

Herramienta para el Análisis Automático de Parches

AAP v0.1b

- <https://github.com/gicsi/aap>

Antonio Castro Lechtaler^{*,1,2}, Julio César Liporace^{†,1}, Marcelo Cipriano^{*,1}, Edith García^{†,1}, Ariel
Maiorano^{†,1}, Eduardo Malvacio^{†,1}, Néstor Tapia^{†,1}
^{*}{acastro,marcelocipriano}@iese.edu.ar

[†]{edithxgarcia,jcliporace,maiorano,edumalvacio,tapianestor87}@gmail.com

¹Grupo de Investigación en Criptografía y Seguridad Informática
(GICSI) - Instituto Universitario del Ejército, Argentina

²FCE - Universidad de Buenos Aires, Argentina

CFR Criptored/RootedCON 2015

Presentación

- 1 **Análisis de código fuente**
 - Calidad del software
 - Revisiones generales de código fuente
 - Revisiones de seguridad de código fuente
- 2 **¿Por qué revisar parches?**
 - Código fuente de Parches
 - Parches: Mejoras, defectos y vulnerabilidades
 - Vulnerabilidades descubiertas a partir de parches
- 3 **Herramienta AAP versión 0.1 beta**
 - Presentación del proyecto
 - Funcionalidad actualmente implementada
 - Funcionalidad planeada y en desarrollo
 - Capturas de pantalla
- 4 **Referencias**
 - Referencias generales

Calidad del software

- Buena práctica: análisis y revisión de código fuente en busca de mejorar la calidad.
- Métrica de calidad: cantidad de *bugs* o defectos. * No único indicador, pero válido para el control y mejora de procesos.
- Objetivo: reducir el costo de calidad, encontrando y corrigiendo defectos de manera temprana y a un costo menor.

Revisiones generales de código fuente

- Estudios académicos (ingeniería del software) acerca del costo de la calidad del software, revisión y corrección de defectos:
 - Demostrado [9] que la calidad depende de los mecanismos de control de calidad empleados como parte integral de los procesos.
 - Máximo recomendado de 200 líneas de código por hora [9].
 - Tasa óptima de 125 líneas por hora [3].
 - "... *it is almost one of the laws of nature about inspections, i.e., the faster an inspection, the fewer defects removed*" [25].

Revisiones de seguridad de código fuente

- Revisiones en la búsqueda de vulnerabilidades y sistematización, según NIST y SEI/CERT:
 - Las herramientas de aseguramiento de calidad son actualmente un recurso fundamental para la mejora de las aplicaciones de software, según una publicación del NIST, donde ya se proponen especificaciones mínimas de funcionalidad para lo que sería un software analizador de código fuente en busca de vulnerabilidades [2].
 - Valerse únicamente de la implementación de políticas, estándares y buenas prácticas de desarrollo de software para el aseguramiento de la calidad sería inadecuado según un estudio del SEI/CERT [30].
- Disponibilidad de herramientas *open source* para la revisión/auditoría de código fuente; listados del SIE, NIST y OWASP [4, 19, 22].

Revisiones de seguridad de código fuente (cont.)

- Problemas en implementaciones de funcionalidad criptográfica:
 - Estudio publicado para la plataforma Android que indica que, de 269 vulnerabilidades reportadas desde enero de 2011 hasta mayo de 2014, sólo el 17% correspondió a defectos en librerías criptográficas, y el 83% restante a usos incorrectos de estas librerías por aplicaciones [14].
 - También en [8] se estudiaron aplicaciones Android; 10327 de un total de 11748 aplicaciones (88%) cometieron al menos un error en su implementación.
 - Este último trabajo presentó una herramienta para la detección automática de estos problemas, pero no se ha distribuido libremente [15].

Código fuente de Parches

- Parches o *patches*: actualizaciones de código fuente de software, típicamente en la forma de un detalle de diferencias en el código.
 - No todos los parches corrigen defectos.
 - No todos los defectos son vulnerabilidades.

Parches: Mejoras, defectos y vulnerabilidades

Mejoras, nuevas funcionalidades, ...

Nuevo código fuente. Prestaciones adicionales del software; nuevo módulos, opciones, ... Prioridad relativa baja o media en los procesos de "emparche" o *patching*.

Parches: Mejoras, defectos y vulnerabilidades

Mejoras, nuevas funcionalidades, ...

