

Subsistema de Recolección de Elementos de Hardware Integrado a un Sistema de Gestión, Inventario y Monitoreo de Hardware con Alertas Automáticas (S.R.E.H.)

Silvia Edith Arias, Fabián Gibellini, Analía L. Ruhl, M. Alejandra Di Gionantonio, Nora V. Flores, Mónica M. Serna, Daniel F. Arch, Milagros Zea Cárdenas, Germán Parisi, Diego Barrionuevo.

Laboratorio de Sistemas / Dpto. de Ingeniería en Sistemas / Universidad Tecnológica Nacional / Facultad Regional Córdoba
Cruz Roja S/N, 5016

s_autn@hotmail.com, fgibellini@bbs.frc.utn.edu.ar, analialorenaruhl, ing.alejandrardg, ingnoraflores, sernamonicam@gmail.com, daniel.arch@pjn.gov.ar, milyzc, germannparisi, santosdiegob@gmail.com

Resumen

S.R.E.H. permite recolectar información de distintos elementos de hardware en una estación de trabajo. Trabaja tanto en sistemas operativos GNU/Linux como en Windows. Estos datos recolectados son base para un monitoreo de los equipos.

Esta herramienta se ha diseñado para interactuar con un sistema de Gestión, Inventariado y Monitoreo de Hardware con Alertas Automáticas, desarrollado durante el período 2014-2016 en el marco del Proyecto “Inventario Seguro en Ambiente Informáticos con Alertas Automáticas y Registro de historial para activos de TI” acreditado y financiado por la Secretaría de Ciencia y Técnica de Córdoba.

el Laboratorio de Ingeniería en Sistemas de Información de la Facultad Regional Córdoba de la Universidad Tecnológica Nacional (Argentina).

Palabras clave: Recolección, Alertas, Inventario, Hardware, Monitoreo, Historial, Gestión.

Contexto

El presente trabajo está inserto dentro del proyecto homologado por la Secretaría de Ciencia y Técnica de la UTN - FRC “Inventario Seguro en Ambiente Informáticos con Alertas Automáticas y Registro de historial para activos de TI” (I.S.A.I.) - Código: EIUTNCO0002226, radicado en el Laboratorio de Ingeniería en Sistemas de Información (LabSis) de la Universidad Tecnológica Nacional – Facultad Regional Córdoba (UTN - FRC), quien es a su vez es el organismo que contribuye a su financiamiento.

En el marco de la gestión de incidentes, el Laboratorio de Sistema (LabSis) tiene como principal objetivo ofrecer disponibilidad y un correcto funcionamiento de los equipos (PCs)

Actualmente cuando un alumno, docente o encargado reporta un funcionamiento no esperado, relacionado al hardware, se notifica a los integrantes del Área Técnica, quienes realizan pruebas sobre el equipo para detectar los componentes afectados. En caso de ser necesario, el equipo queda fuera de cualquier uso hasta su reparación. Es decir, que el personal del LabSis debe, dependiendo de la cantidad de máquinas afectadas, redistribuir los recursos o las aulas. Este proceso de

detección de inconvenientes relacionados al hardware es manual.

De lo antes planteado, derivó la necesidad de que el personal de LabSis conozca en cualquier momento: PCs habilitadas y el funcionamiento de sus respectivos componentes, para determinar con anterioridad a sus posibles usos, si surgió o no, alguna incidencia en las mismas a través de una Gestión, Inventariado y Monitoreo de Hardware que a través de alertas automáticas notifiquen ante cualquier comportamiento no esperado a los usuarios involucrados.

Introducción

La primera fase del proyecto I.S.A.I. consistió en el desarrollo y pruebas de un sistema de Gestión, Inventariado y Monitoreo de Hardware con Alertas Automáticas (G.I.M.H.A.A.) [1], cuya fuente de datos provenían desde un sistema de inventariado, llamado OCS Inventory [2]. Durante esta etapa se detectaron inconsistencias en los datos que reportaba el OCS Inventory, cuyo patrón no fue identificado. Debido a esto surgió la necesidad de identificar un procedimiento más confiable de obtención de datos de cada estación de trabajo.

