Análisis Estático de Seguridad en PHP mediante el Analizador de Código Estático Pixy

Adrian Hernández Yeja[1], Yosbany Tejas de la Cruz[2]
[1]Universidad de las Ciencias Informáticas, La Habana, Cuba, ayeja@uci.cu
[2]Universidad de las Ciencias Informáticas, La Habana, Cuba, ytejas@uci.cu

Web development environments become increasingly complex with a large volume of connected users, many of them considered "hackers", those that cause security problems in their interaction with the Internet, in most cases are due to errors application development, so that it becomes necessary to detect early stages thereof in software development. For the programming language PHP for the Web, there are tools for static code analysis, one of which is Pixy, which has an advanced search algorithm for the detection of vulnerabilities, however, this product stopped his development in 2007 without the possibility of continuing their advance, not providing support to existing elements of language. The presented paper provides an overview of the updates to the static code analyzer Pixy.

INTRODUCCIÓN

La seguridad de las aplicaciones Web es una cuestión de importancia en la actualidad. Existen diversos problemas a los que están sometidas las mismas. La mayoría de estos problemas se deben a errores en el desarrollo de las aplicaciones, por lo que se hace necesaria la detección de los mismos en fases tempranas del ciclo de desarrollo del software. Una de las técnicas que se utilizan con este fin es el análisis estático de código, que consiste en la revisión de las aplicaciones sin llegar a ejecutarlas. En este sentido, muchos autores comparten el criterio de que una de las mejores formas de detectar las debilidades de un software es analizando su código fuente, mediante la utilización de herramientas automáticas, con lo que se minimiza el posible efecto que puede ocasionar determinada vulnerabilidad. En el lenguaje de programación para la Web PHP, existe un analizador denominado Pixy, el cual detecta vulnerabilidades solamente de los tipos Cross-Site Scripting (XSS) e inyecciones SQL (SQLi). Esta herramienta de seguridad presenta algunos problemas como es el análisis de código PHP solamente para la versión 4, lo que provoca la imposibilidad de detectar múltiples amenazas en la escritura del código PHP por parte de los desarrolladores. Esta herramienta detuvo su desarrollo en el año 2007.

SEGURIDAD EN APLICACIONES WEB

Las aplicaciones Web han experimentado un acelerado crecimiento en los últimos años, dado, principalmente, a que se han convertido en el canal de comunicación fundamental entre clientes y proveedores de servicios; vinculado con este gran desarrollo, existe un impacto negativo en la seguridad de las aplicaciones, lo cual compromete la disponibilidad e integridad de la información con la cual se interactúa, lo que conduce a que se constaten enormes pérdidas materiales y a que se generen molestias en los usuarios en sentido general. Según OWASP (Open Web Application Security Project) en su artículo *"The Ten Most Critical Web Application Security Risks -2010"* (OWASP, 2010), los 2 problemas más serios, por orden de incidencia, hasta el año 2010 en las aplicaciones Web están enmarcados en: inyecciones de código y Cross-Site Scripting.

SEGURIDAD EN APLICACIONES PHP

La popularidad que ha adquirido PHP a nivel mundial y los crecientes niveles de ataques que se producen en la red de redes, constituyen aspectos de interés en el estudio de la seguridad de este lenguaje, en el que se observan características de seguridad muy especiales y es importante tenerlas en cuenta por la vitalidad que representa el tema. Las facilidades y gran flexibilidad en la escritura de código que brinda el lenguaje PHP a los desarrolladores, representa en ocasiones un problema. Ello es debido a que para los atacantes no es muy difícil detectar algunas vulnerabilidades de los programadores en sus aplicaciones, en tanto que conocen las fallas de seguridad más usuales y es ahí hacia donde dirigen sus ataques. La seguridad de PHP incluye la minimización de los errores de programación y la colocación del código apropiado en su lugar para eliminar toda posible vulnerabilidad. Los problemas más comunes de las aplicaciones PHP están enfocados en (Consortium PHP Security, 2010): Register Globals, XSS, SQLi y ejecución de código remoto.

ANÁLISIS ESTÁTICO DE SEGURIDAD

Es una técnica de detección de vulnerabilidades a través de la cual se revisa el código fuente de las aplicaciones sin llegar a ejecutarlas. El análisis se realiza principalmente en base a patrones o reglas, mediante el análisis del flujo de los datos o métricas para detectar problemas de seguridad en el desarrollo. Algunas ventajas del análisis estático radican en que se revisan todas las posibles entradas y salidas de una aplicación, se detectan problemas del código en su localización exacta, es rápido si se utilizan herramientas automatizadas, puede ser escaneado completamente el código base, permite encontrar errores en el desarrollo de las aplicaciones bien temprano en el ciclo de desarrollo del software, etc. En el análisis de código

estático existen tres técnicas fundamentales, las que determinarán las capacidades de detección de problemas en las aplicaciones Web, ellas son: (Crespo, 2007a): análisis de flujo de datos, análisis semántico y análisis estructural.

