Tilsor S.A.

Universidad de la República

Facultad de Ingeniería

Tesis de grado

Advanced Threats Information Sharing and Collaboration

Autor:
Julio Saráchaga

Supervisor:
Dr. Gustavo Betarte

Contraparte del cliente:
Ing. Fernanda Molina

Supervisor alterno:
Ing. Marcelo Rodríguez

3 de noviembre de 2014

Índice general

Capítulo 1

Introducción

1.1. Contexto

Este proyecto se desarrolla en el contexto de los temas de investigación y trabajo desarrollados en el Instituto de Computación, en particular dentro del grupo de seguridad informática (GSI) y del equipo de Respuesta a incidentes de Tilsor S.A (CSIRT Tilsor). Se interaccionará y coordinará eventualmente con otras personas y/o organizaciones que estén trabajando en temas afines.

1.2. Motivación

Históricamente, se ha visto un aumento en la sofisticación, velocidad, impacto y cantidad de los ataques informáticos. Por ello es necesario que las estrategias de defensa se adapten a los nuevos actores y ataques. Para responder a esto, las organizaciones han tenido que recurrir al intercambio de información de amenazas con el fin de tener un mejor panorama de las actividades de sus adversarios y así ayudar a administrar sus recursos de forma de obtener el mejor resultado posible de sus defensas.

Las aproximaciones tradicionales de seguridad tienen como objetivo entender y registrar vulnerabilidades, debilidades y configuraciones necesarias pero insuficientes. Para defenderse, las organizaciones tienen estrategias basadas en alertas que buscan bloquear los ataques y arreglar las vulnerabilidades. Si bien dichas estrategias pueden ser efectivas contra algunas amenazas no logran detener ataques avanzados o proveer información sobre las actividades de un atacante luego de que ingresó a la red. Una estrategia más adecuada es basarse en ***cyber kill-chain***, en esta estrategia se busca descomponer las fases de un ataque con la finalidad de obtener una mejor comprensión del

ataque y el atacante, así como mejorar las posibilidades de defensa.

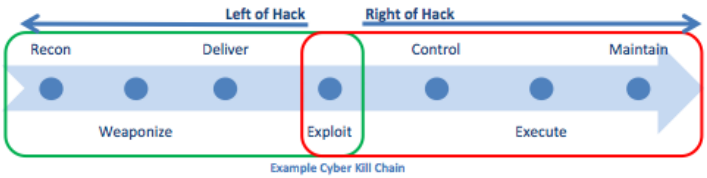

Figura 1.1: Cyber kill-chain [32]

Como se ve en la figura 1.2, los primeros pasos de esta estrategia representan una oportunidad para detectar y mitigar las amenazas de forma proactiva antes de que al adversario realice un acceso no autorizado en los sistemas de la organización. En los pasos posteriores es donde se realiza la detección, respuesta y aseguramiento de los activos más importantes. Al entender al adversario, se puede tener una mejor oportunidad para descubrir sus intenciones y responder al ataque. Entender la amenaza permite la toma de mejores decisiones, dando prioridad a los recursos y así tener una ventaja ante el atacante. El resultado de defensas en base a inteligencia es una mejor opción que las respuestas a los ataques dado que estos ajustan sus operaciones basándose en el éxito o falla de sus intentos. En un modelo como el de la figura 1.2 los intentos del adversario pueden ser reconocidos, logrando que los defensores tengan la posibilidad de ajustar sus tácticas para que al adversario le sea más difícil alcanzar sus objetivos.

Compartir información con socios y comunidades de confianza le permite a las organizaciones tener un conjunto de datos relevante para lograr una identificación precisa de la amenaza. Por medio de este intercambio, cada organización puede entender mejor las amenazas, no solo de forma abstracta sino que también con evidencias específicas que indiquen la presencia del atacante. Las tareas de cyber inteligencia permiten anticipar y mitigar las amenazas antes de que sean difíciles de encontrar y erradicar utilizando los métodos tradicionales de detección y respuesta. Con la información recolectada los analistas pueden agrupar patrones de actividades similares, atribuir actividades a ciertos actores, identificar e implementar estrategias para mitigar ataques de forma rápida y anticiparse al lanzamiento de ataques similares en el futuro. Para aprovechar de forma adecuada los beneficios de la cyber inteligencia, las organizaciones deben compartir la información recolectada con socios de su confianza.

Por medio del análisis del comportamiento de los adversarios en distintos objetivos y en un período de tiempo adecuado, los defensores son capaces de identificar un conjunto importante de indicadores, tácticas, técnicas y procedimientos. De esta forma se obtiene información de los objetivos y las estrategias lo cual permite al defensor predecir el comportamiento del ataque y generar defensas dinámicas. Dada la forma y complejidad con la cual evolucionan las amenazas, la velocidad con la cual ocurren los eventos y la basta cantidad de datos que se deberían intercambiar, es necesario establecer una forma automática para ayudar a los analistas y tomadores de decisión a llevar a cabo acciones defensivas.

Existen múltiples métodos para el intercambio de información y estos juegan un rol importante en los tipos, volúmenes y naturaleza de la información compartida con la comunidad. Algunos de estos medios de intercambio limitan el tipo de contenido que es compartido mientras que otros promueven ciertos tipos de intercambio. Los procesos para compartir información son manuales, llevan mucho tiempo, son repetitivos y en muchos casos requieren que las organizaciones reescriban o traduzcan la información a una amplia variedad de formatos. Otro problema que se presenta cuando se quiere intercambiar información es la utilización de tecnologías y/o formatos propietarios, esto provoca la necesidad de desarrollar scripts y módulos para permitir compartir información por fuera de las comunidades. Para aquellas comunidades con algún grado de automatización, sus modelos son generalmente bajos en prestaciones y usan soluciones propietarias, comerciales o adaptadas a su comunidad.

La mayoría de dichos métodos no permiten el consumo de información de amenazas de forma automática, esto hace que rutinariamente las organizaciones deban tomar dicha información y sintetizarla en sus bases de datos locales. Si bien los métodos utilizados en la actualidad han ayudado a mejorar las capacidades defensivas de numerosas organizaciones, éstas no han logrado explotar su máximo potencial.

La automatización requiere de información de calidad, esto no puede ser logrado adecuadamente con los distintos productos y sistemas de hoy en día. Para lograr un nivel adecuado de automatización es necesario contar con un estándar el cual tenga representaciones estructuradas de información, de ésta forma se puede lograr un mejor aprovechamiento de los datos sin saber de antemano quien los provee. Dicha información debe ser legible por un humano y parseable por una máquina. Estos requerimientos tienen varias justificaciones, primero que nada, un analista podría realizar un análisis que

es inapropiado para ser automatizado o que sea focalizado en tomas de decisiones por parte de personas. También podría ser de interés que un analista tenga conocimiento de la situación actual, de la fidelidad de las fuentes o de los métodos utilizados para producir la información. Por lo dicho anteriormente, es necesario la existencia de un estándar con una representación estructurada de la información, siendo ésta expresiva, flexible, extensible, automatizable y legible. Además se deben contar con medios para permitir el intercambio seguro y confiable de información entre distintas organizaciones.

1.3. Objetivos

Este proyecto tiene como objetivo principal profundizar en el estudio de los mecanismos para intercambiar información entre dos entidades (en particular equipos/centros de repuesta a incidentes) de forma segura, utilizando para ésto protocolos estándares, por ejemplo haciendo implementaciones que utilicen los protocolos STIX y TAXII antes mencionados. Asimismo, se pretende que dicha herramienta sea aplicada al menos en algún caso de estudio.

1.4. Organización del documento

El documento se organiza de la siguiente forma.

En el Capítulo 2 se presentan conceptos generales de la gestión a incidentes y los mecanismos de intercambio de información para poner en contexto el presente trabajo. En particular se mencionan RTIR, TAXII y STIX.

En el Capítulo 3 se presentan el análisis de la investigación realizada y se definen requerimientos necesarios en una herramienta de intercambio de información.

En el Capítulo 4 plantea el diseño de la solución planteada en el capítulo anterior. Se describen los componentes del sistema, las decisiones de diseño realizadas, la arquitectura del sistema y la interacción entre cada uno de los componentes utilizados.

En el Capítulo 6 se explica la implementación del sistema definido.

En el Capítulo 7 se describe el caso de estudio planteado.

En el Capítulo 8 se presentan las conclusiones del trabajo y posibles trabajos a futuro.

Por último se presenta la bibliografía consultada y los anexos, los cuales son referenciados a lo largo del documento.

Capítulo 2

Estado del Arte

2.1. Gestión de incidentes

2.1.1. Herramientas

2.1.1.1. Que es RTIR

RTIR es un sistema de manejo de incidentes diseñado para ser utilizado por los equipos de seguridad de sistemas. Ha sido creado en conjunto con equipos de CSIRT para manejar el creciente número de incidentes reportados. Presenta la ventaja de ser *opensource*, contener una API completa y una comunidad de usuarios grande y experta. Además es simple de integrar con otras herramientas existentes. Está implementado por medio de módulos PERL y las herramientas que provee RT. Se puede pensar en RTIR como una extensión de RT para ser utilizada por CSIRT's.

Existen otras herramientas parecidas a RTIR que se pueden ver en el documento del estado del arte **??**.

2.2. Representación

2.2.1. Modelos de datos

2.2.1.1. IDMEF

Intrusion Detection Message Exchange Format (IDMEF) fue especificado como un protocolo experimental que no especifica ningún estándar. El propósito de IDMEF es definir formatos de datos y procedimientos de intercambio para compartir información de interés con sistemas de detección

de intrusos y sistemas de respuesta con los sistemas de administración que deben interactuar con estos. El RFC de IDMEF describe el modelo de datos para representar información tomada de los sistemas de intrusión. Se busca que dicho formato pueda ser utilizado por los *Intrusion Detection Systems* (IDSs) para reportar alertas sobre eventos que parezcan sospechosos.

2.2.1.1.1. Modelo de datos de IDMEF

El modelo de datos de IDMEF es una representación orientada a alertas enviadas a los administradores desde los sistemas de intrusión. Su diseño busca proveer una representación estándar de las alertas de manera que la información no sea ambigua y que se describa la relación entre alertas simples y complejas.

IDMEF tiene como objetivo proveer una representación estándar de la información analizada por un sistema de intrusión cuando es detectado un evento inusual. Estas alertas pueden ser simples o complejas dependiendo de las capacidades de la herramienta que las creo.

Por medio de IDMEF se pueden referencias contenidos adicionales. es importante debido a que la tarea de clasificar y nombrar vulnerabilidades es difícil y sumamente subjetiva. El modelo de datos no debe ser ambiguo, por lo que mientras que se permite que los análisis sean más o menos precisos entre ellos, no se debe permitir que estos produzcan información contradictoria en dos alertas que describen el mismo evento. De todas formas, siempre es posible insertar toda la información útil de un evento en campos de extensión de la alerta en lugar de en los campos a los que estos pertenecen, sin embargo, dichas prácticas reducen la interoperabilidad y deberían ser evitadas en lo posible.

2.2.1.2. IODEF

Incident Object Description Exchange Format (IODEF) define una representación de datos que provee un framework para el intercambio de información entre CSIRTs, dicho tipo de información de seguridad es comúnmente intercambiado entre este tipo de organizaciones. IODEF provee una representación en XML para transportar información de incidentes entre pares. El modelo de datos provisto por IODEF codifica la información referente a hosts, redes y servicios corriendo en estos sistemas; metodología de ataques y evidencia forense asociada; impacto de la actividad realizada y aproximaciones a los trabajos realizados.

IODEF es compatible y tiene la capacidad de incluir mensajes IDMEF. Los actores principales en IODEF son los CSIRTs, por lo tanto se busca que sea utilizado por personas. Esto causa que IODEF deba ser leíble por humanos además de parseable por computadoras.

Los mensajes IODEF consideran información referente al manejo de incidentes, almacenamiento de dicha información o estadísticas y análisis.

El objetivo primordial de IODEF es mejorar las capacidades operacionales de los CSIRTs. La adopción por parte de la comunidad provee una habilidad mejorada para resolver los incidentes y transmitir información de contexto simplificando la colaboración e intercambio de datos. El formato estructurado provisto por IODEF permite:

- Incrementar la automatización en el procesamiento de los datos.
- Bajar el esfuerzo necesario para normalizar datos similares de diferentes fuentes.
- Un formato común con el cual construir herramientas interoperables para el manejo de incidentes y análisis subsecuente, específicamente cuando los datos provienen de dominios distintos.

La coordinación entre CSIRTs no es un problema estrictamente técnico. La confianza, los procedimientos y las leyes son consideraciones que pueden impedir que las organizaciones intercambien información. IODEF no busca evadir dichas consideraciones, sin embargo, las implementaciones operacionales de IODEF deben considerar este contexto.

2.2.1.3. STIX

Structured Threat Information eXpression (STIX) es un lenguaje para la especificación, captura, representación y comunicación de información de seguridad. Lo realiza de manera estructurada para soportar un mejor manejo de la información así como para permitir procesos automatizados.

Se busca que la información sea representada de forma estructurada, a su vez esta debe ser expresiva, flexible, extensible, automatizable y legible. Esto se presenta como un desafío ya que la información intercambiada debe ser estructurada para que pueda ser parseada por una máquina pero no se debe perder la posibilidad de que un analista la pueda leer y entender. Que

sea entendible por una persona es necesario para que no se pierda el juicio y control que esta pueda tener.

Al convertirse STIX en un estándar para la comunidad se obtendrá información de mejor calidad la que será mejor aprovechada y de la cual no se sabrá la fuente de los datos de antemano.

2.2.1.3.1. Aproximaciones de la actualidad

La información que se intercambia y utiliza actualmente es atómica, inconsistente y muy limitada en sofisticación y expresividad. Además, el uso de estructuras estandarizadas se enfoca en una porción del problema. Generalmente, las actividades de intercambio de indicadores son entre humanos, siendo los indicadores desestructurados o semi-estructurados e intercambiados por medio de portales web o encriptados y enviados vía email. Recientemente se ha visto el surgimiento de transferencias entre máquinas, éstas intercambian conjuntos de indicadores simples de modelos de ataques bien conocidos.

A diferencia de IODEF, STIX provee una lista de elementos para construir indicadores de compromiso y se integra con CyBox, CAPEC, MAEC o CVE. Además posee soporte para tácticas, técnicas y procedimientos del adversario y tareas realizadas por la organización. IODEF fue pensado para compartir información de incidentes y no indicadores de compromiso. STIX permite el intercambio de indicadores referentes a amenazas así como información del contexto en el que éstas se dan. Juntos, permiten tener un conocimiento más rico de las intenciones, capacidades, motivaciones y actividades de un adversario y de esta forma defenderse de él.

STIX busca extender los indicadores para permitir el manejo e intercambio de estos de forma más expresiva y con un espectro más amplio de información.

STIX provee una arquitectura unificada con un conjunto amplio de información para permitir una solución práctica que permita la implementación de distintos casos de uso. El conjunto de información incluye:

- Cyber Observables
- Indicadores
- Incidentes

- Tácticas, técnicas y procedimientos de los adversarios
- Objetivos de exploits
- Cursos de acción
- Campañas de ataques
- Actores de ataques

2.2.1.3.2. Principios de desarrollo de STIX

Con el consenso de la comunidad, se han definido un conjunto de principios para el desarrollo de STIX. Estos principios son los siguientes:

- Expresividad: Con el fin de soportar la diversidad de casos de uso relevantes, STIX apunta a proveer una cobertura expresiva en todos sus casos de uso específicos en lugar de dirigirse específicamente a alguno de ellos.
- Integración en lugar de duplicación: Cuando STIX abarca conceptos de información estructurada para los cuales ya existen representaciones estandarizadas con el consenso adecuado y que se encuentran disponibles, se busca integrar estas representaciones a la arquitectura STIX en lugar de duplicar la información innecesariamente.
- Flexibilidad: Con el fin de soportar un amplio rango de casos e información variable con varios niveles de fidelidad, se ha diseñado a STIX para ofrecer tanta flexibilidad como sea posible. STIX se adhiere a una política de permitir a los usuarios utilizar cualquier porción de representaciones estándar que sean relevantes para un contexto dado y evita elementos obligatorios siempre que sea posible.
- Extensibilidad: Con la finalidad de soportar un amplio rango de casos de uso con un potencial diferente de representación y para facilitar el perfeccionamiento impulsado por la comunidad, se ha diseñado STIX para construir mecanismos de extensión para usos específicos, para usos localizados, para refinamientos del usuario y evolución y para facilidad de refinamiento y evolución centralizada.
- Automatización: El diseño realizado en STIX busca maximizar la estructura y la consistencia para soportar métodos de procesamiento automático por máquinas.

- Lectura: El diseño de STIX busca estructuras del contenido para que no sea únicamente consumible o procesable por máquinas sino que también sea leíble por humanos. Esto es necesario para claridad y comprensibilidad durante las primeras etapas de desarrollo y adopción.
- Implementación: La implementación inicial de STIX utiliza XML como un mecanismo portátil, estructurado y que se puede encontrar en cualquier parte para la discusión, colaboración y refinamiento entre las comunidades involucradas. Está pensado para el desarrollo en colaboración de un lenguaje estructurado sobre información de amenazas entre expertos de la comunidad. Se ha pensado que el uso del lenguaje sea estimulado y soportado por medio del desarrollo de varias herramientas como APIs. Solo por medio de niveles adecuados de colaboración entre miembros de la comunidad y con la utilización de datos reales se puede llegar a que la solución evolucione.

Cuadro 2.1: Comparativa de STIX e IODEF

STIX	**IODEF**
Se representan observables que son propiedades o eventos que se dan durante las operaciones de redes o computadores. Se refiere a información sobre un archivo, una key en la registry, un servicio iniciado, etc. Para su representación se utiliza CyBox. Se da información de eventos en el host y la red por medio de Observables. Se puede considerar información como nombres, hashes, tamaños de archivos. Keys que se cambian en la registry, servicios que se ejecutan o request realizadas a un host (ie: request http)	Descripción de eventos particulares del incidente que se dan en un host o red. También se puede tener información de Log de los eventos o acciones significativas realizadas por los involucrados.
Se representa información de indicadores que afectan a la organización así como nuevos datos conocidos durante la respuesta al incidente. Se dan datos sobre lo que busco realizar un atacante, fuente de la información del incidente y tareas realizadas por los analistas.	Se da una descripción textual del incidentes. Se da información de contacto de las partes que participaron del incidente. Se da información de la razón por la cual se envía el documento y de cuales son las posibilidades que tiene el receptor para divulgar información.

