Sistema de administración y supervisión de equipos en una red de datos*[1]

[Management System and Monitoring Workstations in a Data Network]

Marco Emilio MONTES BOTERO[2] , Julio César GOMEZ CASTAÑO[3]

Recibo: 02.03.2011 - Ajuste: 20.05.2011 - Ajuste: 20.10.2011 - Aprobación: 10.12.2011

Resumen

El sistema de administración y supervisión del hardware y el software de los equipos en una red de datos, es de tipo cliente/servidor y permitirá, a los administradores de red, visualizar en un plano arquitectónico de la empresa, todos los equipos que componen la red, a través de una interfaz gráfica, y sobre ésta tendrán la posibilidad de consultar la información del inventario de hardware y software de cada uno de ellos, los cambios que se presenten en los mismos, y las funciones de supervisión. El desarrollo del proyecto está basado en la metodología ágil Scrum. Las herramientas de desarrollo e implementación utilizadas para la elaboración del sistema web corresponden a software libre; entre ellas Ruby on Rails, Rest, XML, Apache, Passenger y Netbeans. Se logró implementar el sistema de administración y supervisión, utilizando un sistema agente, con diferencias respecto a otras herramientas similares existentes en el mercado.

* Modelo para citación de este artículo de reflexión:
MONTES BOTERO, Marco Emilio y GÓMEZ CASTAÑO, Julio César (2011). Sistema de administración y supervisión de equipos en una red de datos. En: Ventana Informática. No. 25 (jul. – dic., 2011). Manizales (Colombia): Facultad de Ciencias e Ingeniería, Universidad de Manizales. p. 11-24. ISSN: 0123-9678

1 Artículo proveniente del proyecto titulado *Sistema SCADA para la administración y supervisión de equipos en una red de datos*, ejecutado entre octubre de 2009 y febrero de 2011, e inscrito en el grupo de investigación y desarrollo en Informática y Telecomunicaciones, para optar al título, en la Universidad de Manizales, de Ingeniero de Sistemas y Telecomunicaciones por parte del primer autor, bajo la dirección del segundo.

2 Ingeniero de Sistemas y Telecomunicaciones. Gerente, Siete Cumbres S.A.S, Manizales (Colombia). Correo electrónico: marcomontes@gmail.com.

3 Docente, Facultad de Ciencias e Ingeniería, Universidad de Manizales. Correo electrónico: jgomez@umanizales.edu.co.

El proyecto llegó a ser finalista en la cuarta convocatoria del programa nacional Destapa Futuro año 2009-2010 realizado por la Fundación Bavaria.

Palabras Clave: *Supervisión, Administración, Inventario, Alarma, OWASP, Scrum.*

Abstract

The management and monitoring system is client/server, and enable network administrators to view an architectural plan of the company, all computers in the network. On the graphical interface, they can check the information hardware and software inventory of each of these teams, the changes that occur in them, and monitoring functions. The project is based on the Scrum agile methodology and free tools for development and implementation, such as Ruby on Rails, Rest, XML, Apache, Passenger and Netbeans. It was possible to implement the management and monitoring system, using a broker. The article shows the differences of the system with other similar tools on the market. The project was presented to the fourth edition of the national Uncover Future 2009-2010, conducted by the Foundation Bavaria, which was a finalist.

Keywords*: Monitoring, Management, Inventory, Alarm, OWASP, Scrum.*

Introducción

La administración de los activos tecnológicos de una empresa, en muchos casos está mal enfocada o hace uso de medios y metodologías inapropiadas para su buen desempeño. Juega un papel importante el manejo eficiente y seguro de la información relacionada con los equipos de cómputo disponibles en una empresa, ya que se hacen inventarios y revisiones manuales sólo en caso de ocurrir algún incidente en estos equipos.

Este proyecto tiene como sustento la sistematización de estos procesos, mediante actividades de administración y supervisión, como son el control de supervisión, automatización y adquisición de datos, con el fin de agilizar y garantizar la seguridad en la administración de los activos tecnológicos de una empresa, generando innovación y posibilidades de empleo en campos de estudio específicos de la informática como lo son el desarrollo de software, la seguridad informática y el supervisión de equipos.

Además, el uso de la metodología ágil *Scrum*, se adapta perfectamente al desarrollo de un proyecto de este tipo, en el cual los requerimientos surgen o cambian en cualquier momento, permitiendo un alto nivel de adaptabilidad a las circunstancias cambiantes de la evolución del proceso.