ANALIZADOR DE CÓDIGO ESTÁTICO PIXY

Pixy es una herramienta de seguridad desarrollada en Java para el análisis estático de vulnerabilidades en PHP 4. La misma basa su funcionamiento en tres características principales (Jovanovic, et al., 2006): sensibilidad al flujo, carácter interprocedural y sensibilidad al contexto. Cada vulnerabilidad detectada por Pixy genera un grafo de dependencia, en el cual se muestra el origen del problema detectado y con ello se brinda la comprensión o posible solución de la vulnerabilidad detectada al auditor de seguridad. El analizador clasifica las funciones de PHP en cinco categorías, ellas son: fuertes analizadoras, débiles sanitarizadoras, multidependencias, multidependencias inversas y malignas.

MODIFICACIONES REALIZADAS EN EL ANALIZADOR DE CÓDIGO ESTÁTICO PIXY

Los problemas de actualización de Pixy han provocado que esta herramienta de seguridad haya perdido popularidad entre los desarrolladores y auditores de seguridad de aplicaciones Web. A continuación se presentan algunas modificaciones fundamentales efectuadas en el analizador de código estático Pixy, donde se tomó como referencia por el equipo de desarrollo la versión 3.03 del mismo:

- Modificaciones en el parser de Pixy (PHPParser): Pixy realiza el análisis del código fuente que se le proporciona mediante la técnica de análisis de flujo de datos. El parser de la herramienta utiliza el método de análisis ascendente LALR (De Remer, et al., 1982). Pixy poseía algunas limitaciones en la gramática que analiza, las cuales están enfocadas principalmente en problemas con el reconocimiento de estructuras sintácticas de PHP 5 como es el soporte para la Programación Orientada a Objetos. Para que fuera posible el reconocimiento sintáctico de nuevas estructuras, se hizo preciso modificar el escáner y generador de gramática JFLEX (Klein, 2004) y JCUP (Hudson, 1999) respectivamente.
- Generación de reportes en el formato XML: Pixy mostraba el resultado del análisis que realizaba solamente en la consola de comandos. Esta característica del analizador representa un problema debido a que la comunicación con otros sistemas que interactúen con él se torna complejo. Teniendo en cuenta las posibilidades de expansión que brinda el analizador, se hizo posible la generación de los reportes vía archivos XML, por lo que se modificó el analizador en ese sentido para brindar la posibilidad de generación de reportes en ese formato.
- Análisis de vulnerabilidades del tipo inyección de código: El analizador de código estático Pixy solo analizaba vulnerabilidades de los tipos XSS y SQLi. Las aplicaciones Web y en especial las que se realizan en PHP son objeto de las amenazas provocadas por la inyección de código en sentido general. Un analizador de código estático para PHP requiere el análisis de este tipo de vulnerabilidad, de gran impacto en la actualidad. La flexibilidad que presenta Pixy en el procesamiento de "datos manchados" permitió la modelación de la inyección de código en la herramienta.
- Incorporación de nuevas funciones de repercusión en la seguridad de aplicaciones PHP en Pixy: La modelación de las funciones a las que da soporte Pixy es muy limitada debido a que no se tienen en cuenta elementos de seguridad de PHP 5. En total fueron incorporadas 135 funciones que expandieron la capacidad de detección de vulnerabilidades de Pixy.

CONCLUSIONES

El trabajo presentado refleja la incorporación de nuevas funcionalidades y características al analizador de código estático Pixy, las que permiten la ampliación del mismo para la detección de código vulnerable y reconocimiento sintáctico de elementos de PHP 5, en especial lo referente a la Programación Orientada a Objetos. En este sentido, se permitió el análisis de vulnerabilidades en sentencias propias de la versión 5 de PHP, las que no eran soportadas anteriormente en el analizador. De igual forma, se incorporó un nuevo módulo para la detección de vulnerabilidades por inyección de código maligno, así como la generación de reportes que permiten la interacción del analizador con otros sistemas que lo utilicen.

REFERENCIAS

Consortium PHP Security. (2010). "PHP Security Guide". http://phpsec.org/projects/guide, 02/17/2012.

Crespo, J. (2007). "El análisis de código, fuente de seguridad". Revista SIC, nº 74.

De Remer, F., Pennello, T. (1982). "Efficient computation of LALR (1) look-ahead sets", *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 4, no. 4, pp. 615–649, 1982.

Hudson S. E. CUP User's Manual. *Usability Center, Georgia Institute of Technology*, 1999.

Jovanovic N., Kruegel C., Kirda E. (2006). "Pixy: A Static Analysis Tool for Detecting Web Application Vulnerabilities (Short Paper)". http://www.seclab.tuwien.ac.at/papers/pixy.pdf, 02/17/2012.

Klein, G. Jflex user's manual. November, 2004.

OWASP. (2010). "The Ten Most Critical WEB Application Security Risks 2010". http://www.owasp.org/index.php/File:OWASP_T10_-_2010_rc1.pdf, 02/17/2012.