UNIVERSIDAD POLITECNICA DE VALENCIA

DEPARTAMENTO DE COMUNICACIONES

Máster Universitario en Tecnologías, Sistemas y Redes de Comunicaciones

MONITORIZACIÓN Y SEGURIDAD EN REDES DE INFRAESTRUCTURAS CRÍTICAS MEDIANTE EL PROTOCOLO IPFIX

Autor: Hector Luis Castro Granados

Director: Carlos Enrique Palau Salvador

Fecha de comienzo: 1/03/2013

1 Monitorización y seguridad en redes de infraestructuras críticas mediante el protocolo IPFIX

Objetivos

Objetivo general

Diseñar y desarrollar un sistema de monitorización de red para infraestructuras críticas mediante IPFIX, permitiendo la detección efectiva de ataques de seguridad y generando archivos registro que puedan ser accedidos por otras aplicaciones.

Objetivos específicos

- Conceptualizar un marco apropiado para la monitorización de una infraestructura crítica.
- Mostrar las ventajas que supone IPFIX como protocolo para el transporte de registros de datos de monitorización.
- Diseñar una arquitectura de red apropiada para la implementación de IPFIX.
- Analizar los problemas de diseño presentes actualmente en los sistemas de monitorización que utilizan IPFIX.
- Realizar una prueba de concepto del sistema de monitorización planteado, detectando ataques DDoS y escaneo de puertos.
- Programar una función en Java que permita la lectura de los archivos TXT generados por nProbe y los mapee a plantillas XML siguiendo la especificación SML de SWE.

Metodología

Se han llevado 4 partes importantes a la hora de desarrollar el proyecto. El primero fue el de conocer la arquitectura, las estructuras de información la dinámica, las aplicaciones de IPFIX y las herramientas que permiten su implementación. La segunda parte consistió en el diseño de la testbed sobre la cual se puso a prueba IPFIX para la monitorización de redes de infraestructuras críticas y la detección de ataques del tipo DDoS y escaneo de puertos. Por último, se desarrolló un programa en Java que mapeaba los archivos TXT generados como registros de IPFIX a archivos XML siguiendo la especificación SML de SWE; con el fin de que los registros de IPFIX pudieran ser entendidos por un SOS y así ser puestos a disposición de otras herramientas o usuarios que necesiten acceso a dicha información.

Resultados

El sistema de monitorización para la protección de redes de infraestructuras críticas mediante el uso de IPFIX que se presentó en esta tesina de máster es apto para la detección de ciertos tipos de ataques a la red ya que dió como resultado la correcta detección de ataques DDoS y escaneo de puertos.

Además, se demostró la eficiencia, flexibilidad e interoperabilidad de IPFIX en los sistemas de monitorización de redes gracias al uso de plantillas y la generación de archivos de registro. Sin embargo, se encontró el inconveniente de que IPFIX aún no se encuentra incluido en los equipos de red y de hecho existen pocas herramientas software que lo implementen a cabalidad; por lo que es necesario optar por nuevas alternativas de implementación mientras los fabricantes incluyen en sus diseños este protocolo, como la presentada en esta tesina.

Por otra parte, el programa desarrollado en lenguaje Java brinda grandes ventajas debido a que no sólo transcribe correctamente los datos de los ficheros texto a las plantillas XML, sino que además permite el

almacenamiento de datos en variables, dando la posibilidad de crear más aplicaciones Java que puedan aprovechar dichos recursos.

Finalmente, la integración del sistema de monitorización con herramientas de SWE como el SOS dan características de interoperabilidad con otros sistemas, ya que pone a disposición de usuarios y servicios autorizados los datos de monitorización bajo la misma especificación SML.

Líneas futuras

- Desarrollar un software que mejore la implementación y compatibilidad de los elementos de información privados manejados en IPFIX.
- Implementar IPFIX dentro del software de los equipos de red.
- Crear aplicaciones a partir de los registros IPFIX guardados en un SOS bajo el concepto "IPFIX as a sensor".
- Utilizar el sistema de monitorización mediante IPFIX para la detección de otras amenazas de seguridad.

Abstract

La monitorización y seguridad de las infraestructuras críticas son tema fundamental en la seguridad de una nación debido a la escala del impacto económico, político y social que se produciría si fallara su sistema. Además es necesario establecer un marco de monitorización que englobe todos los elementos integrantes de una infraestructura crítica con el fin de brindarle protección frente ataques cibernéticos. En el desarrollo de la presente tesina se propondrá un sistema de monitorización para la protección de infraestructuras críticas mediante el protocolo IPFIX. Para ello, se ha diseñado un "testbed" conformado por un ejemplo controlado de red de ciber-activos críticos la cual se buscó monitorizar y detectar ataques del tipo DDoS y escaneo de puertos. A su vez, se desarrolló un programa en Java el cual se encarga de mapear los archivos txt de registros creados por IPFIX a archivos XML que cumplen con las especificaciones de SML de SWE, permitiendo subir los registros a una plataforma SOS para ser puestos a disposición de otras herramientas o usuarios que necesiten acceso a dicha información.

Autor: Hector Luis Castro Granados email: hectorcastro1993@gmail.com

Director 1: Carlos Enrique Palau Salvador email: cpalau@dcom.upv.es

Fecha de entrega: 02-12-2013

ÍNDICE

I. INTRODUCCIÓN

Las redes de telecomunicaciones juegan un papel fundamental en la gestión de infraestructuras críticas, tanto para garantizar el correcto intercambio de información de control entre las diferentes entidades que componen el sistema crítico. Debido a la convergencia de las tecnologías a "todo basado en IP", han evolucionado los sistemas SCADA pasando de ser soluciones propietarias y dedicadas a ser soluciones distribuidas. Sin embargo, este avance conlleva también problemas de los que se destaca la seguridad.

Hoy en día, los crímenes en el ciberespacio van en aumento y los fallos de seguridad en los sistemas de infraestructuras críticas no son la excepción. Una de las características que define a una infraestructura crítica es la escala del impacto económico, político y social que produciría si fallara su sistema. Está el claro ejemplo de Stuxnet, el cual fue un software dañino que infecto al sistema de una central nuclear en Irán y alteró el comportamiento de las turbinas de refrigeración del reactor.

Por lo anterior se requiere establecer un marco de monitorización que englobe todos los elementos integrantes de una infraestructura crítica con el fin de brindarle protección frente ataques cibernéticos. En el desarrollo de la presente tesina se propondrá un sistema de monitorización para la protección de infraestructuras críticas mediante el protocolo IPFIX.

Ahora, ¿Por qué el uso de IPFIX como protocolo de monitorización?. La respuesta se encuentra en el hecho de que IPFIX permite recopilar datos de todos los paquetes que pasan por la red consumiendo muy pocos recursos de la misma y proporcionando gran escalabilidad. Además, su protocolo de capa de transporte está diseñado para asegurar la correcta llegada de los registros a su destino en caso de congestión. Otra significativa ventaja es el uso de elementos de información tanto los definidos por IANA como los privados definidos por el usuario, entre otras ventajas que serán comentadas en su respectivo capitulo.

En el capítulo I se abarcará una breve descripción de los problemas cibernéticos que acechan a las redes de infraestructuras críticas, ejemplificando y especificando el por qué los nuevos marcos de monitorización deben tener en cuenta las redes de comunicaciones de los diferentes elementos conformantes.

Posteriormente en el capítulo II se hará una corta mención de las principales herramientas software utilizadas para el desarrollo del proyecto.

En el capítulo III se comentarán rápidamente los programas que velan por la seguridad de las infraestructuras críticas: PEPIC para la Unión Europea y CNPIC para España. Luego se mostrará uno de los estándares más confiables del mundo entorno a la seguridad de infraestructuras críticas, NERC CIP.

Los capítulos IV, V, VI, VII y VIII se dedicarán para especificar el trabajo desarrollado en la tesina, explicando la metodología de monitorización en la que se basó el proyecto (BUGYO) [1] y los diferentes bloques que componen el sistema de monitorización planteado: Infraestructuras críticas, protocolo para la monitorización de redes IPFIX, Arquitectura de monitorización y el registro de datos en un SOS. Finalmente se concluirá analizando los resultados obtenidos en cada una de las fases llevadas a cabo y los diferentes inconvenientes de diseño encontrados.

La presente tesina ha sido realizada por Hector Luis Castro Granados para el máster en Tecnologías, servicios y redes de comunicaciones, impartido en la Escuela Técnica Superior de Ingenieros de Telecomunicaciones de la Universitat Politècnica de València. Ha estado bajo la revisión del doctor Carlos Palau Salvador, director de la presente tesina.

II. OBJETIVOS

II.1. *OBJETIVO GENERAL*

Diseñar y desarrollar un sistema de monitorización de red para infraestructuras críticas mediante IPFIX, permitiendo la detección efectiva de ataques de seguridad y generando archivos registro que puedan ser accedidos por otras aplicaciones.

II.1. *OBJETIVOS ESPECÍFICOS*

- Conceptualizar un marco apropiado para la monitorización de una infraestructura crítica.
- Mostrar las ventajas que supone IPFIX como protocolo para el transporte de registros de datos de monitorización.
- Diseñar una arquitectura de red apropiada para la implementación de IPFIX.
- Analizar los problemas de diseño presentes actualmente en los sistemas de monitorización que utilizan IPFIX.
- Realizar una prueba de concepto del sistema de monitorización planteado, detectando ataques DDoS y escaneo de puertos.
- Programar una función en Java que permita la lectura de los archivos TXT generados por nProbe y los mapee a plantillas XML siguiendo la especificación SML de SWE.

III. DESCRIPCIÓN DEL PROBLEMA

Para nadie es un secreto que gran parte del desarrollo tecnológico llevado a través de la historia es debido en gran parte a la guerra y la innovación en las tecnologías de la información no es la excepción. Tanto así que actualmente se habla del ciberespacio como quinto dominio de la guerra junto a la tierra, mar, aire y espacio.

Claros ejemplo de la situación anterior son la estimación realizada en el plan ciberseguridad de la Unión Europea presentado el 7 de febrero de 2013; en el que se calcula que cada día hay hasta 150000 virus informáticos en circulación y hasta 148000 ordenadores infectados; y las predicciones presentadas por el Foro económico mundial de que existe un 10% de probabilidad de interrupción de infraestructuras de información críticas en los próximos diez años; lo cual causaría daños por valor de 250 000 millones de dólares [2].

Las noticias de ataques digitales (ciberataques) a infraestructuras críticas como plantas de energía, centrales nucleares o fábricas de diferente índole se han vuelto habituales. De hecho, el ataque cibernético más grande de la historia se produjo en los sistemas de control de la central nuclear de Bushehr y al menos 14 industrias más en Irán el día 27 de Septiembre de 2010. Detrás de dicho ciberataque se encontró un software dañino de 500 Kb muy avanzado del tipo gusano llamado Stuxnet, el cual se propagaba a través de la red informática de los sistemas SCADA y permitía a sus creadores espiar los sistemas industriales e incluso controlar los sistemas a espaldas de los operarios de las plantas. Eugene Karpesky, fundador de la compañía informática que lleva su nombre, comentó a la revista IEEE Spectrum de marzo de 2013: “Stuxnet no pudo haber sido creado por ningún grupo de hackers, solo algunos estados tienen recursos para montar una operación semejante” [3]. Stuxnet es el primer ejemplo de guerra cibernética.

Una vez Stuxnet se instalaba en el sistema, mediante las redes de comunicaciones, el gusano buscaba los ordenadores y controladores industriales de los sistemas SCADA que poseyeran la vulnerabilidad MS10-0466 de los sistemas operativos Windows CC. La tendencia de la arquitectura de las redes de telecomunicaciones a “todo IP” ha facilitado la proliferación y expansión de este tipo de ataques, por lo que se hace necesario que los planes de monitorización y seguridad de las infraestructuras críticas no se limiten sólo a la protección de sus ciber-activos sino que también se extiendan a la protección de las redes de telecomunicaciones a las que se encuentran conectados, con el fin de prevenir, detectar o generar una respuesta oportuna y eficiente a los fallos internos o ataques deliberados contra las infraestructuras críticas [4].

IV. HERRAMIENTAS

IV.1. *NPROBE*

Es un software sonda desarrollado por ntop y permite la recolección, análisis y exportación de reportes del tráfico de redes utilizando los protocolos de Cisco Netflow v5 y v9, y el protocolo IPFIX. Está disponible para diferentes sistemas operativos como Linux, Windows y MacOSX. Nprobe sobre Windows es activado como un servicio o aplicación que se inicia desde un cmd.exe.

Un ejemplo de configuración de nprobe es el siguiente [5]:

nprobe /i -i 0 -n 192.168.0.1:2055

En dicho comando se configura nprobe para que cree un servicio en la consola, comience a leer todos los paquetes que pasan por la interfaz 0 del dispositivo, cree los paquetes en formato por defecto Netflow v5 y los envíe a la dirección IP 192.168.0.1 por el puerto lógico 2055.

Además permite la selección del protocolo de transporte que quiere usarse para exportar los datos IPFIX. En el anexo A puede observarse un ejemplo de configuración para nProbe utilizando plantillas IPFIX y la explicación de cada uno de sus elementos.

Para conocer las diferentes opciones de configuración que posee nprobe puede introducirse el comendo “nprobe /c -h” o consultar la guía de usuario disponible en la página web de ntop www.ntop.org.

IV.2. *NETFLOW ANALYZER*

Netflow analyzer es una herramienta de monitorización de ancho de banda y análisis de tráfico de red que proporciona visibilidad en tiempo real del rendimiento del ancho de banda de la red. Es un software recopilador, analizador y motor de informes Netflow, IPFIX, sFlow, JFlow, etc, disponible para los sistemas operativos Windows y Linux, y una de sus ventajas representativas es que no necesita de ningún dispositivo hardware adicional para su funcionamiento [6].

Netflow analyzer brinda varios informes instantáneos y cuenta con una interfaz gráfica amigable con el usuario y que se va actualizando cada cierto tiempo configurable.

La mayor ventaja del uso de este software de monitorización es el control de protocolos y aplicaciones, debido a que permite a los administradores de la red la posibilidad de ver la distribución de protocolos y que aplicaciones se están usando en la red con el ancho de banda consumido por aplicación.

III.3. *HPING3*

Hping es un generador de paquetes TCP libre que es utilizado para auditorias de seguridad y para el testeo de cortafuegos y redes. Con este software se pueden trazar rutas, realizar pings y probar el

acceso a host que estén detrás de un firewall. Además de explotar vulnerabilidades de las pilas TCP/IP dando la posibilidad de realizar ataques DoS del tipo TCP Syn flooding [7].

IV.4. *NMAP*

Nmap es una herramienta libre y de código abierto para auditorias de seguridad y descubrimiento de redes. Nmap utiliza paquetes IP para encontrar los host que se encuentran disponibles en la red, los servicios que están utilizando, los sistemas operativos sobre los que funcionan, tipo de filtrado de paquetes o cortafuegos que están en uso, entre otras. Este software se encuentra disponible para los sistemas operativos Windows, Linux y Mac OS X [8].

Nmap puede ser un excelente identificador de fallas y vulnerabilidades de seguridad en la red analizada. Pero por contraparte, puede usarse sólo o en conjunto con otros programas para preparar ataques, como herramienta de intrusión.

V. NORMATIVA PARA LA PROTECCIÓN DE INFRAESTRUCTURAS CRÍTICAS.

V.1. *PEPIC*

El Programa Europeo de Protección de infraestructuras críticas (PEPIC) nace en 2005 como respuesta por parte de la Unión Europea frente a los posibles ataques terroristas que afectan la seguridad de sus infraestructuras críticas, creando un marco para mejorar la prevención, preparación y respuesta de Europa frente a diferentes tipos de amenazas contra dichas infraestructuras, tales como ataques terroristas [9].

V.2. *CNPIC*

Es el órgano creado por la Secretaría de Estado de Seguridad Española que es responsable de la dirección, coordinación y supervisión de la protección de infraestructuras críticas españolas. Desde su creación se ha encargado de la elaboración y actualización del catálogo de infraestructuras críticas y el Plan Nacional de Protección de Infraestructuras Críticas con el fin de dirigir y coordinar las actuaciones en materia de protección de Infraestructuras críticas, previa identificación y designación de las mismas, impulsando a los diferentes propietarios de dichas estructuras a optimizar el grado de protección de éstas contra ataques deliberados de todo tipo [10].

V.3. *NERC CIP*

NERC (North American Electric Reliability Corporation) es una entidad sin ánimo de lucro que se encarga de asegurar la fiabilidad del sistema de volumen de energía en América del Norte. NERC ha ayudado a hacer el sistema de distribución eléctrico Norte Americano el más confiable del

mundo, gracias al desarrollo de un estándar de ciberseguridad permanente que ha influenciado la creación de otros estándares y esquemas de industrias. Dentro del set de estándares desarrollados por NERC se encuentran los NERC CIP, específicamente creados para la Protección de Infraestructuras Críticas [11].

A continuación se presentan los más importantes.