Finalmente, el ciclo de vida del desarrollo del proyecto, adaptado a las metodologías ágiles de desarrollo, recomendaciones de desarrollo más seguro y el seguimiento de estándares y protocolos para intercambio de datos entre aplicaciones, permitirá una perfecta sinergia de herramientas de software nuevas o existentes que requieran el acceso a consultas y/o manipulación de la información generada por este proyecto.

1. Fundamento teórico

1.1 Windows Management Instrumentation

Según AJPDSoft (2010), *Windows Management Instrumentation* (WMI o Instrumental de administración de Windows) es una iniciativa que pretende establecer normas estándar para tener acceso y compartir la información de administración a través de la red de una empresa. WMI incluye un repositorio de objetos, a modo de base de datos de definiciones de objetos, y el administrador de objetos CIM, que controla la recopilación y manipulación de objetos en el repositorio y reúne información de los proveedores de WMI. Los proveedores de WMI actúan como intermediarios entre los componentes del sistema operativo, las aplicaciones y otros sistemas.

Los proveedores proporcionan información acerca de sus componentes, y podrían proporcionar métodos para manipular los componentes, las propiedades que se pueden establecer, o los sucesos que le pueden alertar de las modificaciones efectuadas en los componentes.

Por ejemplo, con WMI se podría obtener información detallada del procesador y de los diferentes componentes de hardware y software del equipo.

1.2 Representational State Transfer, REST

REST es un estilo de arquitectura de software para sistemas hipermedias distribuidos tales como la Web y se refiere estrictamente a una colección de principios para el diseño de arquitecturas en red. Estos principios resumen cómo los recursos son definidos y direccionados. Está basado en estándares HTTP, URL, representación de recursos XML/HTML/GIF/JPEG y tipos MIME text/xml, text/html. Su funcionamiento en la identificación de recursos y manipulación

de ellos a través de representaciones, mensajes autodescriptivos y mecanismos de reconocimiento del estado de la aplicación, como lo expresa Navarro (2007).

1.3 Open Web Application Security Project, OWASP

Consiste en un proyecto de código abierto dedicado a determinar y combatir las causas que hacen inseguro al software. La misión de OWASP (2011) «*es hacer la seguridad en aplicaciones "visible", de manera que las organizaciones pueden hacer decisiones informadas sobre los riesgos en la seguridad de aplicaciones*». La Fundación OWASP, un organismo sin ánimo de lucro que apoya y gestiona proyectos e infraestructura, está formada por empresas, organizaciones educativas y particulares de todo mundo. Juntos constituyen una comunidad de seguridad informática que trabaja para crear artículos, metodologías, documentación, herramientas y tecnologías que se liberan y pueden ser usadas gratuitamente por cualquiera.

1.3.1 OWASP Top 10. Proyecto de la Fundación OWASP que pretende recopilar los diez tipos de vulnerabilidades en aplicaciones web que suponen el mayor riesgo para su seguridad. Según OWASP (2010a) las diez vulnerabilidades más críticas que se han recogido en la versión del año 2010, son las siguientes:

- Inyecciones: Vulnerabilidades de inyección de código, desde SQL hasta comandos del sistema.
- *Cross-site Scripting*: Una de las vulnerabilidades más extendidas y a la par subestimadas, que abarca cualquier ataque que permitiera ejecutar código *scripting*.
- Gestión defectuosa de sesiones y autenticación: Comprende los errores y fallos en las funciones de administración de sesiones y validación de usuario.
- Referencias directas a objetos inseguras: Errores al exponer partes privadas o internas de una aplicación sin control y accesibles públicamente.
- *Cross-site Request Forgery*: Vulnerabilidad consistente en el desencadenamiento de acciones legítimas por parte de un usuario autenticado, de manera inadvertida por este último y bajo el control de un atacante.
- Configuración de seguridad mala o ausente: Más que un error en el código, se trata de la falta o mala configuración de seguridad de todo el conjunto de elementos que comprende el despliegue de una aplicación web, desde la misma aplicación hasta la configuración del sistema operativo o el servidor web.