Nuevo código fuente. Prestaciones adicionales del software; nuevo módulos, opciones, ... Prioridad relativa baja o media en los procesos de "emparche" o *patching*.

Defectos

Correcciones específicas de código fuente. Errores en el software, defectos o *bugs* que podrían generar problemas al utilizarlo. Prioridad media.

Parches: Mejoras, defectos y vulnerabilidades

Mejoras, nuevas funcionalidades, ...

Nuevo código fuente. Prestaciones adicionales del software; nuevo módulos, opciones, ... Prioridad relativa baja o media en los procesos de "emparche" o *patching*.

Defectos

Correcciones específicas de código fuente. Errores en el software, defectos o *bugs* que podrían generar problemas al utilizarlo. Prioridad media.

Vulnerabilidades

Subconjunto de los defectos generales; aquellos que, de explotarse, implicarían un compromiso de seguridad. Prioridad máxima, crítica.

Vulnerabilidades descubiertas a partir de parches

- Errores en la clasificación de defectos como vulnerabilidades.
- Se publica un parche desconociendo al momento que el problema que corrige representaba una vulnerabilidad.

Vulnerabilidades descubiertas a partir de parches (cont.)

- *Hidden impact bugs*, o defectos de impacto oculto
 - Trabajos de Jason Wright
 - Trabajó en el *framework* criptográfico del sistema operativo OpenBSD [13]; publicó recientemente su tesis de maestría [34], que revisa y extiende trabajos anteriores [35, 36, 33].
 - Introduce el concepto de *hidden impact bugs*, o defectos de impacto oculto (ya en [1], con otro nombre): vulnerabilidades que fueron en primera instancia reportadas (y *patchheads*) como *bugs* o defectos sin impacto de seguridad.
 - Retraso de impacto, o *impact delay*: tiempo transcurrido desde la publicación del defecto -en la forma de parche-, y el momento en que se asignó un CVE al *bug*.
 - Estudios sobre el *Kernel* de Linux: de un total de 185 defectos de impacto oculto, 73 de ellos (el 39%) tuvieron un retraso de impacto de al menos 2 semanas.
 - Sobre MySQL: el total de defectos de impacto oculto fue de 29, y de ellos, 19 (65%) tuvo retraso de impacto de al menos 2 semanas.

Vulnerabilidades descubiertas a partir de parches (cont.)

- Problemas ya corregidos en la versión *devel* de NTP
 - Vulnerabilidades críticas descubiertas en diciembre de 2014.
 - Algunos de estos defectos ya habían sido corregidos en la versión de desarrollo o *devel*, años atrás [16].
 - De acuerdo a su *Bugzilla*, esta información estuvo públicamente accesible recién el día 20/12/2014.
 - En los mensajes correspondientes al *bug* 2665 [17] en relación a una llave criptográfica débil por defecto, se comenta que la vulnerabilidad ya había sido "corregida" en la versión de desarrollo anterior del proyecto, ntp-dev (4.2.7p11) del 2010/01/28.
 - Otro ejemplo: en el *bug* 2666 [18], relativo a al generador de números aleatorios, que aunque luego volvió a empatcharse antes del *release* de la versión 4.2.8, se comentó que en la versión de desarrollo 4.2.7p230 (del primero de noviembre de 2011) el problema ya había sido corregido.

Vulnerabilidades descubiertas a partir de parches (cont.)

- Más de un parche para vulnerabilidades *shellshock*
 - Descubiertas también recientemente, en el *shell* de sistemas UNIX bash.
 - El aviso público de Red Hat [26] y los paquetes actualizados, son del día 24 de septiembre de 2014.
 - Sin embargo, luego y durante el lapso de unos pocos días, otros problemas relacionados fueron descubiertos, para los cuales Red Hat no proveyó paquetes actualizados sino hasta el día 26 de septiembre de 2014 [27].

Vulnerabilidades descubiertas a partir de parches (cont.)

- Descubrimiento simultáneo de vulnerabilidad *Heartbleed*
 - Problema en librería de código abierto OpenSSL; Medio millón de sitios afectados indicó Bruce Schneier, calificando el problema como "catastrófico" [29].
 - Según un representante de Red Hat Security [6], la coincidencia de dos hallazgos del mismo problema, al mismo tiempo, incrementa el riesgo de mantener por mayor tiempo esta vulnerabilidad sin publicar los parches que para su corrección.
 - Dado el apuro, los encargados de la "divulgación responsable" no pudieron coordinar como hubiera sido óptimo la disponibilidad de parches y paquetes de actualización para todas las distribuciones de Linux.