- ESTÁNDAR CIP-002-1 IDENTIFICACIÓN DE CIBER-ACTIVOS CRÍTICOS
- ESTÁNDAR CIP-003-1 CONTROLES DE GESTIÓN DE SEGURIDAD
- ESTÁNDAR CIP-004-1 PERSONAL Y ENTRENAMIENTO
- ESTÁNDAR CIP-005-1 PERÍMETRO DE SEGURIDAD ELECTRÓNICO

  Para cumplir con este estándar se requiere la identificación, documentación y protección de un perímetro de seguridad electrónica en la cual residan los ciber-activos críticos y todos sus puntos de acceso electrónico. Dentro de dichos perímetros pueden encontrarse por ejemplo centros de control SCADA, consolas de sistemas de medición, consolas de sistemas de administración, centros de cómputo, equipos de telecomunicaciones, etc. En la siguiente figura se ejemplifica un perímetro de seguridad electrónico de un centro de control SCADA, especificando sus puntos de acceso electrónico.

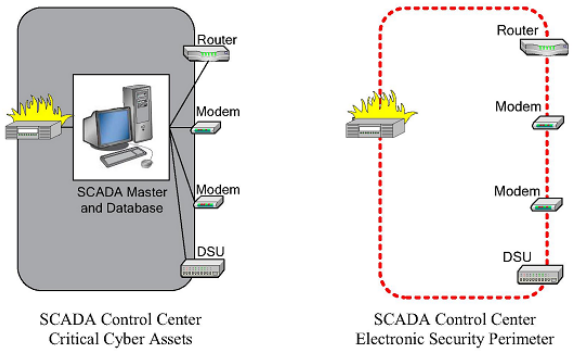


Fig. 1. Ejemplo de perímetro de seguridad electrónico [11].

- ESTÁNDAR CIP-006-1 SEGURIDAD FÍSICA
- ESTÁNDAR CIP-007-1 SISTEMAS DE GESTIÓN DE SEGURIDAD
- ESTÁNDAR CIP-008-1 PLANEAMIENTO DE REPORTE Y RESPUESTA FRENTE A INCIDENTES
- ESTÁNDAR CIP-009-1 PLANES DE RECUPERACIÓN PARA CIBER-ACTIVOS CRÍTICOS.

VI. CONCEPTO

VI.1. *SELECCIÓN DEL MARCO DE MONITORIZACIÓN*

Como se comentó en los apartados anteriores, los sistemas de control para infraestructuras críticas están evolucionando desde soluciones propietarias y dedicadas hacia marcos de funcionamiento basados en IP, trayendo consigo problemas de seguridad a los nuevos escenarios de sistemas SCADA, exponiéndolos a ciber-amenazas, por lo que es necesario la adopción de una estrategia efectiva de seguridad de red para la protección global de las infraestructuras críticas, tales como la red eléctrica, centrales de producción, transporte y distribución de gas y petróleo, redes de distribución de agua.

A su vez, dicha estrategia debe formar parte de un marco de monitorización del sistema, el cual procese y represente la recolección de información, y la provea a los servicios o clientes que hagan uso de esta información de monitorización. Típicamente, la información en bruto es detallada en el lugar donde es recogida y por lo tanto, es utilizada por otros componentes software del sistema para llevar a cabo un tratamiento posterior de datos, tales como visualizaciones para usuarios humanos o sistemas de comando y control de adaptación automática en base al análisis algorítmico [12].

Como referencia para el desarrollo de la tesina se tomará la arquitectura de monitorización BUGYO (Building Security Assurance in Open Infraestructures), la cual provee de un marco de monitorización con formatos de protocolo basados en el uso de XML, que abarca el problema de la garantía de seguridad en los servicios, redes e infraestructura de telecomunicaciones [13]. El principal reto de este marco era el de proporcionar una solución al problema de la recopilación de datos relacionados con los mecanismos de monitorización y seguridad del sistema observado. A continuación se puede observar el esquema básico de la arquitectura de monitorización BUGYO:

Fig. 2. Esquema básico de los actuadores en la arquitectura de monitorización BUGYO.

En donde [14]:

- El marco de monitorización es el responsable de recolectar todos los resultados de los controles efectuados por las sondas y los agrupará al proceso de evalucación de medidas correspondiente. Luego evaluará los resultados de dichos procesos y finalmente enviará los resultados a la cabina o las cabinas de garantía de seguridad para ser incluido o actualizado.
- La cabina de garantía de seguridad es la responsable de recoger los resultados de evaluación de las medidas realizado por el marco de monitorización, el estado de los dispositivos que participan en el modelo de servicio del NOC y las consultas de los usuarios. También se encarga de mantener actualizada la base de datos. Por último, será la encargada de gestionar las consultas de los usuarios y llevar a cabo las acciones apropiadas y mostrar toda la información en la interfáz gráfica del usuario.
- El NOC es el responsable de la supervición del estado de los dispositivos que participan en el modo de servicio, como los datos de inventario y la disponibilidad de las maquinas.
- Los usuarios finales tienen la posibilidad de conectarse con la cabina de garantía de seguridad para configurarla, adaptarla a sus necesidades y solicitarle la información que desea que se muestre.

VI.2. *PLANTEAMIENTO DEL SISTEMA DE MONITORIZACIÓN*

Continuando con la metodología BUGYO, una red de monitorización para la protección de Infraestructuras críticas tendría el siguiente esquema general:

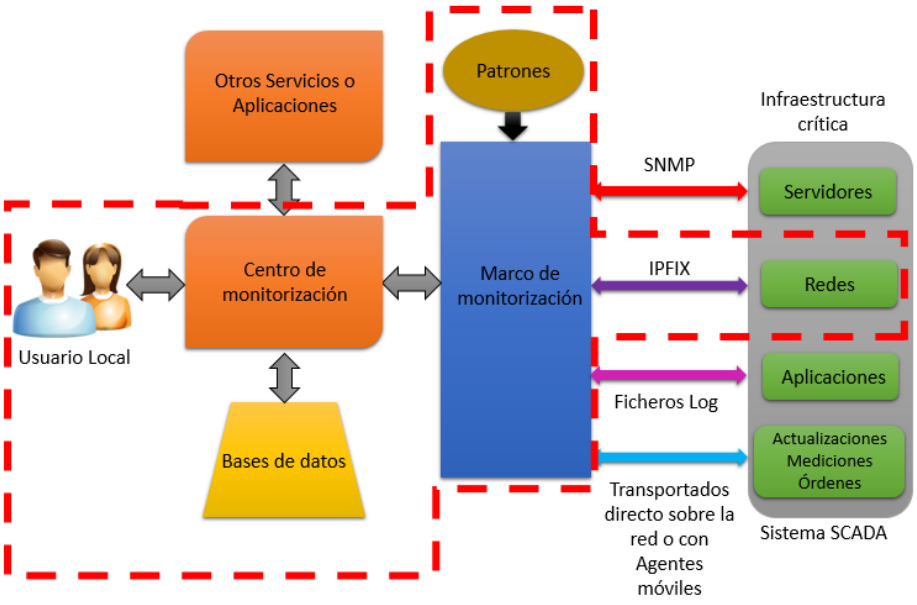


Fig. 3. Aplicación de la metodología BUGYO al entorno de monitorización para CIP propuesto en el actual trabajo.

Analizando los componentes de la anterior figura, encontramos al usuario local y otros servicios o aplicaciones que al tener los permisos necesarios pueden acceder a la información recolectada por el centro de monitorización, el cual concentra toda la información procesada en sus bases de datos. El procesamiento de dicha información es realizado siguiendo el respectivo marco de monitorización teniendo en cuenta una serie de patrones definidos. Y por último, se posee la infraestructura crítica a monitorizar, la cual es en realidad un sistema SCADA. El concepto de sonda definido en BUGYO puede ser desplegado para tomar medidas o transportar información de los servidores, de las redes y las aplicaciones que conforman el sistema SCADA. La monitorización de los servidores podría ser llevada a cabo utilizando el protocolo SNMP; a la vez que los datos como mediciones, órdenes y actualizaciones, pueden ser gestionados mediante agentes móviles o simplemente ser transportados directamente sobre la red; mientras la monitorización de aplicaciones puede ser realizada mediante el estudio de ficheros de log de los procesos, servicios y sistemas operativos.

Sin embargo, como se ha comentado en el capitulo 2, es necesario que la monitorización de una infraestructura crítica extienda su ámbito al sector de las redes de telecomunicaciones. El actual proyecto se centra en la monitorización de este sector presentando como propuesta la implementación de IPFIX. Para ello se propone el siguiente sistema:

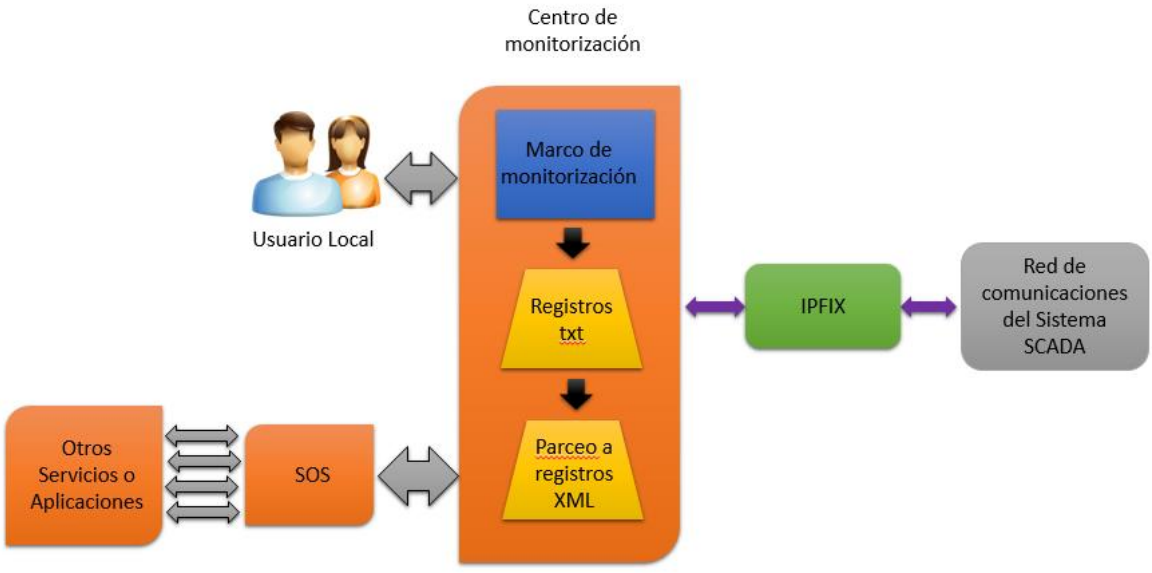


Fig. 4. Sistema de monitorización y seguridad para CIP implementando el protocolo IPFIX.

Entonces mediante IPFIX se buscará monitorizar diferentes parámetros que pueda poseer una red de comunicaciones del sistema SCADA tales como direcciones IP y MAC origen y destino de todos los paquetes enviados, flags de capa TCP, cantidad de bytes transportados, tipos de

protocolo, puertos utilizados, etc. Los datos recolectados por el colector IPFIX entrarán al marco de monitorización donde serán procesados, analizados y presentados en forma visual a los usuarios locales. A su vez, se crearán registros txt entendibles para los usuarios y se parcearan automáticamente a formato XML (siguiendo el estándar OGC SWE) mediante un programa JAVA desarrollado por el autor del presente trabajo. Finalmente los registros en formato XML serán enviados a un SOS en donde serán almacenados, siendo accesibles por aquellos usuarios, servicios o aplicaciones que tengan los correspondientes permisos y requieran el uso de dichos registros.

VII. IPFIX

IPFIX (IP Flow Information Protocol) es un estándar del IETF para la exportación de la información que fluye en redes IP. El protocolo IPFIX es usado para maximizar la flexibilidad y la interoperabilidad en los sistemas de monitorización de redes. Dicho protocolo se creó con los objetivos de ser funcional tanto en redes actuales como futuras y de ser aplicado en la gestión de redes detrás de las capas de red y de transporte según el modelo OSI. IPFIX es un protocolo de transporte independiente y unidireccional con una representación flexible de datos y un modelo de información que intenta suplir la mayoría de las necesidades, en cuanto a gestión de redes, de las capas 3 y 4 siguiendo el modelo OSI [15].

El protocolo Netflow V9 fue seleccionado como base de IPFIX debido a que implementa plantillas que flexibilizan la definición de la información que se está transfiriendo.

La clave de la innovación en IPFIX es la flexible definición de flujo de red y los formatos de registro a través de plantillas basadas en un modelo de información bien definido y extensible, independiente del protocolo de transporte usado o el tipo de trama del mensaje.

VII.1. *ARQUITECTURA DE IPFIX*

La arquitectura de IPFIX busca generalizar las propiedades comunes de las arquitecturas de medida actuales. Se conforma básicamente por tres procesos: procesos de medida, de exportación y de recolección. El proceso de medida genera flujos de información a partir de los paquetes que se están observando, luego el proceso de exportación utiliza IPFIX para enviar los flujos al proceso de recolección. Cada una de las relaciones entre los procesos pueden ser one-to-many, en las que cada nodo puede enviar y recibir paquetes de datos de múltiples nodos [15].

En recientes trabajos sobre IPFIX ha surgido un 4 tipo de proceso denominado proceso intermediario, el cual puede modificar flujos. Dicho proceso surge a partir de diferentes necesidades tales como la compresión de información o la falta de equipos comerciales que puedan implementar el protocolo IPFIX. Entonces un dispositivo, como por ejemplo un router, que no implemente IPFIX puede enviar sus datos en flujos Netflow a un proceso intermediario que se

encargue de la transformación de tramas Netflow a tramas IPFIX y que posteriormente envíe el resultado al exportador IPFIX correspondiente.

VII.2. *ESTRUCTURAS DE INFORMACIÓN EN IPFIX*

El mensaje es la unidad básica de transferencia de IPFIX. Un mensaje contiene una cabecera y uno o más sets, los cuales contienen registros de información. La cabecera del mensaje posee una longitud de 16 bytes y contiene un número de versión, la longitud del mensaje, una marca de tiempo de exportación, un número de secuencia y una identificación del dominio de observación. La identificación del dominio de observación permite conocer el lugar donde el flujo fue observado. El número de secuencia es utilizado para detectar las pérdidas de mensajes.

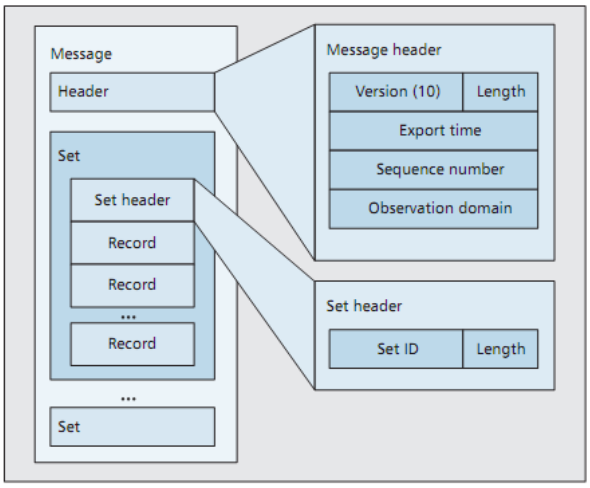


Fig. 5. Estructura de un mensaje IPFIX [15].

Los sets pueden ser clasificados dependiendo de su contenido: sets con plantillas, sets con registros de plantillas de opciones o sets de datos que contienen los registros de datos. Cada set posee su propia cabecera, la cual a su vez posee un campo longitud de set y una identificación de set de 16 bits.

Las plantillas describen el formato con el que se enviarán los registros de datos. Esta es la razón por la cual IPFIX posee su flexibilidad. Una plantilla contiene una lista ordenada de elementos de información, los cuales representan un tipo de dato y significado específico mediante un campo nombrado.

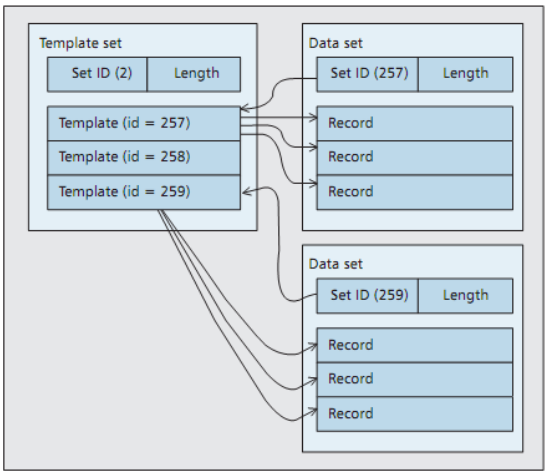


Fig. 6. Organización de los registros en una plantilla IPFIX [15].

Existen además las plantillas de opciones. Como su nombre lo indica, transporta información acerca de la estructura de la plantilla en lugar de elementos de información. Las plantillas de opciones dan la habilidad a los exportadores de enviar información adicional a los colectores.

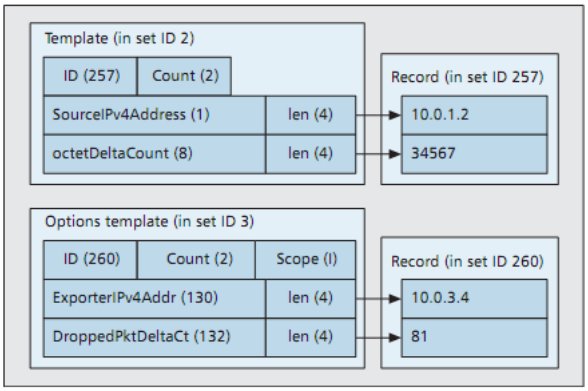


Fig. 7. Estructura de una plantilla y una plantilla de opciones IPFIX [15].