- Almacenamiento con cifrado inseguro: Referida a la ausencia o mal uso de los sistemas de cifrado en relación a los datos almacenados o manejados por la aplicación.
- Falta de restricciones en accesos por URL: Falta de validación en el procesamiento de URL que podrían ser usadas para invocar recursos sin los derechos apropiados o páginas ocultas.
- Protección insuficiente de la capa de transporte: Relacionada con la vulnerabilidad penúltima, pero orientada a la protección del tráfico de red, indica una elección de un cifrado débil o mala gestión de certificados.
- Datos de redirecciones y destinos no validados: Errores en el tratamiento de redirecciones y uso de datos no confiables como destino.

1.4 Scrum

Según Infante (2010), es una metodología ágil, que puede ser usada para manejar el desarrollo de productos complejos de software usando prácticas iterativas e incrementales. Ha sido usado desde proyectos simples hasta en cambios estructurales completos en las empresas para sus negocios. Con él se incrementa significativamente la productividad y reduce el tiempo de espera, para ver los beneficios así como facilitar la adaptación de los sistemas desarrollados. El mencionado autor considera estas características:

- Procesos ágiles para el manejo y control del trabajo de desarrollo.
- Es un contenedor de prácticas de ingeniería existentes.
- Es un enfoque basado en equipos, incrementa el desarrollo cuando los requerimientos cambian rápidamente.
- Es un proceso que controla el caos entre los conflictos de interés y las necesidades.
- Es un camino para mejorar las comunicaciones y maximizar la cooperación.
- Es la vía para detectar la causa y solucionar cualquier problema en el desarrollo.
- Es escalable desde proyectos simples a proyectos completos organizacionales.

Asimismo, expresa que esta metodología consiste en un conjunto de prácticas interrelacionadas y reglas que optimizan el entorno de desarrollo, reducen la sobrecarga organizativa y sincronizan los requisitos del mercado con los prototipos de cada iteración. Se basa en el trabajo en equipo, en reuniones diarias presididas por el máster para establecer el estado del proyecto y en la salida cada 30 días de las características del proyecto finalizadas y listas para trabajar. El corazón

de esta metodología es la iteración; en cada iteración se presenta una mejora del funcionamiento del producto final y se evalúa la tecnología y capacidades requeridas, además, diariamente se puede modificar el enfoque si se encuentran nuevas dificultades.

1.5 Antecedentes

1.5.1 Aranda Asset Management. Herramienta de gestión y soporte que permite tener información actualizada y detallada sobre el estado de la infraestructura tecnológica de forma automatizada, cuyas características, según Aranda (2010), son:

- Inventario actualizado y detallado de hardware, software, periféricos, agendas de bolsillo (PDAs) y activos fijos minimizando costos de inventarios manuales.
- Control de licenciamiento de software en cada equipo y del utilizado por cada usuario.
- Control especial de eventos mediante la generación de alarmas que indican cualquier cambio en el inventario.
- Control y administración remota a estaciones inventariadas y no inventariadas en tiempo real para minimizar los tiempos de soporte.
- Ejecución en línea de tareas de administración remota a estaciones inventariadas y no inventariadas.
- Módulo de reportes que permite unificar y presentar toda la información recogida.
- *PC Web*, alternativa móvil para la visualización de la información de una estación de trabajo en particular y Módulo *PC Browser Web edition*, que permite su administración y control remoto.
- Sincronización bidireccional permanente con el Directorio Activo de Windows.

1.5.2 Dexon Control de Plataforma IT. De acuerdo con Dexon (2010), permite un sistema totalmente integrado para cubrir todas las necesidades de administración y supervisión de una plataforma de la infraestructura tecnológica. Diseñada para asegurar el control, seguimiento y auditoria de todos los componentes tecnológicos que conforman la plataforma IT de cualquier organización. Brindando prácticas de administración basadas en la metodología ITIL (Biblioteca de Infraestructura de Tecnologías de la Información). Ofreciendo servicios sobre Inventarios automáticos, administración de activos fijos, control de licenciamiento de software, auditoría y control remoto, recuperación y respaldo de archivos críticos. Además, por tratarse de un sistema flexible, totalmente Integrado y que dinámicamente se ajusta a las necesidades del negocio, permite la administración y supervisión de Servidores, Impresoras, Dis-

positivos de Red, Recursos de red, Servicios y SNMP. La *suite* Dexon brinda soluciones específicas para:

- Administración de computadoras de escritorio*,*
- Supervisión de redes y servidores,
- Administración de dispositivos de red,
- Distribución e Instalación desatendida de software,
- Restauración y copias de seguridad e imágenes de disco de estaciones de trabajo,
- Control remoto y virtualización del mantenimiento remoto,
- Administración de activos fijos.

