13

Fecha de presentación: septiembre, 2016
Fecha de aceptación: noviembre, 2016
Fecha de publicación: Diciembre, 2016

SUBSISTEMA INFORMÁTICO

PARA LA INTEROPERABILIDAD DE LA PLATAFORMA SIUDERLAN DESARROLLADA EN LA EMPRESA ETECSA

COMPUTER SUBSYSTEM FOR THE SIUDERLAN PLATFORM INTEROPERABILITY DEVELOPED IN THE COMPANY ETECSA

MSc. Denis Morejón López[1]
E-mail: denis.morejon@etecsa.cu
MSc. Anay Carrillo Ramos[2]
E-mail: anayc@ucf.edu.cu
Ing. Darian Enrique Martínez Pombar[2]
[1]División Territorial de ETECSA. Cienfuegos. Cuba.
[2]Universidad de Cienfuegos. Cuba.

¿Cómo referenciar este artículo?

Morejón López, D., Carrillo Ramos, A., & Martínez Pombar, D. E. (2016). Subsistema Informático para la Interoperabilidad de la Plataforma SIUDERLAN desarrollada en la empresa ETECSA. Universidad y Sociedad [seriada en línea], 8 (4), pp. 100-105. Recuperado de http://rus.ucf.edu.cu/

RESUMEN

En el presente trabajo se aborda la implementación de un subsistema para la interoperabilidad de la plataforma SIUDERLAN con otras aplicaciones. Es desarrollado en la división territorial de ETECSA en Cienfuegos. Con este subsistema otras aplicaciones pueden comunicarse con el SIUDERLAN. Si se necesita bloquear el tráfico de una PC, por determinados motivos, puede emitirse la solicitud al SIUDERLAN y este llevar a cabo dicha tarea sin la intervención directa de un operador. En la documentación del subsistema se utiliza el lenguaje de modelado UML y como metodología de desarrollo el software RUP. Como lenguaje de programación se utiliza tanto del lado del cliente como del servidor Python y para la comunicación entre ambos lados se utiliza JSON. Como servidor de base de datos se emplea PostgreSQL 9.1. Todas estas herramientas y lenguajes son de código abierto.

Palabras clave: Subsistema informático, interoperabilidad, plataforma SIUDERLAN.

ABSTRACT

The present project deals with the implementation of a subsystem for SIUDERLAN Platform, with other applications. It was developed in ETECSA, Cienfuegos. With this subsystem other applications can communicate with SIUDERLAN. If, for any reason, it is needed to block traffic to a PC, a request can be issued to SIUDERLAN and it can carry out the task without the direct intervention of an operator. In the subsystem documentation, the UML modeling language is used and as development methodology RUP software. As a programming language, both the client side and server the Python are used and for communication between both JSON is used. As a data base server PostgreSQL 9.1 is used. All languages and tools are of open coding.

Keywords: Computer subsystem, interoperability, SUNDERLAND platform.

INTRODUCCIÓN

Las tecnologías para fabricar computadoras personales, *PC*, y otros elementos para redes locales, como conmutadores y enrutadores, se han desarrollado tanto que cada día es más fácil para las organizaciones montar redes de este tipo. Por tanto, existe una tendencia al crecimiento en el número de redes locales y en tamaño de las mismas (COMER, 2000). El tamaño es proporcional al número de miembros, ***computadoras***, que poseen. Esto ayuda a la productividad de las organizaciones, pero trae aparejados riesgos de seguridad que hay que tener en cuenta para el normal desarrollo de los negocios o actividades de las mismas (Barrientos, 2011).

Existen y se implementan en el mundo muchas medidas para asegurar las redes locales como:

- Sistema Antivirus.
- Sistemas Detectores de Intrusos (IDS, por sus siglas en inglés).
- Cortafuegos perimetrales.
- Sistemas para la supervisión de tráfico.
- Los subsistemas de trazas o historiales que se activan en las aplicaciones fundamentales de la organización.