Las plantillas tienen la misma duración que la sesión de transporte en las que son utilizadas. Si las plantillas después de ser enviadas no necesitan ser reenviadas o reutilizadas, pueden ser retiradas mediante orden o borradas al finalizar la sesión de transporte.

Por ejemplo, para borrar o liberar una plantilla enviada previamente al proceso recolector, el proceso exportador envía un mensaje de retiro de plantilla. Este mensaje contiene uno o más plantillas de registros con un cero en la casilla de contador de elementos información.

VII.3. *DINÁMICA: TRANSPORTE, ALMACENAMIENTO DE DATOS Y GESTIÓN DE PLANTILLAS Y SESIONES*

El intercambio de un conjunto de mensajes en una comunicación IPFIX entre un proceso de exportación y un proceso de recolección se conoce conceptualmente como sesión de transporte. En una sesión de transporte, el proceso de exportación inicia una conexión con el proceso de

recolección, luego se envían las plantillas seguidas por los datos descritos en dichas plantillas. Durante el proceso pueden ser enviadas también órdenes para el re-uso y borrado de plantillas, además de los paquetes concernientes a la capa de transporte cuando se utilizan los protocolos TCP o SCTP para confiabilidad y control de congestión. IPFIX puede trabajar con los protocolos SCTP, TCP y UDP, además de dar la posibilidad de almacenar la información en archivos, los cuales pueden ser guardados, enviados y tratados en otro lugar.

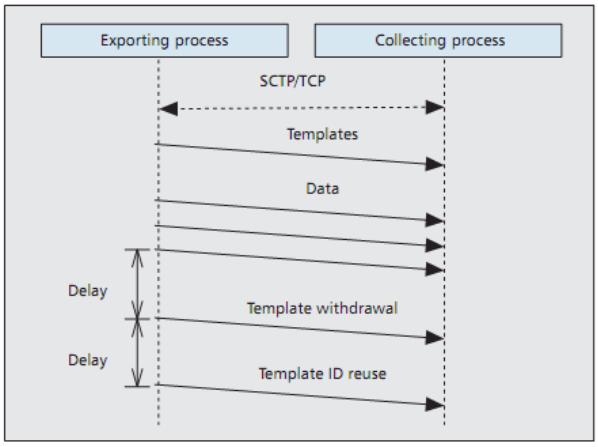


Fig. 8. Dinámica entre los procesos de exportación y recolección de datos IPFIX [15].

En IPFIX, el protocolo SCTP es el más utilizado debido a que este provee de un servicio de envío de paquetes secuencial con confiabilidad selectiva por paquete y control de congestión. La confiabilidad selectiva permite a los mensajes IPFIX con plantillas e información de alta importancia darles mayor prioridad en caso de congestión.

Cada flujo en una asociación SCTP es asociado con un buffer separado, por lo que los mensajes enviados en un flujo no bloquearan a los demás. Cuando se usa SCTP, el campo de número de secuencia en la cabecera de un mensaje IPFIX cuenta los registros de datos enviados por cada flujo de SCTP desde el inicio de la sesión de transporte. Lo anterior puede ser usado para contar los registros faltantes debido a la confiabilidad parcial en base a los demás flujo.

Por otra parte, el orden de los mensajes en los flujos no es necesariamente mantenido; un mensaje IPFIX puede aparecer sobre cualquier flujo IPFIX enviado, creando inconvenientes por ejemplo en el envío de mensajes con plantillas y con retiro de plantillas. Así pues, al no mantenerse un orden de envío y al ser enviado un mensaje con una plantilla y prontamente un mensaje con su retiro, pueden crearse inconsistencias. Por esto, se recomienda a los usuarios dejar un retardo razonable entre el envío de estos dos mensajes.

El uso de TCP como protocolo de transporte para los mensajes IPFIX es idéntico al funcionamiento de IPFIX con SCTP utilizando sólo un flujo y confiabilidad completa para cada mensaje.

El envío de mensajes IPFIX sobre el protocolo de transporte UDP se realiza normalmente sobre redes dedicadas. Cuando se usa UDP el campo de número de secuencia en la cabecera del mensaje cuenta los datos IPFIX desde el inicio de la sesión de transporte. UDP al no poseer confiabilidad y no establecer conexiones, requiere una gestión especial de plantillas.

Los exportadores y colectores IPFIX sobre UDP tienen un retardo de retransmisión de plantillas configurable y un tiempo de vida de plantilla configurable. El primero lo utilizan los exportadores para determinar cada cuanto deben reenviar las plantillas activas y el segundo es usado por los colectores para determinar cada cuanto deben descartar las plantillas que tienen guardadas. Por consecuencia en IPFIX sobre UDP no existen los mensajes de retiro de plantillas. Además estas dos configuraciones requieren calibraciones cuidadosas. Si el tiempo de vida de plantilla se configura muy corto, las plantillas serán borradas prematuramente y si es muy largo impide la reutilización de las plantillas. A su vez, un corto retardo en la retransmisión de las plantillas saturará el ancho de banda de la red y un largo retardo resultarán en pérdidas de datos debido a las pérdidas de las plantillas.

Finalmente, IPFIX tiene la posibilidad de generar archivos según el formato reglamentado en la RFC 5655. Dicho archivo IPFIX es la recopilación de los mensajes enviados utilizando el protocolo, es decir, el archivo será encabezado por las plantillas y estas serán seguidas de los datos. Esta es una forma de protocolo de transporte virtual que posee IPFIX y presenta la ventaja de que los archivos pueden ser almacenados en bases de datos o ser transportados sobre protocolos como HTTP, SMTP, SCP o FTP

VII.4. *APLICACIONES DE MONITORIZACIÓN CON IPFIX*

VII.4.1 *CONTEO Y FACTURACIÓN*

Los procesos de facturación en una red pueden ser basados según el uso ancho de banda (cantidad de paquetes transferidos y recibidos por usuario), la aplicación utilizada (VoIP o servicios web), el momento del día en el que se establecen las comunicaciones, entre otros.

El uso de las herramientas de monitorización con IPFIX, sin saturar el ancho de banda de la red, puede entregar información detallada acerca de la cantidad de paquetes que cruzan por la red, el ancho de banda utilizado por cada usuario y el tipo de aplicación que se está utilizado; facilitando así los procesos de facturación y ayudando a que los datos sean más exactos y de fácil entendimiento para los usuarios [16].

En cuanto a la confiabilidad de los datos, IPFIX ofrece ventajas sobre otros protocolos de exportación de datos IP al utilizar SCTP o TCP en su capa de transporte en lugar de UDP, asegurando que los paquetes con información sensible lleguen a su destino y se pueda realizar una correcta facturación.

VII.4.2. *MONITORIZACIÓN Y ANÁLISIS DE LA SEGURIDAD DE UNA RED*

La monitorización de flujos de red permite al administrador conocer el comportamiento normal de la red y le da la habilidad para establecer diferentes parámetros que le permitan identificar en tiempo real anomalías y problemas de seguridad. A su vez, dichos análisis de flujos pueden ser guardados en bases de datos y ser analizados posteriormente como datos forenses en caso de ciber-ataques [16].

En primer lugar se debe establecer un perfil de actividad normal de la red y luego irlo comparando con diferentes estadísticas del comportamiento actual de la red y dependiendo de diferentes parámetros (cantidad de conexiones, datos transferidos, tipo de tráfico, etc.) determinar si se está realizando un ciber-ataque sobre la red. Algunas de las anomalías que se pueden descubrir utilizando este procedimiento son las siguientes:

- **Escaneo de puertos**
- **Ataques DoS/DDoS**
- **Presencia de un gusano en la red monitorizada**

VII.4.3. *PLANEACIÓN Y CLASIFICACIÓN DE TRÁFICO DE LA RED*

Una de las mayores ventajas que proporciona el protocolo IPFIX a los sistemas de monitorización de redes es la creación y almacenamiento de archivos que recopilan la información de las plantillas y los registros transportados durante la sesión de transporte. Estos archivos pueden ser utilizados para analizar el uso de la red y el comportamiento de los usuarios durante un largo periodo de tiempo, siendo una herramienta importante para una planeación y distribución de recursos más óptima, minimizando costes y maximizando la utilización y rendimiento de la red.

La congestión de la red es crítica para aquellas aplicaciones que consumen altos recursos de red y son sensibles a la latencia y el jitter, tales como video bajo demanda (VoD) o videojuegos en línea. Mediante el conocimiento del uso de la red durante un largo periodo de tiempo pueden encontrarse diferentes parámetros que permiten anticipar momentos de congestión de red como por ejemplo las horas pico de conexión o las aplicaciones más utilizadas en la red, y tomar medidas económicas y eficaces al respecto como corrección de configuraciones o agregación de recursos específicos para la red.

A su vez, la caracterización del tráfico que circula por la red durante largos periodos de tiempo permite identificar el tráfico dándole un perfil según las necesidades del usuario.

VII.4.4. *MAPAS DE RED Y RECONSTRUCCIÓN DE RUTAS DE DATOS*

La información generada y almacenada por el protocolo IPFIX representa la conexión entre dos host particulares que incluyen cuanta información es transferida entre ellos y los tiempos en los que

esto ocurre. Estos mapas de red pueden ser complementados a su vez con información de posición geográfica de los hosts [16].

La reconstrucción de rutas de datos intenta encontrar el camino que toman los flujos desde su fuente origen hasta el destino.

VIII. ARQUITECTURA DE MONITORIZACIÓN MEDIANTE IPFIX

Como idea fundamental del desarrollo del proyecto es la aplicación del protocolo IPFIX en un sistema de monitorización de redes, aprovechando todas sus ventajas para la identificación de ataques de seguridad y el almacenamiento de archivos, que contienen la información de análisis, en bases de datos.

Sin embargo, aunque IPFIX es un protocolo estandarizado en 2007, aún no se encuentra operativo en los diferentes equipos de redes, de hecho es complicado encontrar aplicaciones que lo implementen. Esta falta de desarrollo y motivación por el protocolo es debida a su complejidad de programación proporcionada en su mayor medida por el manejo de las plantillas y por los elementos de información, ya que, como es bien sabido, lograr la compatibilidad de un protocolo con la diversidad de fabricantes de equipos no es una tarea fácil.

Además, muchas de las redes IP aún utilizan el protocolo antecesor, Netflow V9 ya que les provee de las características de monitorización suficientes. Pero como se pudo estudiar en el apartado de IPFIX, este protocolo posee muchas mejoras respecto a Netflow V9, tales como reducción del consumo de ancho de banda, un mejor mecanismo de manejo de plantillas, mayor precisión en los timestamps de los paquetes, y la más importante de todas las mejoras: permite la implementación de los protocolos TCP y SCTP en su capa de transporte y adicionalmente la creación y almacenamiento de archivos de monitorización.

Como programa exportador de paquetes IPFIX se utilizó Nprobe debido a las ventajas y facilidades de configuración que posee, tales como la configuración de plantillas con elementos de información estandarizados y corporativos.

Por otra parte, como recolector de paquetes se utilizó el software de monitorización Netflow analyzer, aprovechando sus herramientas gráficas, interfaz de usuario, generación de paquetes y herramientas de detección de ataques como DoS o escaneo de puertos.

Adicionalmente, se instaló el analizador de protocolos wireshark para tener la posibilidad de analizar los paquetes de datos y plantillas IPFIX intercambiados entre los exportadores y recolectores.

VIII.1. *DESCRIPCIÓN DE LA TESTBED PROPUESTA*

La testbed propuesta para las pruebas del proyecto está conformada por 3 ciber-activos críticos conectados a un switch conformando una LAN, el cual a su vez se encuentra conectado a un router que sirve como puerta de acceso a otras redes (punto de acceso electrónico), por ejemplo una red corporativa. Dichos ciber-activos críticos pueden asemejarse a ordenadores, servidores, centros de control de SCADA, entre otros.

Además en otro lugar se encuentra el sistema de monitorización de redes, en el cual se centraliza todos los datos de monitorización. En la siguiente figura puede observarse la estructura física de red con las direcciones IP de los equipos:

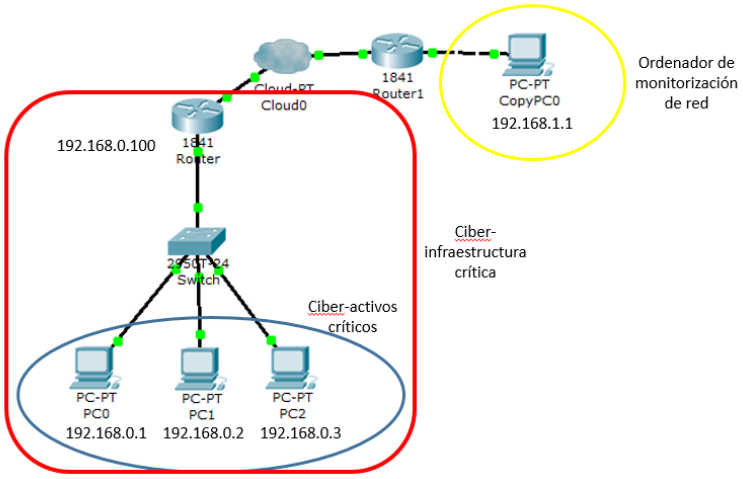


Fig. 9. Red de ciber-activos críticos a monitorizar.

VIII.2. *RED DE MONITORIZACIÓN IDEAL UTILIZANDO EL PROTOCOLO IPFIX*

VIII.2.1 *PLANTEAMIENTO DE LA ARQUITECTURA IDEAL DE LA RED DE MONITORIZACIÓN.*

Para implementar el protocolo IPFIX en la red de ciber-infraestructuras críticas es necesario definir que entidades actuaran como exportadores y colectores. En la figura N, se puede apreciar fácilmente que el único punto de acceso electrónico del perímetro electrónico diseñado es el router con dirección IP 192.168.0.100, el cual sería un perfecto exportador de paquetes IPFIX ya que toda la información que entra a la red de ciber-activos críticos debe pasar por él. Idealmente, el protocolo IPFIX estaría instalado dentro del router.

En cuanto al colector IPFIX, es obvio ubicarlo en el ordenador destinado a recibir los datos de monitorización de la red. Cabe destacar que no siempre es así. Como se estudió en el apartado de arquitectura de IPFIX, se pueden agregar procesos intermedios, los cuales por ejemplo pueden ser de gran utilidad en redes muy grandes donde se monitorice gran número de equipos. Entonces se agregan estos procesos intermedios para recolectar y unificar información IPFIX de ciertas zonas, y luego si enviarla al colector IPFIX central.

Por último, para aprovechar la facilidad que proporciona IPFIX de crear archivos que contengan todos los registros de monitorización, se puede crear una base de datos dentro del ordenador donde se aloja el colector IPFIX o en su defecto agregar un servidor donde se vayan almacenando todos los registros de monitorización y estén disponibles para ser utilizados por otras aplicaciones como se podrá observar en el capítulo 10.

Por lo tanto si se implementara de forma ideal el protocolo IPFIX para la red propuesta anteriormente, se representaría como se muestra a continuación:

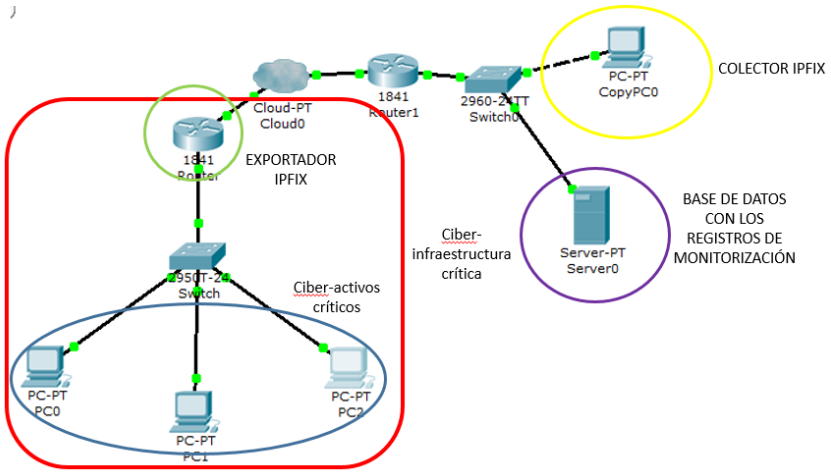


Fig. 10. Red de ciber-activos críticos desde el punto de vista del protocolo IPFIX ideal.

El router con el exportador IPFIX instalado generará las plantillas y los registros de datos de toda la información que entre y salga del perímetro electrónico definido y los enviará utilizando el protocolo SCTP al colector IPFIX que se encuentra instalado en el ordenador de monitorización. Este recibirá todos los datos, los procesará y los presentará en la aplicación de monitorización de red que se esté utilizando. Adicionalmente, generará los archivos respectivos almacenando toda la información de las plantillas y los registros recibidos, y los guardará donde se haya definido, en este caso, un servidor que se encuentra en la misma LAN que el ordenador encargado de la monitorización.