Continua en la página siguiente

Cuadro 2.1 – *Continuación de la página anterior*

STIX	**IODEF**
Se permite la extensión del modelo para representar datos que no están en éste.	Se permite la extensión del modelo para representar datos que no están en éste.
Se da información sobre socios involucrados.	Información de contacto para personas u organizaciones involucradas en el incidente.
La clase Incidents da representaciones de los tiempos de detección reporte, etc del incidentes.	La clases Time dan información sobre el comienzo, fin, detección del incidente.
Se da una representación del modus operandi del adversario. Esto se realiza por medio de Tactics Technics and Procedures (TTP), en estas se busca representar comportamientos específicos que muestra el adversario, recursos que éste tiene como herramientas e infraestructura, información de las victimas, objetivos de exploits, fuentes de la información de los TTP. Se utilizan otros estándares como CVE, CAPEC o MAEC para la representación de los ataques o del malware.	Una representación libre de la metodología utilizada por el adversario. Se representan las técnicas utilizadas por el atacante.
Por medio de la clase incidents se da información de bienes afectados y de la naturaleza del incidente.	Repercusiones técnicas y no técnicas del incidente, ie: impacto monetario, de tiempo, etc.
Las campaigns dan referencia a actividades similares que fueron realizadas.	Referencia a actividades similares.
Representación de campañas, esto son adversarios con una intención. Las campañas consisten de las intenciones del adversario, sus TTP utilizadas en la campaña, los incidentes realizados en la campaña, indicadores asociados a la campaña.	

Continua en la página siguiente

Cuadro 2.1 – *Continuación de la página anterior*

STIX	IODEF
Representaciones de adversarios que tienen ciertas intenciones y han sido observados históricamente. Los adversarios tienen las siguientes características: una representación de identidad, se sospecha de una motivación, una intención de conseguir algo, tienen un historial de TTP, ciertas campaigns son asociadas con un adversario, etc.	
Se representan objetivos de exploits, estos se pueden ver como vulnerabilidades en software, sistemas, configuraciones de red que son objetivos para ser explotados por las TTP de un adversario. Se utilizan CVE, OSVBD (entre otras) para la identificación de vulnerabilidades publicas.	Se da una clasificación de vulnerabilidades conocidas.
Se representan medidas que deberían ser realizadas para prevenir o corregir para evitar ser objetivo de exploits.	Se dan acciones que debería realizar el receptor.
Se representan etiquetas para la información que indican restricciones o datos potencialmente sensibles.	

2.3. Intercambio

2.3.1. Protocolos

2.3.1.1. RID

Real-Time Inter-Network (RID) es un método de comunicación entre redes para facilitar el intercambio de datos de incidentes. A su vez, busca integrar mecanismos existentes de detección, seguimiento, identificación de fuentes y mitigación que aporta una solución al manejo de incidentes. Combinar estas capacidades en un sistema de comunicación permite incrementar el nivel de seguridad en la red. Las políticas para manejar incidentes son recomendadas y pueden ser acordadas por un consorcio utilizando recomen-

daciones y consideraciones de seguridad.

RID ha sido ampliamente utilizando en comunidades de investigación, pero no ha sido muy adoptado por otros sectores. Fue desarrollado como un mecanismo de comunicación para facilitar la transferencia de información entre distintos proveedores de servicios de Internet para trazar precisa y eficientemente el flujo de paquetes nocivos a lo largo de la red.

Los datos en RID son representados como documentos XML utilizando IODEF. De esta forma se simplifica la integración con otros aspectos del manejo de incidentes.

Se deben utilizar conexiones autenticadas y encriptadas entre sistemas RID para proveer confidencialidad, integridad, autenticidad y privacidad de los datos.

RID requiere una especificación de un protocolo de transporte para asegurar la interoperabilidad entre las organizaciones socias. El RFC 6046, especifica el transporte de mensajes RID sobre HTTPS/TLS.

2.3.1.2. TAXII

El objetivo que plantea *Trusted Automated eXchange of Indicator Information* (TAXII) es extender la capacidad de compartir indicadores, siendo dichos intercambios robustos, seguros y de gran volumen de datos. A su vez, los datos intercambiados deberían ser mas expresivos que en la actualidad.

Con TAXII no se busca crear una comunidad para compartir, sino que se le da a las organizaciones una herramienta que facilite el intercambio entre ellas. TAXII mejora las deficiencias existentes dando especificaciones abiertas y comunes para transportar los mensajes con información. También se provee un conjunto de capacidades como encriptación, autenticación, direccionamiento, alertas y pedidos entre sistemas.

TAXII es un conjunto de especificaciones técnicas y de documentación para permitir el intercambio de información procesable entre organizaciones. Para realizar dichos intercambios, se definen protocolos y formatos de datos que permiten intercambiar información de forma segura. Ha sido diseñado para permitir la interoperabilidad de diferentes soluciones en lugar de ligarse a una tecnología o producto en particular. Además, se busca incentivar a los proveedores de tecnología a incorporar soporte para las especificaciones

de TAXII en sus productos. La información intercambiada ayuda a detectar, prevenir y mitigar amenazas informáticas en tiempo real. Es importante recalcar que no se buscan definir acuerdos para el intercambio de información o gobierno. En su lugar, permite a las organizaciones mejorar el contexto en el que se encuentran respecto a las nuevas amenazas y además compartir la información que ellos elijan con las organizaciones que deseen de forma simple y rápida aprovechando las relaciones y sistemas existentes.

Al igual que en STIX, en TAXII se buscó consenso y participación de la comunidad. TAXII permite el intercambio de información sobre amenazas de forma eficiente y comprensiva por medio de *automatización* y *articulación* de un modelo detallado de información. Para lograr esto, se utiliza una representación estándar de información de amenazas y un framework para soportar el intercambio de datos. El modelo permite el envío y recepción de un conjunto amplio de información de seguridad para soportar las necesidades referentes al intercambio de información. Se le da la libertad a los proveedores de determinar como sus productos producen, consumen o toman ventaja de los flujos de información especificados por TAXII.

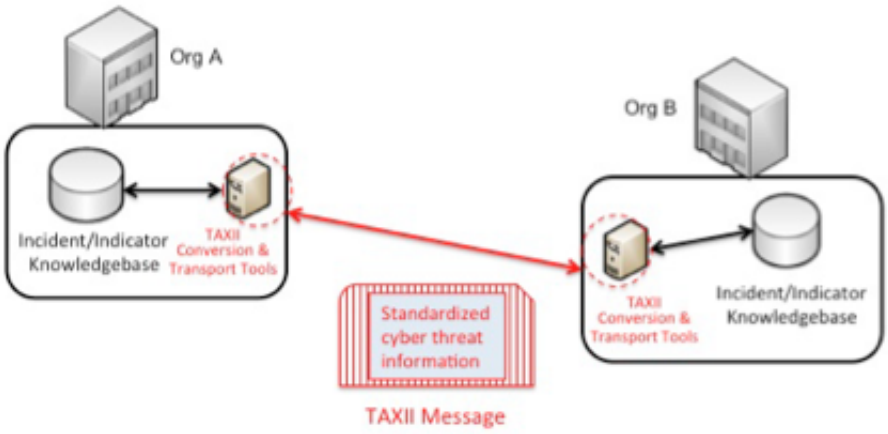

Figura 2.1: Arquitectura de TAXII [32]

TAXII utiliza protocolos y especificaciones existentes siempre que es posible, de esta forma se integra con mecanismos de intercambio de información para reducir los costos de implementación y permitir la adopción rápida por parte de organizaciones ya establecidas y que se encuentran intercambiando información.

Las motivaciones para tener una mejor solución que permita intercambiar información lleva a que en el diseño de TAXII se hayan planteado los siguientes objetivos:

- Permitir el intercambio seguro y rápido de información referente a amenazas entre comunidades de defensores de seguridad.
- Lograr un estándar para permitir compartir indicadores entre organizaciones.
- Extender el intercambio de indicadores para permitir intercambios seguros, robustos y de gran volumen que tengan una expresividad mayor a la actual.
- Soportar un amplio número de casos de uso y prácticas comunes a las comunidades.
- Tomar los estándares existentes que sean adecuados.
- Llegar a una adopción por parte de organizaciones internacionales de estándares.

Para automatizar el intercambio de información, es necesario especificar como ésta es compartida. Para lograr esto, TAXII define especificaciones técnicas y documentación de soporte. En particular, las especificaciones de TAXII definen un conjunto de capacidades necesarias para el transporte exitoso de mensajes. Los mensajes TAXII llevan datos de amenazas informáticas representados por medio de STIX. El conjunto completo de los mensajes incluyen mensajes con datos y de control.

2.3.1.2.0.1. TAXII Toolkit

Es provisto para soportar la adopción de TAXII y asistir en el desarrollo de capacidades compatibles. El *toolkit* provee una colección de implementaciones de referencia, un conjunto de herramientas y una colección de librerías e interfaces.

TAXII está definido por múltiples especificaciones relacionadas. Esta sección describe las especificaciones definidas en TAXII.

- Especificación de servicios: Provee los requerimientos por los cuales se definen los servicios e intercambios de TAXII. No provee detalles respecto al formato de los datos o como los mensajes TAXII son transportados

por la red. Dichos detalles y requerimientos pueden ser encontrados en la especificación de los protocolos de enlace y en la especificación de mensajes de enlace.

- Especificación de protocolos de enlace: Define los requerimientos para transportar mensajes TAXII por la red. Puede haber varias especificaciones creadas para TAXII. Cada especificación define requerimientos para el transporte de mensajes TAXII utilizando protocolos de red y se proveen requerimientos respecto a como los servicios TAXII son soportados por los protocolos de red.

- Especificación de mensajes de enlace: Se definen requerimientos para representar mensajes TAXII en un formato particular. Puede haber múltiples especificaciones para dichos mensajes. Se provee información detallada sobre como las especificaciones definidas en la especificación de servicios son expresadas en los mensajes.

Para dar flexibilidad en el proceso evolutivo de TAXII, se han separado las especificaciones de servicios, de los protocolos de enlace y de los mensajes de enlace. Debido a que las organizaciones generalmente tienen restricciones respecto a los protocolos que soportan, TAXII busca no ligarse a un único protocolo que excluya a una parte de la comunidad. Cuando se ve que la comunidad expresa interés en un nuevo protocolo o tipo de mensaje, TAXII puede dar soporte para ellos sin cambiar los componentes centrales.

Dos grupos que usen el mismo protocolo de red y formato de mensajes serán capaces de intercambios de información estructurada de forma automática. Las políticas de intercambio de los participantes pueden limitar estos intercambios si es necesario, pero el uso de servicios compatibles con TAXII asegura que se puede intercambiar cualquier información con los mecanismos definidos por TAXII. Los grupos que usen diferentes protocolos o formatos de mensajes no serán capaces de comunicarse directamente, pero como están utilizando mensajes y servicios en el núcleo de las comunicaciones de sus comunidades significa que es posible establecer caminos para que ocurra la interacción.

2.3.1.2.0.2. Especificación de Servicios

Esta especificación provee normativas respecto a los servicios, mensajes e intercambios de mensajes en TAXII. No provee detalles respecto a como los mensajes son transportados, dejando eso a la especificación de los protocolos

de enlace. Se da información respecto a los datos presentes en los mensajes TAXII y no a como los mensajes son expresados.

Las unidades funcionales de TAXII representan conjuntos discretos de actividades requeridas para soportar TAXII. Una unidad funcional representa algún componente con un rol bien definido en TAXII.

- TAXII Transfer Agent (TTA): Es una unidad funcional conectada a la red que envía o recibe mensajes TAXII. Una TTA interactúa con otras TTAs por medio de la red y maneja los requerimientos de una o más de las especificaciones de los protocolos de enlace. Una TTA provee un mensaje TAXII a un *TAXII Message Handler* permitiendo que éste último sea independiente del protocolo de red utilizado. De la misma forma, el TTA puede ser independiente del contenido de los mensajes TAXII, dejando el manejo de la información al *TAXII Message Handler*.

- TAXII Messsage Handler (TMH): Es una unidad funcional que produce y consume mensajes TAXII. EL TMH es responsable de parsear y construir mensajes con el formato especificado en uno o más TAXII *Message Binding Specifications*. Un TMH interactúa con un TTA, el cual maneja los detalles necesarios para transmitir mensajes por la red. El Backend TAXII interactúa con el TMH para convertir su contenido en mensajes TAXII y llevar a cabo actividades basadas en los mensajes TAXII que son recibidos por el TMH.

- Backend TAXII: Cubre todas las unidades funcionales distintas al TTA y al TMH. Las especificaciones de TAXII no proveen requerimientos sobre como son implementadas las capacidades en un backend más allá de como debe interactuar con el TMH. Las organizaciones o implementadores pueden decidir que capacidades implementar según los servicios TAXII que deseen soportar o según como quieran dar ese soporte.

- Arquitectura TAXII: Cubre los aspectos de las unidades funcionales de la infraestructura de productor o consumidor que provee o utiliza servicios TAXII. Una arquitectura TAXII incluye una TTA, un TMH y un backend TAXII.

Lo expresado anteriormente se puede ver en la figura 2.2.

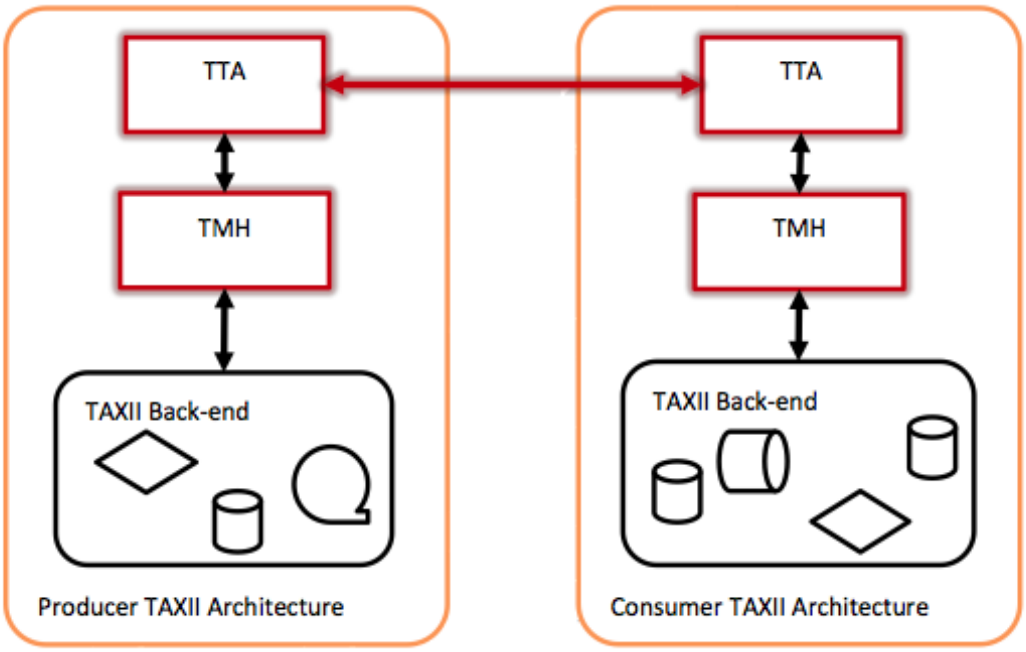

Figura 2.2: Unidades funcionales de TAXII [32]

2.3.1.2.1. Capacidades

TAXII provee capacidades específicas para aquellos que desean compartir información de amenazas cibernéticas. Las capacidades TAXII son el nivel más alto en el cual se pueden expresar las acciones de TAXII. Hay tres capacidades que soporta la actual versión de TAXII, estas son: *push messaging*, *pull messaging* y *discovery*.

En ***push messaging*** la información puede ser enviada de un productor a un consumidor. Esto puede reflejar una relación pre-existente entre el productor y el consumidor en la que el consumidor ha pedido que se le envíen datos desde el productor. También puede usarse en caso de que el consumidor desee aceptar contribuciones de cualquier productor, y estos le envíen datos en cualquier momento.

Pull messaging permite a un consumidor requerir información de un productor. Esto no solo le permite al consumidor el control sobre el momento en el que recibe los datos sino que también le permite hacerlo sin tener que aceptar conexiones entrantes. Así como en *push messaging*, el productor

y consumidor pueden tener acuerdos pre-existentes para que el consumidor tenga acceso a los datos del productor. De forma alternativa, un productor puede hacer su información pública de forma que cualquier consumidor pueda obtenerla. La versión actual de *pull messaging* limita a los consumidores a hacer pedidos por medio de las organizaciones productoras de los datos en lugar de por los datos en si. Toda la información provista por un productor debe estar organizada en grupos llamados "TAXII Data Feeds". Cada elemento en un TAXII data feed es etiquetado utilizando *timestamps*. El productor tiene total dominio sobre como el contenido se mapea en TAXII data feeds y en el significado de los *timestamps*. La capacidad de *pull messaging* está atada a entender el contenido del productor.

Para facilitar las comunicaciones automatizadas, TAXII soporta capacidades para descubrir los servicios específicos que ofrece un servidor o grupo de servidores, así como los protocolos o mensajes que este servidor ofrece. Esto no quita la necesidad de que personas estén involucradas para establecer acuerdos de cooperación lo cual esta por fuera del objetivo de TAXII. Sin embargo, permite el intercambio de información respecto a las capacidades que un productor soporta y cuales son los mecanismos que utiliza para hacerlo.

