Investigación en Progreso: Método de Inclusión de Hacking Ético en el Proceso de Testing de Software

Research in Progress: Method for Inclusion of Ethical Hacking in Software Testing Process

Ariel Giannone[1,2]
1. Programa de Maestría en Ingeniería de Sistemas de Información.
Escuela de Posgrado, Facultad Regional de Buenos Aires. Universidad Tecnológica Nacional. Argentina.
2. Laboratorio de Investigación y Desarrollo en Espacios Virtuales de Trabajo (LIDEVT UNLa).
Grupo de Investigación en Sistemas de Información (GISI). Universidad Nacional de Lanús. Argentina.
giannonearIel@gmail.com

***Resumen*—Debido al crecimiento exponencial de Internet y a que las organizaciones poseen cada vez mas información, se hace imprescindible bloquear y eliminar todas las intrusiones posibles. Gracias al hacking ético es posible detectar y corregir algunas de las vulnerabilidades antes que el sistema salga a la luz. Con el fin de mejorar este proceso se intentan incluir estas técnicas y métodos en el proceso tradicional de testing de software dentro de las organizaciones.**

***Abstract*— Due to the exponential growth of the Internet and to the fact that organizations have an increasing amount of digital information, it is essential to identify so as to prevent all possible intrusions. In order to improve this prevention, it is proposed to incorporate into the Software Testing process all existing Ethical Hacking concepts and tools.**

***Palabras Claves*—Hacking ético, Proceso de Testing**

***Index Terms*— Ethical Hacking, Testing Process**

I. JUSTIFICACIÓN

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. Como ocurre con la mayoría de los avances tecnológicos, también hay un lado oscuro: los hackers. Los gobiernos, las empresas, y ciudadanos privados de todo el mundo están ansiosos por ser parte de esta revolución, pero tienen miedo que algún intruso (hacker) irrumpa en su servidor Web y reemplace su logotipo con pornografía, lea su correo electrónico, robe su número de tarjeta de crédito de un sitio de compras en línea, o implante software encubierto para transmitir secretos de su organización a la Internet abierta. [12] El software inseguro está debilitando las finanzas, salud, defensa, energía, y otras infraestructuras críticas. A medida que la infraestructura digital se hace cada vez más compleja e interconectada, la dificultad de lograr la seguridad en aplicaciones aumenta exponencialmente. [11] La información para la organización es un activo que debe ser protegido del acceso no autorizado de personas y el mal uso de algunas otras cuestiones ilegales, 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. [18]

Si la historia sirve de indicio, la comunidad de tecnología de la información fue incapaz de construir sistemas de información en red que pueden prevenir consistentemente ataques con éxito [5]. La escalada natural de amenazas ofensivas contra las medidas defensivas ha demostrado una y otra vez que no hay sistemas prácticos que se puedan construir que sean invulnerables a los ataques. Incluso una organización tal como el Departamento de Defensa de EE.UU. en la red ha demostrado de forma continua el grado de susceptibilidad que posee ante los ataques.

Se supone que el factor principal que contribuye a la mala situación de la seguridad en Internet es la falta de pruebas de software de calidad. La complejidad intelectual asociada con el diseño de software, codificación, y prueba, prácticamente asegura la presencia de "errores" en el software que puede ser explotada por atacantes. La mayoría del software hoy en día es la prueba de errores por el enfoque penetración-parche; cuando alguien encuentra una seguridad explotable "agujero" del fabricante de software emite un parche. Este enfoque ha demostrado ser insuficiente, ya que después de los hechos de seguridad deja abiertas las vulnerabilidades de errores hasta que sean explotados. Sin embargo, los fabricantes de software sostienen que este enfoque es económicamente atractivo, ¿por qué invertir tiempo y dinero en las pruebas de control si los consumidores no están dispuestos a pagar una prima por software seguro?. Otra variable es el tiempo de salida al mercado dicta que el software se libera en la forma más temprana como sea posible, a menudo con graves defectos no detectados de seguridad [20]. El problema presentado por falta de pruebas de calidad también se agrava ante ataques automatizados, la homogeneidad del sistema operativo y las malas prácticas.[17].

Algunos informáticos tienen la impresión equivocada de que su sitio Web no sería un objetivo citando numerosas razones, tales como "No tiene nada interesante en él" o "los hackers nunca han oído hablar de mi empresa". Lo que estas personas no se dan cuenta es que cada sitio web es un objetivo. El objetivo de muchos hackers es simple: hacer algo espectacular y luego asegurarse de que todos sus amigos sepan que hizo. Mismo, a muchos hackers simplemente no les importa que empresa u organización piratear, solo lo hacen porque pueden. Por ejemplo, los administradores Web de UNICEF (Naciones Unidas para la Infancia) podrían bien

haber pensado que ningún pirata informático los atacaría. Sin embargo, en enero de 1998, su página fue completamente desfigurada. [12].