VIII.2.2. *INCONVENIENTES DE DISEÑO*

Hoy por hoy son muy pocos los equipos de red que ya tienen implementado dicho protocolo, por lo que surge el primer y mayor inconveniente de la aplicación de IPFIX directamente sobre un equipo de red. En consecuencia deben ser exploradas nuevas alternativas para la utilización de dicho protocolo.

VIII.3. *IMPLEMENTACIÓN REAL DE IPFIX SOBRE LA RED DE MONITORIZACIÓN PROPUESTA.*

VIII.3.1 *PLANTEAMIENTO DE LA ARQUITECTURA DE LA RED DE MONITORIZACIÓN.*

En el apartado anterior se propuso un modelo para la monitorización de una red de ciber-activos críticos utilizando IPFIX. Sin embargo se encontró la limitación de que este protocolo aún no se encuentra implementado en casi ningún equipo de red por lo que se hace necesario explorar otras opciones.

Una de ellas es la de buscar un software que funcione como sonda, pueda ser instalado en todos los ciber-activos críticos y genere reportes con el formato especificado por IPFIX. Después de probar diferentes exportadores IPFIX (libipfix, nfdump con implementación IPFIX, ntop, monkit, entre otros), se llegó a la elección de utilizar el software nProbe, ya que es fácil de implementar, es un software actualizado y ofrece múltiples opciones para el uso de IPFIX, tales como la selección del protocolo de transporte, selección de elementos de información tanto corporativos como los estandarizados por IANA, entre otros. Las ventajas del software nProbe fueron comentadas en el apartado de herramientas.

Para la elección de un colector IPFIX se realizaron pruebas con diferente software de monitorización (Ntop, Scrutinizer, nfdump, entre otros) y el seleccionado fue Netflow Analyzer, ya que como se estudió en el apartado de herramientas, Netflow Analyzer es un software de monitorización eficiente, con interfaces gráficas muy descriptivas y fáciles de analizar, sistemas de programación de alertas ante incidentes de seguridad y funcionamiento de la red, entre otras aplicaciones. En la siguiente figura se puede observar la nueva organización de la red a monitorizar:

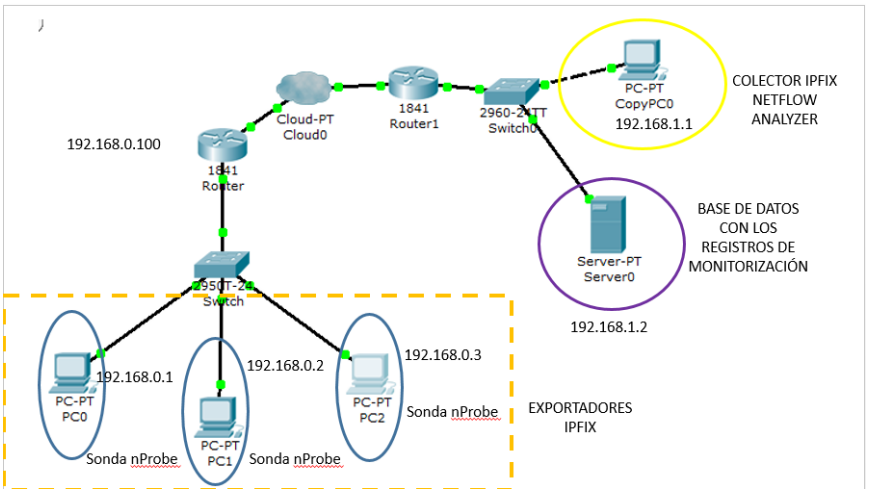


Fig. 11. Red de ciber-activos críticos con la primera propuesta de implementación de IPFIX.

En esta propuesta se buscó instalar nProbe en cada uno de los ciber-activos críticos. Sin embargo, el planteamiento anterior al ser aplicado a una red real de ciber-infraestructuras críticas es poco escalable y de limitada aplicación. La escalabilidad se ve reducida en proporción al número

de sondas que se posean instaladas en el sistema; entre mayor sondas se posea mayor será la información de monitorización generada y se consumirá mayor ancho de banda. A su vez, el esquema anterior posee una limitada aplicación debido a que supone que los ciber-activos críticos son sólo ordenadores con ciertas capacidades de procesamiento y con un sistema operativo, ya sea Linux, Windows o MacOSX. Como se estudió en el apartado de infraestructuras críticas, existe una gran variedad de clases de ciber-activos críticos en las que se incluyen sensores, PLC's, dispositivos móviles y otro tipo de equipos industriales o de uso corporativo que no pueden soportar la instalación de un software como nProbe. De ahí surge la necesidad de plantear una forma de monitorizar el perímetro electrónico propuesto centralizando la información de monitorización en un solo punto, utilizando equipos de red que no soportan IPFIX.

Como solución se plantea la siguiente arquitectura de red:

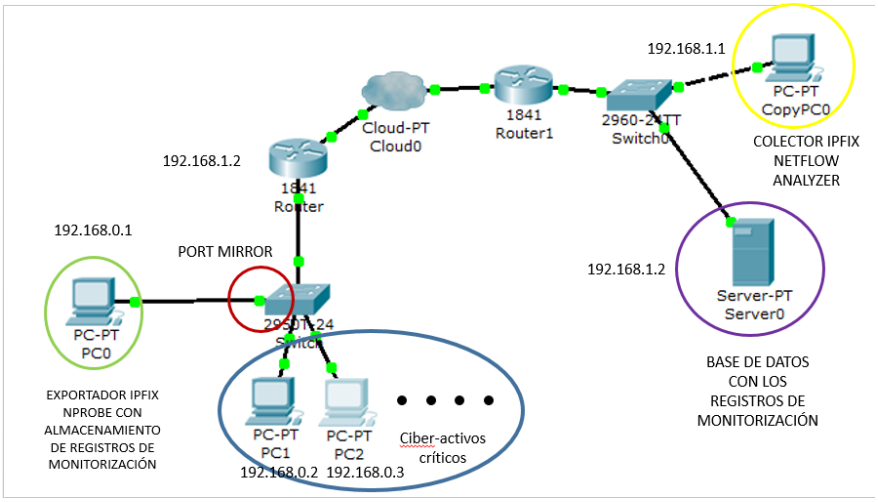


Fig. 12. Red de ciber-activos críticos con la implementación de IPFIX utilizando la opción "port mirror" del switch.

En este nuevo diseño se implementó la configuración de "Port mirror", la cual es implementada por los switches. Entonces se instaló la sonda nProbe en un ordenador que se especificó como exportador IPFIX y el puerto por el que se conectó al switch se habilitó como "port mirror", es decir que todo el tráfico que circula por la red era copiado al "port mirror" y enviado al host exportador IPFIX, el cual recopilaba toda la información, generaba los registros IPFIX y los enviaba al colector IPFIX. A su vez también los almacenaba en memoria en archivos de texto. Con lo anterior se logró monitorizar todo el tráfico que circulaba por el perímetro electrónico propuesto.

VIII.3.1 *PLANTEAMIENTO DE LA ARQUITECTURA DE RED DE MONITORIZACIÓN.*

En primer lugar se instaló nProbe en el host asignado como exportador IPFIX y se configuraron sus parámetros con la siguiente línea de programación:

"*nprobe –i eth0 –V 10 –n udp://192.168.1.1:9996 –b 1 –a 1 –Q 2 –T "%IPV4_SRC_ADDR %IPV4_DST_ADDR %IPV4_NEXT_HOP %OUT_PKTS %OUT_BYTES %FIRST_SWITCHED*

%LAST_SWITCHED %L4_SRC_PORT %L4_DST_PORT %PROTOCOL" –P C:/Registros –D t –dont-nest-dump-dirs"

La explicación de cada uno de los comandos escritos anteriormente puede ser encontrada en el anexo A de este documento. El tráfico que circulaba por la red mientras esta era monitorizada se conformaba por:

- Tráfico ICMP entre los diferentes ciber-activos.
- Para generar tráfico HTTP, se abrió el servidor HTTP interno del router con dirección IP 192.168.0.100, al cual accedía el ciber-activo crítico con dirección IP 192.168.0.1.
- Tráfico SNMP y otro tipo de tráfico propio de los equipos cisco.
- Tráfico SIP.

En cuanto al colector IPFIX, Netflow Analyzer no necesita una configuración previa; basta con ejecutar el software, configurar el puerto por el que se recibirá los registros de IPFIX (puerto 9996) y el programa mismo se pone a la escucha de los paquetes de monitorización.

El switch donde se conectan los ciber-activos críticos se configuró con los siguientes comandos:

No monitor sesión 1

Monitor sesión 1 source interface FastEthernet 0/x

Monitor sesión 1 destination interface FastEthernet 0/y (encapsulation dot1q ingress)

End

Es decir, primero se elimina cualquier configuración de la sesión de monitorización seleccionada, en este caso sesión de monitorización 1. Luego se establece los puertos que se monitorizaran (x) y el puerto a donde irán los datos de monitorización (y), en este caso la copia exacta de todos los paquetes que circulan por el switch.

Una vez configurados los equipos con las direcciones IP correspondientes, con el exportador y el colector IPFIX funcionando, se procedió a poner en marcha el intercambio de información para generar el tráfico de paquetes antes descrito. Entonces la red estuvo en funcionamiento y todos los equipos (ciber-activos) con nProbe instalado primero enviaron al colector la plantilla IPFIX que iban a utilizar, que en este caso es la plantilla que se configuró anteriormente y luego comenzaron el envío de los registros de monitorización. Además cada 5 minutos los equipos fueron generando archivos de texto con la información de los registros de monitorización, los fueron almacenando en el directorio configurado y los fueron organizando en carpetas según el año, mes, día y hora del registro.

A su vez, el colector IPFIX (Netflow Analyzer) recibió las plantillas y luego los registros de monitorización y realizó un análisis de datos y los representó en diferentes gráficos.

VIII.3.2 *OBSERVACIÓN DEL TRÁFICO IPFIX.*

En el ordenador con dirección IP 192.168.0.1 se instaló el software Wireshark con el fin de observar el tráfico generado por el protocolo IPFIX.

VIII.3.2.1 *OBSERVACIÓN DE UNA PLANTILLA IPFIX GENERADA.*

Como era de esperar el ordenador envió en primer lugar la plantilla IPFIX y luego los registros de monitorización. A continuación se puede observar el paquete IPFIX capturado que contiene la plantilla de datos:

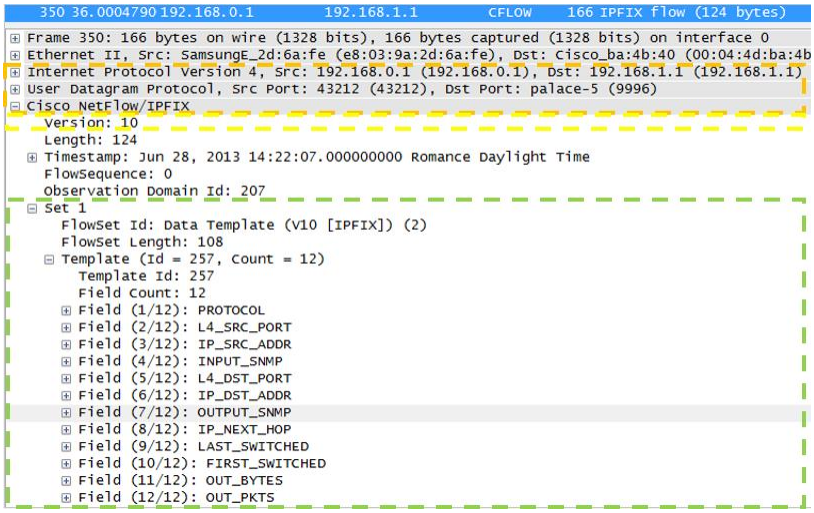


Fig. 13. Paquete con la plantilla IPFIX generada por el exportador y capturada por el Wireshark.

En la anterior figura se destacan 3 zonas. En la zona naranja se puede observar que la plantilla generada fue enviada desde la dirección IP 192.168.0.1 a la dirección IP 192.168.1.1, es decir desde el exportador IPFIX 1 al colector IPFIX. Además se utilizó el protocolo de transporte UDP con el puerto destino por defecto de IPFIX 9996 conocido también con el nombre "Palace-5". Lo más importante de esta parte es que el software Wireshark identifica la trama como el protocolo IPFIX, con lo que se confirma que nProbe genera correctamente tramas IPFIX. Aunque es de destacar que erróneamente Wireshark asocia IPFIX a Cisco. Como se estudió anteriormente, IPFIX es una evolución a partir de Netflow v9 pero es un protocolo libre, por lo que Cisco no es propietario ni desarrollador de dicho protocolo.

En la zona amarilla, puede encontrarse la versión de protocolo configurada en nProbe (versión 10).

Por otra parte, en la zona verde se puede observar la estructura que posee el paquete de IPFIX, es decir, un set con los parámetros que se describieron en el apartado de IPFIX de este trabajo, y

dentro de este la plantilla con cada uno de los elementos de información que se configuraron en el nProbe.

VIII.3.2.2 *OBSERVACIÓN DE UN PAQUETE IPFIX GENERADO.*

Se capturó uno de los paquetes que contiene un registro IPFIX y se analizó su contenido utilizando Wireshark. En la figura siguiente se puede observar el encabezado y uno de los flujos del set 1 de dicho paquete:

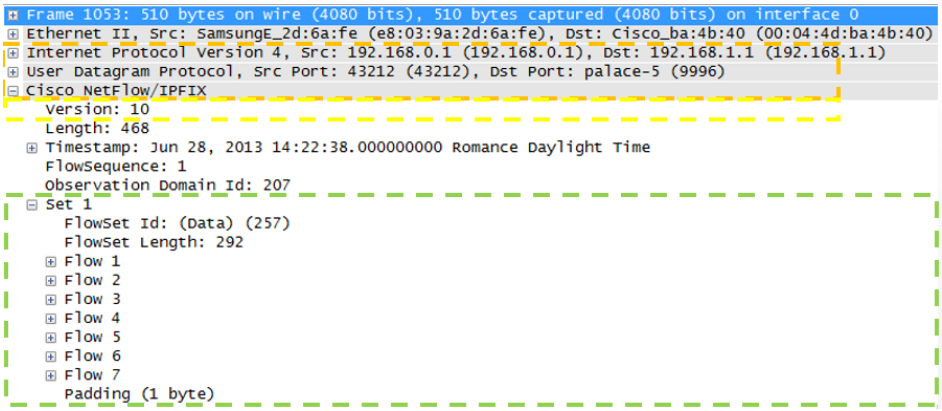


Fig. 14. Paquete con registros IPFIX generado por el exportador y capturado por Wireshark.

En la zona naranja se puede observar que el paquete de registros posee el mismo encabezado que el paquete que contiene la plantilla IPFIX. Sin embargo, en la zona verde se puede observar que cambia el contenido del Set 1. Ahora en vez de aparecer la plantilla aparecen los diferentes registros de monitorización; en la figura anterior sólo se puede observar uno de ellos. A continuación se puede observar un zoom en uno de los registros de configuración:

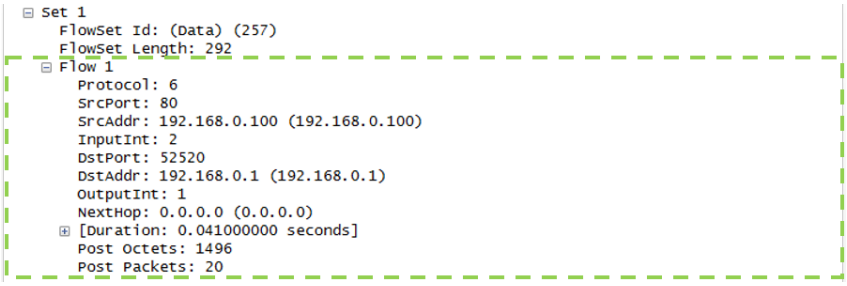


Fig. 15. Contenido de uno de los registros IPFIX generado por el exportador y capturado por Wireshark.

Este registro posee todos los datos de los elementos de información, por ejemplo, por el número de protocolo se puede identificar que es un tráfico TCP y por el puerto fuente 80 y las direcciones IP origen y destino, se puede identificar que el registro corresponde a los paquetes HTTP generados entre el ciber-activo donde se capturó el paquete y el servidor HTTP activado en el router con dirección IP 192.168.0.100.

VIII.3.2.3 *OBSERVACIÓN DE ARCHIVO DE TEXTO GENERADO CON REGISTROS IPFIX.*

De los archivos de texto generados por nProbe se extrajo la plantilla y el primer registro de configuración. El archivo completo se puede observar en el anexo C.

