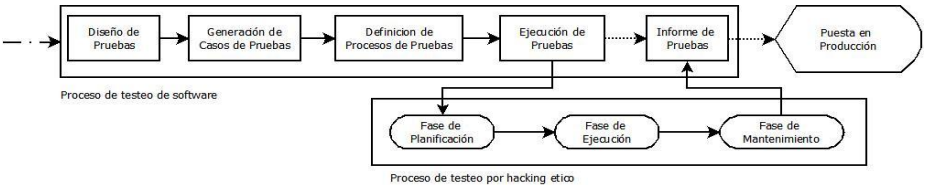

Figura 1. Inclusion del Proceso de Testeo por Hacking Ético

El proceso propuesto está compuesto por tres fases que se muestran en la Figura 2 y detallan en la Tabla 1.

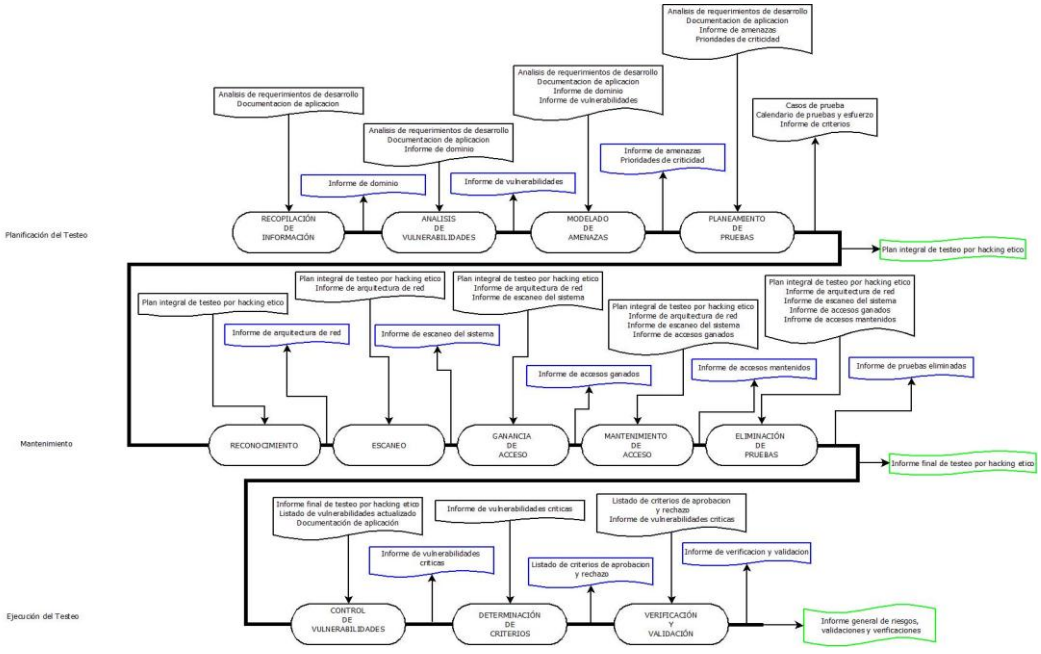

Figura 2. Proceso de Testeo por Hacking Ético

Proceso de testeo de hacking etico			
FASE I: PLANIFICACION DE TESTEO			
ETAPAS	**ACTIVIDADES**	**INSUMOS**	**PRODUCTOS**
RECOPILACION DE INFORMACION	1) Documentar tipo de aplicación 2) Establecer límites de componentes a someter en prueba 3) Delimitar alcance de la prueba 4) Identificar primeros riesgos asociados	• Análisis de requerimientos de desarrollo • Documentación de aplicación	• Informe de dominio
ANALISIS DE VULNERABILIDADES	5) Listar vulnerabilidades 6) Determinar probabilidad de vulnerabilidad 7) Realizar ranking de vulnerabilidades propicias	• Análisis de requerimientos de desarrollo • Documentación de aplicación • Informe de dominio	• Informe de vulnerabilidades
MODELADO DE AMENAZAS	8) Identificar información sensible en el sistema 9) Crear una descripción de la arquitectura 10) Descomponer la aplicación 11) Identificar las vulnerabilidades criticas 12) Documentar las vulnerabilidades criticas 13) Asignar prioridades a las vulnerabilidades criticas	• Análisis de requerimientos de desarrollo • Documentación de aplicación • Informe de dominio • Informe de vulnerabilidades	• Informe de amenazas • Prioridades de criticidad
PLANEAMIENTO DE PRUEBAS	14) Planificar pruebas 15) Armar casos de pruebas 16) Establecer criterios de aprobación y rechazo 17) Establecer tipo de acción (preventiva, correctiva, etc.) 18) Determinar grados de criticidad 19) Calendarizar hitos 20) Estimar esfuerzo 21) Elaborar plan	• Análisis de requerimientos de desarrollo • Documentación de aplicación • Informe de amenazas • Prioridades de criticidad	• Casos de prueba • Calendario de pruebas y esfuerzo (Gantt • Informe de criterios
PRODUCTO FINALDE LA FASE: PLAN INTEGRAL DE TESTEO POR HACKING ETICO			
FASE II: EJECUCION DEL TESTEO			
ETAPAS	**ACTIVIDADES**	**INSUMOS**	**PRODUCTOS**
RECONOCIMIENTO	22) Recopilar información inicial 23) Determinar el tamaño de la red 24) Identificar máquinas activas 25) Descubrir puertos abiertos y puntos de acceso 26) Identificar el sistema operativo 27) Mapear la red completa	• Plan integral de testeo por hacking ético	• Informe de arquitectura de red
ESCANEO	28) Detectar sistemas vivos en la red 29) Descubrir puertos activos 30) Descubrir el sistema operativo 31) Descubrir los servicios ejecutándose y presentes en el sistema 32) Descubrir direcciones IPs	• Plan integral de testeo por hacking ético • Informe de arquitectura de red	• Informe de escaneo del sistema