Vulnerabilidades descubiertas a partir de parches (cont.)

- Vulnerabilidad en OpenBSD luego de parche mal categorizado
 - En el año 2007 fue publicada una vulnerabilidad descubierta en la implementación de IPv6 del *kernel* del sistema operativo OpenBSD [21].
 - La vulnerabilidad fue descubierta al analizar, e intentar reproducir, el problema corregido por un parche no categorizado como vulnerabilidad, sino como un *Reliability fix*, o parche de fiabilidad [20]..

Presentación del proyecto

La herramienta intentaría sistematizar y automatizar al menos los primeros pasos en la revisión de actualizaciones de código fuente para alertar, temprana y automáticamente, acerca de posibles vulnerabilidades, o acerca de código que debería revisarse especialmente.

Objetivo Se pretende que podría servir de ayuda a personas o a equipos que deban realizar estos análisis de forma repetida y continua, e intenta proveer un entorno que facilite el trabajo colaborativo entre los usuarios analistas.

- Aplicación Web, desarrollada en lenguaje Python, utilizando el *framework* Web Django.
- Depende de Git para el manejo del código fuente de los proyectos a analizar.

Funcionalidad actualmente implementada

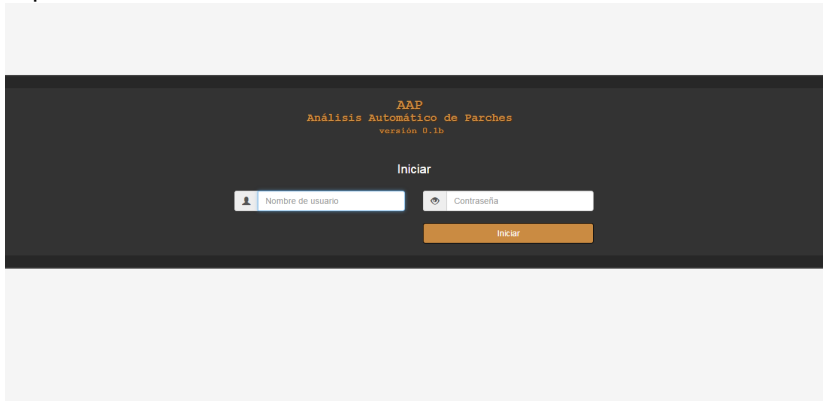
- Revisión y alerta, parametrizadas en la forma de plugins, de parches dentro de una misma rama o *branch*, o de diferencias de código fuente entre ellas, accediendo a sus repositorios Git:
 - Actualizaciones por parte de usuarios específicos.
 - Actualizaciones a archivos y/o directorios específicos.
 - Patrones en código fuente reemplazado y/o reemplazante.
 - Patrones en comentarios de código fuente.
 - Ejecución de herramienta externa Flawfinder [32] para el análisis estático de código fuente.
 - Identificación simple de código fuente que implementa criptografía.

Funcionalidad planeada y en desarrollo

- Actualmente en desarrollo:
 - Integración con técnicas de *text-mining* y *machine learning* para la clasificación de código fuente actualizado o diferente que deba generar alertas.
- Planes, trabajo a futuro, *TODOs*:
 - Mejoramiento de implementaciones actuales de reglas/pluings de análisis.
 - Desarrollo de nuevos plugins; integración con nuevas herramientas externas.
 - Integración con una bases de datos de reportes de vulnerabilidades o *advisories*.
 - Integración con herramientas de comunicación utilizadas por los desarrolladores (foros, Bugzilla, ...)
 - Continuación con la investigación y desarrollo para la aplicación de técnicas de inteligencia artificial.

Capturas de pantalla

Control de acceso para usuarios analistas, administradores y super-administradores



Capturas de pantalla

Ejemplo de detalle de datos de usuario del sistema

AAP
ver. 0.1b

ANÁLISIS DE PARCHES

- Alertas
- Categorías de proyectos
- Lenguajes de desarrollo
- Logs generales
- Notas de analistas
- Proyectos de software
- Ramas o branches
- Reglas configuradas
- Reglas de análisis
- Repositorios
- Sistemas operativos
- Usuarios analistas**

AUTENTICACIÓN Y AUTORIZACIÓN...