Otros sistemas para la gestión de inventario de los recursos de los equipos utilizados son *Lever IT Discovery* (LeverIT, 2010) y *Excalibur* (CMN-Consulting, 2010).

2. Metodología

El trabajo estuvo basado en la metodología ágil de desarrollo *Scrum* (Control Chaos, 2009), que no se basa en el seguimiento de unas fases específicas, sino en la adaptación continua a las circunstancias de la evolución del proyecto, permitiendo de este modo, un desarrollo de carácter adaptable en lugar de predictivo y una estructura de desarrollo ágil e incremental basada en iteraciones y revisiones. Además, se siguieron las recomendaciones de las guías de desarrollo y pruebas propuestas por OWASP (2008, 2009, 2010b).

El producto finalizado fue usado como prototipo de pruebas en la red de datos de la Universidad de Manizales, llevando a cabo un pilotaje específico sobre un segmento de red compuesto por un número reducido de equipos de cómputo bajo la plataforma Windows XP, en tres salas de cómputo de la Facultad de Ciencias e Ingeniería.

El proyecto se realizó en tres fases, siendo la segunda, un ciclo iterativo que se adaptó al modelo de desarrollo ágil, así:

2.1.1 Adecuación de entorno de trabajo. Comprendió la Instalación del entorno de desarrollo, con tres actividades:

- Acondicionamiento de un equipo de cómputo (con las especificaciones de software: Sistema operativo *Microsoft Windows XP*, Entorno de desarrollo integrado Microsoft Visual Studio versión 2008 con soporte para el lenguaje Visual C#, Entorno de desarrollo integrado *Netbeans* versión 6 con soporte para el lenguaje *Ruby*, Distribución binaria compilada del lenguaje de programación *Ruby* para la plata-

forma Windows, *Framework Rails*, Servidor de aplicaciones *Mongrel*, Servidor de base de datos MySQL versión 5, Software para control de versiones *Subversion* y Software para virtualización *VMware Server*).
- Instalación del entorno de pruebas, donde se consideraron un servidor (especificaciones de software para el equipo servidor: Sistema operativo *Ubuntu Server Edition* versión 10.04, Software para control de versiones *Subversion*, Sistema web para la gestión de proyectos y seguimiento de errores Trac, Distribución binaria compilada del lenguaje de programación *Ruby* para la plataforma Linux, *Framework Rails*, Servidor de aplicaciones *Mongrel*, Servidor de base de datos *MySQL* versión 5 y *.NET Framework* versión 3), y varios clientes virtualizados (con Sistemas operativos Windows XP, Windows Vista, Windows Server 2003 y Windows Server 2008 en sus diferentes versiones de *Service Pack*).
- Instalación del entorno de producción, en un servidor virtualizado bajo el entorno de desarrollo (Sistema operativo *Ubuntu Server Edition* versión 10.04, Distribución binaria compilada del lenguaje de programación *Ruby* para la plataforma Linux, *Framework Rails*, Servidor Web Apache + Módulo *Passenger* y Servidor de base de datos *MySQL* versión 5).

2.1.2 Desarrollo Ágil. Patrón de ciclo de vida del modelo de desarrollo ágil, el cual comprendió las siguientes actividades:

- Concepto, o creación de la visión del proyecto y conocimiento del alcance del mismo.
- Especulación, que se repite en cada iteración del desarrollo y teniendo como referencia la anterior actividad y consiste en Desarrollo / revisión de los requisitos generales del proyecto (priorizados), Desarrollo de una lista con las funcionalidades esperadas del proyecto, Construcción de un plan de entrega: Fechas en las que se entregarán las versiones, hitos e iteraciones del desarrollo.
- Exploración, que consiste en el desarrollo de las funcionalidades que, para generar el siguiente incremento de producto, se han determinado en la actividad anterior.
- Revisión de las funcionalidades construidas hasta el momento y puesta en marcha en el entorno de producción para determinar su alineación y dirección con los objetivos.
- Cierre, que implica la fecha de entrega de una versión del proyecto con el producto esperado.