GANANCIA DE ACCESO	33) Recorrer direcciones 34) Investigar puertos 35) Explotar servicios y sistemas	• Plan integral de testeo por hacking ético • Informe de arquitectura de red • Informe de escaneo del sistema	• Informe de accesos ganados
MANTENIMIENTO DE ACCESO	36) Mantener el acceso 37) Asegurar el acceso exclusivo 38) Cargar, descargar y manipular datos, aplicaciones y configuraciones en el sistema 39) Usar el sistema comprometido para lanzar más ataques	• Plan integral de testeo por hacking ético • Informe de arquitectura de red • Informe de escaneo del sistema • Informe de accesos ganados	• Informe de accesos mantenidos
ELIMINACION DE PRUEBAS	40) Ocultar actos maliciosos. 41) Continuar el acceso al sistema de la víctima. 42) Sobrescribir el servidor, los sistemas y el registro de aplicaciones. 43) Elaborar Informe	• Plan integral de testeo por hacking ético • Informe de arquitectura de red • Informe de escaneo del sistema • Informe de accesos ganados • Informe de accesos mantenidos	• Informe de pruebas eliminadas
PRODUCTO FINALDE LA FASE: INFORME FINAL DE TESTEO POR HACKING ETICO			
FASE III: MANTENIMIENTO			
ETAPAS	**ACTIVIDADES**	**INSUMOS**	**PRODUCTOS**
CONTROL DE VULNERABILIDADES	44) Listar vulnerabilidades nuevas 45) Determinar grado de aplicabilidad	• Informe final de testeo por hacking ético • Listado de vulnerabilidades actualizado	• Informe de vulnerabilidades criticas

DETERMINACION DE CRITERIOS	46) Determinar grado de criticidad 47) Establecer criterios de aprobación y rechazo 48) Listar ranking de vulnerabilidades mas riesgosas	• Informe de vulnerabilidades criticas	• Listado de criterios de aprobación y rechazo
VERIFICACION Y VALIDACION	49) Ejecutar pruebas de verificación 50) Comparar con criterios establecidos 51) Determinar criticidad del riesgo 52) Establecer acciones a tomar 53) Validar resultados 54) Elaborar informe	• Listado de criterios de aprobación y rechazo • Informe de vulnerabilidades criticas	• Informe de verificación y validación
PRODUCTO FINALDE LA FASE: INFORME GENERAL DE RIESGOS, VALIDACIONES Y VERIFICACIONES			

Tabla 2. Proceso de Testeo por Hacking Ético

4. Descripción de Etapas y Actividades

Este proceso puede comenzar en paralelo a la fase de ejecución de prueba dentro del proceso mismo de testeo de software.
El modelo de proceso propuesto para el Testeo por Hacking Ético se encuentra dividido en tres fases principales: Fase de Planificación del testeo, Fase de Ejecución del testeo y Fase de Mantenimiento.