Modificar usuario analista

Administrador

Proyecto de software:

Proyecto demo Github

Proyecto de software asignado

Analista:

Administrador

Analista asignado al proyecto de software

Tarea:

Primera revisión y reporte.

Tarea/s asignada/s al analista

☒ Recibe emails

Este analista recibirá emails automáticos?

Historio

Grabar

- Grabar y añadir otro
- Grabar y continuar editando
- Eliminar

Capturas de pantalla

Ejemplo de detalles de un proyecto cargado junto a datos iniciales

The screenshot displays the AAP web interface. On the left is a dark sidebar with the AAP logo (ver. 0.1b) and a menu with categories like 'ANÁLISIS DE PARCHES', 'Alertas', 'Categorías de proyectos', 'Lenguajes de desarrollo', 'Logs generales', 'Notas de analistas', 'Proyectos de software' (highlighted), 'Ramias o branches', 'Reglas configuradas', 'Reglas de análisis', 'Repositorios', 'Sistemas operativos', 'Usuarios analistas', and 'AUTENTICACIÓN Y AUTORIZ...'. The main content area is titled 'Modificar proyecto de software' and shows the breadcrumb 'Análisis de parches / Proyectos de software / Proyecto demo Github'. It contains several form fields: 'Categoría de proyecto de software:' (dropdown with 'Otros'), 'Sistemas operativos:' (dropdown with 'Otros' selected), 'Lenguaje de programación:' (dropdown with 'Otros'), 'Nombre:' (text input with 'Proyecto demo Github'), and 'Descripción:' (text area with 'Proyecto "Hello-world" mantenido por Github para demos y pruebas.'). On the right, there's a sidebar with buttons: 'Historico', 'Grabar', 'Grabar y añadir otro', 'Grabar y continuar editando', and 'Eliminar'.

Capturas de pantalla

Ejemplo de detalles de un repositorio del proyecto

AAP
ver. 0.1b

ANÁLISIS DE PARCHES

Alertas

Categorías de proyectos

Lenguajes de desarrollo

Logs generales

Notas de analistas

Proyectos de software

Ramas o branches

Reglas configuradas

Reglas de análisis

Repositorios

Sistemas operativos

Usuarios analistas

AUTENTICACIÓN Y AUTORIZACIÓN

Modificar repositorio

Administrador

/ Análisis de parches / Repositorios / <https://github.com/octocat/Hello-World>

Proyecto de software:

Proyecto demo Github

Proyecto de software del repositorio

Git URL:

Actualmente: <https://github.com/octocat/Hello-World>

Cambiar:

URL del repositorio (https:// o file://)

Usuario:

Nombre de usuario para acceder a URL

Contraseña:

Contraseña para acceder a URL

Ruta local:

Ruta/path local donde será clonado el repositorio

☒ Activo

¿Es un repositorio activo?

Histórico

Grabar

+ Grabar y añadir otro

Grabar y continuar editando

Eliminar

Capturas de pantalla

Ejemplo de detalles de una rama o *branch* del repositorio

The screenshot displays the AAP web interface. On the left is a dark sidebar with the AAP logo (ver. 0.1b) and a menu with items like 'ANÁLISIS DE PARCHES', 'Alertas', 'Categorías de proyectos', 'Lenguajes de desarrollo', 'Logs generales', 'Notas de analistas', 'Proyectos de software', 'Ramas o branches' (highlighted), 'Reglas configuradas', 'Reglas de análisis', 'Repositorios', 'Sistemas operativos', 'Usuarios analistas', and 'AUTENTICACIÓN Y AUTORIZ...'. The main content area is titled 'Modificar rama o branch' and shows the details for the 'master' branch of the repository 'https://github.com/octocat/Hello-World'. It includes fields for 'Repositorio:', 'Rama:', and 'Observaciones:'. The 'Rama default' checkbox is checked. On the right, there is a 'Historico' button and a 'Grabar' section with options to 'Grabar y añadir otro', 'Grabar y continuar editando', and 'Eliminar'. The bottom of the interface shows 'Observaciones generales' and 'Hash de último commit:'.