2.1.3 Transición. La puesta en marcha del prototipo de pruebas en la red de datos de la Universidad de Manizales, considerando:

- Documentación de Usuario, que comprende la documentación técnica requerida para el despliegue e instalación de componentes de hardware y software en la red de datos,
- Puesta en marcha, con la instalación y ejecución del hardware y software requerido y creado.

3. Resultados y discusión

3.1 Descripción de resultados

Se logró crear un agente para las plataformas Windows que obtiene automáticamente el inventario de hardware y software en los equipos clientes, adquiriendo los parámetros de supervisión a través de un servicio web y por medio del mismo reporta los cambios encontrados en dicho inventario. En la figura 1 (parte superior) se muestran los datos obtenidos para los dispositivos (*Board*, *bios*, arquitectura, RAM, etc.) que están siendo monitoreados en un equipo de prueba.

Se creó una vista que permite la manipulación y visualización en el plano arquitectónico de los equipos registrados, con su ubicación, estado de alerta, funciones para mostrar y ocultar dispositivos, controles de zoom y movimiento del plano, registro e información de equipos. En la figura 1 (parte inferior) se muestra el plano arquitectónico del segundo piso de la Universidad de Manizales, con la ubicación de los dispositivos registrados, su estado y los controles de manipulación.

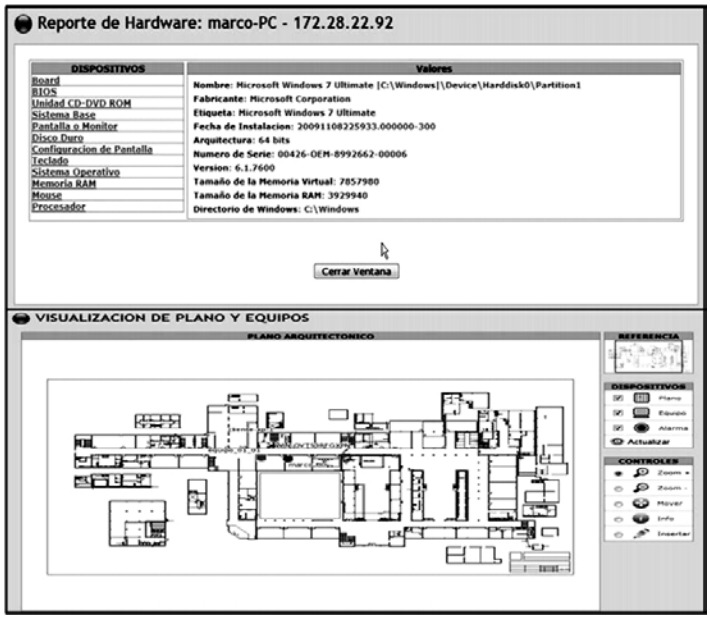

Figura 1. Datos y distribución de los equipos monitoreados

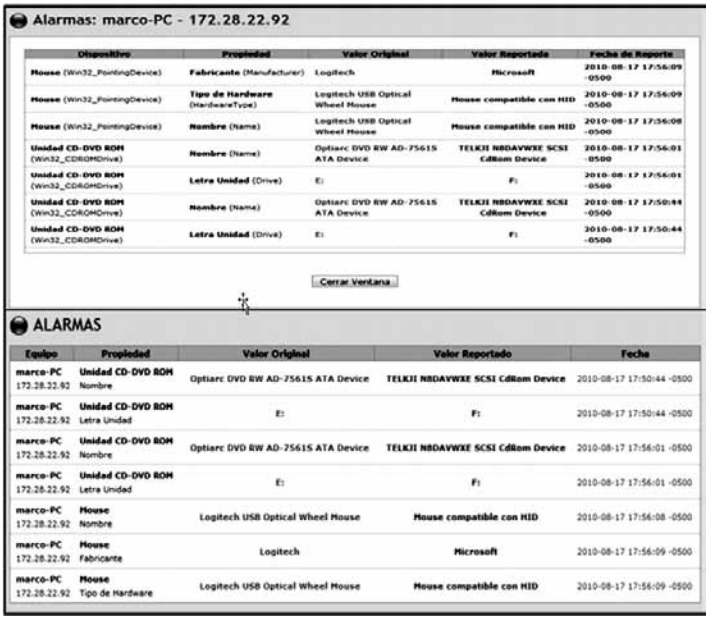

Figura 2. Alarmas por equipo y listado general de alarmas

Además, se creó un módulo de alarmas que reporta, de manera general y por cada equipo, la información de las propiedades no equivalentes, donde aparece el valor original y el valor repostado con su respectiva fecha y hora de reporte, como se observa en la figura 2, donde hay alarmas por los dispositivos *mouse* y Unidad de CD-DVD, así como el listado general de alarmas. De igual manera, se logró la integración de algunas herramientas adicionales que permiten extender las funcionalidades propias del sistema. En la figura 3 se muestra la utilidad *ping* ejecutada sobre uno de los equipos registrados y con el respectivo despliegue de resultados.