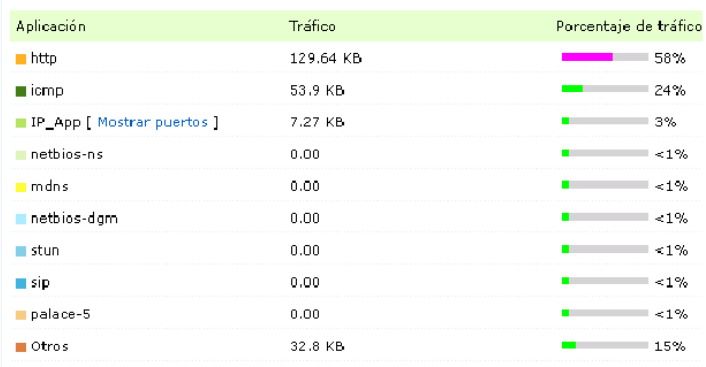
```
"IPV4_SRC_ADDR|IPV4_DST_ADDR|IPV4_NEXT_HOP|OUT_PKTS|OUT_BYTES|FIRST_SWITCHED|LAST_SWITCHED|L4_SRC_PORT|L4_DST_PORT|PROTOCOL

192.168.0.1|192.168.1.1|0.0.0.0|1|2|1372422399.136577|1372422399.136606|43212|9996|17"
```

En la primera parte se puede observar la plantilla configurada en nProbe especificando el nombre de cada uno de los elementos de información. En seguida se encuentra de forma organizada los datos tomados para generar el registro. En este caso el registro describe unos paquetes que viajan desde el exportador IPFIX hacia el colector IPFIX, con protocolo UDP (17) y utilizando el puerto 9996. Como se puede intuir, corresponde a un registro de monitorización de los paquetes IPFIX que se han enviado desde el exportador hacia el colector.

Sin embargo estos archivos de texto son almacenados en el host que contiene la sonda IPFIX, por lo que es necesario implementar alguna herramienta que se encargue de enviar los archivos hasta el host monitor o hasta el servidor con la base de datos de los registros de monitorización.

IX.3.2.4 *ANÁLISIS DE LOS GRÁFICOS OBTENIDOS POR NETFLOW ANALYZER.*

Luego de poner en marcha la red, el software de monitorización Netflow Analyzer comenzó a recibir los paquetes de monitorización enviados por el exportador IPFIX. Luego el software analizó los datos y los represento en varias gráficas. A continuación se observa la primera de ellas:



Aplicación	Tráfico	Porcentaje de tráfico
http	129.64 KB	58%
icmp	53.9 KB	24%
IP_App [Mostrar puertos]	7.27 KB	3%
netbios-ns	0.00	<1%
mdns	0.00	<1%
netbios-dgm	0.00	<1%
stun	0.00	<1%
sip	0.00	<1%
palace-5	0.00	<1%
Otros	32.8 KB	15%

Fig. 16. Porcentaje de tráfico generado por cada aplicación.

La anterior figura muestra los resultados del valor de tráfico y el porcentaje de tráfico que consume cada una de las aplicaciones utilizadas en la red. Entre estas se encuentran los protocolos HTTP, ICMP, SIP, palace-5 (identificación del tráfico IPFIX) y otros protocolos propios de los equipos utilizados. De las dos anteriores gráficas se puede observar que el tráfico de mayor consumo fue el de HTTP y que la exportación de los registros IPFIX consume menos del 1% del tráfico total que circula por la red.

Principal destino

Destino	Tráfico	Porcentaje de tráfico
192.168.0.100	96.84 KB	43%
192.168.0.1	45.38 KB	20%
192.168.1.1	14.39 KB	6%
192.168.0.2	13.46 KB	6%
192.168.0.3	13.45 KB	6%
0.0.0.0	7.27 KB	3%
82.129.27.63	0.00	<1%
11.0.0.3	0.00	<1%
192.168.0.255	0.00	<1%
224.0.0.251	0.00	<1%
Otros	32.8 KB	15%

Fig. 17. Cantidad de tráfico dirigido a los diferentes destinos.

En esta tabla, se puede ver cuánto tráfico es encaminado a cada uno de los destinos. Como es de esperar el mayor porcentaje de tráfico tiene como destino el servidor WEB interno del router.

Principal origen

Origen	Tráfico	Porcentaje de tráfico
192.168.0.1	109.37 KB	49%
192.168.0.100	32.79 KB	15%
192.168.1.1	14.39 KB	6%
192.168.0.3	13.51 KB	6%
192.168.0.2	13.46 KB	6%
0.0.0.0	7.27 KB	3%
Otros	32.8 KB	15%

Fig. 18. Principal origen del tráfico que circula por la red.

En esta figura, se muestra que el principal origen de los paquetes que circulan por la red es la dirección IP del host con la sonda IPFIX, ya que éste es el que hacia las peticiones TCP para conectarse con el servidor interno y además generaba los reportes IPFIX.

Conversación de máximo consumo

Origen	Destino	Aplicación	Tráfico
192.168.0.1	192.168.0.100	http	96.84 KB
192.168.0.100	192.168.0.1	http	32.79 KB
0.0.0.0	0.0.0.0	IP_App [Mostrar puertos]	7.27 KB
192.168.1.1	192.168.0.3	icmp	7.19 KB
192.168.1.1	192.168.0.2	icmp	7.19 KB
192.168.0.2	192.168.1.1	icmp	7.19 KB
192.168.0.3	192.168.1.1	icmp	7.19 KB
192.168.0.3	192.168.0.1	icmp	6.32 KB
192.168.0.1	192.168.0.2	icmp	6.26 KB
192.168.0.2	192.168.0.1	icmp	6.26 KB
Otros	Otros	Otros	39.06 KB

Fig. 19. Conversación de máximo consumo en la red.

En la anterior tabla, se observa que la conversación con mayor tráfico fue el de la conversación HTTP entre el servidor instalado en el router y el host que se intentaba conectar con él, es decir, el host con dirección IP 192.168.0.1.

VIII.4. *APLICACIÓN DE IPFIX PARA LA DETECCIÓN DE ATAQUES.*

Con el fin de mostrar la funcionalidad de una red de monitorización utilizando el protocolo IPFIX, se plantearon dos ciber-ataques: un ataque de denegación de servicio TCP SYN flooding y un escaneo de puertos. Para ello se utilizaron dos herramientas, HPING3 y NMAP, respectivamente. A continuación se puede observar el planteamiento del ataque:

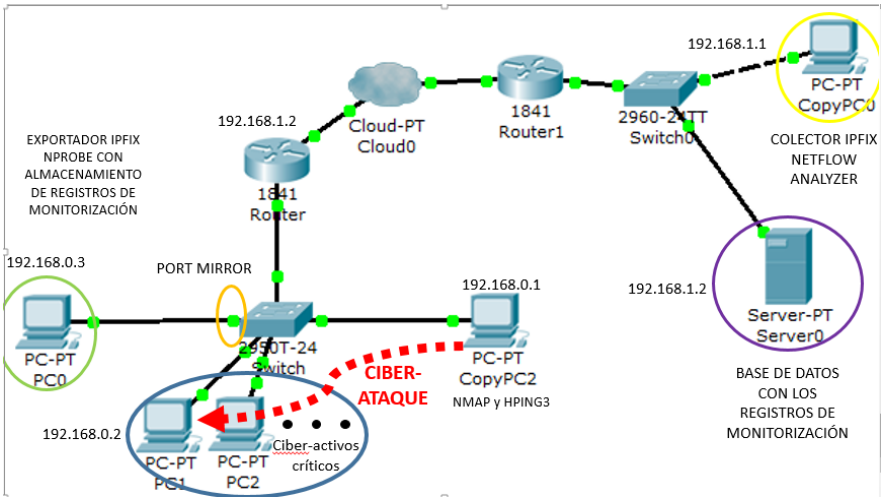


Fig. 20. Ciber-ataque en la red monitorizada usando el protocolo IPFIX.

En esta figura se encuentra la red propuesta anteriormente, con la diferencia de que ahora existe un host infectado (dirección IP 192.168.0.1) que se encuentra realizando ciber-ataques al ciber-activo críticos con dirección IP 192.168.0.2, del tipo TCP Syn flooding y escaneo de puertos. Ahora el host que posee instalado nProbe tiene la dirección IP 192.168.0.3.

VIII.4.1 *DETECCIÓN DE ATAQUES DE ESCANEO DE PUERTOS UTILIZANDO IPFIX.*

Se instaló la herramienta Nmap en el host con dirección IP 192.168.0.1 y se configuró con la siguiente instrucción: “nmap –sA –T4 192.168.0.2”. Este comando ordena al host infectado a usar mensajes de ACK para lograr que el ciber-activo con dirección IP 192.168.0.2 responda y así determinar que puertos están abiertos y descubrir debilidades en la seguridad del host.

Entonces, se puso en funcionamiento la red propuesta para la prueba de seguridad y en un determinado momento se ordenó al host infectado realizar un escaneo de puertos al ciber-activo crítico, como se mencionó anteriormente. Para el análisis del ataque se supondrá que el monitor de red desconoce la circunstancia del ataque y por lo tanto su origen, su víctima y las características

del mismo, con lo que se intentará averiguar todas estas incógnitas a partir de la información transportada por IPFIX y colectada por Netflow Analyzer.

Revisando el análisis realizado por el software Netflow Analyzer, estos fueron los resultados:

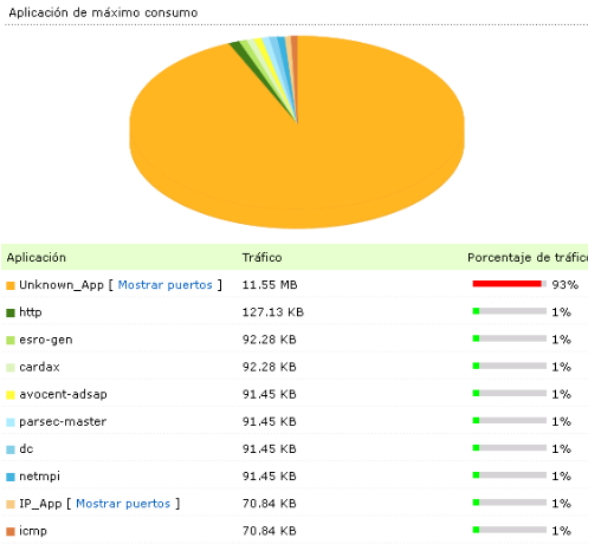


Fig. 21. Aplicación de máximo consumo cuando existe un ciber-ataque de escaneo de puertos en búsqueda de vulnerabilidades.

En la anterior figura se muestra el porcentaje de tráfico circulante por la red de cada protocolo utilizado. Sin embargo, se destaca que existe un tráfico sin identificación que conformó el 93% del tráfico total de la red, lo cual es la primera bandera de alerta. Luego se analizó la gráfica de los protocolos de transporte utilizados en la red en el tiempo de monitorización:

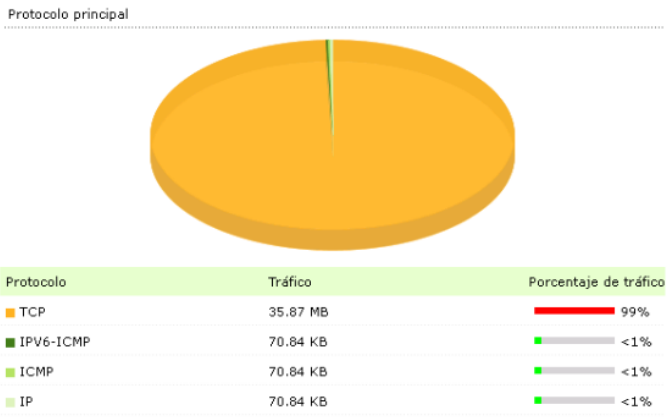


Fig. 22. Protocolos de transporte utilizados en la red mientras se realizó el ciber-ataque.

En esta gráfica se puede observar que el tráfico desconocido de 35.87 MB detectado anteriormente utilizó el protocolo TCP como capa de transporte. Por lo que por ejemplo, ya se puede sospechar de un ataque de denegación de servicio o un escaneo de puerto.

En continuación con la detección del ataque se observaron los datos de los principales orígenes, los principales destinos, y la conversación de máximo consumo, dependiendo de la cantidad de tráfico recibido o generado por cada uno de los host.

Principal origen

Origen	Tráfico	Porcentaje de tráfico
192.168.0.1	35.8 MB	99%
0.0.0.0	141.68 KB	<1%
192.168.0.100	70.84 KB	<1%
192.168.0.3	70.84 KB	<1%

Principal destino

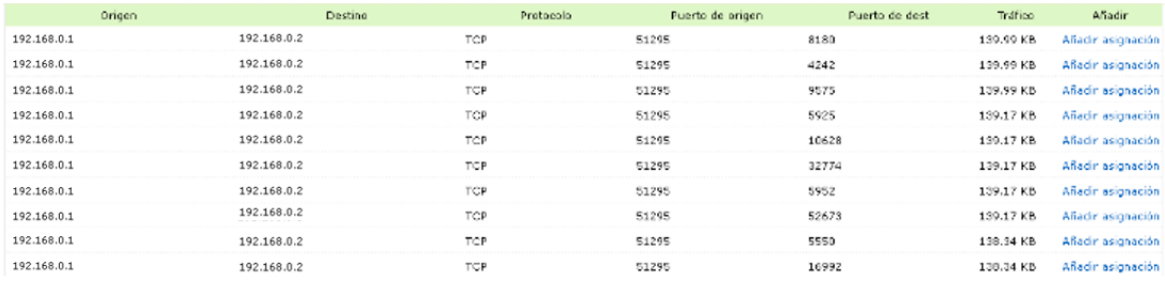
Destino	Tráfico	Porcentaje de tráfico
192.168.0.2	35.8 MB	99%
0.0.0.0	141.68 KB	<1%
54.246.145.162	70.84 KB	<1%
192.168.0.1	70.84 KB	<1%

Conversación de máximo consumo

Origen	Destino	Aplicación	Tráfico
192.168.0.1	192.168.0.2	Unknown_App [Mostrar puertos]	11.55 MB

Fig. 23. Principales orígenes y destinos de los tráficos que circularon por la red atacada.

En continuación, estudiando las direcciones IP origen y destino con mayor tráfico generado o recibido respecitvamente, se puede determinar fácilmente que el tráfico fue generado por la dirección IP 192.168.0.1 y el host que recibió los paquetes fue la dirección IP 192.168.0.2. Con esto ya se concreta un funcionamiento incorrecto de la red, y por la gran cantidad de tráfico transportado, este incidente directamente puede ser asociado con un ciber-ataque, por lo que se debe entrar más a fondo en el análisis de dicho tráfico estudiando el puerto utilizado por el tráfico desconocido para establecer la comunicación. Para ello, se utiliza la herramienta de Netflow Analyzer que permite ver los puertos utilizados por cada tráfico. A continuación se puede observar parte de la tabla del uso de puertos detectado por Netflow Analyzer a partir de la información entregada con el protocolo IPFIX:



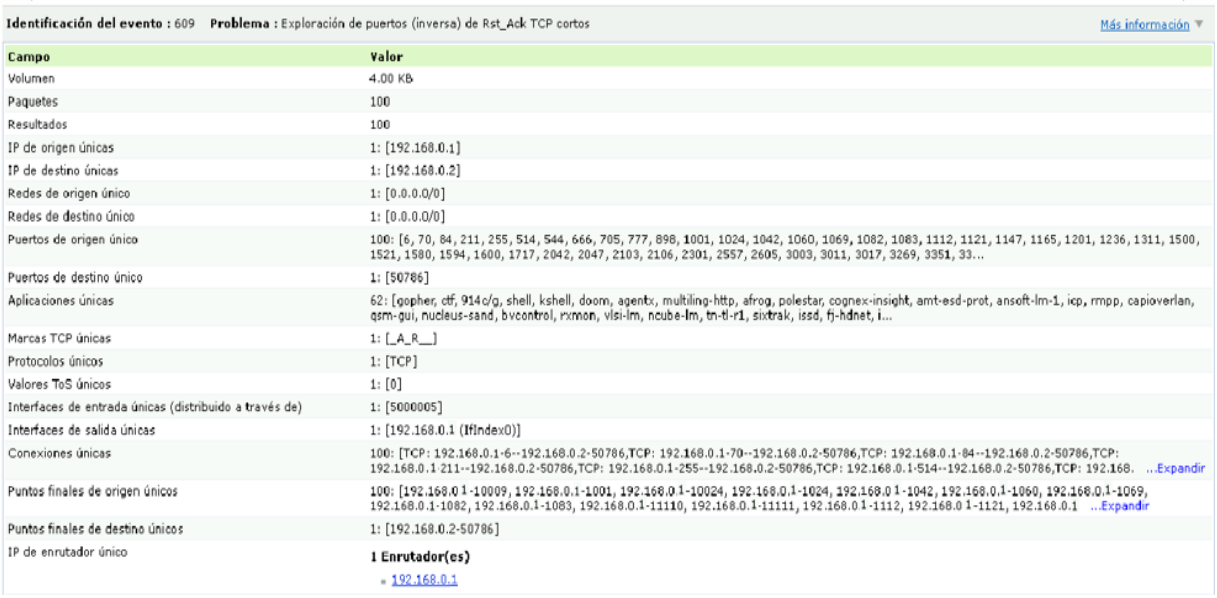
Origen	Destino	Protocolo	Puerto de origen	Puerto de dest	Tráfico	Añadir
192.168.0.1	192.168.0.2	TCP	51295	8180	139.99 KB	Añadir asignación
192.168.0.1	192.168.0.2	TCP	51295	4242	139.99 KB	Añadir asignación
192.168.0.1	192.168.0.2	TCP	51295	9575	139.99 KB	Añadir asignación
192.168.0.1	192.168.0.2	TCP	51295	5925	139.17 KB	Añadir asignación
192.168.0.1	192.168.0.2	TCP	51295	10628	139.17 KB	Añadir asignación
192.168.0.1	192.168.0.2	TCP	51295	32774	139.17 KB	Añadir asignación
192.168.0.1	192.168.0.2	TCP	51295	5952	139.17 KB	Añadir asignación
192.168.0.1	192.168.0.2	TCP	51295	52673	139.17 KB	Añadir asignación
192.168.0.1	192.168.0.2	TCP	51295	5550	138.34 KB	Añadir asignación
192.168.0.1	192.168.0.2	TCP	51295	16992	138.34 KB	Añadir asignación

Fig.24. Parte de la tabla de puertos utilizados por el tráfico desconocido detectado en la red monitorizada.