Fase de Planificación del Testeo

Tiene como objetivo interpretar la aplicación, identificar amenazas y vulnerabilidades, planificar y calendarizar las pruebas, entre otras acciones. Posee como entradas principales los Análisis de Requerimientos previos al desarrollo y la Documentación de la aplicación a someter al testeo, generando como producto final el Plan Integral de Testeo por Hacking Ético (PITHE). Esta fase se encuentra conformada por cuatro etapas que se explican a continuación.

Recopilación de Información: se encuentran las tareas correspondientes al entendimiento e interpretación de la aplicación que se someterá al testeo, por lo que se plantean las siguientes actividades: Documentación del tipo de aplicación, Establecimiento de límites de componentes a someter a pruebas, Delimitación del alcance de la prueba e Identificación de primeros riesgos asociados. Dando como producto final el Infrome de Dominio.

Análisis de vulnerabilidade: aparecen las actividades que ayudan a identificar las debilidades a las que la aplicación sería más propicia. Las tareas que la conforman son: Listado de vulnerabilidades, Determinación de probabilidad de vulnerabilidad y Realización de ranking de vulnerabilidades más propicias. Otrogando como producto final el Infrome de vulnerabilidades.

Modelado de Amenazas: donde se realiza una descomposición detallada y minuciosa de la aplicación, identificando y documentando la criticidad de las vulnerabilidades. Para llevar esto a cabo se plantean las siguientes actividades: Identificación de la información sensible dentro de la aplicación, Descripción de la arquitectura, Descomposición de la aplicación, Identificación de las vulnerabilidades críticas, Documentación de las vulnerabilidades y Asignación de prioridades de las vulnerabilidades críticas, donde los productos finales son el Informe de Amenazas y las Prioridades de Criticidad.

Planeamiento de Pruebas: la cual tiene como objetivo el ordenamiento, armado y planificación de las pruebas a realizar, esta es constituida por las actividades: Planificación de pruebas, Armado de casos de pruebas, Establecimiento de criterios de aprobación y rechazo y de tipo de acción, Determinación de grados de criticidad, Calendarización de hitos, Estimación de esfuerzos y por último, la Elaboración del plan integral de testeo por hacking ético. Dando como productos, los Casos de Prueba, Calendario de pruebas y esfuerzo y el Informe de criterios.
Habiendo cumplido con todas estas tareas y armado de productos, obtendremos como salida final el *Plan Integral de Testeo por Hacking Ético (PITHE).* Siendo precisamente el input para la fase siguiente.