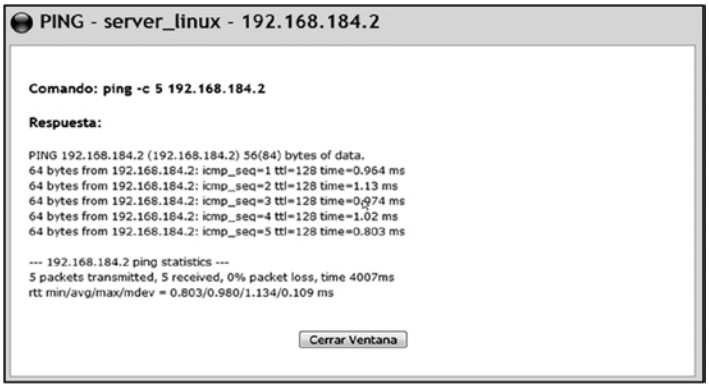

Figura 3. Repuesta enviada por la herramienta ping ejecutada sobre un equipo

Otro aspecto conseguido es el servicio web que permite el intercambio de datos entre el agente instalado en los equipos cliente y el servidor. En la figura 4 se muestra una parte de la estructura XML entregada por el servicio web al agente instalado en los equipos cliente y que contiene la configuración de los parámetros de supervisión.

3.2 Discusión de resultados

El sistema de administración y supervisión del hardware y el software de los equipos que conforman una red de datos, utiliza un sistema agente instalado en los equipos cliente, un sistema web que permite la visualización y manipulación de la información reportada y un servicio web que permite la comunicación e intercambio de datos entre los clientes y el servidor.

Las herramientas de desarrollo e implementación utilizadas para la elaboración del sistema web y el servicio web son todas de libre uso y se usan sobre el sistema operativo Linux; entre ellas *Ruby on Rails*, *Rest*, XML, Apache, *Passenger*, y *Netbeans*.

El sistema agente que se ejecuta en los equipos cliente fue desarrollado con tecnologías .NET, que permite un mejor acoplamiento con la plataforma Windows y su instrumental de administración. En la figura 5 se muestra el software agente instalado como un servicio en el sistema operativo Windows.

```
- <clients type="array">
  - <client>
      <id type="integer">391</id>
      <wmi-class-id type="integer">2</wmi-class-id>
      <wmi-class-name>Win32_BaseBoard</wmi-class-name>
      <wmi-namespace-id type="integer">1</wmi-namespace-id>
      <wmi-namespace-name>root\CIMV2</wmi-namespace-name>
      <wmi-property-id type="integer">1</wmi-property-id>
      <wmi-property-name>Caption</wmi-property-name>
    </client>
  - <client>
      <id type="integer">392</id>
      <wmi-class-id type="integer">2</wmi-class-id>
      <wmi-class-name>Win32_BaseBoard</wmi-class-name>
      <wmi-namespace-id type="integer">1</wmi-namespace-id>
      <wmi-namespace-name>root\CIMV2</wmi-namespace-name>
      <wmi-property-id type="integer">2</wmi-property-id>
      <wmi-property-name>Manufacturer</wmi-property-name>
    </client>
  - <client>
      <id type="integer">393</id>
      <wmi-class-id type="integer">2</wmi-class-id>
      <wmi-class-name>Win32_BaseBoard</wmi-class-name>
      <wmi-namespace-id type="integer">1</wmi-namespace-id>
      <wmi-namespace-name>root\CIMV2</wmi-namespace-name>
      <wmi-property-id type="integer">3</wmi-property-id>
      <wmi-property-name>Product</wmi-property-name>
    </client>
  - <client>
      <id type="integer">394</id>
      <wmi-class-id type="integer">3</wmi-class-id>
      <wmi-class-name>Win32_BIOS</wmi-class-name>
      <wmi-namespace-id type="integer">1</wmi-namespace-id>
```


Figura 4. Documento XML obtenido por el servicio web.