Las inconsistencias encontradas se listan a continuación:

Duplicaba equipos. Durante las pruebas realizadas desde agosto hasta diciembre del 2016 surgió que en algunos casos el agente OCS (cliente instalado en las estaciones de trabajo a ser inventariadas) duplicaba las máquinas al momento de reportar su inventario al módulo OCS servidor, esto ocurrió en un aula del LabSis y en todos los casos bajo el sistema operativo Linux. En el resto de las aulas con equipos idénticos reportaba con normalidad.

En equipos con sistema operativo Windows el software reportaba en distintos tiempos, no seguía la configuración establecida.

No hay compatibilidad entre versiones. Cuando era necesario actualizar los agentes o el servidor a una versión más reciente no había compatibilidad con la versión anterior,

por lo que ante estas situaciones era obligatorio actualizar tanto el agente OCS de cada estación de trabajo como el módulo servidor que brinda servicios a los agentes OCS.

Los puntos mencionados indujeron a la decisión estratégica de desarrollar un subsistema Recolector de Elementos de Hardware (S.R.E.H.) que alimenta al subsistema G.I.M.H.A.A. La integración de ambos subsistemas conforman el sistema de nuestro proyecto I.S.A.I. Se realizó un análisis para determinar cuántos y cuáles serían los elementos de hardware a ser monitoreados, además de definir las características de cada elemento a ser recolectados.

Líneas de Investigación, Desarrollo e Innovación

Este proyecto se inscribe dentro de esta línea de investigación, enfocado en la gestión de activos de T.I. y que afecta a los ámbitos académico, gubernamental y empresarial. Favorece ampliamente al crecimiento de la gestión de activos de T.I. principalmente en las instituciones públicas del país, donde se sufren permanentemente sustracciones indetectables.

Al tratarse de un software con una licencia libre [3], todo aquel que desee implementar el sistema podrá acceder a la aplicación y su código, como así también adaptarlo para la estructura del ambiente informático sobre el cual lo desee trabajar.

Resultados y Objetivos

El objetivo de la fase II fue desarrollar el sistema S.R.E.H., que sea de libre uso y de fácil acceso para el LabSis y para cualquier organismo que lo requiera, haciendo hincapié en que este constituye al sistema I.S.A.I. brindando la base para dar soporte a una Gestión de Configuración y Activos TI del Servicio dentro de estas subredes, permitiendo monitorear, controlar, prevenir, proteger, notificar (a través del G.I.M.H.A.A) y tomar decisiones en tiempo y forma sobre

los equipos con los que cuenta el laboratorio y lograr una mejora continua de los servicios brindados a alumnos y docentes.

El sistema completo I.S.A.I. está conformado el subsistema G.I.M.H.A.A. y el subsistema S.R.E.H., este último fue desarrollado recientemente y ambos se encuentran actualmente en pruebas beta.

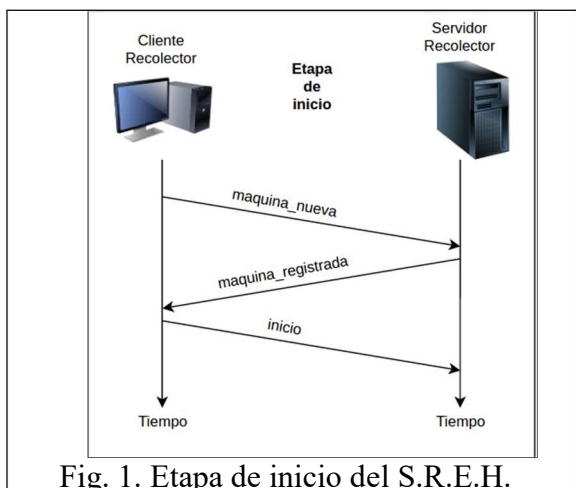


Fig. 1. Etapa de inicio del S.R.E.H.

Los elementos seleccionados a inventariar en cada estación de trabajo son:

- Almacenamiento: Disco duro (unidades físicas)
- Procesamiento: CPU (velocidad, caché, generación, fabricante, etc.).
- Memorias RAM: Memoria (slot que usa, tipo, fabricante, entre otros).

Por otro lado, también se obtienen datos del equipo, como nombre de la máquina, arquitectura (32 o 64 bits), nombre y versión del SO.

Estos datos y sus características, fueron seleccionados a partir de una priorización de componentes de hardware según las necesidades del LabSis en base a un estudio y análisis de factibilidad y confiabilidad de la obtención de los datos según cada sistema operativo.