Fase de Ejecución del Testeo

La cual tiene como objetivo principal la realización de las pruebas de forma ordenada, pautada y regida por las herramientas que se proponen para llevar adelante y arribar de la mejor forma al Informe Final del Testeo (IFT). Para lograr lo descripto anteriormente, esta fase es compuesta por cinco etapas complementarias.
Reconocimiento: se busca entender, descubrir y recopilar toda la información necesaria para llevar adelante las pruebas sobre la aplicación. Para lograr esto, se plantean estas actividades: Recopilación de información inicial, Determinación del tamaño de la red, Identificación de las máquinas activas, Descubrimiento de puertos abiertos y puntos de acceso, Rastreo del sistema operativo y Mapeo de la red completa. Obteniendo como producto el Informe de Arquitectura de Red.
Escaneo: donde se busca descubrir y determinar todas las características propias del sistema, las tareas que la conforman son: Detección de sistemas vivos en la red y Descubrimiento de puertos activos, sistema operativo, servicios en ejecución y presentes en el sistema y direcciones IPs.
Ganancia de acceso: donde se plantean las siguientes tareas: Escaneo de direcciones, Investigación de puertos y Explotación de servicios y sistemas. Dejando como producto el Informe de Escaneo de sistema. Generando como producto el Informe de Accesos Ganados.
Mantenimiento de acceso: la cual es integrada por las tareas: Mantenimiento de acceso, Aseguramiento de acceso exclusivo, Carga, descarga y manipulación de datos, aplicaciones y configuraciones en el sistema y Utilizar el sistema para lanzar más ataques. Produciendo el Informe de Accesos Mantenidos.
Eliminación de Pruebas: la cual tiene como objetivo el ocultamiento de huellas y la apertura para futuros accesos al sistema.Para esto tiene como tareas las siguientes: Ocultamiento de actos maliciosos, Continuación del acceso al sistema víctima, Sobre escritura del servidor, los sistemas y el registro de aplicaciones y el armado del informe de Pruebas Eliminadas.
Todas estas etapas, tareas, actividades y herramientas, nos llevan a obtener el *Informe Final de Testeo por Hacking Ético (IFTHE).* Este informe contiene la información de que tan crítico, vulnerable o protegido esta nuestro sistema próximo a salir a Producción.

Fase de Mantenimiento

La cual vela por que los sistemas que no son vulnerables en su puesta productiva, no sean posibles objetivos ante nuevas amenazas. Para lograr esto se proponen 3 etapas.
Control de Vulnerabilidades: busca constantemente actualizarse con las vulnerabilidades nuevas que surgen y mide que tan críticas son en nuestras

aplicaciones. Para esto tiene las siguientes actividades ordenadas: Listado de vulnerabilidades nuevas, Determinación de grado de aplicabilidad. Dando como producto el Informe de Vulnerabilidades criticas.
Determinación de Criterios: en ella las tareas Determinación del grado de criticidad, Establecimiento de criterios de aprobación y rechazo y Listado del ranking de vulnerabilidades más riesgosas, buscan determinar que tan crítico y con qué prioridad se deben corregir las vulnerabilidades, donde se genera un Listado de Criterios de Aprobación y Rechazo.
Verificación y Validación: donde las tareas a realizarse son: Ejecución de pruebas de verificación, Comparación con criterios establecidos, Determinación de la criticidad del riesgo, Establecimiento de las acciones a tomar, Validación de los resultados y la Elaboración del informe.
Esta etapa final tiene da como producto el Informe de Verificación y Validación. Todo este proceso, proporcionará como salida un *informe general de riesgos, validaciones y verificaciones. (IGRVV)*, el cual resumirá los riesgos, criticidades y consejos a seguir para el software puesto a prueba.

5. Métricas de vulnerabilidad

Durante la fase de Ejecución del Testeo se plantea el uso de métricas que nos indiquen el grado de vulnerabilidad del sistema al ser desarrollado. Las métricas propuestas son las siguientes:
Índice de posibilidad de acceso (IPA): es directamente proporcional a la cantidad de accesos ganados (CAG) e inversamente proporcional a la cantidad de puertos activos detectados (PAD). Es un índice que tiene un valor entre 0 y 1, siendo los valores cercanos a 0 los que nos dan una medida de un sistema más seguro y un valor cercano a la unidad un sistema más vulnerable.

$$IPA = \frac{CAG}{PAD} \qquad 0 <= IPA <= 1 \qquad \left| \qquad IPA\ \% = \frac{CAG}{PAD} * 100\ \% \qquad IPA << 1 \right.$$

Fórmula 3. Índice de posibilidad de acceso

Índice de acceso real (IAR): directamente proporcional a la cantidad de accesos ganados (CAG) e inversamente proporcional a los puertos activos reales (PAR) del sistema. Es un índice que tiene un valor entre 0 y 1, siendo los valores cercanos a 0 los que nos dan una medida de un sistema más seguro y un valor cercano a la unidad un sistema más vulnerable.

$$IAR = \frac{CAG}{PAR} \qquad 0 <= IAR <= 1 \qquad \left| \qquad IAR\ \% = \frac{CAG}{PAR} * 100\ \% \qquad IAR << 1 \right.$$

