Métodos, Técnicas y Herramientas para la Protección de Sistemas de Software

José Pedro Montejano Massa, Mario Marcelo Berón, German Montejano, Daniel Riesco

Universidad Nacional de San Luis

Facultad de Ciencias Físico Matemáticas y Naturales

Área de Programación y Metodologías de Desarrollo de Software

mberon@unsl.edu.ar, {jose.p.montejano, german.a.montejano, riesco.daniel}@gmail.com

Resumen

Se presenta una línea de investigación y desarrollo que se basa en la seguridad informática y en la protección de los sistemas informáticos. La seguridad informática se ha vuelto cada vez más importante en los últimos años debido a la creciente cantidad de ataques cibernéticos y vulnerabilidades de seguridad en las aplicaciones de software.

Una de las principales amenazas a la seguridad de una aplicación de software es la explotación de vulnerabilidades de seguridad en el código fuente de la aplicación. Estas vulnerabilidades pueden ser explotadas por los atacantes para obtener acceso no autorizado al sistema, robar información confidencial, o incluso causar daños irreparables al sistema.

Es fundamental que los desarrolladores de software presten especial atención a la seguridad de sus aplicaciones y tomen medidas para proteger el código fuente de las mismas. Esto da lugar a una necesidad y es la que se pretende cubrir en esta línea de investigación. Se pretende encontrar soluciones informáticas que deriven en la implementación de técnicas y herramientas de protección de código para ser brindadas al desarrollador como un instrumento que puedan sumar en el desarrollo de sus proyectos de software para aumentar la seguridad de los mismos.

Palabras clave

Seguridad informática – Vulnerabilidad – Protección de código fuente.

Contexto

La línea de investigación descripta en este artículo se desarrolla en el Laboratorio de Calidad e Ingeniería de Software (LaCIS) de la Universidad Nacional de San Luis; y se encuentra enmarcada dentro del proyecto: "Ingeniería de Software: Estrategias de Desarrollo, Mantenimiento y Migración de Sistemas en la Nube", perteneciente a la misma. Dicho proyecto, es reconocido por el programa de incentivos, y es la continuación de diferentes proyectos de investigación de gran éxito a nivel nacional e internacional.

1. Introducción

El apogeo del software es, sin lugar a dudas, cada día más tangible y evidente. Es posible encontrar aplicaciones de software en: una casa, en la calle, en los campos, en plataformas marítimas y hasta en el espacio exterior. Es decir, el software no solo se encuentra inmerso en cada aspecto del planeta tierra, sino que es absolutamente necesario para el día a día.

El uso del software soluciona gran parte de los problemas de la vida cotidiana, partiendo de la base de que el usuario lleva siempre un celular, el cual se usa para cosas tan simples como chequear el clima, la hora, o también para cosas tan complejas como hacer transferencias bancarias. El uso del software soluciona necesidades a gran escala que son absolutamente necesarios para el correcto funcionamiento de la población, como sistemas de gestión de hospitales, sistemas bancarios, sistemas de aeronáuticas, sistemas embebidos en maquinaria industrial, etc.

Como es posible observar, el software es fundamental para el bienestar social con lo cual es altamente requerido por la comunidad. Por esta razón, en la actualidad, se ha producido una alta demanda de profesionales de la informática y asociado con ello aumentó masivamente la producción de software. Estos nuevos profesionales deben ser capaces de desarrollar software de calidad. Esta relevante característica, conduce a que haya un conocimiento absoluto del uso de la información que pasa por dicho software; entiéndase por información tanto a la interacción de un usuario con un juego de celular como a el almacenamiento y tratado de cuentas bancarias, información de salud personal, etc. Por ende, los sistemas desarrollados deben ser seguros.

Lograr desarrollar software seguro no es tan simple, dado el crisol de metodologías de desarrollo de software y estilos de programación diferentes que existen [1]. Muchas veces, un uso indebido de una metodología, o el poco soporte que proporciona para el desarrollo de software seguro, puede llevar a no garantizar la integridad y confidencialidad tanto de la información como de los datos que se utilizan en un sistema [2].

Por supuesto que ningún software es inviolable y no se puede garantizar la protección total de un sistema, pero si es fundamental ofrecer una base de protección del código fuente para prevenir un ataque indebido o un acceso no autorizado. Ya no es una tarea aislada desarrollar software y pensar en la seguridad del mismo, es decir estos dos conceptos van intrínsecamente conectados. Es tan relevante la aseveración antes mencionada que muchas empresas de software han adaptado sus metodologías de desarrollo de software para producir software seguro, como es el caso de Microsoft con SDL (Security Development Lifecycle) [3], Oracle con OSSA (Oracle Software Security Assurance) [4], entre otras.