2.3.1.2.2. Servicios TAXII

Los servicios TAXII representan un conjunto de mecanismos necesarios para soportar capacidades TAXII. Una implementación TAXII pudiera implementar alguno, todos o incluso ninguno de los servicios definidos. TAXII define los siguientes servicios:

- Servicio de descubrimiento: Es utilizado para recibir y responder a mensajes que requieren información sobre los servicios ofrecidos.
- Feed Managment Service: Es utilizado para recibir o responder a mensajes utilizados para el manejo de subscripciones a TAXII Data Feed.
- Inbox Service: Es utilizado para recibir información de amenazas cibernéticas por medio de intercambios iniciados por el productor en intervalos dictados por este.
- Poll Service: Es utilizado para recibir y responder a mensajes de pedido a el TAXII Data Feed iniciados por el consumidor.

A continuación se describen los distintos servicios.

2.3.1.2.2.1. Discovery Service

Es un mecanismo para comunicar información referente al uso de servicios TAXII y a su disponibilidad. Para un pedido al servicio, se retorna una lista de los servicios TAXII y como estos pueden ser invocados. Un solo servicio de descubrimiento puede reportar servicios TAXII en diferentes equipos finales o incluso en múltiples organizaciones, los propietarios del servicio pueden definir su alcance a gusto. Un servicio de descubrimiento puede utilizar varios factores para determinar cuales servicios revelar ante una petición, incluyendo, pero no limitado a la entidad del cliente TAXII. El servicio de descubrimiento debe soportar "Discovery Message Exchange".

2.3.1.2.2.2. Feed Managment Service

Es el mecanismo con el cual un consumidor pide información referente a TAXII Data Feeds, pidiendo subscripciones a estos, o modificando las existentes. Este servicio facilita el intercambio de mensajes para manejar las subscripciones. No se entrega contenido de los TAXII Data Feed, en su lugar se envía contenido del TAXII Data Feed al servicio de Inbox de un consumidor en intercambios iniciados por un productor o en respuesta directa a un pedido del consumidor al servicio de poll. Dicho servicio debe implementar soporte para *subscription managment exchange* y podría implementar soporte de *feed information exchange*.

2.3.1.2.2.3. Inbox service

Este servicio es el mecanismo con el cual un consumidor acepta los mensajes en un intercambio iniciado por el productor. Un consumidor puede implementarlo para recibir datos del TAXII Data Feed. El servicio de inbox debe implementar soporte para *Data Push Exchange*.

2.3.1.2.2.4. Poll service

Es provisto por un productor para permitir pedidos al TAXII Data Feed iniciados por el consumidor. Un consumidor contacta a este servicio explícitamente pidiendo el contenido del TAXII Data Feed. Los productores podrían ofrecer Data Feeds combinando envíos al Inbox service del consumidor o por medio de pedidos al servicio de poll del productor. Una implementación de este servicio debe dar soporte a *Data Poll Exchange*.

2.3.1.2.3. Intercambio de mensajes TAXII

Esta sección describe los mensajes intercambiados que son necesarios para soportar los servicios definidos antes. Estos intercambios solo consideran mensajes TAXII y son independientes a los protocolos sobre los cuales viajan los mensajes. En particular, esos protocolos podrían requerir intercambios de red adicionales antes de transmitir mensajes TAXII o romper un mensaje TAXII en múltiples mensajes del protocolo subyacente que son transmitidos independientemente.

2.3.1.2.3.1. Data Push Exchange

En este intercambio, un mensaje STIX es transmitido desde un cliente a un servidor inbox que está esperando. El mensaje STIX puede ser solicitado o no solicitado. El servidor inbox puede ser capaz de filtrar mensajes según la autenticidad del emisor.

2.3.1.2.3.2. Discovery Exchange

Un cliente TAXII pide información sobre el servicio TAXII ofrecido por un productor. El discovery server del productor responde con una lista de servicios.

El cliente TAXII envía un pedido de descubrimiento al servidor. El Backend TAXII podría utilizar esta información junto a su propia política de control de acceso para crear una lista de servicios a ser retornada. Estos son empaquetados en una respuesta de discovery la cual es enviada al cliente TAXII. El cliente TAXII recibe esa respuesta y la pasa la información del servicio a su propio BackEnd para ser procesado.

2.3.1.2.3.3. Feed Information Exchange

En este intercambio un cliente TAXII pide información sobre fuentes de datos disponibles en un Feed Server. El servidor responde con una lista de fuentes de datos de las que dispone. Dicha respuesta es realizada por el backend y en ella se pueden considerar decisiones de control de acceso.

2.3.1.2.3.4. Subscription Managment Exchange

En este un cliente intenta establecer, borrar, pausar, resumir o modificar una subscripción a un TAXII Data Feed conocido enviando un mensaje subscription managment request al servidor. El servidor pasa la request al Backend TAXII el cual determina la respuesta, la cual es luego enviada al

cliente. El backend TAXII puede usar dicha información junto con sus políticas de control de acceso y las funcionalidades que posea para determinar si la acción está permitida o no.

2.3.1.2.3.5. Feed Poll Exchange Es utilizado por un consumidor para pedir contenido de un productor de datos. El TAXII Data Feed content es enviado al consumidor en el mismo intercambio. El cliente consumidor inicia el El backend TAXII evalua el pedido de información información para determinar la respuesta. La respuesta retorna mensajes STIX con el contenido que pidió el cliente.

2.3.2. Modelos

2.3.2.1. Comunidades

Actualmente el número de organizaciones que buscan compartir información de amenazas es creciente. Esto lleva a que el número y tipo de comunidades que busca hacerlo también se incremente. En la actualidad se identifican tres tipos de comunidades:

- Pares
- Comerciales
- Gubernamentales

Para que se pueda compartir información entre dichas organizaciones es necesario cierto nivel de confianza dado que compartir información sensible podría exponer a las organizaciones a daños en su reputación, demandas o advertir a un atacante de la investigación que se lleva a cabo haciendo que el trabajo realizado haya sido inútil. Se deben definir medidas para la protección de los datos como restricciones en su manejo, sanitización y el establecimiento de confianza entre las dos organizaciones. Lo mencionado anteriormente es particularmente importante cuando las organizaciones forman parte de varias comunidades que intercambian información, se encuentran casos en los que datos compartidos con una organización no deberían ser compartidos con otra.

Las comunidades entre **pares** son las más comunes, estas organizaciones o individuos tienen el propósito común de mejorar las defensas colectivas contra adversarios que tienen en común. La información compartida por dichas organizaciones es más especifica que la provista por organizaciones comerciales.

Las comunidades **comerciales** son anónimas y los miembros poseen algún tipo de acuerdo común, por ejemplo el pago de cuotas para pertenecer a la comunidad. La organización comercial maneja la información de forma centralizada y la distribuye entre los miembros de la organización. El acceso a la información por parte de los socios es rápido teniendo la posibilidad de que dicha información sea más amplia que la compartida por pares y que además no siempre sea aplicable a las necesidades de la organización.

Las comunidades **gubernamentales** son establecidas y manejadas por el gobierno, son voluntarias u obligatorias e incluyen participantes tanto del gobierno como de la industria privada. En ellas el gobierno controla la información y la distribución de esta. Así como en las comunidades comerciales, la información y los participantes son altamente confidenciales.

2.3.2.2. Modelos

Se pueden identificar tres modelos para el intercambio de información entre organizaciones:

- Hub and Spoke
- Peer to peer
- Source/subscriber

En el método **hub and spoke**, la entidad hub controla la recepción y diseminación de los datos, además se encarga de mantener anónimos y proveer un análisis adicional de los datos recolectados para luego diseminarlos entre los participantes. Este modelo es comúnmente visto en comunidades de gobierno o comerciales.

En el modelo **peer to peer**, los participantes intercambian y reciben información directamente de los otros participantes. La información es compartida entre todos los miembros de la comunidad por igual y la fuente está claramente identificada.

Source/subscriber es utilizado por las comunidades comerciales que proveen de información. El proveedor de información envía regularmente información a todos los subscriptores y estos podrían eventualmente enviarle información a la fuente. Generalmente, la información se codifica de una manera propietaria y puede faltar información esencial sobre algunos intentos

de irrupción. Presenta la ventaja de que se tiene acceso rápido a un conjunto de datos amplio y es útil para organizaciones con recursos limitados.

Capítulo 3

Análisis

A continuación se presentan los aspectos más importantes que se tuvieron en cuenta para el desarrollo del proyecto. Se presentan características deseadas en un sistema de intercambio de información de seguridad entre organizaciones.

3.1. Análisis de requerimientos

Existen distintos problemas que obstaculizan el intercambio de información entre las organizaciones. Dichas problemáticas afectan la reputación, seguridad y la capacidad de trabajo de las organizaciones.

Las organizaciones pueden intercambiar grandes volúmenes de datos los cuales pueden provenir de diferentes fuentes, cada una de estas fuentes puede utilizar una representación propia para los datos. Esto puede generar problemas para interpretar infromación proveniente de otra organización, y por ello es deseado que las organizaciones utilicen un estándar aceptado por todas. La representación utilizada debería permitir el desarrollo de herramientas que ayuden en la estructuración de los datos de forma de facilitar el trabajo de los analistas.

Otro de los problemas que interfieren en el intercambio de información es el riesgo a la seguridad y reputación de la organización, dada por la divulgación de información sensible o privada. Por esto la información intercambiada debe pasar por procesos que controlen los datos que se intercambien evitando dicha problemática.

Estas son algunas de las problemáticas que se pueden identificar referentes

al intercambio de información de seguridad. También existen otros problemas que no serán analizados en este documento, como por ejemplo establecer un criterio referente a políticas organizacionales que ayuden a identificar organizaciones de confianza. Este problema no será analizado en este documento.

La herramienta que se desea desarrollar busca ayudar a las organizaciones a solucionar algunos de los problemas planteados anteriormente y que afectan el intercambio de información con sus pares. Además es deseado que la herramienta pueda ser extendida en un futuro con nuevas funcionalidades que solucionen otras problemáticas no identificadas o que no se desarrollen en el transcurso de este proyecto.

Se desea desarrollar una herramienta que se integre con alguna de las aplicaciones de gestión de incidentes existentes, proveyéndole así la capacidad de intercambiar información de seguridad. Dicha herramienta debería estructurar y organizar la información de forma de facilitar el intercambio. Además se debería dar la posibilidad de correlacionar la información de forma de facilitar el trabajo de los analistas. Una funcionalidad importante en una herramienta de estas características es la sanitiziación de los datos compartidos, con el fin de proteger la integridad de la organización durante los intercambios.

Los intercambios podrían proveer información sobre la identificación de nuevas vulnerabilidades o entidades maliciosas, soluciones a problemas, prevención de problemas, etc.

El intercambio de información no es un problema estrictamente técnico ya que hay procedimientos y consideraciones legales y de confianza que podrían afectar el intercambio de información entre organizaciones. Durante el estado del arte se investigaron distintos protocolos y lenguajes para la representación de la información, se llegó a la conclusión de que ninguno de estos daba soluciones referentes a las políticas organizacionales que solucionaran problemas como la confianza entre las organizaciones o la información que debe ser compartida.

A pesar de lo mencionado anteriormente, es deseable que un sistema que comparta información de seguridad respete y aplique las políticas organizacionales. Por ello es necesario que el sistema aplique políticas definidas por los administradores para sanitizar y anonimizar la información con el fin de remover datos confidenciales o sensibles antes de que sean compartidos. Para resolver este problema se debe evaluar la protección que se le quiere dar a la

información y considerar a su vez cuan útil es dicha información luego de ser sanitizada.

Del análisis anterior se desprende la necesidad de contar con un módulo que permita a la aplicación sanitizar la información. Como se mencionó anteriormente, los administradores definen dichas políticas en el sistema. La finalidad del módulo es analizar la información y filtrar datos que pudieran ser sensibles y que pusieran en riesgo los intereses de la organización.

Durante el intercambio de información, se pueden obtener datos provenientes de diversas fuentes que se refieren a distintos tipos de información, pero que guarden una relación entre ellos. Por ello es necesario contar con un módulo que se encargue de relacionar la información por medio de la aplicación de estrategias. El resultado de la aplicación de estrategias es la agrupación de los datos, dicha agrupación permite a los analistas manejar una menor cantidad de datos y de esta forma simplificar su trabajo. Esto ayuda a bajar el periodo de tiempo entre la detección del problema y su solución. A su vez, permite detectar ataques que pudieran pasar desapercibidos en volúmenes muy grandes de información.

Si bien la información correlacionada es de utilidad para los analistas, es necesario contar con toda la información recibida para poder hacer un análisis de los datos originales por parte de los analistas.

Además de recibir información proveniente de otras organizaciones, es deseable que se pueda ingresar nueva información al sistema. Dicho ingreso de información se pretende realizar por medio del sistema de gestión de incidentes, a su vez se desea mantener una representación estructurada de la información ingresada en el incidente. Mantener la información de forma estructurada permite realizar la correlación con datos recibidos de otra organización. Esto permitiría por ejemplo ayudar a solucionar un problema para el cual otra organización tenga una solución.

De lo anterior se pueden identificar los siguientes requerimientos funcionales:

Requerimientos funcionales
▪ La herramienta debe implementar un modelo peer-to-peer de intercambio de información entre organizaciones. ▪ Se debe dar la posibilidad de sanitizar la información intercambiada por medio de políticas definidas por el administrador. ▪ Es deseable contar con un módulo para correlacionar la información de los incidentes configurable por el administrador. ▪ Dar la posibilidad de agregar, editar y borrar información de seguridad al sistema.

Tabla 1 - Requerimientos funcionales del sistema.

También se pueden ver los siguientes requerimientos no funcionales:

Requerimientos no funcionales
▪ Extensibilidad: Debe ser posible extender la herramienta con nuevos módulos que implementen nuevas funcionalidades. ▪ Independencia del sistema de gestión de incidentes para que exista la posibilidad de utilizar otra herramienta. ▪ Utilizar estandares para la representación y el transporte de la información de seguridad.

Tabla 2 - Requerimientos no funcionales del sistema.

3.2. Herramientas

En esta sección se muestran las herramientas utilizadas. STIX fue elegida porque provee una representación estructurada y estándar de la información. Junto con STIX se eligió TAXII el cual ha sido diseñado para intercambiar información de seguridad representada por medio del lenguaje STIX y que tiene consideraciones para realizar el intercambio.

Otra de las razones por las cuales se decidieron utilizar dichas herramientas fueron su aceptación por parte de la comunidad y el trabajo que se puede llegar a desarrollar en base a ellas en un futuro.

3.2.1. RTIR

RTIR es un sistema de manejo de incidentes diseñado para ser utilizado por CSIRTs para manejar el creciente número de incidentes reportados. Si bien existen otras herramientas similares, RTIR presenta la ventaja de ser *opensource* y contar con una API que permite extender la herramienta de forma sencilla. También es posible desarrollar *plugins* para extender las funcionalidades de la herramienta. RTIR cuenta además con una comunidad de usuarios grande cuya característica principal es el nivel técnico de estos.

Distintos CSIRTs han contribuido en el desarrollo de la herramienta, el resultado ha sido una herramienta que posee un *workflow* para el manejo de incidentes de seguridad. Dicho *workflow* facilita el trabajo de los CSIRTs.

Como se mencionó en el capítulo 2, RTIR no cuenta con una representación estructurada de la información ni tampoco con un método automático para compartir información de seguridad. Una integración con TAXII y STIX le permitiría cubrir estas dos falencias.

El interés de usar RTIR proviene de que el CSIRT-Tilsor tiene la herramienta instalada y la utiliza para sus operaciones. Además hay miembros del equipo que tienen experiencia en su uso.

A su vez, al ser una herramienta con una profunda inserción en la comunidad, es esperable que sea más fácil la aceptación de una extensión basada en TAXII y STIX que la creación de una nueva herramienta a la que los usuarios deberán adaptarse.

Si bien RTIR fue una premisa dentro de los objetivos del proyecto, se evaluaron durante el estado del arte otras herramientas que pudieran tomar su lugar. De todas formas, del análisis realizado, se eligió RTIR por las razones dadas anteriormente. A pesar de utilizarse RTIR, es deseable que la herramienta desarrollada no sea dependiente de RTIR, esto quiere decir que se pueda utilizar una herramienta con funcionalidades similares en el futuro.

3.2.2. STIX y TAXII

Se decidió utilizar STIX por la facilidad con la que permite representar información de seguridad de forma estructurada y estándar. Por medio de CybOX se permite describir evidencia en la forma de observables, artefactos y/o comportamientos presentes en un sistema. La representación de forma precisa se debe a la gran gama de objetos distintos que permite describir, entre ellos, el nombre de procesos ejecutándose, hash de archivos o mensajes ICMP. Como se vio en el estado del arte, STIX también permite representar incidentes, tácticas, técnicas y procedimientos de los adversarios, actores maliciosos entre otros conceptos que permiten representar adecuadamente información de seguridad.

Otras de las características que posee STIX son la extensibilidad, la simpleza y la facilidad de procesamiento. Dichas características son propias de un lenguaje XML.

TAXII define un conjunto de servicios e intercambios de mensajes que permiten el intercambio de información de seguridad. En la especificación de TAXII se establece que la información de seguridad es representada por medio de STIX. Además TAXII posee consideraciones de seguridad para realizar el intercambio como lo son encriptación y autenticación.

TAXII realiza el intercambio de conjuntos de información de seguridad llamados "TAXII Data Collections". Dichas colecciones de datos pueden ser conjuntos ordenados de información ("TAXII Data Feeds"), en los cuales el criterio de ordenación es un *timestamp*, o conjuntos desordenados ("TAXII Data Sets"). La información en dichos conjuntos es representada de forma

estructurada utilizando el lenguaje STIX. Durante los intercambios de información es necesario que el cliente pida información de una de las colecciones de las que dispone el productor. Por ello es necesario que se de en el sistema la posibilidad de gestionar las colecciones a las que un cliente se suscribe.

STIX también integra con otras iniciativas de MITRE e incluso se integra con lenguajes de otras organizaciones como IODEF de IEEE y OpenIOC de Mandiant.

Es importante mencionar que STIX ha tenido un fuerte apoyo de la comunidad y busca convertirse en un estándar. Actualmente existen esfuerzos para crear herramientas que utilicen el lenguaje STIX y que realice un intercambio de información por medio de TAXII.