II. Fundamentos

Inicialmente las intrusiones en computadoras eran bastante inocentes, siendo el mayor daño el robo de tiempo de procesamiento. Otras veces, estos ataques serian en forma de bromas. Sin embargo, estas intrusiones no permanecieron así por mucho tiempo. En ocasiones, los intrusos menos talentosos, o menos cuidadosos, accidentalmente "bajaban" un sistema o dañaban sus archivos, y los administradores de sistemas tendrían que reiniciar o hacer reparaciones. Otras veces, cuando se descubrieron sus actividades y se les negó el acceso, los intrusos reaccionaron con acciones aun más destructivas. 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" (Sustantivo. Persona que disfruta el aprendizaje de los detalles de los sistemas informáticos y cómo ampliar sus capacidades, en contraste con la mayoría de los usuarios de computadoras, que prefieren aprender sólo el mínimo cantidad necesaria.) [13] para referirse a las personas que irrumpen en los ordenadores para la diversión, la venganza, o su propia ganancia. Dado que llamar a alguien "hacker" fue originalmente concebido como un cumplido, profesionales de la seguridad informática prefieren utilizar el término "cracker" o "intruso" para los hackers que giran hacia el lado oscuro de la informática, la piratería. [12].

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 se han implementado, vulnerabilidades residuales probablemente permanezcan haciendo ineficaz la seguridad de la información y por lo tanto los incidentes son aun mas posibles. Esto puede llegar a tener efectos negativos tanto directos e indirectos sobre las operaciones de negocio de una organización. Además, es inevitable que se produzcan nuevos casos de amenazas no identificadas previamente. 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. [7]

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. En su lugar, permiten evaluar la seguridad de los sistemas de destino e informar de a los propietarios sobre las vulnerabilidades encontradas junto con las instrucciones de cómo remediarlos.

En resumidas palabras, la evaluación de la seguridad de un sistema por parte de un hacker ético busca responder 3 preguntas básicas:

- ¿Qué puede ver un intruso en los sistemas atacados?
- ¿Qué puede hacer un intruso con esa información?
- ¿Hay alguien en el sistema atacado que se dé cuenta de los ataques o éxitos del intruso?

Este proceso debe ser planificado con antelación. Todos los aspectos técnicos, de gestión y estratégicos deben estar sumamente cuidados. La planificación es importante para cualquier todas las pruebas, ya sea desde un simple análisis de contraseña a una prueba de penetración completa en una aplicación web. El resguardo de datos debe garantizarse, de lo contrario la prueba puede volverse en contra si alguien afirma que nunca se autorizaron las pruebas. Por lo tanto, un alcance bien definido implica la siguiente información:

- Sistemas específicos para probar.
- Estimar los riesgos que están involucrados.
- Tiempo que llevara la prueba y evaluación del calendario general.
- Recoger y explorar el conocimiento de los sistemas que tenemos antes de la prueba.
- Entrega de informes específicos incluyendo informes de evaluación de la seguridad y un informe de nivel superior describiendo las vulnerabilidades generales que deben abordarse, junto con las medidas correctivas que se deben implementar.

Ahora bien 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, comerciales o privadas, dado que necesita imitar un ataque de la mejor manera y con el máximo nivel posible. Para eso, tendrá que emplear todos los recursos de inteligencia 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. Continuamente se deberá enfrentar a etapas que le demanden la mayoría de estas aptitudes [19].

- Definir patrones de conducta y acción.
- Hacer relevamientos pasivos de información.
- Interpretar y generar código y cifrado de datos.
- Descubrir manualmente descuidos en el objetivo.
- Descubrir vulnerabilidades presentes de todo el escenario técnico.
- Proyectarse sobre la marcha en modo abstracto, táctica y estratégicamente.
- Ser exhaustivo, pero a la vez saber cuándo es el momento de recurrir a la distensión para no agotar la mente.

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.

III. Objetivo

El presente plan de trabajo de investigación propone la incorporación del método de hacking ético para la evaluación de vulnerabilidades dentro del procedimiento mismo de Testeo de un sistema. Se intenta, de esta manera, aportar a los encargados de testing en sectores de Seguridad Informática de 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.

Se intentaran identificar y describir las falencias que existen en el proceso actual de testeo sobre las vulnerabilidades de hackeo, detallar métodos y herramientas que ayudaran en la detección de vulnerabilidades en el proceso de testeo, efectuar una prueba comparativa de concepto para demostrar la validez de su aplicación y enunciar las conclusiones sobre el tema y aportes a futuro.

IV. Metodología de desarrollo

Para construir el conocimiento asociado al presente proyecto de investigación, se seguirá un enfoque de investigación clásico [14, 4] con énfasis en la producción de tecnologías [16]; identificando métodos, materiales y abordaje metodológico necesarios para desarrollar el proyecto:

- Métodos:
  - Revisiones Sistemáticas:

Las revisiones sistemáticas [2] de artículos científicos siguen un método explícito para resumir la información sobre determinado tema o problema. Se diferencia de las revisiones narrativas en que provienen de una pregunta estructurada y de un protocolo previamente realizado.