En la tabla generada por Netflow Analyzer antes comentada, se puede ver claramente que el la dirección IP 192.168.0.1 utilizando el protocolo TCP y el puerto 51295 envió múltiples paquetes de aproximadamente el mismo tamaño a una gran cantidad de puertos de la dirección IP 192.168.0.2.

Con toda la información obtenida anteriormente utilizando el protocolo IPFIX y el colector Netflow Analyzer se puede concluir fácilmente que el host 192.168.0.1 realizó un escaneo de puertos al ciber-activo crítico con dirección IP 192.168.0.2.

Por otra parte, el análisis realizado anteriormente paso a paso es imposible de realizar por un monitor de red cada vez que se presenta una situación sospechosa. El software Netflow Analyzer posee herramientas para la generación de alertas, por lo que puede realizar las operaciones anteriores automáticamente y avisar en que momento se producen ataques, el tipo de ataque y los origenes y destinos. A continuación se puede observar una de las diferentes alertas generadas por Netflow Analyzer a partir de la información IPFIX colectada:



Identificación del evento : 609 **Problema :** Exploración de puertos (inversa) de Rst_Ack TCP cortos — Más información

Campo	Valor
Volumen	4.00 KB
Paquetes	100
Resultados	100
IP de origen únicas	1: [192.168.0.1]
IP de destino únicas	1: [192.168.0.2]
Redes de origen único	1: [0.0.0.0/0]
Redes de destino único	1: [0.0.0.0/0]
Puertos de origen único	100: [6, 70, 84, 211, 255, 514, 544, 666, 705, 777, 898, 1001, 1024, 1042, 1060, 1069, 1082, 1083, 1112, 1121, 1147, 1165, 1201, 1236, 1311, 1500, 1521, 1580, 1594, 1600, 1717, 2042, 2047, 2103, 2106, 2301, 2557, 2605, 3003, 3011, 3017, 3269, 3351, 33...
Puertos de destino único	1: [50786]
Aplicaciones únicas	62: [gopher, ctf, 914c/g, shell, kshell, doom, agentx, multiling-http, afrog, polestar, cognex-insight, amt-esd-prot, ansoft-lm-1, icp, rmpp, capioverlan, qsm-gui, nucleus-sand, bvcontrol, rxmon, vlsi-lm, ncube-lm, tn-tl-r1, sixtrak, issd, fj-hdnet, i...
Marcas TCP únicas	1: [_A_R__]
Protocolos únicos	1: [TCP]
Valores ToS únicos	1: [0]
Interfaces de entrada únicas (distribuido a través de)	1: [5000005]
Interfaces de salida únicas	1: [192.168.0.1 (IfIndex0)]
Conexiones únicas	100: [TCP: 192.168.0.1-6--192.168.0.2-50786,TCP: 192.168.0.1-70--192.168.0.2-50786,TCP: 192.168.0.1-84--192.168.0.2-50786,TCP: 192.168.0.1-211--192.168.0.2-50786,TCP: 192.168.0.1-255--192.168.0.2-50786,TCP: 192.168.0.1-514--192.168.0.2-50786,TCP: 192.168. ...Expandir
Puntos finales de origen únicos	100: [192.168.0.1-10009, 192.168.0.1-1001, 192.168.0.1-10024, 192.168.0.1-1024, 192.168.0.1-1042, 192.168.0.1-1060, 192.168.0.1-1069, 192.168.0.1-1082, 192.168.0.1-1083, 192.168.0.1-11110, 192.168.0.1-11111, 192.168.0.1-1112, 192.168.0.1-1121, 192.168.0.1 ...Expandir
Puntos finales de destino únicos	1: [192.168.0.2-50786]
IP de enrutador único	**1 Enrutador(es)** • 192.168.0.1

Fig. 25. Alerta generada por Netflow Analyzer a partir de la información IPFIX que colecta.

En esta alerta generada automáticamente por Netflow Analyzer se especifica el tipo de ataque que sucedió (Exploración de puertos (inversa) de Rst_Ack TCP cortos), las direcciones IP atacante y víctima, los protocolos y los puertos utilizados, entre otros datos de interés.

Con los resultados anteriores se demostró una de las posibles implementaciones de IPFIX para el sector de monitorización y seguridad de redes, ya que mediante este protocolo se pudo descubrir un ataque del tipo escaneo de puertos.

VIII.4.2. *DETECCIÓN DE ATAQUES DOS TCP SYN FLOODING UTILIZANDO IPFIX.*

Se instaló la herramienta HPING3 en el host con dirección IP 192.168.0.1 y se configuró con la siguiente instrucción: “hping3 –p 80 –S --faster 192.168.0.2”. Este comando ordena al host infectado a generar paquetes TCP con el flag SYN activado y enviarlos a máxima velocidad al host 192.168.0.2.

Se puso en funcionamiento la red propuesta para la prueba de seguridad y en un determinado momento se ordenó al host infectado realizar un ataque de denegación de servicio TCP Syn flooding a un ciber-activo crítico, como se mencionó anteriormente. Para el análisis del ataque se supondrá que el monitor de red desconoce la circunstancia del ataque y por lo tanto su origen, su víctima y las características del mismo, con lo que se intentará averiguar todas estas incógnitas a partir de la información transportada por IPFIX y colectada por Netflow Analyzer.

Revisando el análisis realizado por el software Netflow Analyzer, estos fueron los resultados:

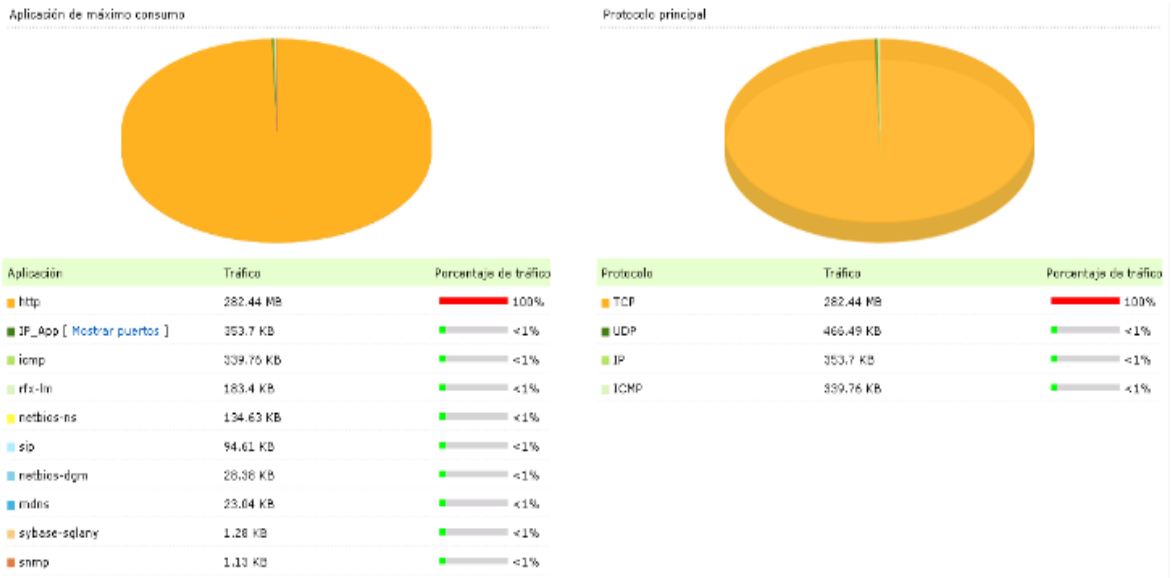


Fig. 26. Aplicación de máximo consumo y protocolo de transporte principal cuando existe un ciber-ataque del tipo TCP syn flooding.

En la anterior figura se muestra el porcentaje de tráfico circulante por la red de cada protocolo utilizado. Sin embargo, se destaca que existe un tráfico HTTP de 282.44 MB que conformó el 99% del tráfico total de la red, lo cual es un tráfico excesivo para la cantidad de datos transportada normalmente. Además, como es de esperarse, el protocolo TCP fue el principal protocolo de transporte utilizado. Entonces es necesario observar el tiempo por el que se realizó el ataque:

* La utilización se calcula con la velocidad de los enlaces de 1.0 Mbps

Categoría	Total	Máx.	Minuto	Prom.	Desviación estándar	95º percentile
ENTRANTE	278.83 MB	1053.12% ⚠	0.00%	60.95%	201.74	491.81%
SALIENTE	0.00	-	-	-	0.00	-

Fig. 27. Gráfica de tráfico del ciber-activo crítico atacado con intervalos de 5 minutos.

En esta gráfica se puede observar que existieron 2 momentos en los que se generaron picos de tráfico muy altos en sentido entrante y nada de tráfico en sentido saliente, lo que demuestra que la conexión del ciber-activo crítico estuvo saturada debido a los inesperados paquetes transmitidos sólo en sentido entrante. Con los datos anteriores se intuye directamente que no se trató de un fallo de sistema sino de un ciber-ataque que generó en ciertos momentos altas ráfagas de paquetes a sólo un ciber-activo crítico utilizando el protocolo TCP, saturando la conexión de dicho dispositivo, quitándole la posibilidad de que este transmitiera información. En otras palabras, se trata de un ataque de denegación de servicio del tipo TCP syn flooding.

Como comprobación del resultado se puede observar una de las diferentes alertas generadas por Netflow Analyzer a partir de la información IPFIX colectada:

Identificación del evento : 310 **Problema :** Desbordamiento entrante de sintaxis TCP Más información

Campo	Valor
Volumen	1.40 MB
Paquetes	3115
Resultados	100
IP de origen únicas	1: [192.168.0.1]
IP de destino únicas	1: [192.168.0.2]
Redes de origen único	1: [0.0.0.0/0]
Redes de destino único	1: [0.0.0.0/0]
Puertos de origen único	100: [2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2646, 2648, 2649, 2650, 2651, 2652, 2653, 2654,...
Puertos de destino único	1: [80]
Aplicaciones únicas	1: [http]
Marcas TCP únicas	1: [____S_]
Protocolos únicos	1: [TCP]
Valores ToS únicos	1: [0]
Interfaces de entrada únicas (distribuido a través de)	1: [5000005]
Interfaces de salida únicas	1: [192.168.0.3 (IfIndex0)]
Conexiones únicas	100: [TCP: 192.168.0.1-2481--192.168.0.2-80,TCP: 192.168.0.1-2482--192.168.0.2-80,TCP: 192.168.0.1-2483--192.168.0.2-80,TCP: 192.168.0.1-2484--192.168.0.2-80,TCP: 192.168.0.1-2485--192.168.0.2-80,TCP: 192.168.0.1-2486--192.168.0.2-80,TCP: 192.168.0.1-2487 ...Expandir
Puntos finales de origen únicos	100: [192.168.0.1-2481, 192.168.0.1-2482, 192.168.0.1-2483, 192.168.0.1-2484, 192.168.0.1-2485, 192.168.0.1-2486, 192.168.0.1-2487, 192.168.0.1-2488, 192.168.0.1-2573, 192.168.0.1-2574, 192.168.0.1-2575, 192.168.0.1-2576, 192.168.0.1-2577, 192.168.0.1-257 ...Expandir
Puntos finales de destino únicos	1: [192.168.0.2-80]

Fig. 28. Alerta de ataque DoS generada por Netflow Analyzer a partir de la información IPFIX que colecta.

En esta alerta generada automáticamente por Netflow Analyzer se especifica el tipo de ataque que sucedió (Desbordamiento entrante de sintaxis TCP), las direcciones IP atacante y víctima, los protocolos y los puertos utilizados, entre otros datos de interés. Profundizando, se puede observar como el atacante envió los paquetes TCP con el flag syn activado desde múltiples puertos hacia únicamente el puerto 80 del ciber-activo crítico víctima.

Con los resultados anteriores se demostró una de las posibles implementaciones de IPFIX para el sector de monitorización y seguridad de redes, ya que mediante este protocolo se pudo descubrir un ataques DoS del tipo TCP syn flooding.

IX. REGISTRO DE DATOS EN EL SOS: "IPFIX AS A SENSOR"

En el apartado anterior se utilizaron los registros enviados por el exportador IPFIX para la monitorización y seguridad de la infraestructura crítica propuesta. Sin embargo, IPFIX proporciona una ventaja adicional: la creación de ficheros. El software nProbe utilizado permite generar estos ficheros con formato binario o de texto. Para este caso se utilizará el formato texto. Un ejemplo de este formato puede ser encontrado en el anexo C de este documento. En este apartado se presentará el término "IPFIX as a sensor" y la forma en la que se realizó el parceo del archivo de texto a un archivo XML en terminos de lenguaje SML para su posterior envío a un SOS, permitiendo que los diferentes registros queden a disposición de otras aplicaciones o usuarios.

IX.1. *SWE: SML y SOS*

SWE (Sensor Web Enablement) es una iniciativa creada por OGC (Open Geospacial Consortium) que se encarga de desarrollar estándares para acceder y controlar sensores y redes de sensores a través de internet. Lo anterior promueve la interoperabilidad y la definición de los distintos servicios y componentes. SWE posee varias especificaciones según su modelo, pero para el desarrollo del proyecto interesan dos en concreto [16]:

- SML: es un lenguaje que describe un formato común para la descripción de sensores y sistemas de sensores. El principal objetivo es el de que los sensores puedan ser modelados como procesos.
- SOS: Es el principal modelo de servicio del framework SWE y su objetivo es el de dar acceso a las observaciones de los sensores y sistemas de sensores de forma estándar. SOS cumple con la especificación SML para modelar sensores y sistemas de sensores.

En el presente proyecto se utilizaron dos plantillas XML con lenguaje SML para poder registrar el sensor "IPFIX" y poder ingresar los registros de datos, cuyos nombres son *RegisterSensor.xml* y *InsertObservation_GenericObs_Derwent.xml* respectivamente. Las plantillas generadas pueden ser observadas en el anexo D de este documento.

IX.2. *PROGRAMA DE PARCEO DE TXT A XML*

Utilizando la herramienta eclipse se desarrolló un programa en Java que realiza la lectura de los archivos TXT generados por nProbe, almacena los valores en vectores, luego

reemplaza los valores en las plantillas XML comentadas en el apartado anterior y finalmente genera los archivos XML con la especificación SML.

Fig. 29. Esquema general de funcionamiento del programa desarrollado en lenguaje JAVA para el parceo de archivos TXT a XML.

Entonces, al final de la ejecución del programa, se generan las dos plantillas XML *RegisterSensor.xml* e *InsertObservation_GenericObs_Derwent.xml* con los datos y listas para ser enviadas al SOS. El esquema de la plantilla *RegisterSensor.xml* se puede ver a continuación:

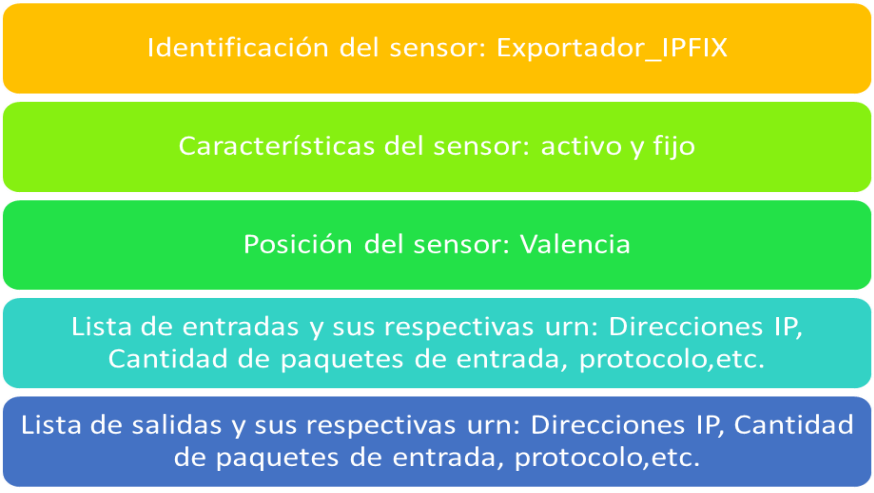


Fig. 30. Esquema general de la plantilla *RegisterSensor.xml*.

El esquema de la plantilla *InsertObservation_GenericObs_Derwent.xml* es el siguiente:

Fig. 31. Esquema general de la plantilla *InsertObservation_GenericObs_Derwent.xml*.