Todos estos sistemas son capaces de detectar anomalías en la red y de saber la identificación o número IP de la computadora que la provoca. En redes pequeñas, ***en espacio y número de integrantes,*** este dato puede bastar para que el administrador de red ubique físicamente la PC infractora, porque puede hasta memorizar sus respectivos identificadores y el lugar donde están instaladas. Incluso si se tratara de una computadora portátil externa a la organización que fuera insertada desde uno de esos locales, basta con recorrerlos para encontrarla y tomar medidas administrativas en caso de necesidad. Pero este proceder no es efectivo aplicarlo cuando se trata de redes de más de 200 PC distribuidas en más de 3 edificios, que a su vez poseen más de 20 locales cada uno, por citar un ejemplo. El tiempo invertido es muy prolongado y si se trata de un intruso que agrede intencionalmente la red, esta puede retirarse a tiempo después de cumplir su objetivo antes de ser ubicado.

En la división territorial de ETECSA en Cienfuegos se desarrolla una aplicación nombrada: Sistema Informático para la Ubicación De Estaciones en una Red LAN (en inglés, Local Area Network), como parte de un proyecto de investigación en el que se demuestra cómo son insuficientes las soluciones existentes para detectar la entrada y ubicación de estaciones a la red, y por consiguiente emprender el desarrollo de este nuevo sistema que aún en la actualidad se sigue explotando y perfeccionando continuamente.

En la división territorial de ETECSA en la provincia de Cienfuegos, el departamento de Tecnologías de la Información lleva a cabo el desarrollo de la versión 0.3 del sistema SIUDERLAN. este se emplea para localizar estaciones en una red LAN cableada y es capaz de detectar máquinas nuevas que intentan utilizar la red con cualquier fin. Esta localización se hace definiendo reglas de manera gestionable a través de la interacción con un usuario, pero no es capaz de definir reglas a través de la interacción con otra aplicación.

Por ejemplo, si un antivirus detecta un programa maligno en una PC, este no puede aislar o desconectar la PC de la red. En el ejemplo anterior el programa maligno puede ser peligroso para la red y atacar los servicios críticos de la misma dejando la empresa sin la posibilidad de realizar las operaciones que tiene automatizada.

DESARROLLO

La arquitectura del subsistema fue concebida con 2 módulos; uno de interfaz de líneas de comandos para interactuar tanto con usuarios como con aplicaciones locales, y otro de servicio web para interactuar con aplicaciones remotas. Ambos módulos interactúan con la base de datos del SIUDERLAN, almacenada en un gestor Postgresql, a través de sus modelos de datos ya definidos en el framework de python (Jackson, 2013) Django (Holovaty & Kaplan Moss, 2010), como se aprecia en la figura 1.

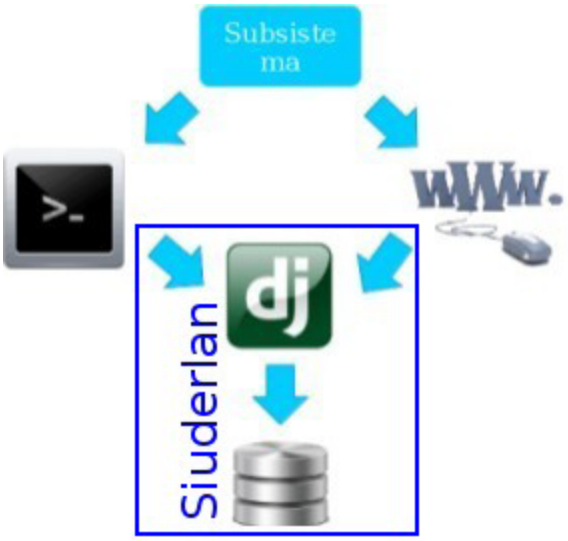

Figura 1. Diagrama del subsistema para la interoperabilidad del sistema SIUDERLAN.

Gracias a dicha arquitectura se pueden dividir los dos módulos de tal manera que el funcionamiento incorrecto

de uno no afecta al otro, además de poder instalar cada módulo en ordenadores diferentes. Esto permite ajustar cada módulo a los intereses más importantes de cada empresa. Significa que cada módulo se encarga de realizar particularmente una función concreta, permite a la empresa instalar el módulo de mayor interés en la mejor de sus PC, se aumenta el rendimiento del mismo.

Por otra parte se toma como principio implementar las mismas funcionalidades en ambos módulos. O sea, se pueden realizar las mismas operaciones tanto con el módulo de servicio web como con el de lineas de comandos.