Con las herramientas mencionadas anteriormente podemos ver un diagrama de bloques como el de la figura 3.1. En la figura se puede ver que se cuenta con una instalación de la herramienta RTIR que será la encargada de la gestión de incidentes y por medio de la cual se dará de alta la información en el sistema. Además en el bloque "TAXII App" realiza el intercambio de información con otras organizaciones por medio de TAXII y representa la información utilizando STIX. Además dicho bloque es el encargado de realizar la sanitización y correlación de información.

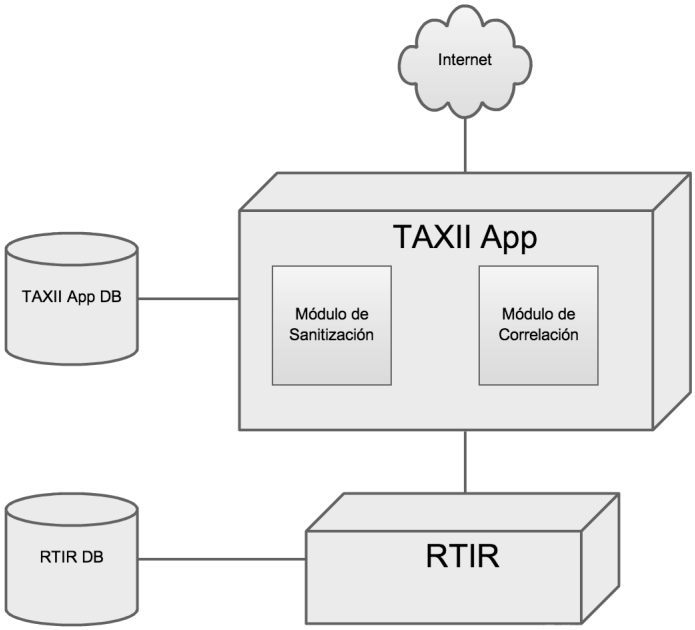

Figura 3.1: Diagrama de Bloques del sistema

3.3. Actores y Casos de Uso

3.3.1. Actores

Actor	Analista
Descripción	Este actor tiene la posibilidad de ingresar nueva información en el sistema. Dicha información puede ser intercambiada con otro sistema. Con la información que se ha intercambiado el actor puede realizar un análisis de ella y hacer un manejo de los casos creados en el RTIR.

Actor	Cliente TAXII
Descripción	Este actor es el que interactúa con el sistema para intercambiar datos por medio del protocolo TAXII. El sistema tiene que dar soporte para dicho protocolo para que el intercambio sea exitoso.

3.3.2. Casos de uso y diagramas de secuencia

3.3.2.1. ABM de políticas de sanitización

Nombre	ABM de políticas de sanitización
Actor	Analista
Descripción	Estos casos de uso comienzan cuando el analista desea realizar el alta, baja o modificación de las políticas de sanitización. Por medio de estas se filtra la información que se desea intercambiar con otras organizaciones.

En la figura 3.2 se especifica el caso de uso, en este un analista ingresa a RTIR y da de alta en el sistema una política de sanitización. Dicha política es utilizada para realizar la sanitización de la información intercambiada por la organización. Las políticas son registradas en TAXII App.

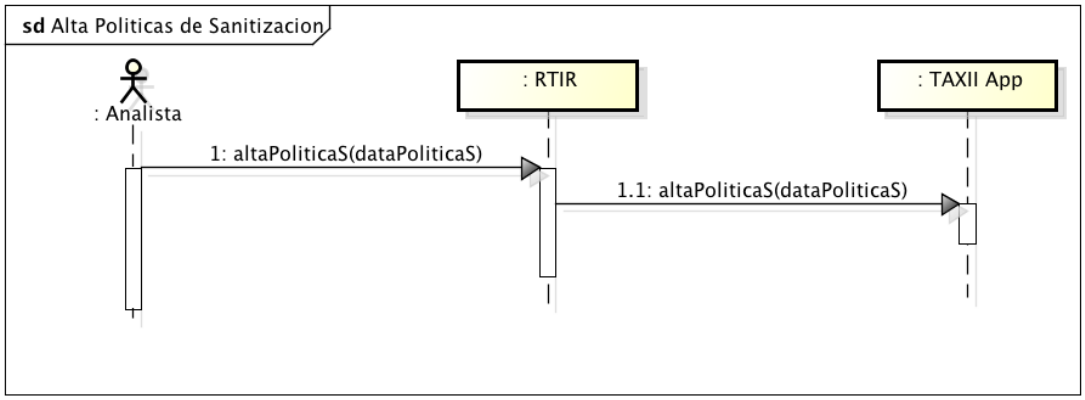

Figura 3.2: Caso de Uso alta políticas de sanitización

En la figura 3.3 se especifica el caso de uso de borrado de políticas de sanitización, en este un analista ingresa a RTIR y lista todas las políticas

disponibles en el sistema. Luego selecciona la política que desea dar de baja para luego borrarla.

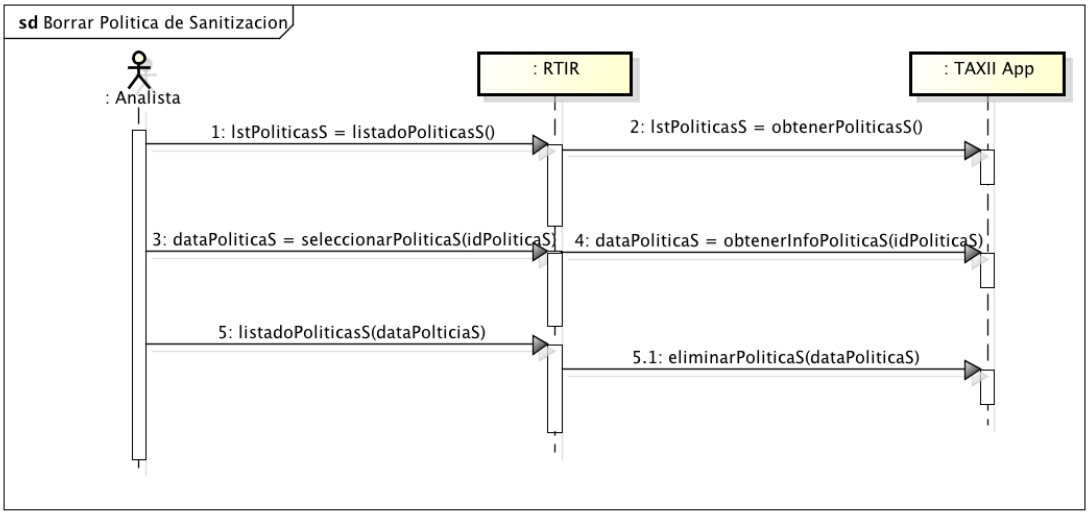

Figura 3.3: Caso de uso borrado políticas de sanitización

En la figura 3.4 se especifica el caso de uso de modificación de políticas de sanitización, en este un analista ingresa a RTIR y lista todas las políticas disponibles en el sistema. Luego selecciona la política que desea modificar.

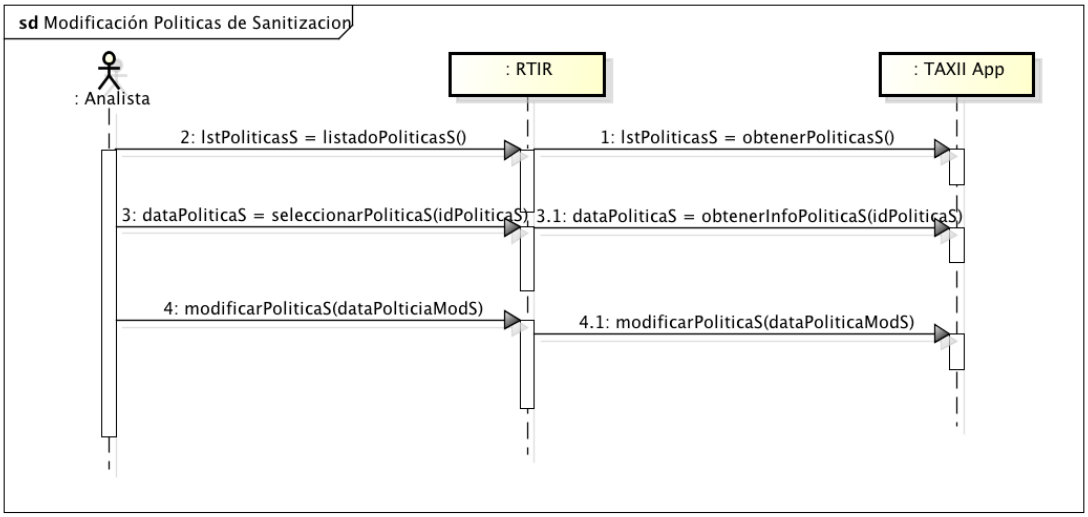

Figura 3.4: Caso de uso modificación de políticas de sanitización

3.3.2.2. ABM de Políticas de Correlación

Nombre	ABM de políticas de correlación
Actor	Analista
Descripción	Estos casos de uso comienzan cuando el analista desea realizar el alta, baja o modificación de las políticas de correlación. Por medio de estas se agrupa la información según los datos existentes en el sistema.

En la figura 3.5 se especifica el caso de uso de borrado de políticas de correlación, en este un analista ingresa a RTIR y lista todas las políticas disponibles en el sistema. Luego selecciona la política que desea dar de baja para luego borrarla.

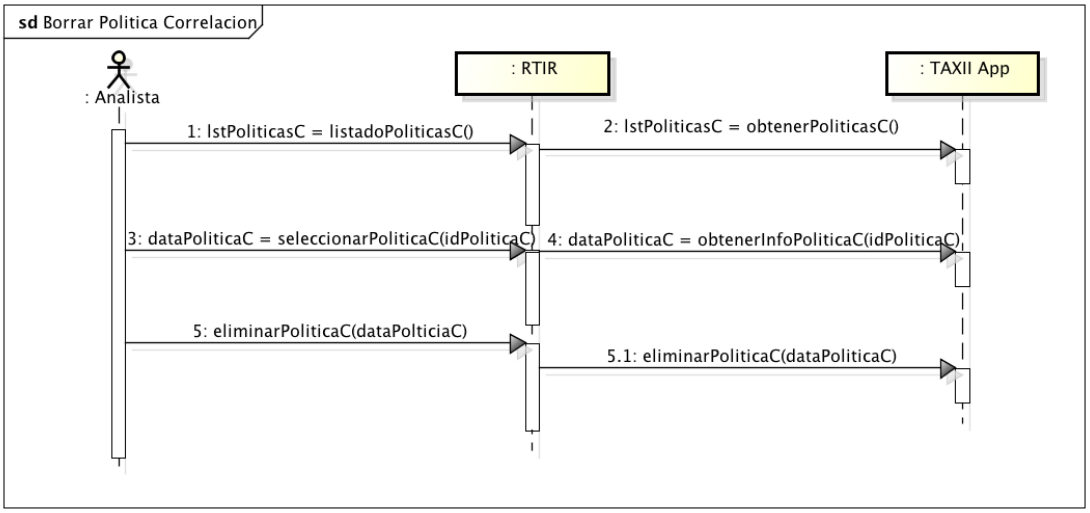

Figura 3.5: Caso de uso borrado políticas de correlación

En la figura 3.6 se especifica el caso de uso de modificación de políticas de correlación, en este un analista ingresa a RTIR y lista todas las políticas disponibles en el sistema. Luego selecciona la política que desea modificar.

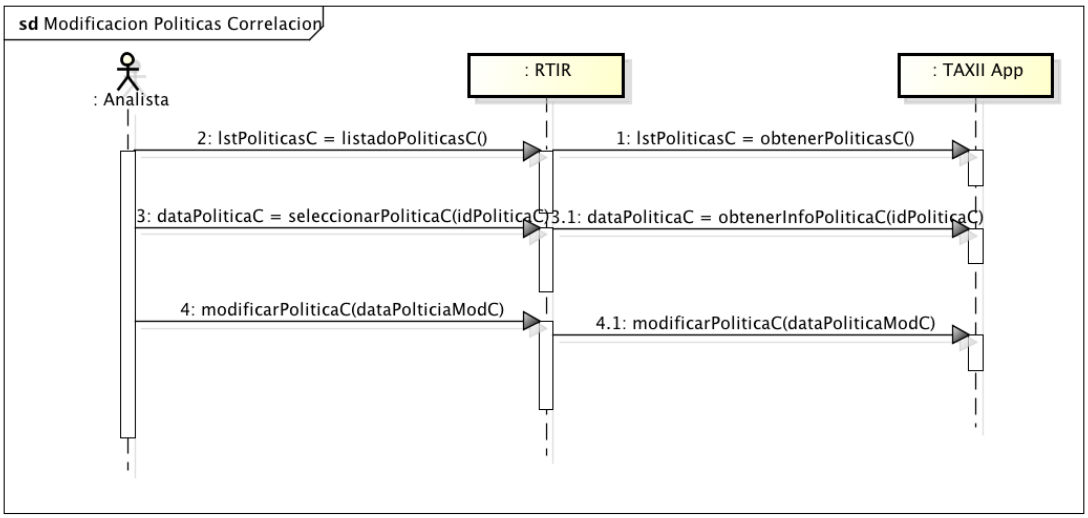

Figura 3.6: Caso de uso modificación políticas de correlación

En la figura 3.7 se especifica el caso de uso de alta de políticas de correlación, en este un analista ingresa a RTIR y da de alta en el sistema una política de correlación. Dicha política es utilizada para realizar la correlación de la información presente en el sistema. Las políticas son registradas en TAXII App.

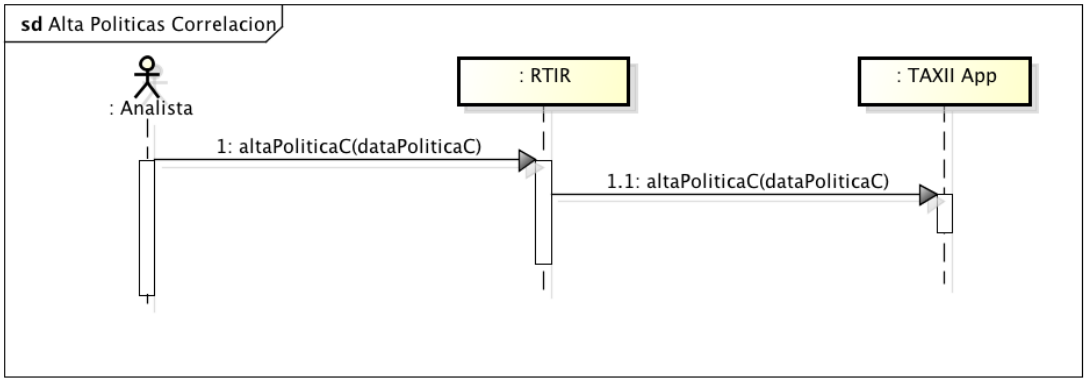

Figura 3.7: Caso de uso alta políticas de correlación

3.3.2.3. ABM de Servicios TAXII

Nombre	ABM de servicios TAXII
Actor	Analista
Descripción	Estos casos de uso comienzan cuando el analista desea realizar el alta, baja o modificación de servicio TAXII de otras organizaciones en el sistema. Estos serán utilizados para lograr el intercambio de información.

En la figura 3.8 se especifica el caso de uso de alta de servicios TAXII, en este un analista ingresa a RTIR y da de alta en el sistema un servicio TAXII, con dichos servicios se realizara el intercambio de información con otra organición

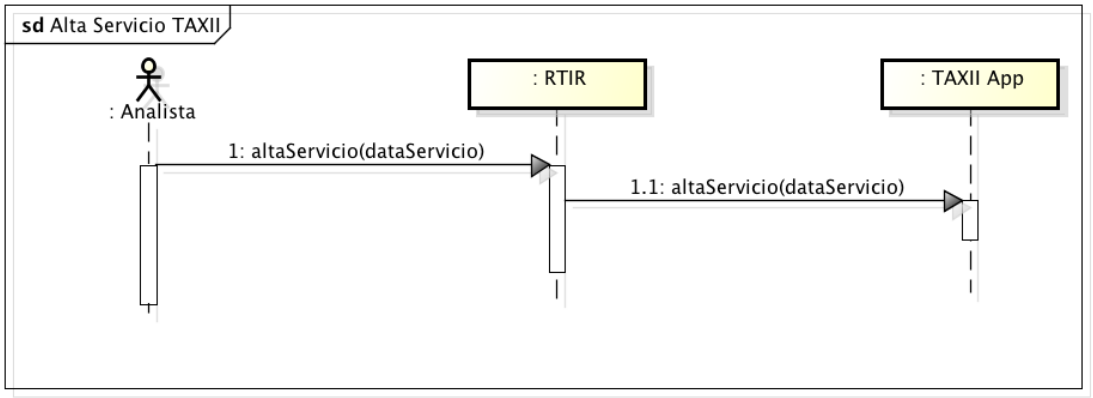

Figura 3.8: Caso de uso alta servicio TAXII

En la figura 3.9 se especifica el caso de uso de borrado de servicios TAXII, en este un analista ingresa a RTIR y lista todas las servicios disponibles en el sistema. Luego selecciona el que desea dar de baja.

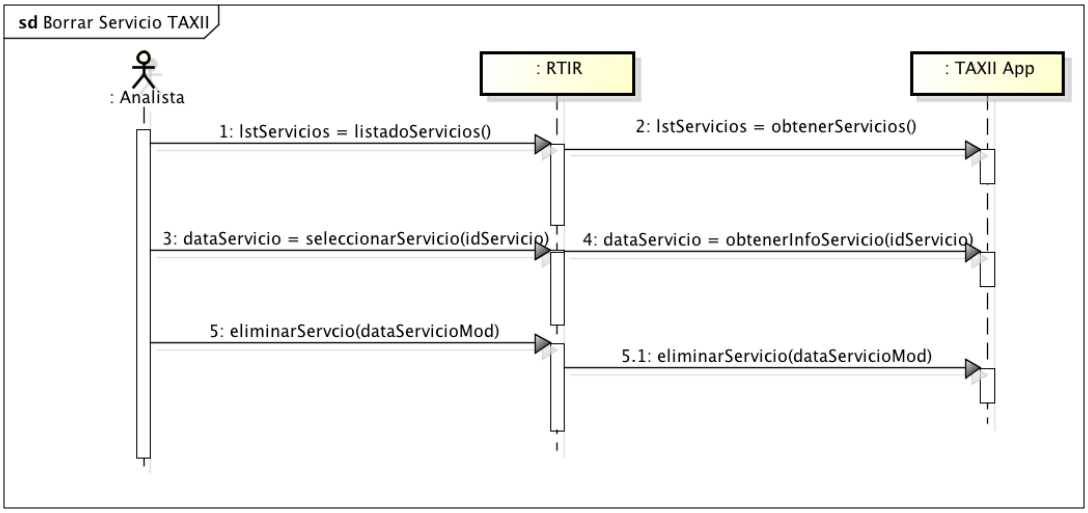

Figura 3.9: Caso de uso borrar servicio TAXII

En la figura 3.10 se especifica el caso de uso de modificación de servicios TAXII, en este un analista ingresa a RTIR y lista todos los servicios disponibles. Luego selecciona el que será modificado.