Figura 5. Agente de supervisión como servicio de Windows

Para el sistema de autenticación en web se usó el componente *Restful Authentication* que permite el manejo de usuarios con autenticación por contraseña, *cookies* del *browser*, o básica por medio del protocolo HTTP, con características de validación, autorización y cifrado de datos. El sistema se diferencia de otras herramientas similares existentes en el mercado en:

- Funcionalidades de visualización y administración basadas en un plano arquitectónico.
- Acceso administrativo vía web a diferencia de otras herramientas que disponen únicamente de una aplicación de escritorio.
- El acceso al código fuente permite tareas de modificación de funcionalidades, programación de nuevos módulos e integración de nuevas herramientas.
- Se permite la personalización y configuración de los parámetros de supervisión, a diferencia de la supervisión preestablecida que usan las otras herramientas.

El proyecto se da a conocer en el IV Encuentro Regional Semilleros de Investigación, celebrado en la ciudad de Cartago (Valle del Cáuca, Colombia), en el año 2009, además de participar en la cuarta convocatoria del programa nacional Destapa Futuro año 2009-2010, realizado por la Fundación Bavaria, llegando a ser finalista, luego de concluir varias etapas de selección entre más de 7000 proyectos de emprendimiento, procedentes de las diferentes regiones del territorio colombiano.

4. Conclusiones

- El software agente instalado en los equipos clientes funciona como un servicio del sistema operativo Windows, opera en forma automatizada y transparente para al usuario, pues no requiere de su intervención para la recopilación de información del sistema y envío de reportes. El software agente solo requiere una intervención previa por parte del usuario administrador, quien debe realizar la instalación de las librerías necesarias para asegurar el correcto funcionamiento del agente, y además, de forma manual, establecerá las configuraciones necesarias en el sistema para evitar que los usuarios sin privilegios de administración puedan manipular el servicio.
- La integración del módulo de visualización de planos empresariales y la ubicación visual de los equipos en dicho plano representa una gran diferencia competitiva ante otros sistemas similares existentes en el mercado, pues permite mayor agilidad y una mejor usabilidad en las tareas de administración que ofrece el sistema de monitoreo, evitando la clásica revisión de extensas listas y hojas de datos.
- La información correspondiente al inventario de hardware y software de cada equipo cliente está disponible en el sistema y es obtenida de forma automática a través del software agente, de esta forma se garantiza la agilidad y exactitud de la información en el proceso de recolección de datos.
- La información del inventario obtenida a través del software agente no permite alteración por parte de los usuarios del sistema; de este modo se garantiza la seguridad en el control y supervisión de datos y alarmas generadas en el momento de realizarse un cambio en el hardware o el software de un equipo cliente.
- El uso del *framework Ruby on Rails* permitió agilidad en el desarrollo del software y facilitó un rápido acoplamiento ante los cambios y novedades que surgieron en el transcurso del desarrollo del proyecto sin mayores contratiempos.
- El desarrollo bajo el *framework Ruby on Rails* se incorpora perfectamente a la metodología ágil *Scrum* en sus distintas fases y actividades, permite una fácil adaptación y reacción ante los cambios y da continuidad al proyecto para desarrollar nuevas versiones.
- El uso de técnicas de seguridad en programación permitió que el sistema en general sea más confiable y seguro, aplicando métodos de protección ante las amenazas y vulnerabilidades especificadas en el Top 10 de OWASP.

Bibliografía

AJPDSOFT (2010). Enciclopedia Definición WMI [en línea]. Murcia (España): Proyecto Ajpdsoft. <http://www.ajpdsoft.com/modules.php?name=Encyclopedia&op=content&tid=827> [consulta: 01/11/2010]

ARANDA SOFTWARE CORPORATION (2010). Asset Management [on line]. Miami (USA): Aranda. <http://www.arandasoft.com/solucion_aam.php> [consult: 01/11/2010]

AUTOMATAS.ORG (2006). Sistemas Scada [en línea]. Valladolid (España): Autómatas Industriales<http://www.automatas.org/redes/scadas.htm> [consulta: 01/11/2010]