La figura 2 muestra un ejemplo de la utilización del subsistema. Una aplicación externa, como es el caso de un servicio centralizado de antivirus Kaspersky, puede ser configurada para ejecutar un script cliente del servicio web. El objetivo es indicar al SIUDERLAN el bloqueo de una PC infectada. Por otra parte la interfaz de lineas de comandos puede ser utilizado directamente por un usuario o por una aplicación interna.

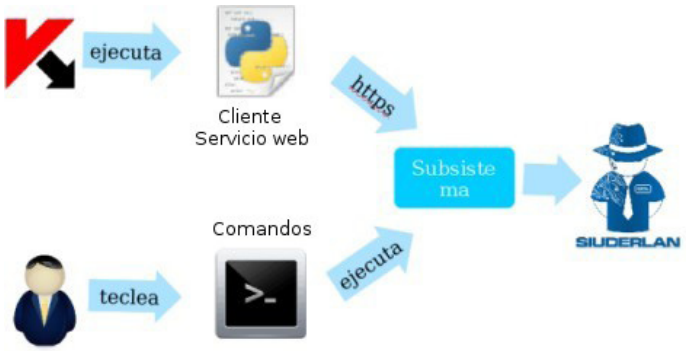

Figura 2. Ejemplo de uso del subsistema para la interoperatividad del SIUDERLAN.

Servicios web

Los servicios web son un conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web. Estas intercambian datos entre sí con el objetivo de ofrecer servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios los solicitan. Se llama a estos procedimientos a través de la Web. A su vez proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones, que interactúan entre sí para presentar información dinámica al usuario. Para proporcionar interoperabilidad y extensibilidad entre estas aplicaciones, y que al mismo tiempo sea posible su combinación para realizar operaciones complejas, es necesaria una arquitectura de referencia estándar.

Tipos de servicios web

El concepto ha sido perfilado en varios trabajos del comité Web Service Activity perteneciente al consorcio de web ***W3C***, particularmente con la propuesta del protocolo SOAP ha sido utilizado desde su concepción para automatizar el intercambio empresarial. No obstante el concepto se ha enriquecido con la profundización de las nociones de recurso y de estado, dentro del comité de modelación REST y en la profundización de la noción de servicio con el advenimiento de SOA.

SOAP

El protocolo simple de acceso a objetos o SOAP (siglas de Simple Object Access Protocol) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. Este protocolo se deriva de un protocolo creado por David Winer en 1998, llamado XML-RPC. SOAP ,creado por Microsoft, IBM y otros fabricantes. Está actualmente bajo el auspicio del consorcio W3C y es uno de los protocolos utilizados en los servicios web para intercomunicar aplicaciones.

REST

REST (Representational State Transfer) es un estilo de arquitectura de software para sistemas hipermedias distribuidos tales como la Web. El término es introducido en la tesis doctoral de Roy Fielding en 2000, quien es uno de los principales autores de la especificación de HTTP. En realidad, REST se refiere estrictamente a una colección de principios para el diseño de arquitecturas en red. Estos principios resumen cómo los recursos son definidos y diseccionados. El término frecuentemente es utilizado en el sentido de describir a cualquier interfaz que transmite datos específicos de un domino sobre HTTP sin una capa adicional, como hace SOAP. Estos dos significados pueden chocar o incluso solaparse. Es posible diseñar un sistema de gran tamaño de acuerdo con la arquitectura propuesta por Fielding sin utilizar HTTP o sin interactuar con la Web (Marset, 2006.). Así como también es posible diseñar una simple interfaz XML+HTTP que no ha de seguir los principios REST, y en cambio seguir un modelo RPC. Cabe destacar que REST no es un estándar, ya que es tan solo un estilo de arquitectura. Aunque REST no es un estándar, está basado en estándares:

- HTTP.
- URL.
- Representación de los recursos: XML/HTML/GIF/ JPEG/.
- Tipos MIME: text/xml, text/html.

Características de REST y SOAP

En la siguiente tabla se muestran las características entre REST y SOAP, así como también las ventajas y desventajas de estos.

Tabla 1. Características, ventajas y desventajas de REST y SOAP.