Fórmula 2. Índice de acceso real

Índice de detección de puertos (IDP): directamente proporcional a la cantidad de puertos activos detectados (PAD) e inversamente proporcional a los puertos activos reales (PAR) del sistema. Es un índice que tiene un valor entre 0 y 1, siendo los valores cercanos a 0 los que nos dan una medida de un sistema más seguro y un valor cercano a la unidad un sistema más vulnerable.

$$IDP = \frac{PAD}{PAR} \qquad 0 <= IDP <= 1 \qquad \left| \qquad IDP\ \% = \frac{PAD}{PAR} * 100\ \% \qquad IDP << 1 \right.$$

Fórmula 3. Índice de detección de puertos

6. Validación

Para validar la propuesta se construyó una maqueta la cual contiene una maquina virtual Linux donde se instaló la distribución Kali del sistema operativo y por otro lado dos maquinas virtuales mas con una distribución Ubuntu de Linux y Windows XP respectivamente.
Se realizaron dos aplicaciones web para someter a diferentes pruebas, las cuales son muy diferentes conceptualmente, ya que una es un blog de posteos y la otra un gestionador de archivos con login y ftp de archivos.
Tomando como base el modelo de proceso propuesto, se llevaron a cabo las tareas propuestas en cada una de las etapas de las tres fases, comenzando por la planificación, definiendo la estructura (creación de plantillas a seguir durante las subsiguientes fases) del proceso de testeo.
En primer lugar junto al análisis de requerimientos y la documentación de la aplicación, se obtuvo el Informe de dominio, una vez obtenido esto ayuda a realizar la segunda etapa, en la cual se realiza el informe de vulnerabilidades, donde se determina la probabilidad de vulnerabilidades y se ordenan generando un ranking de acuerdo a su propensión. Luego de tener las posibles debilidades listadas, se identificaron y documentaron las mas criticas armando el informe de amenazas y prioridades de criticidad.
Junto a toda esta información, se realiza la última etapa del planeamiento donde se armaron los casos de prueba, se calendariza y se identifican los criterios de aceptación y rechazo.
En las pruebas llevadas a cabo dentro de las vulnerabilidades detectadas hay muchas coincidentes, pero otras que son propias de cada escenario.
Para la fase de ejecución, se utilizaron múltiples herramientas para cada etapa, las cuales se brindan como ejemplo a continuación.
Para recopilar la arquitectura de la red y realizar el escaneo de puertos y servicios activos, se utilizaron herramientas instaladas en la distribución ya mencionada tales como Nmap, masscan SQLmap y wireshark . Los resultados son reflejados en los infromes de arquitectura y escaneo respectivamente.
Para las etapas de ganancia de acceso y mantenimiento, ambos escenarios fueron sometidos a las mismas herramientas, intersec, powersploit, inundator y Slowhttptest,

donde además de recorrer y explotar puertos, se intentó inundar la comunicación y denegar el servicio.
Por último se probaron herramientas forenses para eliminar las pruebas de intrusión.
Recopilando todos los informes se redacta el informe final de testeo por hacking ético, donde se detallan las vulnerabilidades detectadas y en qué medida se cumplen los criterios de rechazo y aprobación.
Como última fase, y ya pensada luego de la puesta en producción, se encuentra la fase de mantenimiento, donde se reutilizan algunas de las herramientas ya mencionadas y se busca mantener la aplicación libre de vulnerabilidades.

7. Conclusiones y futuras líneas de investigación

Durante las pruebas realizadas se pudo comprobar la utilidad de planificar las tareas de testeo de hacking ético, al tener una guía de actividades y consulta de herramientas para las tareas de testeo, recolección de información, análisis y toma de medidas correctivas.
Algunas de estas tareas pueden llegar a pasarse por alto de no contar con una planificación adecuada, junto a un conjunto de medidas a tener en cuenta y herramientas para utilizar.
Si bien se han realizado solamente dos casos de validación, la utilización de las métricas de vulnerabilidad ofrecen una visión general del estado inicial del sistema al ser desarrollado (en cuanto a vulnerabilidades se trate). El empleo de estas métricas y análisis de más proyectos de desarrollo de software siguiendo el presente modelo de proceso para incorporar el haking ético durante el testeo de software, harán posible el desarrollo de indicadores sobre estas métricas para, incluso, poder llegar a realizar una clasificación de sistemas de acuerdo a su grado de vulnerabilidad necesario y el que posee realmente. Así, un sistema de home banking tendrá un indicador de las métricas de vulnerabilidad mucho más ajustado que el de un blog hobista, por indicar un ejemplo.