Finalmente, los archivos XML son enviados al SOS. Primero se envía la plantilla *RegisterSensor.xml* para registrar el sensor IPFIX junto con todos los identificadores de los datos que se enviarán en los registros. Una vez el sensor se encuentra registrado se comienzan a enviar plantillas *InsertObservation_GenericObs_Derwent.xml* con los diferentes datos.

X. CONCLUSIONES

En el desarrollo de la tesina se presentó un sistema de monitorización para la protección de redes de infraestructuras críticas mediante el uso de IPFIX. Entrando en más detalle, se presentó la importancia de que el marco de monitorización de dichas estructuras abarque sus elementos integrantes además de las redes de telecomunicaciones que los interconectan para la detección de ataques de seguridad cibernéticos.

La arquitectura de monitorización BUGYO es una buena base para el planteamiento de sistemas de monitorización ya que es fácil de asimiliar y cubre de forma general los requisitos necesarios para la protección de una infraestructura crítica.

IPFIX demuestra ser un protocolo eficiente, con gran la flexibilidad y la interoperabilidad en los sistemas de monitorización de redes gracias al uso de plantillas y la generación de archivos de registro. A su vez garantiza la correcta llegada de los registros a su destino en caso de congestión gracias al protocolo SCTP. Además de poseer la ventaja de que es un protocolo abierto, es decir, no tiene propietario (a diferencia de su antecesor Netflow V9).

Sin embargo, a pesar de que se encuentra estandarizado, IPFIX es un protocolo que aún no se encuentra incluido en los equipos de red y de hecho existen pocas herramientas software que lo implementen a cabalidad; por lo que es necesario optar por nuevas alternativas de implementación mientras los fabricantes incluyen en sus diseños este protocolo, como la presentada en esta tesina.

Por otra parte, el programa desarrollado en lenguaje Java brinda grandes ventajas debido a que no sólo transcribe correctamente los datos de los ficheros texto a las plantillas XML, sino que además permite el almacenamiento de datos en variables, dando la posibilidad de crear más aplicaciones Java que puedan aprovechar dichos recursos.

A su vez, la integración del sistema de monitorización con herramientas de SWE como el SOS dan características de interoperabilidad con otros sistemas, ya que pone a disposición de usuarios y servicios autorizados los datos de monitorización bajo la misma especificación SML.

Finalmente se concluye que el sistema de monitorización funciona y es apto para la detección de ciertos tipos de ataques a la red ya que dió como resultado la correcta detección de ataques DDoS y escaneo de puertos, haciendo del proyecto una propuesta aceptable para la monitorización y seguridad de redes de infraestructuras críticas.

REFERENCIAS

[1] Technical Report, ETSI TR 187 023 V1.1.1 (2012-13) http://www.etsi.org/deliver/etsi_tr/187000_187099/187023/01.01.01_60/tr_187023v010101p.pdf

[2] "Ciberseguridad, retos y amenazas a la seguridad nacional en el ciberespacio", Ministerio de Defensa, España, Diciembre, 2010.

[3] "The real story of Stuxnet", IEEE spectrum, March 2013.

[4] A. Cristina, *Wide-Area Situational Awareness for Critical Infrastructure Protection*, Universidad de Malaga, España, 2013.

[5] D. Luca, Guía de usuario nProbe, 2010.

[6] http://www.manageengine.com/products/netflow/spanish/

[7] http://www.hping.org/hping3.html

[8] www.nmap.org

[9] http://europa.eu/legislation_summaries/justice_freedom_security/fight_against_terrorism/l33260_es.htm

[10] http://www.cnpic-es.es/

[11] T. Tim y D. David, *Communications technologies and practices to satisfy NERC Critical Infrastructure Protection (CIP), Shweitzer Engineering Laboratories Inc., USA, 2006.*

[12] T. K. Reijo y S. H. Artur, *An adaptative and dependable distributed monitoring framework. International Journal on Advances in Security, 2011*.

[13] O. Moussa, K. Djamel, entre otros, *Deployment of a security assurance monitoring framework for telecommunication service infrastructure on a VoIP service, University of East London, England, 2008.*

[14] P. Christophe, H. Perttu, entre otros, BUGYO beyond – security assurance cockpit features specifications, 2013.

[15] T. Brian y B. Elisa, *An introduction to IP Flow Information Export (IPFIX),* Zurich, 2011.

[16] K. Radek, *Network Traffic Collection with IPFIX Protocol,* Master's Thesis, Brno 2009.

[17] G. Pablo, *Simulador web de redes de sensores para la automatización industrial,* tesina de máster, 2011.

ANEXOS

ANEXO A. CONFIGURACIÓN DE NPROBE Y EXPLICACIÓN DE LOS COMANDOS UTILIZADOS.

A continuación se encuentra un ejemplo de configuración de nProbe para utilizar el protocolo IPFIX con determinada plantilla:

"*nprobe –i eth0 –V 10 –n udp://192.168.1.1:9996 –b 1 –a 1 –Q 2 –T "%IPV4_SRC_ADDR %IPV4_DST_ADDR %IPV4_NEXT_HOP %OUT_PKTS %OUT_BYTES %FIRST_SWITCHED %LAST_SWITCHED %L4_SRC_PORT %L4_DST_PORT %TCP_FLAGS %IN_SRC_MAC %OUT_DST_MAC %TOTAL_BYTES_EXP %TOTAL_PKTS_EXP %TOTAL_FLOWS_EXP %IN_BYTES %IN_PKTS %SRC_TOS" –P C:/Registros –D t –dont-nest-dump-dirs"*

En donde:

- "-i" indica la interfaz que se va a monitorizar. En este caso se monitorizará la interfaz eth0
- "-V" especifica la versión de protocolo de monitorización que se utilizará. En este el 10 significa que se trabajará con el protocolo IPFIX.
- "-n" se utiliza para definir la dirección IP del colector IPFIX, el puerto lógico por el que se quiere transmitir y el protocolo de transporte a utilizar. En este caso la dirección IP del colector IPFIX es la del ordenador donde se encuentra instalado el Netflow Analyzer (192.168.1.1), el número de puerto 9996 es el que utiliza IPFIX por defecto, y el protocolo de transporte usado es UDP.
- "-b 1" selecciona el detalle con la que se envían los datos, en este caso 1 significa que se utilizará un detalle óptimo para aplicaciones de estadísticas de tráfico.
- "-a 1" este flag cuando se activa permite que todos los flujos de información sean enviados a todos los colectores IPFIX definidos.
- "-Q 2" es un flag usado como índice del dispositivo de salida utilizado en los flujos emitidos (tráfico saliente).
- "-T" es el comando que se utiliza para definir la plantilla del protocolo de monitorización utilizado, en este caso debe ser una plantilla de IPFIX.
- "-P" es usado para indicar la dirección donde se almacenarán los registros generados por IPFIX. En este caso se almacenarán en la carpeta con nombre "Registros" que se encuentra en el disco local C.
- "-D t" es utilizado para indicar el formato en el que se quieren generar los registros. En este caso se indica que se generen archivos de texto.

- "--dont-nest-dump-dirs" es un comando utilizado para que el propio software IPFIX organice los archivos en carpetas indicando el año, el mes, el día y la hora en la que se generó el registro.

En la tabla a continuación se muestran los elementos de información utilizados en la plantilla definida en la configuración de nProbe realizada:

"%IPV4_SRC_ADDR	**Dirección IPv4 fuente**
%IPV4_DST_ADDR	Dirección IPv4 destino
%IPV4_NEXT_HOP	Dirección IPv4 del siguiente salto
%OUT_PKTS	Flujos de paquetes de salida desde el destino hacia la fuente
%OUT_BYTES	Flujos de bytes de salida desde el destino hacia la fuente
%FIRST_SWITCHED	SysUpTime (ms) del primer paquete del flujo
%LAST_SWITCHED	SysUpTime (ms) del último paquete del flujo
%L4_SRC_PORT	Puerto fuente
%L4_DST_PORT	Puerto destino
%TCP_FLAGS	Acumulado de todas las banderas del flujo TCP
%IN_SRC_MAC	Dirección MAC fuente
%OUT_DST_MAC	Dirección MAC destino
%TOTAL_BYTES_EXP	Total de bytes exportados
%TOTAL_PKTS_EXP	Total de paquetes exportados
%TOTAL_FLOWS_EXP	Total de flujos exportados
%IN_BYTES	Bytes de flujos entrantes (fuente a destino)
%IN_PKTS	Número de paquetes de flujos entrantes (fuente a destino)
%SRC_TOS	Tipo de byte de servicio

Tabla. N. Elementos de información configurados en el nProbe para las pruebas con cada uno de sus significados.

ANEXO C. ARCHIVO DE TEXTO GENERADO POR NPROBE

```
   IPV4_SRC_ADDR|IPV4_DST_ADDR|IPV4_NEXT_HOP|INPUT_SNMP|OUTPUT_SNMP|OUT_PKTS|OUT_BYTES|FIRS
T_SWITCHED|LAST_SWITCHED|L4_SRC_PORT|L4_DST_PORT|PROTOCOL

   192.168.0.1|192.168.1.1|0.0.0.0|1|2|0|0|1372422399.136577|1372422399.136606|43212|9996|1
7

   192.168.0.3|192.168.0.255|0.0.0.0|1|2|0|0|1372422407.058413|1372422407.058413|138|138|17

   0.0.0.0|0.0.0.0|0.0.0.0|1|2|0|0|1372422425.541957|1372422425.541957|0|0|0

   192.168.0.3|192.168.1.1|0.0.0.0|1|2|120|7200|1372422335.482175|1372422454.479218|0|0|1

   192.168.1.1|192.168.0.3|0.0.0.0|2|1|120|7200|1372422335.482561|1372422454.479578|0|0|1

   0.0.0.0|0.0.0.0|0.0.0.0|1|2|59|4214|1372422335.275114|1372422335.275114|0|0|0

   0.0.0.0|0.0.0.0|0.0.0.0|2|1|1|60|1372422337.275086|1372422453.276188|0|0|0

   192.168.1.1|192.168.0.2|0.0.0.0|1|2|120|7200|1372422335.137405|1372422454.136053|0|0|1

   192.168.0.2|192.168.1.1|0.0.0.0|2|1|120|7200|1372422336.137077|1372422455.135587|0|0|1

   0.0.0.0|0.0.0.0|0.0.0.0|1|2|2|116|1372422416.761378|1372422416.761378|0|0|0

   0.0.0.0|0.0.0.0|0.0.0.0|2|1|1|56|1372422442.153205|1372422446.761987|0|0|0

   192.168.0.1|192.168.1.1|0.0.0.0|1|2|0|0|1372422430.136460|1372422438.137205|43212|9996|1
7

   192.168.0.3|82.129.27.63|0.0.0.0|1|2|0|0|1372422353.418967|1372422452.916530|36951|3478|
17

   192.168.0.100|192.168.0.3|0.0.0.0|1|2|0|0|1372422353.419872|1372422459.542361|0|0|1
```

ANEXO D. PLANTILLAS XML GENERADAS EN JAVA Y SIGUIENDO LA ESPECIFICACIÓN SML

Plantilla RegisterSensor.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><RegisterSensor
xmlns="http://www.opengis.net/sos/1.0" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:om="http://www.opengis.net/om/1.0"
xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:sml="http://www.opengis.net/sensorML/1.0.1"
xmlns:swe="http://www.opengis.net/swe/1.0.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" service="SOS"
version="1.0.0" xsi:schemaLocation="http://www.opengis.net/sos/1.0
http://schemas.opengis.net/sos/1.0.0/sosRegisterSensor.xsd
http://www.opengis.net/om/1.0
http://schemas.opengis.net/om/1.0.0/extensions/observationSpecialization_override
.xsd">

  <!-- Sensor Description parameter; Currently, this has to be a sml:System -->
  <SensorDescription>

    <sml:SensorML version="1.0.1">
    <sml:member>
```

```
      <sml:System>

      <!--sml:identification element must contain the ID of the sensor-->
      <sml:identification>
        <sml:IdentifierList>
          <sml:identifier>
            <sml:Term definition="urn:ogc:def:identifier:OGC:uniqueID">
              <sml:value>Exportador_IPFIX</sml:value>
            </sml:Term>
          </sml:identifier>
               </sml:IdentifierList>
      </sml:identification>

      <!-- sml:capabilities element has to contain status and mobility
information -->
      <sml:capabilities>
        <swe:SimpleDataRecord>
          <!-- status indicates, whether sensor is collecting data at the moment
(true) or not (false) -->
          <swe:field name="status">
            <swe:Boolean>
              <swe:value>true</swe:value>
            </swe:Boolean>
          </swe:field>
          <!-- status indicates, whether sensor is mobile (true) or fixed (false)
-->
          <swe:field name="mobile">
            <swe:Boolean>
              <swe:value>false</swe:value>
            </swe:Boolean>
          </swe:field>
        </swe:SimpleDataRecord>
      </sml:capabilities>

      <!-- last measured position of sensor -->
      <sml:position name="sensorPosition">
            <swe:Position referenceFrame="urn:ogc:def:crs:EPSG::4326">
                <swe:location>
                    <swe:Vector gml:id="STATION_LOCATION">
                        <swe:coordinate name="easting">
                            <swe:Quantity>
                                <swe:uom code="degree"/>
                                <swe:value>39.46</swe:value>
                              </swe:Quantity>
              </swe:coordinate>
                            <swe:coordinate name="northing">
                              <swe:Quantity>
                                  <swe:uom code="degree"/>
                                  <swe:value>22.5</swe:value>
                              </swe:Quantity>
                            </swe:coordinate>
                            <swe:coordinate name="altitude">
                              <swe:Quantity>
                                  <swe:uom code="m"/>
                                  <swe:value>1</swe:value>
                                </swe:Quantity>
                            </swe:coordinate>
            </swe:Vector>
          </swe:location>
        </swe:Position>
      </sml:position>

            <!-- list containing the input phenomena for this sensor system -->
            <sml:inputs>
                 <sml:InputList>
                      <sml:input name="IPV4_SRC_ADDR">
                           <swe:ObservableProperty
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:IPV4_SRC_ADDR"/>
```

```
                        </sml:input>
                        <sml:input name="IPV4_DST_ADDR">
                             <swe:ObservableProperty
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:IPV4_DST_ADDR"/>
                        </sml:input>
                        <sml:input name="IPV4_NEXT_HOP">
                             <swe:ObservableProperty
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:IPV4_NEXT_HOP"/>
                        </sml:input>
                        <sml:input name="OUT_PKTS">
                             <swe:ObservableProperty
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:OUT_PKTS"/>
                        </sml:input>
                        <sml:input name="OUT_BYTES">
                             <swe:ObservableProperty
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:OUT_BYTES"/>
                        </sml:input>
                        <sml:input name="FIRST_SWITCHED">
                             <swe:ObservableProperty
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:FIRST_SWITCHED"/>
                        </sml:input>
                        <sml:input name="LAST_SWITCHED">
                             <swe:ObservableProperty
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:LAST_SWITCHED"/>
                        </sml:input>
                        <sml:input name="L4_SRC_PORT">
                             <swe:ObservableProperty
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:L4_SRC_PORT"/>
                        </sml:input>
                        <sml:input name="L4_DST_PORT">
                             <swe:ObservableProperty
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:L4_DST_PORT"/>
                        </sml:input>
                        <sml:input name="PROTOCOL">
                             <swe:ObservableProperty
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:PROTOCOL"/>
                        </sml:input>
                        <sml:input name="TCP_FLAGS">
                             <swe:ObservableProperty
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:TCP_FLAGS"/>
                        </sml:input>
                        <sml:input name="IN_SRC_MAC">
                             <swe:ObservableProperty
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:IN_SRC_MAC"/>
                        </sml:input>
                        <sml:input name="OUT_DST_MAC">
                             <swe:ObservableProperty
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:OUT_DST_MAC"/>
                        </sml:input>
                        <sml:input name="TOTAL_BYTES_EXP">
                             <swe:ObservableProperty
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:TOTAL_BYTES_EXP"/>
                        </sml:input>
                        <sml:input name="TOTAL_PKTS_EXP">
                             <swe:ObservableProperty
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:TOTAL_PKTS_EXP"/>
                        </sml:input>
                        <sml:input name="TOTAL_FLOWS_EXP">
                             <swe:ObservableProperty
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:TOTAL_FLOWS_EXP"/>
                        </sml:input>
                        <sml:input name="IN_BYTES">
                             <swe:ObservableProperty
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:IN_BYTES"/>
                        </sml:input>
                        <sml:input name="IN_PKTS">
                             <swe:ObservableProperty
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:IN_PKTS"/>
```

```
                    </sml:input>
                    <sml:input name="SRC_TOS">
                         <swe:ObservableProperty
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:SRC_TOS"/>
                    </sml:input>
               </sml:InputList>
          </sml:inputs>