Capturas de pantalla

Listado de reglas/plugins de análisis en datos iniciales

The screenshot displays the AAP web application interface. On the left is a dark sidebar with the AAP logo (ver. 0.1b) and a menu for 'ANÁLISIS DE PARCHES' containing items like Alertas, Categorías de proyectos, and Reglas de análisis (which is highlighted). The main content area has a header 'Escoja Regla de análisis a modificar' and a breadcrumb 'Análisis de parches / Reglas de análisis'. Below this is a search bar with the text 'Buscar "Reglas de análisis"', a dropdown menu, and a 'Ir' button. A status bar indicates 'seleccionados 0 de 8'. The central part of the interface is a table listing eight analysis rules, each with a checkbox and a right-pointing arrow icon. The rules are: 'Regla de análisis', 'DETECCIÓN DE ACTUALIZACIÓN POR PARTE DE USUARIO/S ESPECÍFICO/S', 'DETECCIÓN DE CÓDIGO FUENTE IMPLEMENTANDO CRİPTOGRAFÍA', 'DETECCIÓN DE PATRONES EN COMENTARIO DE CÓDIGO FUENTE', 'DETECCIÓN DE PATRONES EN CÓDIGO FUENTE REEMPLAZADO Y REEMPLAZANTE', 'DETECCIÓN DE PATRONES EN CÓDIGO FUENTE REEMPLAZANTE', 'DETECCIÓN DE PATRONES EN NOMBRES DE ARCHIVOS Y DIRECTORIOS', and 'EJECUCIÓN DE HERRAMIENTA FLAWFINDER'. At the bottom of the table, it says '8 Reglas de análisis' with an upward arrow icon.

AAP
ver. 0.1b

ANÁLISIS DE PARCHES

- Alertas
- Categorías de proyectos
- Lenguajes de desarrollo
- Logs generales
- Notas de analistas
- Proyectos de software
- Ramas o branches
- Reglas configuradas
- Reglas de análisis**
- Repositorios
- Sistemas operativos
- Usuarios analistas

AUTENTICACIÓN Y AUTORIZ...

Escoja Regla de análisis a modificar

Administrador

🏠 / Análisis de parches / Reglas de análisis

Buscar "Reglas de análisis" 🔍

seleccionados 0 de 8

<input type="checkbox"/>	Regla de análisis	➡
<input type="checkbox"/>	DETECCIÓN DE ACTUALIZACIÓN POR PARTE DE USUARIO/S ESPECÍFICO/S	➡
<input type="checkbox"/>	DETECCIÓN DE CÓDIGO FUENTE IMPLEMENTANDO CRİPTOGRAFÍA	➡
<input type="checkbox"/>	DETECCIÓN DE PATRONES EN COMENTARIO DE CÓDIGO FUENTE	➡
<input type="checkbox"/>	DETECCIÓN DE PATRONES EN CÓDIGO FUENTE REEMPLAZADO Y REEMPLAZANTE	➡
<input type="checkbox"/>	DETECCIÓN DE PATRONES EN CÓDIGO FUENTE REEMPLAZANTE	➡
<input type="checkbox"/>	DETECCIÓN DE PATRONES EN NOMBRES DE ARCHIVOS Y DIRECTORIOS	➡
<input type="checkbox"/>	DETECCIÓN DE PATRONES EN NOMBRES EN CÓDIGO FUENTE REEMPLAZADO	➡
<input type="checkbox"/>	EJECUCIÓN DE HERRAMIENTA FLAWFINDER	➡

8 Reglas de análisis

FIN DE PRESENTACIÓN

MUCHAS GRACIAS

- [1] J. Arnold, T. Abbott, W. Daher, G. Price, N. Elhage, G. Thomas, A. Kaseorg *Security Impact Ratings Considered Harmful*. Proceedings 12th Conference on Hot Topics in Operating Systems. USENIX. Mayo de 2009. [en línea: <http://www.inl.gov/technicalpublications/Documents/5588153.pdf> - accedido el 29/12/2014].
- [2] P. Black, M. Kass, M. Koo, M. Fong. *Source Code Security Analysis Tool Functional Specification Version 1.1*. NIST Special Publication 500-268 v1.1. Febrero de 2011. [en línea: http://samate.nist.gov/docs/source_code_security_analysis_spec_SP500-268_v1.1.pdf - accedido el 29/12/2014].
- [3] F. Buck. *Indicators of Quality Inspections*. IBM Technical Report TR21.802, Systems Comm. Diciembre de 1981.