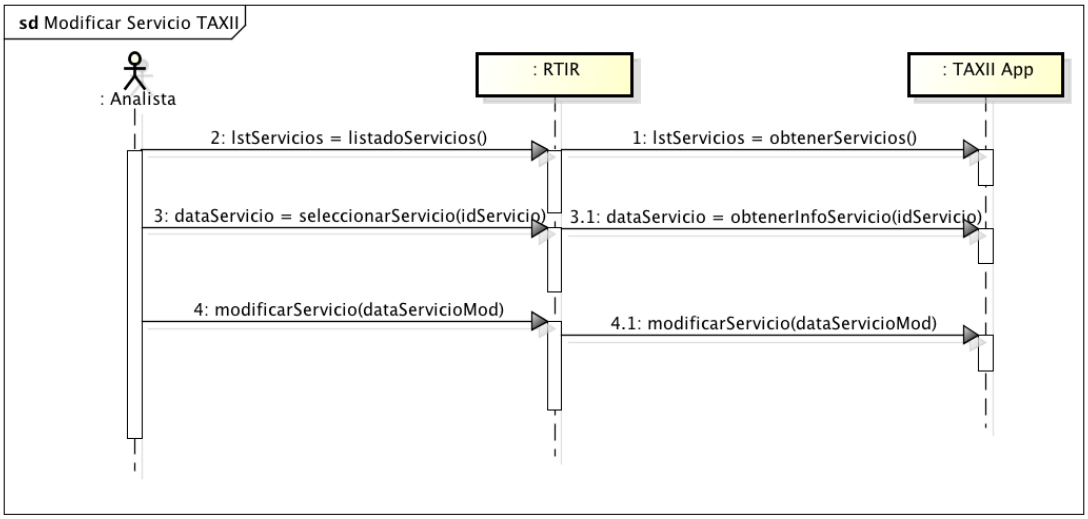

Figura 3.10: Caso de uso modificar servicio TAXII

3.3.2.4. Alta de información

Nombre	Alta de información RTIR
Actor	Analista
Descripción	Este caso de uso comienza cuando el analista desea registrar nueva información en el sistema. Es deseado que se pueda dar de alta información referente a cyber observables como por ejemplo IPs, hash de archivos, descripciones de amenazas, etc.

En la figura 3.11 se ve el caso de uso de alta de información en RTIR. En este un analista ingresa al sistema un cyber observable que será dado de alta en el sistema.En TAXII App se da de alta la información representando y almacenándola de forma estructurada por medio de STIX.

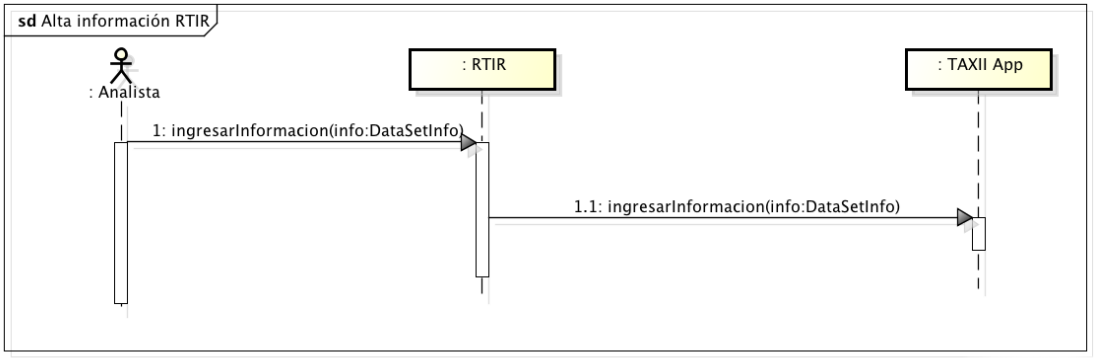

Figura 3.11: Caso de uso alta de información RTIR

3.3.2.5. Asociación de información

Nombre	Asociación de información
Actor	Analista
Descripción	Este caso de uso comienza cuando el analista desea asociar contenido existente en TAXII App a un Ticket de RTIR. La información asociada al caso de RTIR puede ser utilizada durante la resolución de un incidente.

En la figura 3.12 se ve el caso de uso de asociación de información. En este un analista asocia un ticket existente en RTIR con información estructurada almacenada en TAXII App. Como se dijo anteriormente dicha información es almacenada utilizando STIX.

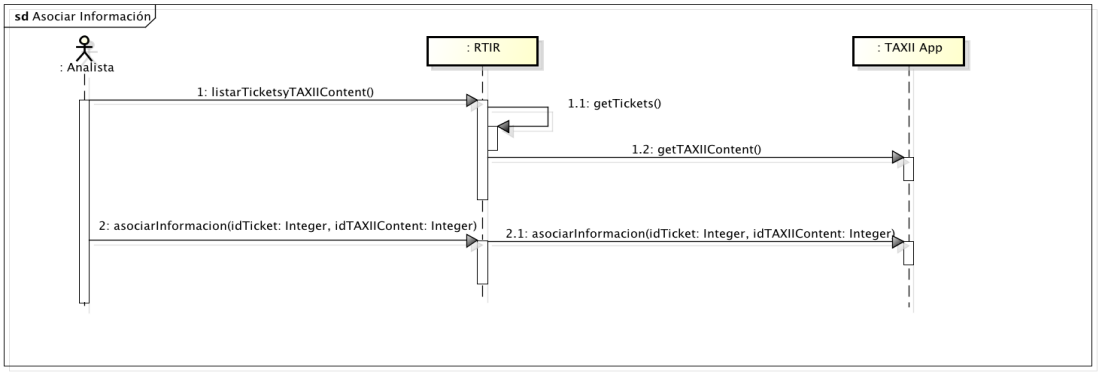

Figura 3.12: Caso de uso asociación de información

3.3.2.6. Subscripción a TAXII Data Feed

Nombre	Subscripción a Taxii Data Feed
Actor	Analista
Descripción	Con este caso de uso un analista selecciona un data feed en otro sistema al que quiere subscribirse. Esto se realiza por medio del Feed Managment Service de los sistemas.

En este caso de uso el analista desea subscribirse a uno de los TAXII data feed provistos por otra organización. Para ello obtiene un listado de los clientes TAXII con los que se relaciona para luego obtener un listado de los Data Feeds disponibles en ese cliente. Finalmente selecciona el TAXII data feed deseado.

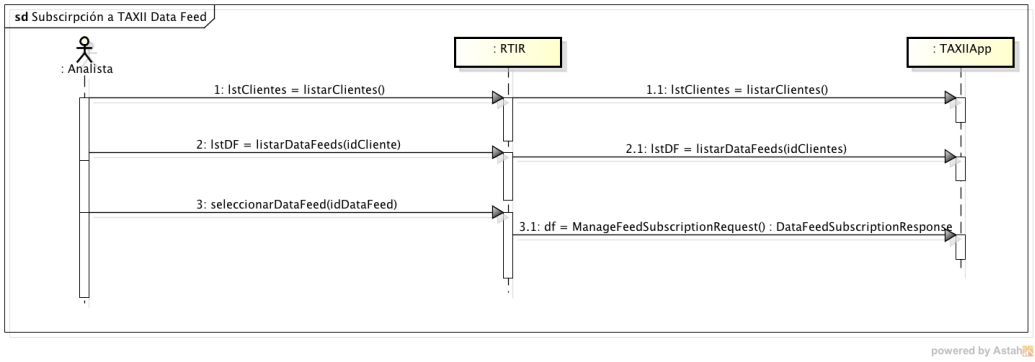

Figura 3.13: Caso de uso subscripción a TAXII Data Feed

3.3.2.7. Recepción de información

Nombre	Recepción de información
Actor	Cliente TAXII
Descripción	Este caso de uso se da cuando un cliente TAXII desea enviarle información a nuestro sistema. El envío de información se realiza porque un analista se subscribió a un data feed en el cliente. La recepción de información se realiza por medio del Inbox Service de nuestro sistema.

En este caso de uso un cliente TAXII en otra organización envía información a TAXII App. Luego la aplicación correlaciona la nueva información con la ya existente en el sistema.

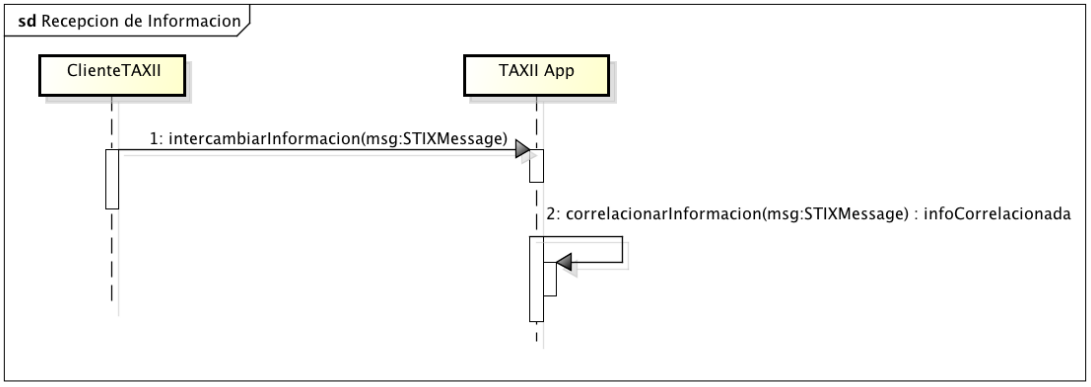

Figura 3.14: Caso de uso de recepción de información

3.3.2.8. Envío de información

Nombre	Envío de información
Actor	TAXII App
Descripción	Este caso de uso se da cuando el sistema desea enviar información a otro cliente TAXII. El envío de información se realiza porque el cliente se subscribió al TAXII Data Feed del sistema. Esto se realiza por medio del Inbox Service del cliente. El intercambio es iniciado por el sistema.

En este caso de uso TAXII App se comporta como el Actor y desea enviar nueva información a un cliente TAXII. Para ello lo primero que debe hacer es sanitizar la información. Luego envía la información representada en la forma de un mensaje STIX. Esto se ve en la figura 3.15.

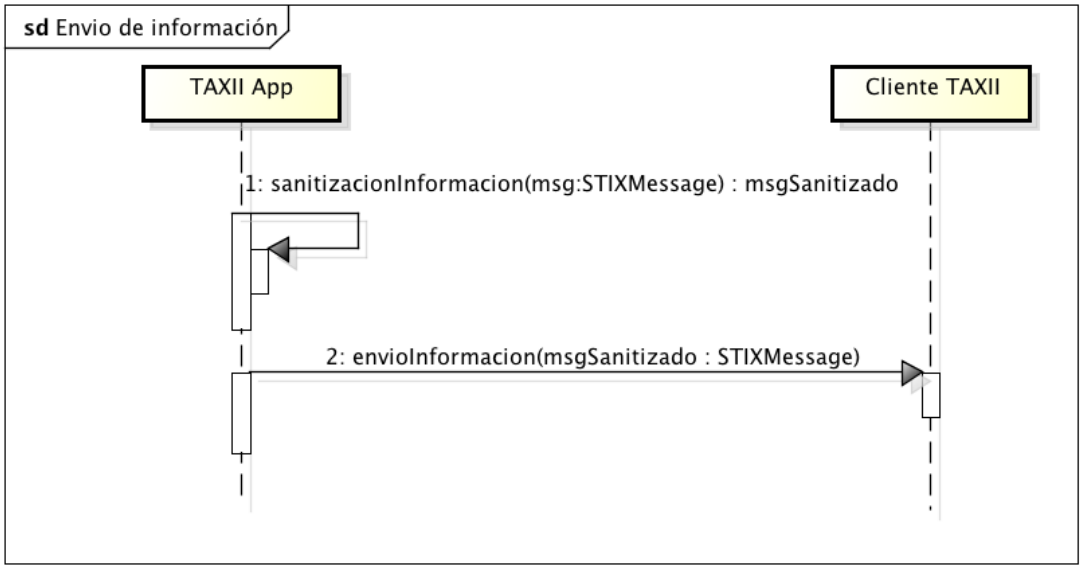

Figura 3.15: Caso de uso de envío de información

3.3.2.9. Poll de información

Nombre	Poll de información
Actor	TAXII App
Descripción	Este caso de uso se da cuando un cliente desea recibir información de un productor TAXII, en este los intercambios son iniciados por el cliente que contacta al Poll Service del productor.

En la figura 6.9 se ve el caso de uso de poll de información. En este TAXII App realiza el pedido de información a otro cliente TAXII por medio de mensajes Poll Request. Cuando obtiene la información esta se correlaciona con los datos existentes en el sistema.

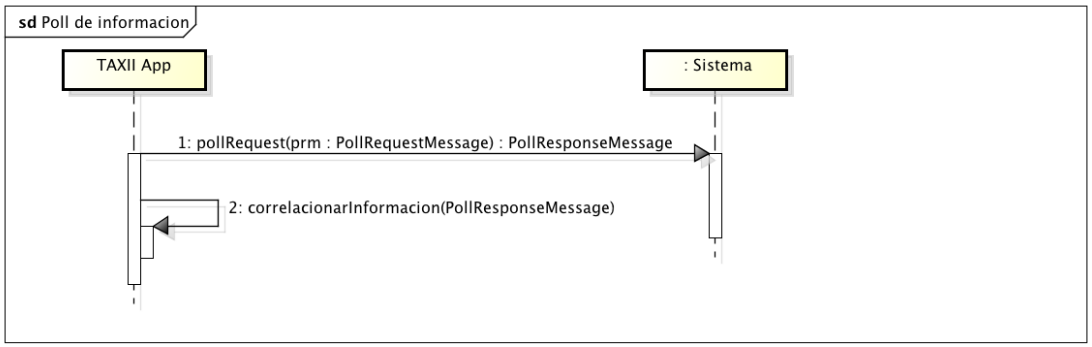

Figura 3.16: Caso de uso de poll de información

También se deben considerar los casos de uso provistos por RTIR para el seguimiento y manejo de los incidentes los cuales no serán especificados en este documento ya que se pueden encontrar en [RTIR]. Dichos casos de uso permiten el manejo de *tickets*, *queues* y gestión de usuarios.

Con RTIR se especifica un *workflow* para el trabajo con los *tickets* en organizaciones de seguridad. Dicho *workflow* comienza cuando se reporta un incidente, dicho reporte de incidente se asocia a un incidente o se crea uno nuevo. Los incidentes tratan de registrar toda la información necesaria para resolver el problema. De los incidentes se pueden iniciar investigaciones para trabajar con otras organizaciones. También se pueden crear *blocks* para mantener un registro de las acciones realizadas para mitigar el incidente.

Capítulo 4

Diseño

En este capítulo se propone una solución que respeta los requerimientos que se especificaron durante el capítulo anterior. Además se describe el funcionamiento de la herramienta propuesta identificando los componentes que participan en dicho proceso. También se explica la arquitectura de la aplicación con la función de cada una de sus componentes. Finalmente se detalla el modelo de datos creado para la base de datos.

4.1. Contexto para el uso del sistema

Por medio del sistema propuesto es posible ingresar información de seguridad, correlacionarla con información existente en el sistema, y, luego de ser sanitizada, compartirla con organizaciones socias.

El sistema ha sido diseñado para que sea utilizado por grupos de seguridad. Se busca que dichos grupos puedan compartir información de incidentes de seguridad representada con el lenguaje STIX y que sea distribuida utilizando TAXII. El diseño y desarrollo de STIX y TAXII ha sido realizado por MITRE con el apoyo de la comunidad. El intercambio de información da a las organizaciones la posibilidad de identificar adecuadamente las amenazas dando evidencias específicas de su existencia.

Se busca que la herramienta ayude a solucionar algunas de las problemáticas a las que se enfrentan las organizaciones durante el intercambio de información de seguridad. Dichas problemáticas se abordan durante el capítulo 3.

4.2. Componentes del sistema

En esta sección se presenta el sistema diseñado, mostrando en la figura 4.1 su arquitectura. El sistema busca facilitar a las organizaciones el intercambio de información de seguridad. Se busca integrar a RTIR dicha capacidad utilizando las herramientas mencionadas anteriormente.

Para ello se busca extender RTIR para que el usuario ingrese la información al sistema, con dicha información se crea un ticket en RTIR y a su vez se envía a TAXII App para crear una representación utilizando STIX. La aplicación TAXII es un desarrollo nuevo que utiliza librerías provistas por MITRE, estas son utilizadas para representar e intercambiar la información.