A partir de lo antes mencionado, es importante recalcar que no debe lanzarse el despliegue de un sistema o módulo de software sin poder garantizar su seguridad.

En esta línea de investigación y desarrollo se está abordando la temática de la seguridad de los sistemas informáticos a través del desarrollo de herramientas y técnicas que permitan la protección del código fuente de un programa.

Se considera que la investigación en el campo de la seguridad informática es de vital importancia en la actualidad, y se espera que este trabajo pueda contribuir a mejorar la protección de los sistemas informáticos y prevenir los ataques cibernéticos y las vulnerabilidades de seguridad en las aplicaciones de software.

2. Líneas de investigación y desarrollo

Las vulnerabilidades en informática [5] son, hoy por hoy, un eje central a tener en cuenta en el desarrollo de cualquier proyecto de software. La norma ISO/IEC 27005:2008 [6] define una vulnerabilidad como “Una debilidad de un activo o grupo de activos que pueden ser explotadas por una o más

amenazas, donde un activo es cualquier cosa que tiene valor para la organización, sus operaciones comerciales y su continuidad, incluidos los recursos informáticos que respaldan la misión de la organización". En cualquier proyecto uno de los activos [7] principales es el código fuente del software en cuestión, esto da paso al eje central de esta línea de investigación que se basa en la protección de dicho código fuente.

Siguiendo con esto, se llega a la deducción de que sería de gran valor, para cualquiera de las partes interesadas en un proyecto de software, lograr proteger el código fuente [8]. Pero es necesario ahondar un poco más y ver cuáles serían posibles beneficios de lograr esto. La protección del código de un programa puede ofrecer varios beneficios, entre ellos:

- Protección de la propiedad intelectual: Al proteger el código fuente, se protege la propiedad intelectual del creador o propietario del programa, lo que puede impedir la copia o uso no autorizado del software [9]. Esto puede ser especialmente importante para los desarrolladores que buscan monetizar su trabajo a través de la venta de licencias de software [10].
- Seguridad: La protección del código fuente puede ayudar a proteger contra ataques malintencionados [11]. Si los detalles internos del código fuente son de conocimiento público, los atacantes pueden explorar el código en busca de vulnerabilidades y usar esta información para comprometer el sistema o el software en sí. Al proteger el código fuente, se hace más difícil para los atacantes identificar las debilidades y crear ataques [12].
- Ventaja competitiva: La protección del código fuente puede otorgar una ventaja competitiva a una empresa o desarrollador de software. Si el código fuente es inaccesible, otras empresas o desarrolladores no podrán estudiarlo para replicar la funcionalidad del software, lo que puede ayudar a mantener una ventaja en el mercado [13].

Ahora bien, se sabe que es imposible garantizar la seguridad de un programa al cien por ciento, entonces el objetivo es dificultar la tarea de un usuario que intenta robar o manipular código fuente lo máximo posible para que fracase en su intento. Y para esto se pueden utilizar de base innovadoras estrategias que ya han sido desarrolladas como Ofuscación de Código, Marca de Agua, Marca de Nacimiento, Protección de Software por Hardware, entre otras tantas aproximaciones orientadas a proteger los activos contenidos en los sistemas. Cada estrategia tiene su finalidad, por ejemplo, la ofuscación intenta confundir al atacante de forma tal que no pueda identificar parte del sistema que desea vulnerar [14]. La marca de agua es una técnica usada para prevenir los ataques a través de la inserción identificadores confiables que representan al propietario del sistema. Si un atacante modifica el software, el propietario puede extraer el identificador para verificar si es el software original o el mismo ha sido modificado por el atacante [15]. Otras técnicas como la marca de nacimiento se basan en encontrar propiedades innatas del software el cual cuando es modificado cambia la marca indefectiblemente [16]. Finalmente, las estrategias de protección basadas en hardware se basan en la elaboración de dispositivos electrónicos que son necesarios para que el sistema funcione [17].

Según esta línea de investigación, las herramientas que se han encontrado en la literatura implementan sofisticadas técnicas de protección, y la finalidad principal de ellas es la de probar el nivel de fortaleza de la estrategia en sí [18]. Mientras que el enfoque de esta línea de investigación se centra en encontrar una solución más general la cual permita aplicar primitivas de protección a

diferentes lenguajes y paradigmas de programación.

3. Resultados obtenidos/esperados

A través de los trabajos realizados por los integrantes de esta línea de investigación se han podido obtener diferentes resultados. Se desarrolló una aplicación web que permite proteger sistemas escritos en Java a través del uso de técnicas de protección de software que pueden ser desarrolladas por el mismo usuario de la herramienta según su necesidad particular.