          <!-- list containing the output phenomena of this sensor system;
ATTENTION: these phenomena are parsed and inserted into the database; they have
to contain offering elements to determine the correct offering for the sensors
and measured phenomena -->
          <sml:outputs>
               <sml:OutputList>
                    <sml:output name="IPV4_SRC_ADDR">
                         <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:IPV4_SRC_ADDR">
                              <gml:metaDataProperty>
                    <offering>
                      <id>PaqueteIPFIX</id>
                      <name>PaqueteIPFIX</name>
                    </offering>
               </gml:metaDataProperty>
                              <swe:uom code=""/>
                         </swe:Quantity>
                    </sml:output>
                     <sml:output name="IPV4_DST_ADDR">
                         <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:IPV4_DST_ADDR">
                              <gml:metaDataProperty>
                    <offering>
                      <id>PaqueteIPFIX</id>
                      <name>PaqueteIPFIX</name>
                    </offering>
               </gml:metaDataProperty>
                              <swe:uom code=""/>
                         </swe:Quantity>
                    </sml:output>
               <sml:output name="IPV4_NEXT_HOP">
                         <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:IPV4_NEXT_HOP">
                              <gml:metaDataProperty>
                    <offering>
                      <id>PaqueteIPFIX</id>
                      <name>PaqueteIPFIX</name>
                    </offering>
               </gml:metaDataProperty>
                              <swe:uom code=""/>
                         </swe:Quantity>
                    </sml:output>
               <sml:output name="OUT_PKTS">
                         <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:OUT_PKTS">
                              <gml:metaDataProperty>
                    <offering>
                      <id>PaqueteIPFIX</id>
                      <name>PaqueteIPFIX</name>
                    </offering>
               </gml:metaDataProperty>
                              <swe:uom code=""/>
                         </swe:Quantity>
                    </sml:output>
               <sml:output name="OUT_BYTES">
                         <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:OUT_BYTES">
                              <gml:metaDataProperty>
                    <offering>
                      <id>PaqueteIPFIX</id>
```

```
                        <name>PaqueteIPFIX</name>
                    </offering>
                </gml:metaDataProperty>
                            <swe:uom code=""/>
                        </swe:Quantity>
                    </sml:output>
                <sml:output name="FIRST_SWITCHED">
                        <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:FIRST_SWITCHED">
                            <gml:metaDataProperty>
                    <offering>
                      <id>PaqueteIPFIX</id>
                      <name>PaqueteIPFIX</name>
                    </offering>
                </gml:metaDataProperty>
                            <swe:uom code=""/>
                        </swe:Quantity>
                    </sml:output>
                <sml:output name="LAST_SWITCHED">
                        <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:LAST_SWITCHED">
                            <gml:metaDataProperty>
                    <offering>
                      <id>PaqueteIPFIX</id>
                      <name>PaqueteIPFIX</name>
                    </offering>
                </gml:metaDataProperty>
                            <swe:uom code=""/>
                        </swe:Quantity>
                    </sml:output>
                <sml:output name="L4_SRC_PORT">
                        <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:L4_SRC_PORT">
                            <gml:metaDataProperty>
                    <offering>
                      <id>PaqueteIPFIX</id>
                      <name>PaqueteIPFIX</name>
                    </offering>
                </gml:metaDataProperty>
                            <swe:uom code=""/>
                        </swe:Quantity>
                    </sml:output>
                <sml:output name="L4_DST_PORT">
                        <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:L4_DST_PORT">
                            <gml:metaDataProperty>
                    <offering>
                      <id>PaqueteIPFIX</id>
                      <name>PaqueteIPFIX</name>
                    </offering>
                </gml:metaDataProperty>
                            <swe:uom code=""/>
                        </swe:Quantity>
                    </sml:output>
                <sml:output name="PROTOCOL">
                        <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:PROTOCOL">
                            <gml:metaDataProperty>
                    <offering>
                      <id>PaqueteIPFIX</id>
                      <name>PaqueteIPFIX</name>
                    </offering>
                </gml:metaDataProperty>
                            <swe:uom code=""/>
                        </swe:Quantity>
                    </sml:output>
                <sml:output name="TCP_FLAGS">
                        <swe:Quantity
```

```
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:TCP_FLAGS">
                                    <gml:metaDataProperty>
                        <offering>
                          <id>PaqueteIPFIX</id>
                          <name>PaqueteIPFIX</name>
                        </offering>
                    </gml:metaDataProperty>
                                    <swe:uom code=""/>
                                </swe:Quantity>
                        </sml:output>
                    <sml:output name="IN_SRC_MAC">
                                <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:IN_SRC_MAC">
                                    <gml:metaDataProperty>
                        <offering>
                          <id>PaqueteIPFIX</id>
                          <name>PaqueteIPFIX</name>
                        </offering>
                    </gml:metaDataProperty>
                                    <swe:uom code=""/>
                                </swe:Quantity>
                        </sml:output>
                    <sml:output name="OUT_DST_MAC">
                                <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:OUT_DST_MAC">
                                    <gml:metaDataProperty>
                        <offering>
                          <id>PaqueteIPFIX</id>
                          <name>PaqueteIPFIX</name>
                        </offering>
                    </gml:metaDataProperty>
                                    <swe:uom code=""/>
                                </swe:Quantity>
                        </sml:output>
                    <sml:output name="TOTAL_BYTES_EXP">
                                <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:TOTAL_BYTES_EXP">
                                    <gml:metaDataProperty>
                        <offering>
                          <id>PaqueteIPFIX</id>
                          <name>PaqueteIPFIX</name>
                        </offering>
                    </gml:metaDataProperty>
                                    <swe:uom code=""/>
                                </swe:Quantity>
                        </sml:output>
                    <sml:output name="TOTAL_PKTS_EXP">
                                <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:TOTAL_PKTS_EXP">
                                    <gml:metaDataProperty>
                        <offering>
                          <id>PaqueteIPFIX</id>
                          <name>PaqueteIPFIX</name>
                        </offering>
                    </gml:metaDataProperty>
                                    <swe:uom code=""/>
                                </swe:Quantity>
                        </sml:output>
                    <sml:output name="TOTAL_FLOWS_EXP">
                                <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:TOTAL_FLOWS_EXP">
                                    <gml:metaDataProperty>
                        <offering>
                          <id>PaqueteIPFIX</id>
                          <name>PaqueteIPFIX</name>
                        </offering>
                    </gml:metaDataProperty>
                                    <swe:uom code=""/>
```

```
                              </swe:Quantity>
                         </sml:output>
                    <sml:output name="IN_BYTES">
                              <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:IN_BYTES">
                                   <gml:metaDataProperty>
                         <offering>
                           <id>PaqueteIPFIX</id>
                           <name>PaqueteIPFIX</name>
                         </offering>
                    </gml:metaDataProperty>
                                   <swe:uom code=""/>
                              </swe:Quantity>
                         </sml:output>
                    <sml:output name="IN_PKTS">
                              <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:IN_PKTS">
                                   <gml:metaDataProperty>
                         <offering>
                           <id>PaqueteIPFIX</id>
                           <name>PaqueteIPFIX</name>
                         </offering>
                    </gml:metaDataProperty>
                                   <swe:uom code=""/>
                              </swe:Quantity>
                         </sml:output>
                    <sml:output name="SRC_TOS">
                              <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:SRC_TOS">
                                   <gml:metaDataProperty>
                         <offering>
                           <id>PaqueteIPFIX</id>
                           <name>PaqueteIPFIX</name>
                         </offering>
                    </gml:metaDataProperty>
                                   <swe:uom code=""/>
                              </swe:Quantity>
                         </sml:output>
                    </sml:OutputList>
               </sml:outputs>

                     </sml:System>
          </sml:member>
          </sml:SensorML>
     </SensorDescription>

     <!-- ObservationTemplate parameter; this has to be an empty measurement at the
moment, as the 52N SOS only supports Measurements to be inserted -->
     <ObservationTemplate>
       <om:Measurement>
         <om:samplingTime/>
         <om:procedure/>
         <om:observedProperty/>
         <om:featureOfInterest/>
         <om:result uom=""/>
       </om:Measurement>
     </ObservationTemplate>

</RegisterSensor>
```

Plantilla InsertObservation_GenericObs_Derwent.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><InsertObservation
xmlns="http://www.opengis.net/sos/1.0" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:om="http://www.opengis.net/om/1.0"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:sa="http://www.opengis.net/sampling/1.0"
```

```
xmlns:sos="http://www.opengis.net/sos/1.0"
xmlns:swe="http://www.opengis.net/swe/1.0.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" service="SOS"
version="1.0.0" xsi:schemaLocation="http://www.opengis.net/sos/1.0
http://schemas.opengis.net/sos/1.0.0/sosInsert.xsd
http://www.opengis.net/sampling/1.0
http://schemas.opengis.net/sampling/1.0.0/sampling.xsd
http://www.opengis.net/om/1.0
http://schemas.opengis.net/om/1.0.0/extensions/observationSpecialization_override
.xsd">


  <AssignedSensorId>Exportador_IPFIX</AssignedSensorId>

  <om:Observation>

         <om:samplingTime>
          <gml:TimePeriod xsi:type="gml:TimePeriodType">
          <gml:beginPosition>2011-10-31T17:05:20+0200</gml:beginPosition>
          <gml:endPosition>2011-10-31T17:10:20+0200</gml:endPosition>
          </gml:TimePeriod>
          </om:samplingTime>


       <om:procedure xlink:href="Exportador_IPFIX"/>
       <om:observedProperty>
       <swe:CompositePhenomenon dimension="1" gml:id="cpid0">
       <gml:name>resultComponents</gml:name>
       <swe:component xlink:href="urn:ogc:data:time:iso8601"/>
       <swe:component
xlink:href="urn:ogc:def:phenomenon:FASYS:1.0.30:IPV4_SRC_ADDR"/>
       <swe:component
xlink:href="urn:ogc:def:phenomenon:FASYS:1.0.30:IPV4_DST_ADDR"/>
       <swe:component
xlink:href="urn:ogc:def:phenomenon:FASYS:1.0.30:IPV4_NEXT_HOP"/>
       <swe:component xlink:href="urn:ogc:def:phenomenon:FASYS:1.0.30:OUT_PKTS"/>
       <swe:component
xlink:href="urn:ogc:def:phenomenon:FASYS:1.0.30:OUT_BYTES"/>
       <swe:component
xlink:href="urn:ogc:def:phenomenon:FASYS:1.0.30:FIRST_SWITCHED"/>
       <swe:component
xlink:href="urn:ogc:def:phenomenon:FASYS:1.0.30:LAST_SWITCHED"/>
       <swe:component
xlink:href="urn:ogc:def:phenomenon:FASYS:1.0.30:L4_SRC_PORT"/>
       <swe:component
xlink:href="urn:ogc:def:phenomenon:FASYS:1.0.30:L4_DST_PORT"/>
       <swe:component xlink:href="urn:ogc:def:phenomenon:FASYS:1.0.30:PROTOCOL"/>
       <swe:component
xlink:href="urn:ogc:def:phenomenon:FASYS:1.0.30:TCP_FLAGS"/>
       <swe:component
xlink:href="urn:ogc:def:phenomenon:FASYS:1.0.30:IN_SRC_MAC"/>
       <swe:component
xlink:href="urn:ogc:def:phenomenon:FASYS:1.0.30:OUT_DST_MAC"/>
       <swe:component
xlink:href="urn:ogc:def:phenomenon:FASYS:1.0.30:TOTAL_BYTES_EXP"/>
       <swe:component
xlink:href="urn:ogc:def:phenomenon:FASYS:1.0.30:TOTAL_PKTS_EXP"/>
       <swe:component
xlink:href="urn:ogc:def:phenomenon:FASYS:1.0.30:TOTAL_FLOWS_EXP"/>
       <swe:component xlink:href="urn:ogc:def:phenomenon:FASYS:1.0.30:IN_BYTES"/>
       <swe:component xlink:href="urn:ogc:def:phenomenon:FASYS:1.0.30:IN_PKTS"/>
       <swe:component xlink:href="urn:ogc:def:phenomenon:FASYS:1.0.30:SRC_TOS"/>
       </swe:CompositePhenomenon>
       </om:observedProperty>


         <om:featureOfInterest>
```

```
   <gml:FeatureCollection>
    <gml:featureMember>
     <sa:SamplingPoint gml:id="foi_1001">
      <gml:name>foi_1001</gml:name>
      <sa:sampledFeature xlink:href=""/>
      <sa:position>
       <gml:Point>
        <gml:pos srsName="urn:ogc:def:crs:EPSG::4326">-42.91 147.33 </gml:pos>
       </gml:Point>
      </sa:position>
     </sa:SamplingPoint>
    </gml:featureMember>
   </gml:FeatureCollection>
  </om:featureOfInterest>



   <om:result>
        <swe:DataArray>

        <swe:elementCount>
        <swe:Count>
        <swe:value>1</swe:value>
        </swe:Count>
        </swe:elementCount>

    <swe:elementType name="Components">
     <swe:DataRecord>
      <swe:field name="Time">
       <swe:Time definition="urn:ogc:data:time:iso8601"/>
      </swe:field>
      <swe:field name="feature">
       <swe:Text
definition="urn:ogc:dahttp://www.opengis.net/def/property/OGC/0/FeatureOfInterest
ta:feature"/>
      </swe:field>
      <swe:field name="IPV4_SRC_ADDR">
       <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:IPV4_SRC_ADDR">
        <swe:uom code=""/>
       </swe:Quantity>
      </swe:field>
      <swe:field name="IPV4_DST_ADDR">
       <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:IPV4_DST_ADDR">
        <swe:uom code=""/>
       </swe:Quantity>
      </swe:field>
      <swe:field name="IPV4_NEXT_HOP">
       <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:IPV4_NEXT_HOP">
        <swe:uom code=""/>
       </swe:Quantity>
      </swe:field>
      <swe:field name="OUT_PKTS">
       <swe:Quantity definition="urn:ogc:def:phenomenon:FASYS:1.0.30:OUT_PKTS">
        <swe:uom code=""/>
       </swe:Quantity>
      </swe:field>
      <swe:field name="OUT_BYTES">
       <swe:Quantity definition="urn:ogc:def:phenomenon:FASYS:1.0.30:OUT_BYTES">
        <swe:uom code=""/>
       </swe:Quantity>
      </swe:field>
      <swe:field name="FIRST_SWITCHED">
       <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:FIRST_SWITCHED">
        <swe:uom code=""/>
```

```
        </swe:Quantity>
       </swe:field>
       <swe:field name="LAST_SWITCHED">
        <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:LAST_SWITCHED">
         <swe:uom code=""/>
        </swe:Quantity>
       </swe:field>
       <swe:field name="L4_SRC_PORT">
        <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:L4_SRC_PORT">
         <swe:uom code=""/>
        </swe:Quantity>
       </swe:field>
       <swe:field name="L4_DST_PORT">
        <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:L4_DST_PORT">
         <swe:uom code=""/>
        </swe:Quantity>
       </swe:field>
       <swe:field name="PROTOCOL">
        <swe:Quantity definition="urn:ogc:def:phenomenon:FASYS:1.0.30:PROTOCOL">
         <swe:uom code=""/>
        </swe:Quantity>
       </swe:field>
       <swe:field name="TCP_FLAGS">
        <swe:Quantity definition="urn:ogc:def:phenomenon:FASYS:1.0.30:TCP_FLAGS">
         <swe:uom code=""/>
        </swe:Quantity>
       </swe:field>
       <swe:field name="IN_SRC_MAC">
        <swe:Quantity definition="urn:ogc:def:phenomenon:FASYS:1.0.30:IN_SRC_MAC">
         <swe:uom code=""/>
        </swe:Quantity>
       </swe:field>
       <swe:field name="OUT_DST_MAC">
        <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:OUT_DST_MAC">
         <swe:uom code=""/>
        </swe:Quantity>
       </swe:field>
       <swe:field name="TOTAL_BYTES_EXP">
        <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:TOTAL_BYTES_EXP">
         <swe:uom code=""/>
        </swe:Quantity>
       </swe:field>
       <swe:field name="TOTAL_PKTS_EXP">
        <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:TOTAL_PKTS_EXP">
         <swe:uom code=""/>
        </swe:Quantity>
       </swe:field>
       <swe:field name="TOTAL_FLOWS_EXP">
        <swe:Quantity
definition="urn:ogc:def:phenomenon:FASYS:1.0.30:TOTAL_FLOWS_EXP">
         <swe:uom code=""/>
        </swe:Quantity>
       </swe:field>
       <swe:field name="IN_BYTES">
        <swe:Quantity definition="urn:ogc:def:phenomenon:FASYS:1.0.30:IN_BYTES">
         <swe:uom code=""/>
        </swe:Quantity>
       </swe:field>
       <swe:field name="IN_PKTS">
        <swe:Quantity definition="urn:ogc:def:phenomenon:FASYS:1.0.30:IN_PKTS">
         <swe:uom code=""/>
        </swe:Quantity>
```

```
      </swe:field>
      <swe:field name="SRC_TOS">
       <swe:Quantity definition="urn:ogc:def:phenomenon:FASYS:1.0.30:SRC_TOS">
        <swe:uom code=""/>
       </swe:Quantity>
      </swe:field>
     </swe:DataRecord>
    </swe:elementType>

    <swe:encoding>
     <swe:TextBlock blockSeparator=";" decimalSeparator="." tokenSeparator=","/>
    </swe:encoding>

    <swe:values>2011-10-
31T17:06:20+0200,foi_1001,192.168.0.2,192.168.0.3,0.0.0.0,0,0,1373356362.359375,1
373356362.359375,51295,49176,6,2,C8:60:00:A3:55:54,C8:60:00:A3:56:80,71744,102,10
29,44,1,0;</swe:values>

   </swe:DataArray>
  </om:result>


  </om:Observation>

</InsertObservation>
```