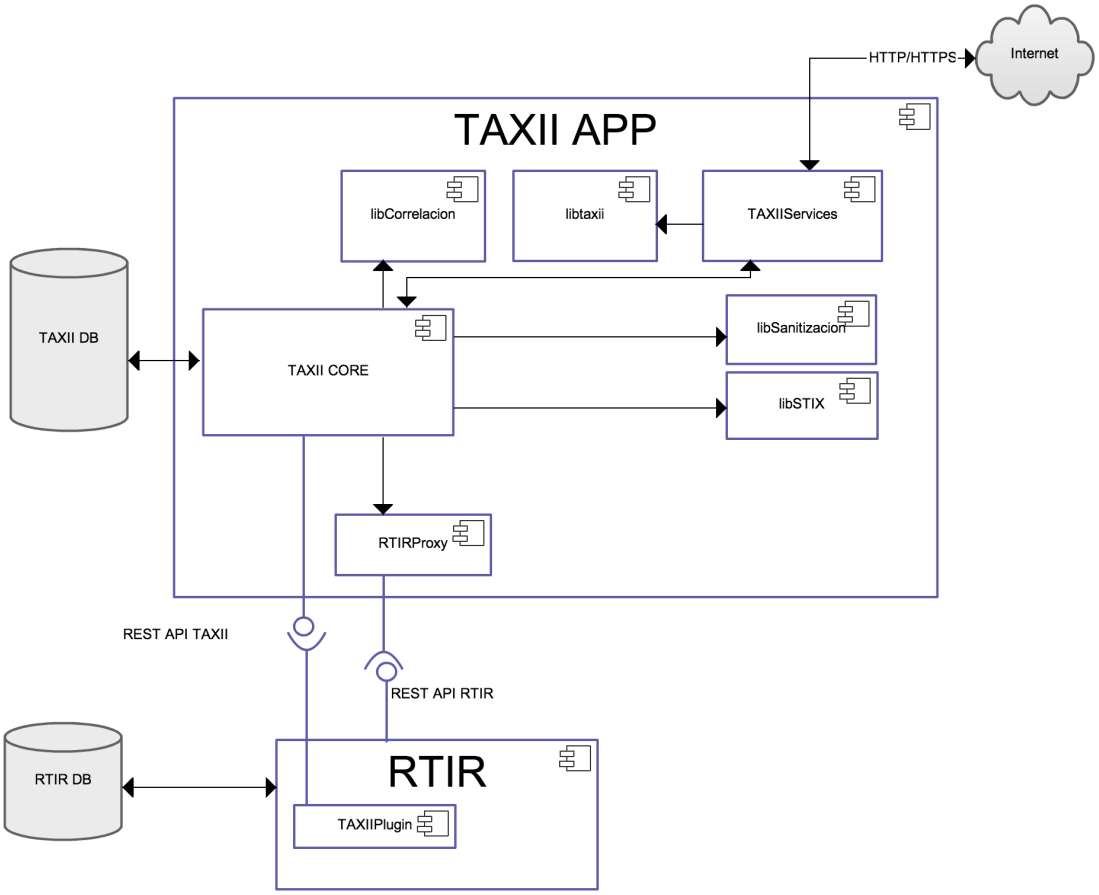

Figura 4.1: Arquitecture del sistema

4.3. Aspectos generales de las componentes del sistema

En la figura 4.1 se pueden ver dos bases de datos: TAXII DB y RTIR DB. A su vez se cuenta con dos aplicaciones: RTIR y TAXII App. RTIR es la aplicación para manejo de incidentes desarrollada por la empresa Bestpractial [2] y elegida para el desarrollo de este proyecto. TAXII App es una aplicación nueva que será desarrollada durante el proyecto y se encarga de representar la información de forma estructurada e intercambiar dicha información de seguridad.

La aplicación RTIR recibe datos ingresados por un analista y los transmite a la aplicación TAXII App. Para realizar dicha transmisión es necesario extender RTIR para consumir una API REST publicada por el cliente TAXII. La aplicación RTIR también puede recibir de la aplicación TAXII App información, la cual puede estar relacionada a tickets existentes y contener soluciones a problemas, identificación de atacantes, etc.

La aplicación TAXII App recibe datos de RTIR así como también de organizaciones socias. Luego de que los datos son recibidos y almacenados se correlacionan con los ya existentes en por TAXII App. De la correlación se pueden obtener soluciones a problemas encontrados, información sobre ataques realizados, identificación de atacantes, etc. En caso de que exista información de utilidad se pueden ingresar notas a tickets o relacionar tickets existentes en el RTIR.

Con el sistema diseñado se permite el intercambio de información de seguridad con otras organizaciones. Para realizar dicho intercambio se representa la información utilizando el lenguaje STIX. Los intercambios son realizados por medio del protocolo TAXII. Para realizar la representación de la información así como el intercambio se utilizaron librerías provistas por MITRE.

La aplicación TAXII App es independiente de la aplicación RTIR, siendo posible reemplazar esta última por otra herramienta con funcionalidades similares. Es deseado que la herramienta que pueda reemplazar a RTIR pueda ser extendida de forma de invocar a las funcionalidades desarrolladas en TAXII App. Además deberia contar con un método por el cual la TAXII App pueda utilizar la funcionalidades de la herramienta que sustituya a RTIR.

4.3.1. RTIR

Este componente presenta las funcionalidades originales de la herramienta RTIR con un agregado específico implementado para este proyecto. RTIR es el componente de la aplicación encargado de realizar el manejo de los incidentes. Se agrega una extensión que permite invocar una API REST en la aplicación TAXII.

RTIR cuenta con su propia base de datos, RTIR DB, a la cual solo accede la componente RTIR, que no será modificada para la realización de este proyecto.

4.3.2. TAXII App

La segunda componente presente es TAXII App, está compuesta por otros componentes implementados durante el proyecto así como por librerías provistas por MITRE. Los componentes provistos por MITRE son libTAXII y libSTIX, a estos no se le realizarán cambios debido a que cuentan con las funcionalidades necesarias para llevar a cabo el proyecto. En el caso de libSTIX permite representar la información recibida por medio del lenguaje STIX, por otro lado libTAXII permite la creación de mensajes TAXII para que estos sean intercambiados. El componente TAXII App utiliza los componentes TAXIICore, RTIRProxy y TAXIIServices, estos son implementados durante el transcurso del proyecto.

4.3.2.1. libTAXII

libTAXII [15] es una librería que provee una representación de objetos de los mensajes TAXII, cuenta además con una serie de métodos para el manejo de dichos mensajes. Provee clientes para http y https. La librería se utiliza para generar los mensajes y dispone métodos para transformarlos a xml, estos xml son utilizados en los intercambios entre cliente y servidor.

4.3.2.2. libSTIX

libSTIX es una librería provista por MITRE para parsear, manipular y generar contenido STIX.

4.3.2.3. RTIRProxy

RTIRProxy es la componente de TAXII App encargada de integrarse con RTIR, para ello consume la API REST provista por RTIR para el manejo de tickets. Por ser una API REST la comunicación está encapsulada en el protocolo http. Esta componente busca que se tenga independencia de RTIR permitiendo el uso de otro sistema en su lugar. Lo que se logra es que si otro sistema tiene una API o un método de integración distinto al de RTIR se re implemente esta componente. La única premisa que se debe cumplir es que la interfaz con la cual se comunique TAXIICore se mantenga. Dentro de dichos tipos de integración podría encontrarse otra API REST o web services con SOAP.

4.3.2.4. TAXIICore

Es necesario que este componente tenga una interfaz de web services REST para realizar la comunicación con el RTIR. En dicha interfaz se deben dar operaciones para realizar el alta de información, obtener información de servicios publicados por contrapartes, enviar y recibir información de seguridad entre contrapartes.

Además debe proveer una interfaz para que la componente TAXIIServices envíe la información recibida en la aplicación TAXII, en dicha interfaz se deben recibir tanto paquetes STIX así como información adicional utilizada por el protocolo TAXII.

La información tomada de la base de datos es enviada en forma de objetos STIX junto a las políticas definidas por los administradores para que sea sanitizada. Luego de que la información es sanitizada es enviada a TAXIIServices para que sea compartida con las organizaciones socias.

4.3.2.5. libSanitizacion

Esta es la componente encargada de realizar la sanitización de la información que se desea intercambiar entre los sistemas. Para realizar dicha sanitización se podrían utilizar librerías ya existentes o implementar una propia. La componente se alimenta de políticas definidas por los administradores para realizar la sanitización.

4.3.2.6. libCorrelacion

Es la componente encargada de la correlación de los datos del sistema, dicha correlación se realiza por medio de políticas ingresadas por los admi-

nistradores del sistema. La componente ayuda a agrupar datos que tengan características similares lo que facilita el trabajo de los analistas y da más valor a la información recibida. Se pueden utilizar distintos métodos para realizar las correlaciones y de ahí se desprende la necesidad de que sea un componente separado del Core del sistema. Los datos correlacionados son los almacenados en TAXII DB.

4.3.2.7. TAXIIServices

Esta componente es una implementación de los servicios TAXII como se especifica en [35] y [37]

TAXIIServices provee los siguientes servicios:

- Inbox: Por medio de este servicio el sistema acepta mensajes enviados por otro cliente TAXII en intercambios iniciados por este.
- Poll: Este servicio es el que utiliza el sistema para consumir datos provenientes de un TAXII Data Feed en un cliente TAXII.
- Discovery: Es el servicio encargado de la comunicación de información referente a la disponibilidad y uso de los servicios TAXII en el sistema.

Utiliza la componente libTAXII para crear mensajes TAXII y de esa forma realizar el intercambio de documentos STIX recibidos de libSanitizacion. Los mensajes que son recibidos de otros socios son pasados a la componente TAXIICore para que sea almacenada y luego correlacionada con datos existentes.

4.4. Comunicación entre los componentes

En la comunicación entre RTIR y la aplicación TAXII se utiliza REST (*Representational State Transfer*), en este tipo de servicio web se trata de emular el protocolo http o algún protocolo similar usando la restricción de proveer una interfaz a un conjunto conocido de operaciones estándar (GET, PUT, etc). Se utiliza este tipo de servicio web debido a que RTIR provee una API de este tipo.

El intercambio entre las distintas organizaciones se realiza por medio de HTTP/HTTPS. Esto se debe a que hasta el momento MITRE solo ha especificado el intercambio por medio de dicho protocolo. Se podrían utilizar

otros protocolos para realizar el intercambio pero sería necesario que MITRE definiera las especificaciones para su uso, un ejemplo de ellos podría ser SOAP.

4.5. Modelo de datos

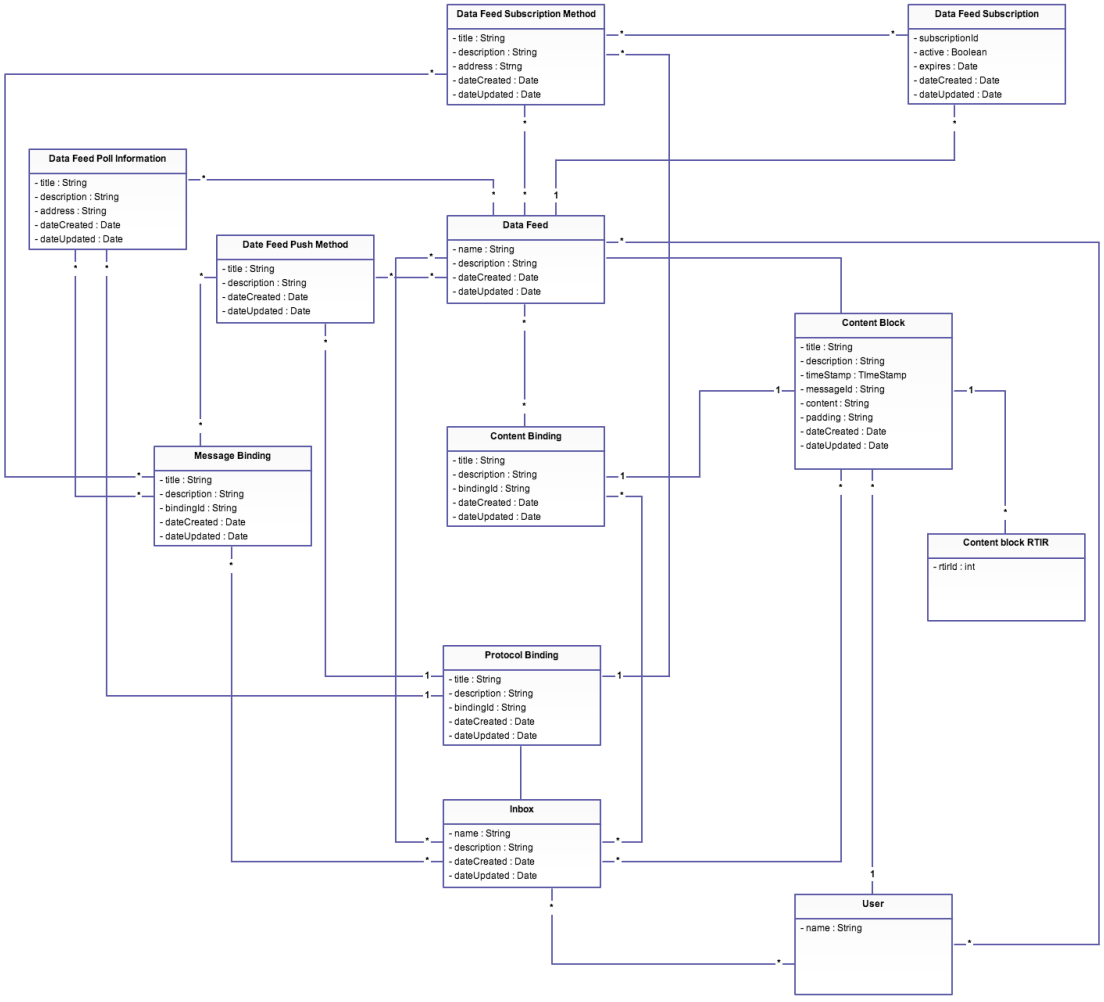

Figura 4.2: Diagrama de datos

El modelo de datos diseñado se adapta a las necesidades de TAXII, dichas necesidades están definidas en [38], además se agrega una tabla para realizar

el mapeo entre los elementos de RTIR y los elementos TAXII intercambiados. En la figura 4.2 se ve el modelo de datos y a continuación se explica cada una de sus componentes.

- Protocol Binding: Es un elemento utilizado para establecer el protocolo de intercambio soportado por una implementación de TAXII.
- Content Binding: Es utilizado para establecer el tipo de contenido soportado para un cierto intercambio realizado con TAXII, por ejemplo: *Poll*, *Inbox*, etc.
- Message Binding: Es utilizado para establecer el tipo de sintaxis para un intercambio realizado con TAXII.
- Data Feed Push Method: Es utilizado para establecer los protocolos que pueden ser utilizados para enviar contenido por medio de una subscripción. Esto se utiliza en un mensaje de tipo *Feed Information Response*. Es definido en [37].
- Data Feed Poll Information: Tiene la finalidad de establecer los protocolos soportados y las direcciones de una fuente de información (*Data Feeds*). Es utilizado en los mensajes de *Feed Information Response* como se define en [37].
- Data Feed Subscription Method: Es utilizado para identificar el protocolo y la dirección de un servicio de *Feed Managment* que pueda procesar las subscripciones a fuentes de información TAXII.
- Content Block: Representa el contenido de los mensajes *Poll Response* o *Inbox Messages*. Es necesario señalar que el campo *content* representa un mensaje STIX. Dicho mensaje representa la información de seguridad estructurada, la cual fue representada utilizando el lenguaje STIX.
- Data Feed: Representa una fuente de información TAXII, los usuarios se suscriben a dichas fuentes de información para recibir datos que sean de su interés.
- Data Feed Subscription: Representa una subscripción a fuentes de datos TAXII.

- Inbox: Representa un *inbox* TAXII, estos son el mecanismo por el cual los consumidores reciben información de otros clientes TAXII que nos envían información. La representación permite que un *inbox* este asociado a ningún o muchas fuentes de datos.
- Content Block RTIR: Representa el nexo entre los contenidos TAXII y los tickets existentes en RTIR.

Capítulo 5

Implementación

5.1. Aspectos Generales

En esta sección se presentan las herramientas y metodologías elegidas para realizar la implementación de la herramienta.

5.1.1. Entorno de desarrollo

El entorno desarrollado durante la implementación del sistema y desarrollo del caso de estudio consta de:

- Sistemas operativos: Mac OS 10.9.3 (Maverick) y Linux Mint 15
- Sistemas operativos utilizados en el testeo y caso de estudio: Linux Mint 15
- Lenguajes de desarrollo: Perl y Python (web framework Django)
- Entorno de desarrollo: vim[1] con plugins provistos por spf13[2]
- Base de datos: Mysql versión 14.14 distribución 5.5.34
- Sistema de versionado: git
- Servidor web: apache2
- Versión de RT: RT4
- Testeo de APIs Rest: Postman[3]

[1]Vim es un editor de texto que está diseñado para usarse tanto desde una interfaz de línea de comandos o como una aplicación independiente en una interfaz gráfica de usuario.

[2]http://vim.spf13.com/ centraliza un conjunto de plugins para vim.

[3]Es una herramienta de chrome utilizada para probar, desarrollar y documentar APIs.

5.1.2. Metodología

Se utilizó el framework web Django por permitir desarrollar aplicaciones web de forma rápida y con la escritura de poco código. Además busca automatizar parte del desarrollo al basarse en el principio DRY (*Don't Repeat Yourself*). Presenta la ventaja de utilizar el patrón de arquitectura MVC (*Model View Controller*). La ventaja de utilizar este patrón es que divide la aplicación en tres partes interconectadas y da una base inicial con la cual desarrollar el trabajo. A su vez, utilizar un patrón MVC presenta la ventaja de que se tiene experiencia desarrollando aplicaciones utilizando dicho patrón lo cual facilita el trabajo. La existencia de librerías implementadas en python para trabajar con STIX y TAXII influyo fuertemente en la decisión de utilizar Django. Otra de las razones fue la existencia de una implementación de referencia con la cual se puede testear el desarrollo realizado.

Además de utilizar Django se utilizó el lenguaje de programación PERL para el desarrollo de un plugin para la herramienta RT.

La metodología utilizada fue la de un desarrollo incremental, esto permitió realizar una validación de la arquitectura propuesta. Luego de realizar dicha validación se siguió implementando la herramienta propuesta.

5.1.3. Librerias utilizadas

Durante el transcurso de la implementación se utilizaron distintas librerías Perl y Python para el desarrollo de la herramienta.

5.1.3.1. Django Rest Framework

Django Rest Framework [24] es una librería para Django que hace simple el construir APIs REST de forma sencilla y rápida. La librería se utilizó para implementar la API REST que se consume desde el plugin RT-TAXII, dicho plugin se explicará mas adelante.