Esta aproximación que permite la aplicación automatizada de técnicas de protección al código fuente de un sistema está dirigida a profesionales de la informática y cuenta con dos funcionalidades principales: la primera de ellas permite que las técnicas de protección sean subidas a la herramienta por los usuarios, mientras que la segunda posibilita la aplicación de estas técnicas al sistema que se desea proteger.

La aplicación es fácil de usar y permite proteger el código fuente de las aplicaciones Java de una manera simple y efectiva. También cuenta con algunas técnicas de protección ya implementadas que están listas para usar, las mismas fueron analizadas y escogidas para que sirvan de la mejor manera como ejemplo de caso práctico. Entre estas técnicas de protección implementadas se encuentra una técnica de marca de agua y una de señuelo.

Se espera desarrollar nuevas estrategias de protección que involucren la elaboración de métodos para detectar las partes vulnerables de un sistema, elaboración de estrategias de reingeniería que dado un sistema detecte la parte vulnerable y lo proteja. Estas soluciones pueden ser utilizadas para detectar patrones en los datos y comportamientos sospechosos en los sistemas, lo que permitiría una respuesta más rápida y eficiente a las amenazas de seguridad.

4. Formación de recursos humanos

Las tareas realizadas en el contexto de la presente línea de investigación están siendo desarrolladas como parte de un Trabajo Final Integrador para optar al grado de Ingeniería en Informática. En el futuro se piensa generar diferentes tesis de maestría y doctorado usando como base parte de esta investigación. Se busca, entre otras cosas, escalar la implementación de la herramienta para que no solo proteja sistemas escritos en Java, sino que también proteja sistemas escritos en varios lenguajes.

5. Bibliografía

[1] Grembi, J., & Chandra, P. (2015). Secure Software Development: A Security Programmer's Guide. Syngress.

[2] Waters, K. (2012). The Dark Side of Agile Software Development. Routledge.

[3] Blokdyk, G. (2021b). Microsoft Security Development Lifecycle a Complete Guide - 2019 Edition. 5STARCooks.

[4] Oracle Software Security Assurance. Oracle Corporation. (2023, 7 marzo). https://www.oracle.com/security/assurance/

[5] Thompson, H. H., & Chase, S. G. (2007). The Software Vulnerability Guide. Laxmi Publications Pvt Limited.

[6] ISO/IEC, "Information technology -- Security techniques-Information security risk management" ISO/IEC FIDIS 27005:2008.

[7] Holsing, N. F., & Yen, D. C. (1999). Software Asset Management. Information Resources Management Journal, 12(3), 14-26. https://doi.org/10.4018/irmj.1999070102

[8] J. Yang, C. Barrientes, J. Sanchez and Y. R. Kim, "Source Code Analysis for Secure Programming Practices," 2018 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2018, pp. 819-824, doi: 10.1109/CSCI46756.2018.00164.
[9] Publications, LandMark. (2019). Software Copyright (English Edition). LandMark Publications.
[10] Información legislativa. (2023, 7 marzo). servicios.infoleg.gob.ar. http://servicios.infoleg.gob.ar/infolegInternet/anexos/40000-44999/42755/texact.htm
[11] Ransome, J., & Misra, A. (2019). Cybersecurity for Software Engineers. Boca Raton, FL: CRC Press.
[12] Nagra, J. & Collberg, C. (2009b). Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection (English Edition) (1.a ed.). Addison-Wesley Professional.
[13] Michael Porter. (1985). Competitive Advantage: Creating and Sustaining Superior Performance. Free Press.
[14] Brunton, F. & Nissenbaum, H. (2016). Obfuscation: A User's Guide for Privacy and Protest. The MIT Press.
[15] Arnold, M., Wolthusen, S. D. & Schmucker, M. (2003). Techniques and Applications of Digital Watermarking and Content Protection (Ilustrado.). Artech House Publishers.
[16] Park, H., Lim, H. I., Choi, S. & Han, T. (2008). A Static Java Birthmark Based on Operand Stack Behaviors. 2008 International Conference on Information Security and Assurance (isa 2008). https://doi.org/10.1109/isa.2008.15
[17] Ju, H., Jeon, Y. & Kim, J. (2015). A Study on the Hardware-Based Security Solutions for Smart Devices. 2015 International Conference on Computational Science and Computational Intelligence (CSCI). https://doi.org/10.1109/csci.2015.105
[18] Wagner, D., & Jones, A. K. (2010). Software Protection and Simulation on Code Virtualization. Springer.