CMN CONSULTING (2010). Excalibur [en línea]. Bogotá (Colombia): CMN Consulting. <http://www.cmn-consulting.com/soluciones/soluciones_index.htm> [consulta: 01/11/2010]

CONTROL CHAOS (2009). Scrum [en línea]. s.l: Advanced Development Methods. <http://www.controlchaos.com> [consulta: 01/12/2009]

DAVE, Thomas (2005). Programming Ruby: The Pragmatic Programmers Guide. 2 ed. Lewisville (Texas, USA): The Pragmatic Programmers. 833 p. ISBN: 0-9745140-5-5

DAVE, Thomas and HEINEMEIER HANSSON, David (2006). Agile Web Development with Rails. 2 ed. Lewisville (Texas, USA): The Pragmatic Programmers. 715 p. ISBN: 978-0-9776-1663-3

DEXON SOFTWARE(2010).DexonImaxt It Management Suite [en línea]. Bogotá (Colombia): Dexon Software Inc. <http://dexon.us/soluciones/dexon_it_asset.html> [consulta: 01/11/2010].

INFANTE, Luis (2010).Metodología ágil SCRUM [en línea]. Monterrey (México): Business Intelligence Latin America <http://www.bi-la.com/profiles/blog/list?user=1v7ecmaqe9u7d> [consulta: 01/11/2010].

JANOFF, Norman (2000). The Scrum Software Development Process for Small Teams [on line]. Phoenix (USA): IEEE Software. <http://members.cox.net/risingl1/Articles/IEEEScrum.pdf> [consult: 28/11/2008]

KNIBERG, Henrik (2007). Scrum and XP from the Trenches [on line]. Ontario (Canada): Java Zealot Solutions. <http://www.infoq.com/minibooks/scrum-xp-from-the-trenches> [consult: 01/11/2010].

LEVER IT (2010). Discovery Advantage [on line]. Bogotá (Colombia): LeverIT Asset Management IT. <http://www.leverit.com> [consult: 01/11/2010].

MARSHALL, Kevin (2006). Web Services on Rails. Sebastopol (California, USA): O'Reilly. 32 p. ISBN: 0-596-52796-9

MARSHALL, Kevin. (2007). PRO Active Record: Databases with Ruby on Rails. New York (USA): Apress. 280 p. ISBN: 1-59059-847-4

NAVARRO, Rafael (2007). REST vs Web Services [en línea]. Valencia (España): Universidad Politécnica de España. <http://users.dsic.upv.es/~rnavarro/NewWeb/docs/RestVsWebServices.pdf> [Consulta: 01/11/2010].

OWASP FOUNDATION (2008). OWASP Testing Guide v3.0 [on line]. Columbia (USA): OWASP. <http://www.lulu.com/items/volume_66/5691000/5691953/9/print/5691953.pdf> [consult: 01/11/2010]

OWASP FOUNDATION (2009). OWASP Ruby on Rails Security Guide [on line]. Columbia (USA): OWASP. <http://www.lulu.com/items/volume_64/5811000/5811294/3/print/5811294.pdf> [consult: 01/11/2010]

OWASP FOUNDATION (2010a). OWASP Top 10 [on line]. Columbia (USA): OWASP. <http://www.lulu.com/items/volume_68/1400000/1400974/5/print/1400974.pdf> [consult: 01/11/2010].

OWASP FOUNDATION (2010b). OWASP Guide 2.1: A Guide to Building Secure Web Applications and Web Services [on line]. Columbia (USA): OWASP/VRG Editions. <http://www.fecj.org/extra/Owasp2.1.0.pdf> [consult: 01/11/2010]

OWASP FOUNDATION (2011). OWASP - The Open Web application Security Project [on line]. Columbia (USA): OWASP/VRG Editions. <https://www.owasp.org/index.php/Main_Page> [Consult: 18/11/2011]

PALACIO, Juan (2007) Flexibilidad con Scrum [en línea]. Zaragoza (España): Navegapolis. <http://www.navegapolis.net/files/navegapolis_2009.pdf> [consulta: 28/11/2008]

SCHWABER, Ken (2004). Agile Project Management with Scrum. Redmond (USA): Microsoft Press. 156 p. ISBN: 0-7356-1993-X

SCHWABER, Ken (2007). The Enterprise and Scrum. Redmond (USA): Microsoft Press. 176 p. ISBN: 0735623376