	REST	SOAP
Características	Las operaciones se definen en los mensajes. Una dirección única para cada instancia del proceso. Cada objeto soporta las operaciones estándares definidas. Componentes débilmente acoplados.	Las operaciones son definidas como puertos WSDL. Dirección única para todas las operaciones. Múltiples instancias del proceso comparten la misma operación. Componentes fuertemente acoplados.
Ventajas declaradas	Bajo consumo de recursos. Las instancias del proceso son creadas explícitamente. El cliente no necesita información de enrutamiento a partir de la URI inicial. Los clientes pueden tener una interfaz de escucha genérica para las notificaciones. Generalmente fácil de construir y adoptar.	Fácil generalmente de utilizar. La depuración es posible. Las operaciones complejas pueden ser escondidas detrás de una fachada. Envolver APIs existentes es sencillo. Incrementa la privacidad. Herramientas de desarrollo.
Posibles desventajas	Gran número de objetos. Manejar el espacio de nombres (URIs) puede ser engorroso. La descripción sintáctica /semántica informal, orientada al usuario. Pocas herramientas de desarrollo.	Los clientes necesitan saber las operaciones y su semántica antes del uso. Los clientes necesitan puertos dedicados para diferentes tipos de notificaciones. Las instancias del proceso son creadas implícitamente.

Implementación del módulo servicio web

En este trabajo se selecciona para el SIUDERLAN el servicio web de tipo REST debido a los siguientes aspectos:

- En el envío y recepción de datos del Servicio Web se utilizan mensajes con formato JSON que es un formato asociado al lenguaje javascript; y como la interfaz visual de la plataforma SIUDERLAN está desarrollada con ExtJs, que es un framework de javascript, esta puede enriquecerse en el futuro con la utilización también del Servicio Web REST.
- REST puede utilizar el protocolo HTTP como medio de transporte. Así que puede ser accedido al pasar a través de las configuraciones estándares de los cortafuegos.
- REST es un Servicio Web fácil de construir y de adoptar.

1. Se utiliza una librería del framework Django que devuelve los datos que se consultan en formato json. Las consultas deben realizarse con la autentificación de un usuario que previamente debe registrarse con este objetivo en el SIUDERLAN. La operación de más utilidad es la de retirar la confianza a una PC a través de su dirección MAC. Cuando se le retira la confianza a una PC el SIUDERLAN puede bloquear el puerto por el que se conecta si está configurado de esa manera dicho switch.
2. Se programa en python un cliente de este servicio web para que pueda ser utilizado por aplicaciones externas. No obstante los usuarios pueden crear sus propios clientes en el lenguaje de programación que desean. En la figura 3 se muestra la ejecución de dicho cliente, nombrado ***siuderlanWebserviceClient,*** con la intención de retirar la confianza de una PC con una determinada dirección MAC.

```
darian@isis-laptop:~$ siuderlanWebserviceClient -u root -p root --untrustmac 00:
00:00:00:00:00
{"msg": "trust to the mac address: 00:00:00:00:00:00 is removed", "data": "empty
", "success": true}
darian@isis-laptop:~$
```

Figura 3. Ejemplo de uso del servicio web para retirar la confianza a una PC.

Implementación del módulo interfaz de linea de comandos

La Interfaz de Líneas de Comandos o Command Line Interface, CLI por sus siglas en ***inglés***, es un método para manipular con instrucciones escritas al programa que subyace debajo. A esta interfaz se le acostumbra a llamar también consola de comandos. Se interactúa con la información de la manera más simple posible, sin gráficas ni nada más que el texto de las instrucciones. Las órdenes se escriben como líneas de texto, y si los programas responden, generalmente lo hacen ubicando la información de respuesta en las líneas de abajo.

Una CLI es usualmente utilizada directamente por usuarios, pero también puede ser usada por parte de una aplicación externa local. Es por esta razón que el subsistema diseñado comprende tanto un módulo de Servicio Web como un módulo de CLI.