El subsistema S.R.E.H utiliza una arquitectura cliente - servidor. El servidor se desarrolló en python 2.7 [4]. Para el desarrollo del cliente se tuvo en cuenta la heterogeneidad de sistemas operativos utilizados en las aulas del LabSis, por lo que se desarrollaron clientes tanto para sistemas

operativos Windows como para GNU/Linux, específicamente Debian [5].

Según el sistema operativo objetivo, los clientes utilizan distintas herramientas que permiten la obtención de datos de los componentes seleccionados, para el caso de Windows existe "Windows Management Instrumentation" (WMI) [6] que provee una interfaz para acceder a datos de administración del sistema y su red, lo que nos permite obtener datos del hardware y software entre otros. Esta información es simple de entender ya que se muestra de manera muy similar a un motor de base de datos relacional, pues se basa en SQL.

En el caso de GNU/Linux se usó inicialmente la herramienta dmidecode [7] con permisos de superusuario para los tres componentes (almacenamiento, procesador y memoria).

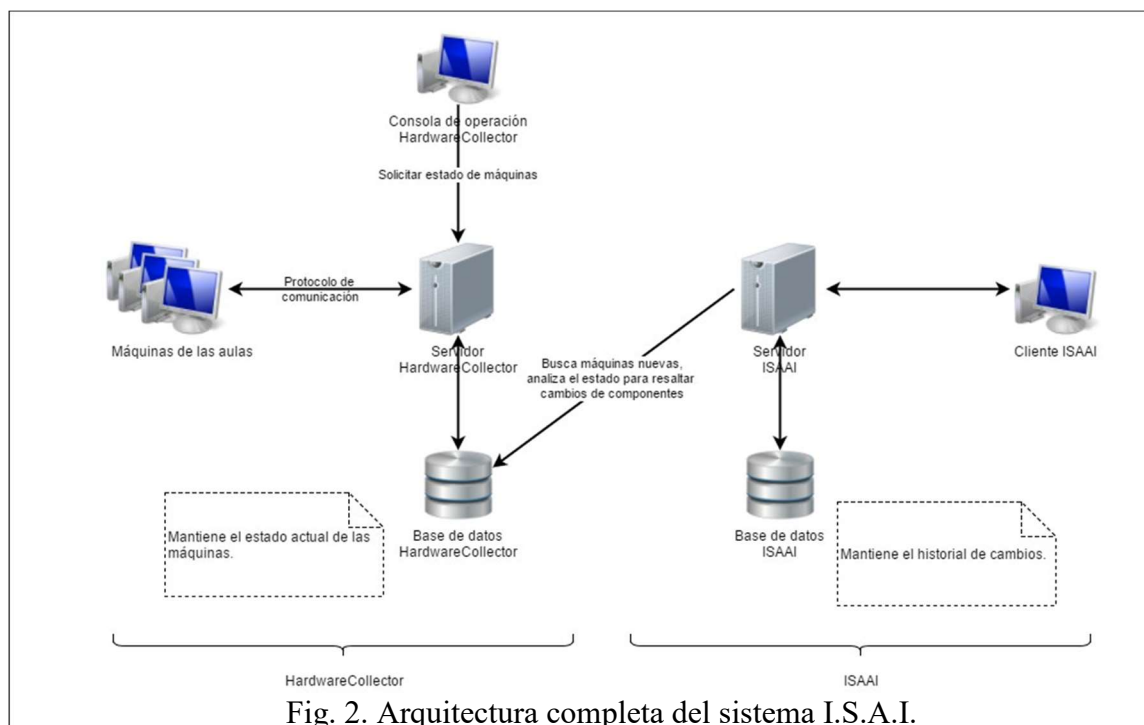
Actualmente, debido a la necesidad de minimizar el uso de permisos de usuario y otorgar al subsistema mayor flexibilidad, se usan distintos archivos y herramientas: módulo de python: platform [8], /proc/cpuinfo [9], /sys/devices/system/cpu/cpu/{id_procesador_fisico}/cpufreq/scaling_max_freq [10], dmidecode, lsblk [11], /sbin/udevadm, /sys/class/block/{nombre_dispositivo}/device /vendedor.

Para la comunicación entre cliente y servidor se ha creado un protocolo que se ejecuta sobre el protocolo TCP [12], utilizando sockets [13] como canal de comunicación.