5.1.3.2. LWP

LWP se refiere a LibWww-Perl [16] y es un conjunto de módulos Perl que proveen una API simple para desarrollar clientes web y desarrollar servidores http. En particular se utilizó el módulo LWP::UserAgent que permite realizar con facilidad pedidos web. Con esta librería se realizan los pedidos que se hacen a la componente libTAXII.

5.1.3.3. libTAXII

libTAXII [15] es una librería Python para manejar mensajes TAXII e invocar servicios TAXII. Uno de sus objetivos es ser fiel a las especificaciones de TAXII y a las buenas prácticas de Python. La librería provee:

1. Una representación de objetos de los mensajes TAXII.
2. Permite transformar XML en objetos Python y viceversa.
3. Un cliente HTTP/HTTPS TAXII.

Esas tres características se utilizaron durante el desarrollo de la herramienta. Debido a que primero se crea un objeto Python que representa el mensaje TAXII, dicho objeto es transformado a XML para ser enviado por el cliente TAXII, la respuesta al mensaje enviado también es un objeto XML, dicho XML es transformado a un objeto Python para ser procesado. A su vez cuando se recibe un XML, este se transforma a un objeto Python para ser procesado, luego del procesamiento se devuelve una respuesta XML por medio del cliente HTTP/HTTPS.

5.1.3.4. pythonStix

PythonStix [12] es una librería Python que provee una API para parsear, manipular y consumir contenido STIX. Los desarrolladores pueden utilizar la API para desarrollar aplicaciones que crean, consumen, traducen o procesen contenido STIX. Se utilizo en taxiiApp para crear paquetes STIX a partir de un cyber observable que ingrese el usuario en RTIR. Además durante los intercambios se obtiene información de los paquetes STIX para lo cual es necesario parsear la información del paquete STIX.

5.1.3.5. YETI

YETI [29] es una prueba de concepto de TAXII creado para ayudar a los desarrolladores a implementar y testear sus propias aplicaciones TAXII. A su vez ayuda a aprender más sobre el funcionamiento de TAXII. Fue utilizada en etapas tempranas del proyecto como referencia y para testear el desarrollo realizado. De esta forma mientras se avanzó en el desarrollo se pudo validar de forma adecuada que se respetara adecuadamente la especificación de TAXII.

5.1.4. Módulos desarrollados

A continuación se describen lo módulos implementados para el funcionamiento de la herramienta, así como las consideraciones necesarias para extender la herramienta con nuevas funcionalidades.

5.1.4.1. RT-TAXII

Se desarrolló un plugin para RTIR que es el encargado de consumir la API rest provista por TAXIIApp y que extiende la interfaz web de RTIR para presentarle al usuario los formularios necesarios para que interactúe con el sistema. Se incluyó en el menú de RTIR una nueva sección en la que se especifican las distintas acciones que se pueden realizar referentes a TAXII. Cada una de estas acciones invoca a un script desarrollado en PERL que presenta el formulario al usuario. Luego de que el usuario ingresa la información deseada y realiza el envío de la información. El sistema invoca a la API Rest provista por TAXII App. Es necesario que se configure la dirección del host en la que se encuentra ejecutando TAXIIApp. La estructura de directorios de RT-TAXII es la que se muestra en la figura 5.1.

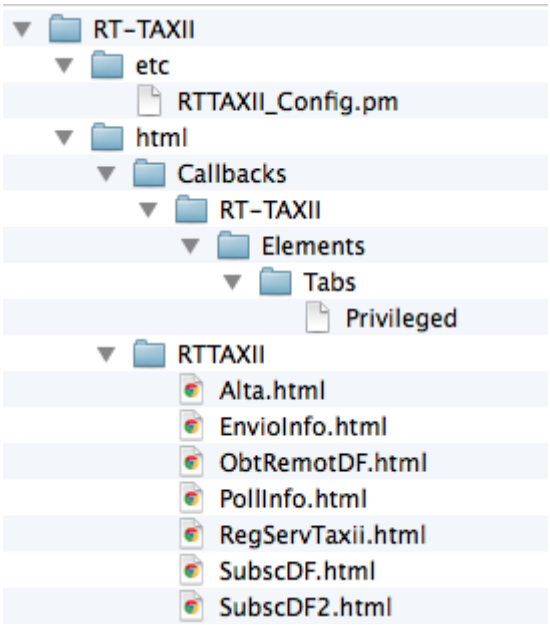

Figura 5.1: Estructura de directorios de plugin RT-TAXII

El archivo RTTAXII_Config.pm es utilizado para guardar las configuraciones del plugin. Dentro de dichas configuraciones se encuentran parámetros

necesarios para el funcionamiento del sistema. En el archivo Privileged se realiza la configuración del menú que contiene las funcionalidades del sistema. Los archivos incluidos en el directorio RTTAXII son los encargados de presentar los formularios y realizar la lógica correspondiente a las llamadas a TAXIIApp. Dichos archivos están implementados en PERL y contienen HTML para realizar la presentación al usuario.

Para extender la herramienta con nuevas funcionalidades es necesario crear un nuevo script PERL en el directorio RTTAXII. Dicho script debe implementar las funcionalidades que se deseen. Como RT está desarrollado en PERL se cuenta con todos los módulos disponibles de PERL.

5.1.4.2. TAXIIApp

A continuación se describirán los componentes implementados y como estos interactúan con las librerías utilizadas. También se mencionarán las consideraciones que se deben tener para expandir la herramienta. En la figura 5.2 se ven los distintos módulos implementados durante el proyecto.

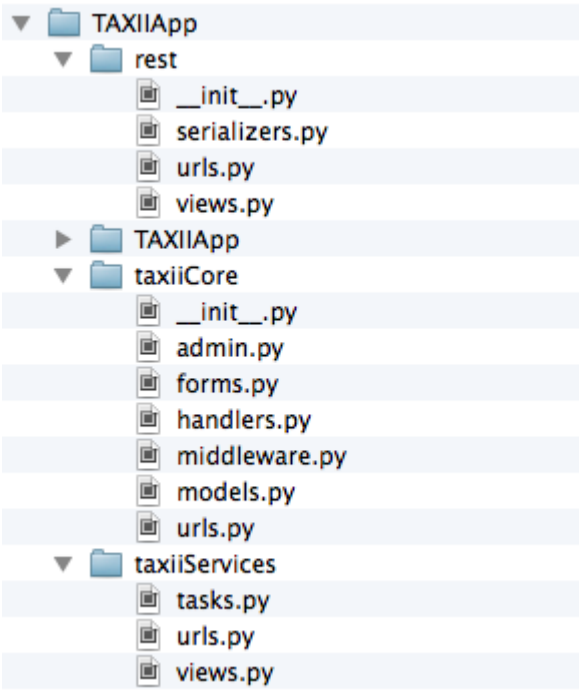

Figura 5.2: Aplicación TAXII implementada junto con sus módulos

5.1.4.2.1. TAXIICore

TAXII Core es la componente encargada del procesamiento de la información, es invocada por la API Rest y por TAXII Services y actúa dependiendo de la información recibida. Además es la encargada de la comunicación con la base de datos por medio del modelo de datos desarrollado. A su vez se tuvo consideración durante el desarrollo que este componente sería el encargado de dialogar con los módulos de sanitización y correlación de información. En el anexo 8 se pueden ver las entidades creadas para modelar la base de datos así como los métodos implementados para enviar y recibir información del resto de las componentes.

5.1.4.2.2. TAXIIServices

Este módulo es el encargado de la comunicación con otra organización por medio del protocolo TAXII. Se desarrollaron los distintos métodos necesarios para implementar los servicios TAXII exceptuando el discovery service. Se utiliza libTAXII para crear los mensajes a intercambiar y para realizar la invocación a los servicios de otras organizaciones. Los mensajes luego de recibidos son pasados a TAXIICore para su procesamiento. En caso de que se crearan nuevos servicios o que se quisiera implementar por ejemplo el discovery service esto se realizaría en este módulo. En el anexo 8 se pueden ver las firmas de los servicios implementados.

5.1.4.2.3. REST API

La API Rest desarrollada provee las funcionalidades necesarias para que RT-TAXII interactúe con TAXII App. Dentro de dichas interacciones se permite el manejo de cualquiera de los elementos del modelo en TAXIICore. Además se crearon métodos específicos para el uso de TAXII. Se permite el envío y poll de información, obtención de data feeds así como la subscripción a estos.

Para la implementación se utilizo la librería auxiliar *Django Rest framework* que simplifica el desarrollo de la API. Si bien se implementaron todos los elementos necesarios para realizar la interacción con la base de datos, si existiera un cambio en esta (i.e: se crea una nueva entidad) sería necesario extender esta API para poder realizar la interacción entre los sistemas. A su vez si se crea un nuevo servicio TAXII y se desea proveer una interfaz en RTIR para que los usuarios interactúen con esta sería necesario agregar los métodos necesarios en esta API. En el anexo 8 se puede ver la firma de la API desarrollada.

Capítulo 6

Caso de estudio

Para mostrar el uso de la herramienta se simulo en un ambiente virtualizado la interacción y colaboración entre varios CSIRTs. Mediante esta prueba se realizó el intercambio de cyber observables entre las distintas organizaciones. Dicho intercambio se realizo utilizando los distintos servicios provistos por TAXII.

6.1. Objetivos

El caso de estudio fue ideado con la finalidad de validar la implementación del protocolo TAXII realizada así como las distintas interacciones entre RTIR y la aplicación TAXII.

Para realizar la validación de las interacciones entre RTIR y la aplicación TAXII se tuvó como objetivo recorrer cada una de las interfaces implementadas.

La validación del protocolo TAXII se realizó mediante la invocación de cada una de las funcionalidades que implementan el intercambio de información. Dicha invocación se realizó desde la interfaz de RTIR mediante la API REST implementada durante el proyecto. El protocolo implementado se validó de dos maneras, primero mediante dos instancias de la aplicación TAXII implementada y luego por medio de YETI. Como se dijó anteriormente YETI es una implementación de referencia del protocolo TAXII provista por MITRE.

6.2. Escenario del caso de estudio

Se utilizaron dos máquinas vistuales para realizar la prueba de concepto, cada una de ellas tiene instalada Linux Mint 17. En dichos equipos se

Equipo	Hostname	Ip	Puerto	Aplicación
CSIRT 1	csirt1.com	172.16.59.219	8080	RTIR
CSIRT 1	csirt1.com	172.16.59.219	8001	TAXIIApp
CSIRT 2	csirt2.com	172.16.59.218	8080	RTIR
CSIRT 2	csirt2.com	172.16.59.218	8001	TAXIIApp
CSIRT 2	csirt2.com	172.16.59.218	9080	YETI

Tabla 6.1: Información de cada equipo

instaló RTIR junto con TAXIIApp. Además en la máquina virtual correspondiente al CSIRT2 se instalo YETI. Cada una de estas máquinas virtuales tiene incluida una base de datos con esquemas correspondientes a cada una de las aplicaciones. En la figura 6.2 se ve un esquema representando el despliegue realizado.

Figura 6.1: Despliegue realizado para el caso de estudio

La tabla que se presenta a continuación indica la información respecto al laboratorio implementado.

En la tabla 6.1 se muestran las direcciones ip para cada uno de los equipos utilizados en el caso de estudio. En el CSIRT 1 se tiene instalado RTIR con el *plugin* implementado durante el proyecto para comunicarse con TAXIIApp. También se tiene instalado TAXIIApp para realizar los intercambios de información con el protocolo TAXII. RTIR escucha en el puerto 8080 mientras que TAXIIApp en el 8001. Dichas aplicaciones se encuentras instaladas también en el CSIRT 2 escuchando en los mismos puertos. A su vez en el CSIRT 2 se encuentra una instalación de YETI. Dicha instalación se encuentra escuchando en el puerto 8080. YETI y TAXIIApp se encuentran corriendo ambas en el mismo equipo para no tener una maquina virtual extra. En cada uno

de los equipos se encuentra instalada una base de datos Mysql en la que se encuentran los esquemas de base de datos utilizados por cada una de las aplicaciones.

6.3. Ejecución del caso de estudio

La figura 6.2 siguiente muestra cuales fueron los pasos para realizar el intercambio entre las distintas organizaciones.

Figura 6.2: Flujos realizados por el caso de estudio

En los pasos 1 a 4 de la figura 6.2 agregamos en el CSIRT 1 los servicios TAXII correspondientes a TAXII App y a YETI, ambos ejecutándose en el CSIRT 2. En el caso de TAXII App se agregaron todos los servicios necesarios para realizar el intercambio. Sin embargo, la implementación de YETI solo cuenta con el *inbox service*, *poll service* y *discovery service*, por lo tanto se agregaron *inbox service* y *poll service* como los servicios en YETI. La implementación de TAXII realizada durante el proyecto no implementó el *discovery service* y por ello fue necesario implementar un caso de uso para ingresar los servicios.

En la figura 6.3 observamos el ingreso de los servicios TAXII y en la figura 6.4 el listado de los servicios ingresados anteriormente.

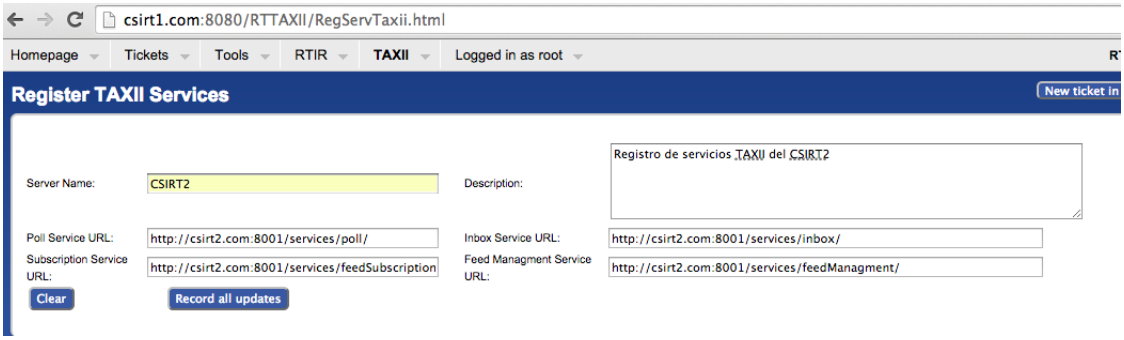

Figura 6.3: Registro de Servicios TAXII

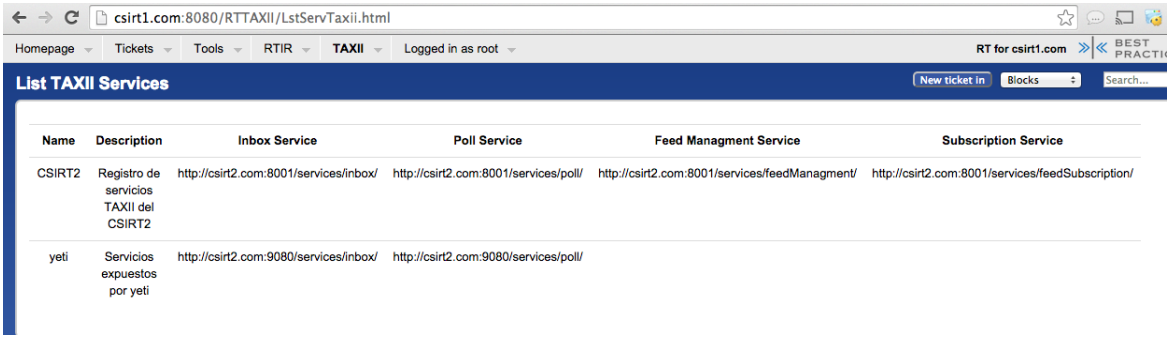

Figura 6.4: Listado de Servicios TAXII

En los pasos 5 a 7 de la figura 6.2 se ingresa nueva información en el CSIRT 1, a su vez se comprueba que la información se halla dado de alta

adecuadamente en el sistema. Lo mismo se realiza en los pasos 11 a 13, 16 y 17, la diferencia que esta vez se realiza desde el CSIRT 2, dicha información será luego enviada al CSIRT 1 por medio del poll service así como del inbox service. En la figura 6.5 se ve el ingreso de información, se da de alta un Cyber observable que representa un correo electrónico que podría provenir por ejemplo de un ataque de phishing. En el anexo se pueden ver los distintos cyber observables dados de alta, los cuales se obtuvieron de MITRE([19]).

En la figura 6.6 se observa información dada de alta en el sistema, también se ve información que fue obtenida por medio del poll service del data feed "default" del CSIRT 2.