El módulo de CLI creado realiza las mismas funciones que el módulo de servicio web, pero debe ser utilizado sólamente desde el mismo servidor que hospeda al SIUDEDERLAN. Por tanto no necesita autentificación de usuario, se asume que el usuario se ha autentificado en el sistema operativo para poder ejecutar los comandos. En el ejemplo de la figura 4 se consulta la ubicación de todos los switches de la red que se han agregado al SIUDERLAN.

```
darian@isis-laptop:~$ siuderlancli --showdevices
The action was success, message: list of all switches with his information
format: <building>; <office>; <rack>; <inventory>; <model>; <manufacturer>
Division Territorial; Office1; Rack1; 192EPL45; FN42; Huawei
Division Territorial; Office1; Rack1; PRT6734; Model; Cisco
Division Territorial; Office1; Rack1; 0293kd; model; allied telesyn
darian@isis-laptop:~$
```

Figura 4. Ejemplo del uso de la interfaz de comandos para obtener información acerca de la ubicación de los switches en la red.

Resumen de funcionalidades del subsistema

La tabla X muestra las principales funcionalidades que ejecutan ambos módulos. Es posible crear nuevas consultas en el futuro, tanto de obtención de datos como de interacción activa sobre el SIUDERLAN.

Tabla 2. Resúmen de las más importantes funcionalidades desarrolladas por el subsistema.

Funcionalidad	Descripción
trustmac / untrustmac	Agregar o retirar la confianza a una PC dada su dirección MAC
trustip / untrustip	Agregar o retirar la confianza a una PC dada su dirección IP
showdevices	Mostrar la ubicación de los dispositivos de conmutación de la red
allip / allmac	Mostrar todas las direcciones IP o direcciones MAC aprendidas
officeip / officemac	Mostrar las direcciones IP o MAC que están en cada oficina
ipswitch / macswitch	Mostrar las direcciones IP o MAC que están en cada switch

En este trabajo se realiza un total de 20 pruebas funcionales y 3 escenarios de prueba. Luego fue implementado totalmente y se encuentra en explotación en la división territorial de ETECSA en Cienfuegos con una valoración económica estimada en 10500 pesos.

CONCLUSIONES

El estudio de los servicios web y las interfaces de líneas de comandos permite que se elijan estas tecnologías como formas eficaces para interconectar al sistema SIUDERLAN con otras aplicaciones externas. Se diseña una arquitectura para el subsistema que cuenta con dos módulos funcionales que permiten la interacción con aplicaciones tanto locales como externas al servidor. Se implementa un módulo de servicio web con la utilización de tecnología REST y otro módulo de interfaz de línea de comandos, y se logra la interoperabilidad de la plataforma SIUDERLAN. Se valida el subsistema mediante las pruebas funcionales, minimizando la posibilidad de errores y elevando la calidad del software.

REFERENCIAS BIBLIOGRÁFICAS

Barrientos, F. J. (2011). ***Seguridad informática Ethical Hacking.*** Barcelona: Ediciones ENI.

Comer, D. E. (2000). Redes globales de información con internet y TCP/IP. México D. F: Prentice Hall Hispanoamericana S.A.

Hernández, J. V. (s.f.). SOA: ¿Qué es? ¿para qué sirve? ¿quién lo necesita? Madrid: Cognicase Management Consulting.

Holovaty, A., Kaplan Moss, J. (2010). La guía definitiva de django. Madrid: Anaya Multimedia-Anaya Interactiva.

Jackson, C. (2013). Learning to Program Using Python. Recuperado de https://www.ida.liu.se/~732A47/literature/PythonBook.pdf

León, J. A. (s.f.). ***Sistema informático para la ubicación de estaciones en redes LAN en la dirección territorial de etecsa en cienfuegos.*** Cienfuegos: Universidad de Cienfuegos.

Navarro Marset, R. (2006). REST vs Web Services. Recuperado de http://users.dsic.upv.es/~rnavarro/NewWeb/docs/RestVsWebServices.pdf

República de Cuba. Ministerio de Informática y Comunicaciones. (2016a). ***Línea de comandos***. Enciclopedia colaborativa en la red cubana. Recuperado de http://www.ecured.cu/Línea_de_comandos

República de Cuba. Ministerio de Informática y Comunicaciones. (2016b). ***Servicio Web***. Enciclopedia colaborativa en la red cubana. Recuperado de http://www.ecured.cu/Servicio_Web