El protocolo creado tiene tres etapas: La etapa de inicio, la cual se lleva a cabo cuando se instala un cliente en el equipo a ser inventariado, el servidor ejecuta un proceso que configura al cliente, la Fig. 1. muestra la dinámica de esta etapa. Esta configuración permite explicitar si se van a reportar los tres componentes o alguno en particular. Por último en la etapa de funcionamiento, existen dos modos de actuar, el modo pasivo y el activo. En el modo pasivo el cliente, según su configuración, reporta al servidor, y en el modo activo el servidor solicita un reporte, al cual el cliente responde.

Todos estos elementos constituyen el del subsistema Recolector para integrarlo al sistema I.S.A.I. logrando de esta manera el primer sistema autóctono de inventario

incorporarse a la carrera de investigador y un estudiante investigador de la carrera de Ingeniería en Sistemas de Información y becarios que forman parte del equipo.



automatizado con alertas automáticas ante un cambio en el hardware, permitiendo que el LabSis tenga conocimiento del estado de sus equipos en uso, cuando sea requerido y al menos una vez por cada uso de cualquier equipo. Las alertas son disparadas ante cualquier cambio de hardware en un equipo respecto a la última vez que este fue usado. A través de esto se logra visibilidad de los activos en uso en las aulas de LabSis.

La arquitectura final del sistema ISAAI se muestra en la Fig. 2.

Si bien esta segunda fase priorizo tres elementos de hardware, posteriormente se puede ampliar a otros componentes e inclusive se puede extender a un inventario del software de cada estación de trabajo.

Formación de Recursos Humanos

El grupo está compuesto por un Director, un Co-Director, dos profesores investigadores de apoyo, tres profesores aspirantes a

Este proyecto contribuirá a la formación y crecimiento de la carrera de investigador de los integrantes del mismo.

Además existe una colaboración interproyecto “Generación de modelos descriptivos para la prevención de incidentes en equipos informáticos en el contexto del Laboratorio de Sistemas (Fase II)” [14].

Referencias

- [1] “Gestión, Inventario y Monitoreo Hardware con Alertas Automáticas”. Workshop de investigadores en Ciencias de la Computación. Concordia. Entre Ríos. 2016.
- [2] OCS Inventory, <http://wiki.ocsinventoryng.org/index.php/Documentation:Administration> Última Visita: 18-08-2016.
- [3] Yingkui, Z., Jing, Z., Liye, W. “Justification of Free Software and its Enlightenment”. Econ. & Manage, Beijing Univ. of Chem. Technol., Beijing, China. Software Engineering (WCSE). Volume: 2. Page(s):171 – 173. ISBN: 978-1-4244-9287-9 DOI: 10.1109/WCSE.2010.47. IEEE. 19-20 Dec. 2010.

- [4] Python. Página oficial.
<https://www.python.org/>. Última visita: 13-03-2017.
- [5] Debian. Página oficial
<https://www.debian.org/intro/about>. Última visita 13-03-2017.
- [6] Windows Management Instrumentation
[https://msdn.microsoft.com/en-us/library/aa394582\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/aa394582(v=vs.85).aspx). Última visita: 13-03-2017
- [7] Dmidecode.
<https://linux.die.net/man/8/dmidecode>. Última visita: 13-03-2017
- [8] Plattform. Página oficial
<https://docs.python.org/3/library/platform.html>.
 Última visita 13-03-2017.
- [9] “The /proc/cpuinf file”.
http://www.linfo.org/proc_cpuinfo.html. Última visita 13-03-2017.
- [10] “CPU frequency and voltage scaling code in the Linux(TM) kernel”.
<https://www.kernel.org/doc/Documentation/cpu-freq/user-guide.txt>. Última visita: 13-03-2017.
- [11] lsbblk. <https://linux.die.net/man/8/lsblk>.
 Última visita: 13-03-2017
- [12] “Transmission Control Protocol”. RFC 793.
- [13] “The Definition of a Socket”. RFC 147. [14] “Generación de Modelo Descriptivo para la prevención de incidentes en equipos informáticos en el contexto del laboratorio de sistemas (Fase II)”. Corso, C., Maldonado, C., Gibellini, F., Ciceri, L., Martinez, G., Pereyra, F., Donnet, M. Departamento de Ingeniería en Sistemas Universidad Tecnológica Nacional, Facultad Regional Córdoba.