- [4] CERT Division - Secure Coding *Secure Coding Tools*. CERT, Software Engineering Institute (SEI), Carnegie Mellon University. [en línea: <http://www.cert.org/secure-coding/tools/index.cfm> - accedido el 29/12/2014].
- [5] C. Corley, L. Etzkorn, N. Kraft, S. Lukins. *Recovering Traceability Links between Source Code and Fixed Bugs via Patch Analysis*. University of Alabama. 2008. [en línea: <http://www.cs.wm.edu/semeru/tefse2011/papers/p31-corley.pdf> - accedido el 29/12/2014].
- [6] M. Cox *Heartbleed*. Mark J. Cox Google+. [en línea: <https://plus.google.com/+MarkJCox/posts/TmCbp3BhJma> - accedido el 29/12/2014].
- [7] Django Framework. *Django overview*. Django Software Foundation. [en línea:

<https://www.djangoproject.com/start/overview/> -
accedido el 29/12/2014].

- [8] M. Egele, D. Brumley, Y. Fratantonio, C. Kruegel. *An empirical study of cryptographic misuse in android applications*. CCS '13 Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. Pages 73-84. 2013. EE.UU. [en línea: http://www.cs.ucsb.edu/~chris/research/doc/ccs13_cryptolint.pdf - accedido el 29/12/2014].
- [9] C. Kemerer, M. Paulk. *The Impact of Design and Code Reviews on Software Quality: An Empirical Study Based on PSP Data*. IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 35, NO. XX Abril de 2009. [en línea: http://www.pitt.edu/~ckemerer/PSP_Data.pdf - accedido el 29/12/2014].



- [10] Git SCM. *About Git*. Git - Software Freedom Cnonservancy.
[en línea: <http://git-scm.com/about> - accedido el 29/12/2014].
- [11] C. Guarnieri *One Flew Over the Cuckoo's Nest*. Hack In The Box 2012. Mayo de 2012. Holanda. [en línea: <http://sebug.net/paper/Meeting-Documents/hitbsecconf2012ams/D1T1%20-%20Claudio%20Guarnieri%20-%20ne%20Flew%20over%20the%20Cuckoos%20Nest.pdf> - accedido el 29/12/2014].
- [12] Gerrit Website *Gerrit Code Review*. Google Inc. [en línea: <https://code.google.com/p/gerrit/> - accedido el 29/12/2014].
- [13] A. Keromytis, J. Wright, T. de Raadt. *The Design of the OpenBSD Cryptographic Framework*. International Conference on Human System Interactions (HSI). Junio de 2012.

Australia. [en línea:

<http://www.thought.net/papers/ocf.pdf> - accedido el 29/12/2014].

- [14] D. Lazar, H. Chen, X. Wang, N. Zeldovich. *Why does cryptographic software fail?: a case study and open problems*. Proceedings of 5th Asia-Pacific Workshop on Systems Article No. 7. 2014. EE.UU. [en línea: <http://pdos.csail.mit.edu/papers/cryptobugs:apsys14.pdf> - accedido el 29/12/2014].
- [15] A. Mujic. *Reimplementation of CryptoLint tool*. Blog for and by my students. Diciembre de 2013. [en línea: <http://sgros-students.blogspot.com.ar/2013/12/reimplementation-of-cryptolint-tool.html> - accedido el 29/12/2014].

- [16] Network Time Protocol project. *NTP Security Notice*. NTP support website. Network Time Foundation. [en línea: <http://support.ntp.org/bin/view/Main/WebHome> - accedido el 29/12/2014].
- [17] Network Time Protocol project. *Bug 2665 - Weak default key*. NTP Bugzilla. Network Time Foundation. [en línea: http://bugs.ntp.org/show_bug.cgi?id=2665 - accedido el 29/12/2014].
- [18] Network Time Protocol project. *Bug 2666 - non-cryptographic random number generator with weak seed*. NTP Bugzilla. Network Time Foundation. [en línea: http://bugs.ntp.org/show_bug.cgi?id=2666 - accedido el 29/12/2014].

- [19] NIST *Source Code Security Analyzers*. SAMATE - NIST. [en línea: http://samate.nist.gov/index.php/Source_Code_Security_Analyzers.html - accedido el 29/12/2014].
- [20] A. Ortega *OpenBSD Remote Exploit*. Core Security. Julio de 2007. [en línea: <https://www.blackhat.com/presentations/bh-usa-07/Ortega/Whitepaper/bh-usa-07-ortega-WP.pdf> - accedido el 29/12/2014].
- [21] A. Ortega, G. Richarte. *OpenBSD Remote Exploit*. Core Security. Abril de 2007. [en línea: <https://www.blackhat.com/presentations/bh-usa-07/Ortega/Whitepaper/bh-usa-07-ortega-WP.pdf> - accedido el 29/12/2014].
- [22] OWASP Wiki. *Source Code Analysis Tools*. The Open Web Application Security Project (OWASP). Últ. mod. 29/10/2014.  