  - Prototipado Evolutivo Experimental (Método de la Ingeniería):

El prototipado evolutivo experimental [3] consiste en desarrollar una solución inicial para un determinado problema, generando su refinamiento de manera evolutiva por prueba de aplicación de dicha solución a casos de estudio (problemáticas) de complejidad creciente. El proceso de refinamiento concluye al estabilizarse el prototipo en evolución.

- Materiales:

Para el desarrollo de los formalismos y procesos propuestos se utilizarán:

Formalismos de modelado conceptual usuales en la Ingeniería de Software [15, 9] y en la Ingeniería del Conocimiento [6].

Modelos de Proceso usuales en Ingeniería de Software [1, 8, 10].

Referencias

[1] ANSI/IEEE, "Draft IEEE Standard for software and system test documentation". ANSI/IEEE Std P829-2007, 2007.

[2] J. Argimón, "Métodos de Investigación Clínica y Epidemiológica". Elsevier España, S.A. ISBN 9788481747096, 2004.

[3] V. Basili, "The Experimental Paradigm in Software Engineering. En Experimental Software Engineering Issues: Critical Assessment and Future Directions". Lecture Notes in Computer Science, Vol. 706. ISBN 978-3-540-57092-9, 1993.

[4] J. Creswell, "Educational Research: Planning, Conducting, and Evaluating Quantitative and Qualitative Research". Prentice Hall. ISBN 10: 01-3613-550-1, 2002.

[5] B. Evans, "The Sorry State of Software". InformationWeek 112, 2001.

[6] R. García Martínez , P. Britos, "Ingeniería de Sistemas Expertos". Editorial Nueva Librería. ISBN 987-1104-15-4, 2004.

[7] ISO 27035:2011, "Information technology – Security techniques – Information security incident management". [Online]. http://www.iso.org/iso/catalogue_detail?csnumber=44379. Página Válida a 10/2016, 2010

[8] IEEE, "IEEE Standard for Developing Software Life Cycle Processes". IEEE Std 1074-1997 (Revision of IEEE Std 1074-1995; Replaces IEEE Std 1074.1-1995), 1997

[9] I. Jacobson, P. Ng, P. McMah, C. Jaramillo, "La esencia de la ingeniería de software: El núcleo de Semat". Revista Latinoamericana de Ingeniería de Software, 1(3), 71-78, 2013.

[10] H. Oktaba, F. Garcia, M. Piattini, F. Ruiz, F. Pino, C. Alquicira, "Software Process Improvement: The Competisoft Project". IEEE Computer, 40(10): 21-28. ISSN 0018-9162, 2007.

[11] "OWASP Top 10" - 2013. [Online]. http://www.owasp.org. Página Válida a 10/2016, 2013.

[12] C. Palmer. "Ethical hacking", IBM Systems Journal, Vol. 40, N°3, 2001.

[13] E. Raymond, "The New Hacker's Dictionary", MIT Press, Cambridge, MA, 1991.

[14] H. Riveros y L. Rosas, "El Método Científico Aplicado a las Ciencias Experimentales". Editorial Trillas. México. ISBN 96-8243-893-4, 1985.

[15] J. Rumbaugh, I. Jacobson, G. Booch, "The Unified Modeling Language, Reference Manual". Addison Wesley, ISBN-10: 02-0130-998-X, 1999

[16] J. Sabato, M. Mackenzie, "La Producción de Tecnología: Autónoma o Transnacional". Instituto Latinoamericano de Estudios Transnacionales - Technology & Engineering. ISBN 9789684293489, 1982.

[17] B. Schneier, "Secrets & Lies: Digital Security in a Networked World". John Wiley & Son, 2000.

[18] Sheoran, Pankaj & Singh, Sukhwinder, "Applications of Ethical Hacking". International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463 Vol. 3 Issue 5, May-2014, pp: (112-114), Impact Factor: 1.252, Available online at: www.erpublications.com Page | 112, 2014.

[19] C. Tori, "Hacking Ético" (1ra Ed). Buenos Aires: Mastroianni, 2008.

[20] C. Zimmerman, "Race to Deploy May Magnify Software Bugs". InternetWeek 13, 2001.

Ariel Giannone. Es Analista Programador en Desarrollo de aplicaciones y Licenciado en Informática por la Universidad Católica de Salta. Es Candidato del Programa de Magister en Ingeniería de Sistemas de Información de la Escuela de Postgrado de la Facultad Regional Buenos Aires de la Universidad Tecnológica Nacional. Es Investigador Tesista del Laboratorio de Investigación y Desarrollo en Espacios Virtuales de Trabajo (LIDEVT UNLa) del Departamento de Desarrollo Productivo y Tecnológico de la Universidad Nacional de Lanús. Argentina.