8. Bibliografía

ANSI/IEEE, (2007). Draft IEEE Standard for software and system test documentation. ANSI/IEEE Std P829-2007.

Comunidad Linux. 2014 <http://www.linux.org/> Página Válida a 05/2018

Coronel Suarez, I.A. (2016). Aplicar hackeo ético para detección de vulnerabilidades mediante herramientas Open Source en las aplicaciones web de una institución de educación superior. Tesis Maestría de la Escuela Superior Politécnica del Litoral.

Evans, Bob (2001). The Sorry State of Software. InformationWeek 112.

GNU (2014) Operating System Sponsored by the Free Software Foundation. 2014/05/15 <http://www.gnu.org/> Página Válida a 05/2018

Hurtado Sandoval, M.E., Mendaño Mendaño, L.A. (2016). Implementación de técnicas de hacking ético para el descubrimiento y evaluación de vulnerabilidades de la red de una cartera de Estado. Tesis Grado. Escuela Politécnica Nacional de Ecuador.

ISO 27035:2011 (2010). Information technology – Security techniques – Information security incident management [Online]. http://www.iso.org/iso/catalogue_detail?csnumber=44379.

López Alvarez, D.M. (2015). Hacking ético para detección de vulnerabilidades de una empresa del sector de telecomunicaciones. Tesis Maestría de la Escuela Superior Politécnica del Litoral.

López Vallejo, M.R. (2017) Hacking ético. Vulnerabilidad de Sistemas Operativos en el acceso por contraseñas. Revista Publicando, 4 No 10. (1). 2017, 31-51. ISSN 1390-9304. <http://rmlconsultores.com/revista/index.php/crv/article/view/407/pdf_259> Página Válida a 05/2018

Mayorga Jácome, T., Quisaguano Belduma, F.J. (2015). Implementación de hacking ético para el análisis de vulnerabilidades, métodos de prevención y protección aplicados a la infraestructura de red de la empresa Construlec Cía. Ltda. en Quito Ecuador. Editorial: Quito: Universidad Israel.

OISSG (2012) Open Information Systems Securty Group. 2003 – 2012 <http://www.oissg.org/> Página Válida a 05/2018

Onofa Calvopiña, F.O., Pilatuña Chica, I. (2016). Análisis y evaluación de riesgos y vulnerabilidades del nuevo portal web de la Escuela Politécnica Nacional, utilizando metodologías de hackeo ético. Tesis Grado. Escuela Politécnica Nacional de Ecuador.

OWASP Top 10 - 2013 (2013). [Online]. http://www.owasp.org. Página Válida a 06/2018.

Palmer, Charles (2001). Ethical hacking, IBM Systems Journal, Vol. 40, N°3

Raymond E (1991). The New Hacker's Dictionary, MIT Press, Cambridge, MA

Santos Castañeda, D.M. (2016). Análisis y diagnóstico de vulnerabilidades informáticas en la red de datos de la empresa YOUPHONE Cía. Ltda. Utilizando Hacking Ético. Tesis Grado. Institucional de la Universidad de las Fuerzas Armadas ESPE.

Sheoran, Pankaj & Singh, Sukhwinder (2014). Applications of Ethical Hacking, International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463 Vol. 3 Issue 5, 05/2014 112-114

Sommerville, I., 2005. Ingeniería del Software. Séptima edición. Capítulo 1. Pearson Addison. ISBN: 84-7829-074-5.

Tenable Network Security (2014), provedora de la herramienta Nessus, 2014 <http://www.tenable.com/products/nessus?gclid=CK2xwJGavr4CFScHwwod9VMAZA> Página Válida a 05/2018

Tori C. (2008). Hacking Ético (1ra Ed). Buenos Aires: Mastroianni.