[en línea: https://www.owasp.org/index.php/Source_Code_Analysis_Tools - accedido el 29/12/2014].

- [23] Phabricator Website. *Phabricator, an open source, software engineering platform..* Phacility, Inc. [en línea: <http://phabricator.org/> - accedido el 29/12/2014].
- [24] Python Website. *About Python.* Python Software Foundation. [en línea: <https://www.python.org/about/> - accedido el 29/12/2014].
- [25] R. Radice. *High Quality Low Cost Software Inspections.* Paradoxicon Publishing. 2002.
- [26] Red Hat. Seguridad. Base de datos de CVE. *CVE-2014-6271.* Red Hat Customer portal. 24 de Septiembre de 2014. [en línea: <https://access.redhat.com/security/cve/CVE-2014-6271> - accedido el 29/12/2014].

- [27] Red Hat. Seguridad. Base de datos de CVE. *CVE-2014-7169*. Red Hat Customer portal. 24 de Septiembre de 2014. [en línea: <https://access.redhat.com/security/cve/CVE-2014-7169> - accedido el 29/12/2014].
- [28] The Register. J. Leyden. *Patch Bash NOW: 'Shellshock' bug blasts OS X, Linux systems wide open*. The Register online tech publication. 24 de Septiembre de 2014. [en línea: http://www.theregister.co.uk/2014/09/24/bash_shell_vuln/ - accedido el 29/12/2014].
- [29] B. Schneier. *Heartbleed*. Schneier on Security, Blog. Abril de 2014. [en línea: <https://www.schneier.com/blog/archives/2014/04/heartbleed.html> - accedido el 29/12/2014].

- [30] R. Seacord, W. Dormann, J. McCurley, P. Miller, R. Stoddard, D. Svoboda, J. Welch *Source Code Analysis Laboratory (SCALE)*. CERT, Software Engineering Institute (SEI), Carnegie Mellon University. Abril de 2012. [en línea: https://resources.sei.cmu.edu/asset_files/TechnicalNote/2012_004_001_15440.pdf - accedido el 29/12/2014].
- [31] W. Weimer. *Patches as Better Bug Reports*. University of Virginia. 2006. [en línea: <https://www.cs.virginia.edu/~weimer/p/p181-weimer.pdf> - accedido el 29/12/2014].
- [32] D. Wheeler. *Flawfinder*. David A. Wheeler's Personal Home Page - Flawfinder Home Page. [en línea: <http://www.dwheeler.com/flawfinder/> - accedido el 29/12/2014].

- [33] D. Wijayasekara, M. Manic, J. Wright, M. McQueen. *Mining Bug Databases for Unidentified Software Vulnerabilities*. Proceedings International Conference on Human System Interactions (HSI). Junio de 2012, Perth, Australia. [en línea: <http://www.inl.gov/technicalpublications/Documents/5588153.pdf> - accedido el 29/12/2014].
- [34] J. Wright. *Software Vulnerabilities: Lifespans, Metrics, And Case Study*. Master of Science Thesis. University of Idaho. Mayo de 2014. [en línea: <http://www.thought.net/papers/thesis.pdf> - accedido el 29/12/2014].
- [35] J. Wright, J. Larsen, M. McQueen. *Estimating Software Vulnerabilities: A Case Study Based on the Misclassification of Bugs in MySQL Server*. Proceedings International Conference of Availability, Reliability, and Security (ARES). Septiembre de

2013. pp. 72-81. Regensburg, Alemania. [en línea: <http://www.inl.gov/technicalpublications/Documents/5842499.pdf> - accedido el 29/12/2014].

- [36] J. Wright, M. McQueen, L. Wellman. *Analyses of Two End-User Software Vulnerability Exposure Metrics (Extended Version)*. Information Security Technical Report, 17(4), Elsevier. Abril de 2013. pp. 44-55. [en línea: <http://www.thought.net/papers/INL-JOU-12-27465-preprint.pdf> - accedido el 29/12/2014].