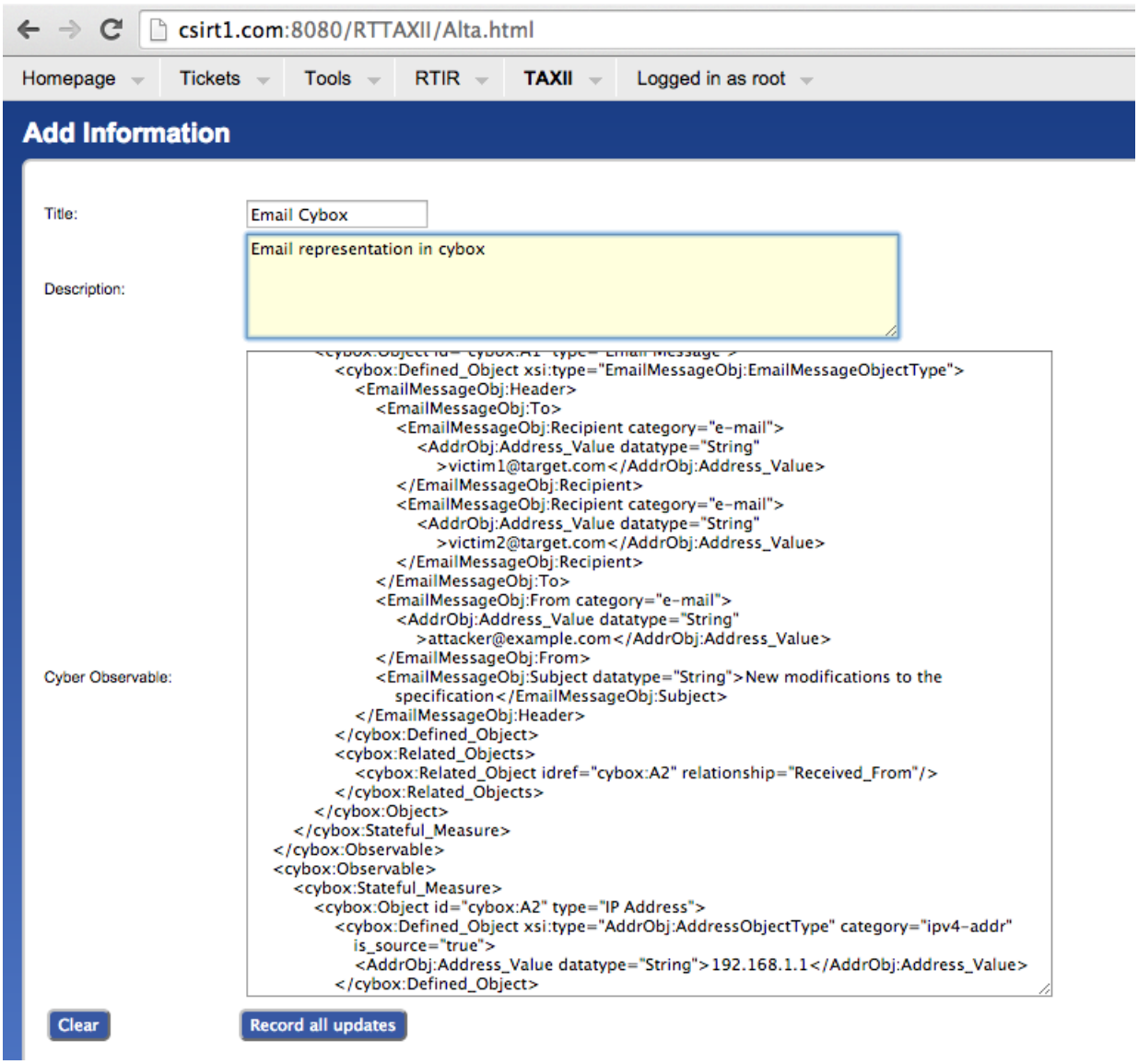

Figura 6.5: Alta de información de correo electrónico

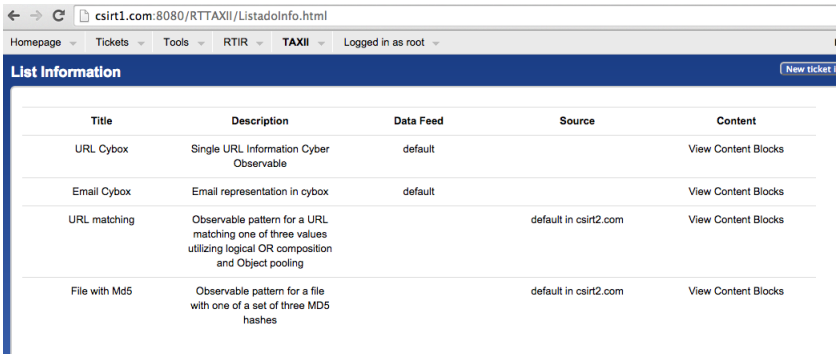

Figura 6.6: Listado de información en el sistema

Los pasos 8 a 10 se encargan de la obtención de los data feeds en el CSIRT 2 y a su vez se comprueba que los data feeds se hayan obtenido correctamente. Para obtener los data feeds se envía un mensaje desde TAXII App en el CSIRT 1 solicitándolos al CSIRT 2.

La figura 6.7 muestra la obtención de los data feeds remotos en el CSIRT 2, luego en la figura 6.8 se ve un listado del data feed obtenido. A su vez se ve el data feed de YETI.

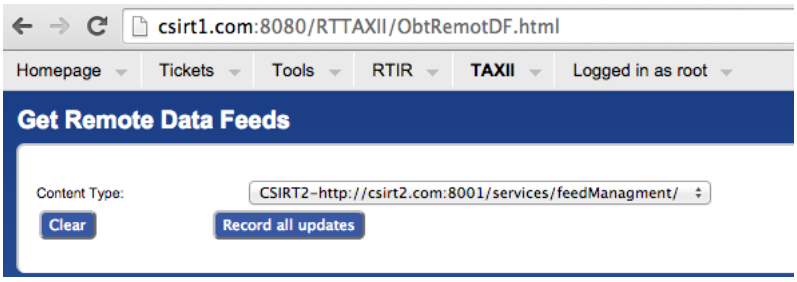

Figura 6.7: Obtención de data feeds remotos

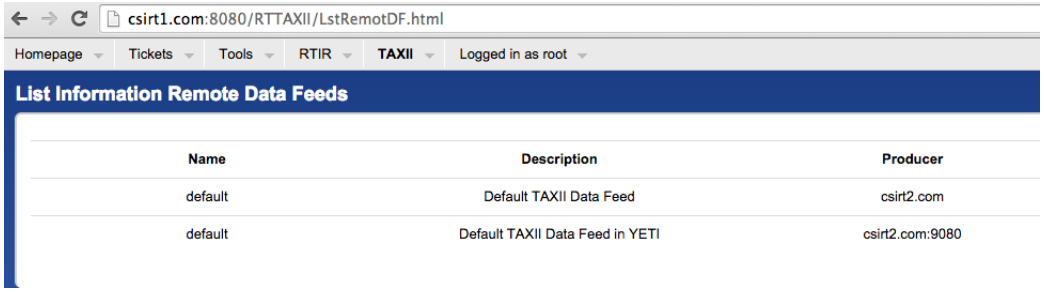

Figura 6.8: Data feeds en otras organizaciones

En el paso 14 se realiza el poll de información, para ello se invoca al poll service en el CSIRT 2, esto devuelve toda la información en el data feed especificado. Luego de realizado este paso, se lista nuevamente la información para ver cuales fueron los datos recibidos. En la figura 6.9 se ve el polling de información. Previamente se vio el listado de información luego de hacer el poll.

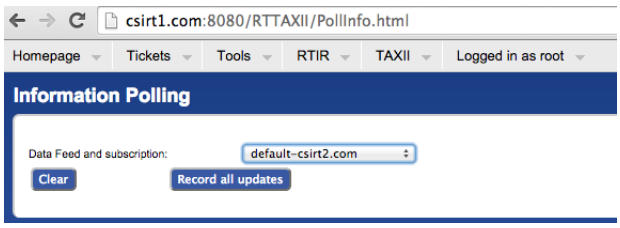

Figura 6.9: Poll de información

Como se dijo anteriormente, en los pasos 16 y 17 se ingresa nueva información al CSIRT 2, dicha información es agregada para luego ser enviada al inbox del CSIRT 1. Previo a ser enviada, el CSIRT 1 debe registrarse para indicar que desea recibir información al inbox service. Dicha subscripción se realiza en el paso 18 invocando al servicio de feed managment del CSIRT 2. Luego de esto, en el paso 19, se envía la información al inbox service del CSIRT 1 desde el CSIRT 2. En el paso 20 se observan los nuevos datos obtenidos luego de que se envíe la información al inbox service del CSIRT 1.

En las figuras 6.10 y 6.11 se ve la subscripción a un data feed en otra organización, mientras que en la 6.13 se ve el envío de información que se realizará por medio de inbox service como se dijo anteriormente.

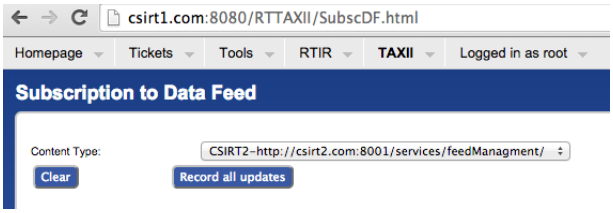

Figura 6.10: Primer paso de subscripción a data feeds

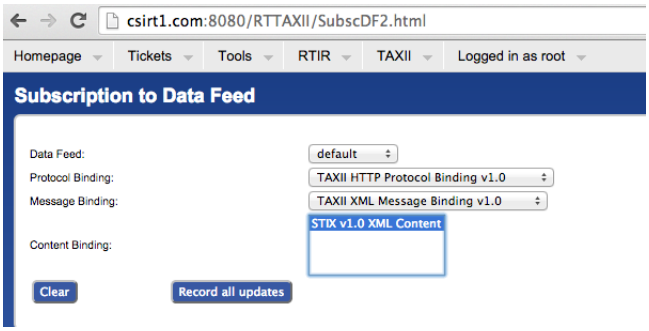

Figura 6.11: Segundo paso de subscripción a data feeds

Figura 6.12: Envio de información

En el paso 6.12 se crea un nuevo ticket en el RTIR, dicho ticket será asociado con información existente en TAXII App. Dicha información puede haber sido obtenida de uno de los otros CSIRTs o de ingresada por un analista localmente.

En la figura 6.14 se ve la asociación entre tickets del RTIR e información almacenada en TAXIIApp.

Figura 6.13: Asociación entre Tickets y Content blocks

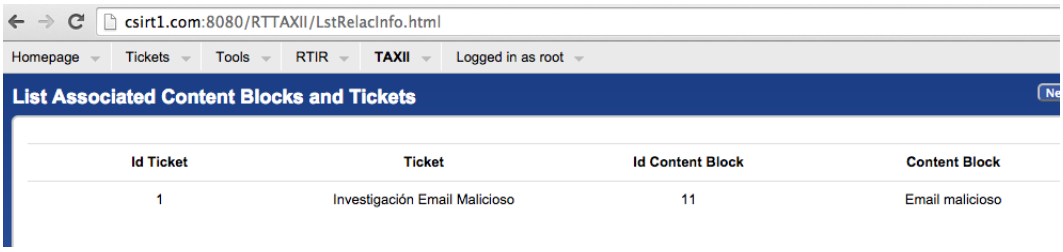

Figura 6.14: Listado de asociaciones existentes

Finalmente se invoca al poll service de YETI para obtener información y luego se envía información a YETI utilizando su inbox service. Como se dijo anteriormente, YETI no tiene todos los servicios implementados y además no puede iniciar el intercambio de información. Como YETI no implementa el manejo de subscripciones u obtenciones de data feeds, se ingreso previamente dicha información de YETI en el CSIRT 1.

Capítulo 7

Conclusiones y trabajo futuro

En este capítulo se presentan las principales conclusiones y dificultades más relevantes del proyecto. Por último, se proponen posibles líneas de trabajo a futuro.

7.1. Trabajo Futuro

Durante el transcurso del proyecto se identificaron distintas líneas de trabajo futuro sobre las cuales se puede trabajar para mejorar la herramienta. A continuación delineamos algunos de los trabajos que se podrían seguir realizando en el marco de este proyecto.

Como es posible que se intercambien grandes volúmenes de datos entre organizaciones, es necesario que se diseñe e implemente un método para correlacionar la información intercambiada, de esta forma se agrupan los datos, facilitando de esta forma el trabajo de los analistas. Con dicha información correlacionada se pueden presentar reportes a los analistas resumiendo el contenido existente en el sistema. Durante el diseño y análisis de la herramienta se tuvo en consideración dicha funcionalidad pero no se implementó y no hubo una investigación profunda referente a esto, por lo cual se deja para un trabajo a futuro. De la misma forma se consideró durante el diseño y análisis la sanitización de la información pero tampoco se implementó en el prototipo dicha funcionalidad quedando para un trabajo a futuro. Dicha funcionalidad permite filtrar datos sensibles que podrían comprometer la imagen o la integridad de las organizaciones.

Respecto a la implementación realizada, queda planteado realizar una mejor interacción entre RTIR y el *plugin* de forma de extender las funcionalidades ya existentes en RTIR así como agregar algunas nuevas. Entre

dichas mejoras se encuentra crear los incidentes y relacionarlos con los cyber observables existentes en TAXII App. Además permitir agregar nuevos cyber observables a incidentes ya existentes. Esto, junto con una interfaz más amigable facilitaría el trabajo de los analistas.

Otro de los puntos a mejorar en la implementación del cliente TAXII es el seguimiento de los usuarios u organizaciones con las que se intercambia la información de seguridad y filtrar a que datos pueden acceder los distintos usuarios. De esta forma se podría realizar un modelo RBAC para el acceso a la información.

7.2. Conclusiones

En los últimos año se ha visto un incremento en la cantidad de equipos conectados a las redes de las organizaciones, a su vez han aumentado los usuarios malintencionados. Estos cuentan cada vez con mejores herramientas y procedimientos mas sofisticados con los cuales realizar ataques. Debido a esto es necesario que las estrategias de defensa se adapten a los nuevos actores y ataques. Para responder a esto, las organizaciones han recurrido al intercambio de información de amenazas con la finalidad de identificar de mejor forma las actividades de sus adversarios con la finalidad de obtener los mejores resultados posibles de sus defensas.

Durante el estado de arte se vieron diferentes problemas que dificultan el intercambio de información de seguridad entre las organizaciones. Estos van desde problemas que son estrictamente técnicos ha otros que son de índole política. En este punto se decidió centrarse en los problemas que son estrictamente técnicos y se identificaron cuales son los enfoques utilizados en la actualidad para el intercambio de información. Se vio que las herramientas que son utilizadas actualmente se enfocan en una parte del problema y que en muchos casos el intercambio de información es realizado entre humanos.

Durante el transcurso del proyecto se vio el fuerte interés que existe en las organizaciones por llegar a un consenso respecto a estándares utilizados en el intercambio de información de seguridad. En particular resultaron llamativas las iniciativas de MITRE que cuentan con el apoyo tanto de organizaciones provenientes del sector publico así como del sector privado y las cuales provienen de distintos sectores de la industria.

La existencia de estos estándares es fundamental para lograr un buen nivel de automatización. Si bien se puede automatizar sin que un estándar

sea apoyado por la comunidad, el esfuerzo que se debe realizar para poder integrar conocimientos es muy grande. Esta integración de conocimientos es clave para fomentar el avance de las herramientas, técnicas y metodologías, que permitan contrarrestar el impacto de los ataques automatizados.

El esfuerzo encabezado por MITRE busca desarrollar un conjunto de estándares referentes a la caracterización e intercambio de información de seguridad. Intenta proveer lenguajes que permitan facilitar la comunicación de la información de seguridad de forma precisa. Algunos de los lenguajes son CYBOX, STIX, CAPEC, entre otros. Junto con estos esfuerzos utilizados para la representación de la información se encuentra TAXII. TAXII también es un esfuerzo de MITRE que busca estandarizar el intercambio confiable y automático de información. TAXII tiene como finalidad ayudar en el intercambio de información entre las organizaciones que lo deseen utilizando relaciones y sistemas existentes. No se busca que TAXII defina métodos de confianza entre las organizaciones o cualquier aspecto no técnico del intercambio de información de seguridad.

También durante el estudio del arte se estudio RTIR y algunas otras herramientas que cuentan con características similares. RTIR ha sido una herramienta desarrollada junto con CSIRTs para el manejo de incidentes de seguridad. Permitiendo que se agregue información asociada a estos.

El estudio del arte no fue una tarea sencilla debido a la amplitud de la temática tratada y por la cantidad de información encontrada. Se tuvo a disposición mucha información de múltiples fuentes. Se vieron temas referentes a la correlación de la información así como de la sanitización de ésta y como estas dos tareas afectan fuertemente el trabajo y la reputación de las organizaciones. A su vez se dio mucha importancia a la representación de información de forma estructurada.

Luego de terminado el estado del arte se realizo el análisis evaluando los distintos problemas identificados durante el estado del arte. Se identificaron algunos problemas los cuales era de interés para el proyecto. De dichos problemas se identificaron requerimientos que son deseables en una herramienta que intercambia información de seguridad.

Se realizó el diseño de la herramienta utilizando STIX, TAXII y RTIR y centrándose en el intercambio y la representación de información de forma estructurada. El diseño realizado buscó solucionar los requerimientos planteados en el análisis. De ésta forma, se tuvieron en cuenta durante el diseño problemáticas referentes a la correlación y sanitización de la información que

se dejaron planteados como trabajos futuros. La arquitectura planteada busca ser extensible y modular. Permitiendo la interacción con herramientas de gestión de incidentes distintas a RTIR y la extensión de las funcionalidades ya existentes en el sistema.

Luego de realizado el diseño realizó un prototipo de la herramienta propuesta. Esta intercambia información utilizando el protocolo TAXII y representando la información por medio de STIX. El manejo de la información se realiza desde RTIR dando de alta los datos necesarios para realizar el intercambio. Es necesario destacar la dificultad en la implementación de un *plugin* para RTIR que no fue una tarea sencilla y que se realizó por medio de ingeniería reversa de un *plugin* ya existente. Además se implementó un cliente que implementa el protocolo TAXII para realizar el intercambió que además cuenta con una API REST que puede consumir el *plugin* RTIR.

Finalmente se ideó un caso de estudio en el que se intercambian cyber observables entre dos organizaciones por medio del protocolo TAXII utilizando los distintos servicios provistos en su especificación.

Bibliografía

[1] Apache software foundation. http://www.apache.org/.

[2] Bp. www.bestpractical.com.

[3] A brief introduction to rtir. http://www.slideshare.net/obrajesse/a-brief-introduction-to-rtir-presentation.

[4] Building security in us. https://buildsecurityin.us-cert.gov/.

[5] Capec. http://capec.mitre.org/.

[6] Celery. http://www.celeryproject.org/.

[7] Cve mitre. https://cve.mitre.org/.

[8] Cwe. http://cwe.mitre.org/.

[9] Cybox. https://cybox.mitre.org/language/version2.1/#CYBOX%20samples.

[10] Django. https://www.djangoproject.com/.

[11] Documentation of pythonstix. http://stix.readthedocs.org/en/latest/.

[12] Git of pythonstix. https://github.com/STIXProject/python-stix.

[13] Internet engineering task force. https://www.ietf.org/.

[14] Json. http://json.org/.

[15] libtaxii. https://github.com/TAXIIProject/libtaxii.

[16] Lwp. http://search.cpan.org/~mschilli/libwww-perl/lib/LWP.pm.

[17] Maec. http://maec.mitre.org/.

[18] Making security measurable. http://makingsecuritymeasurable.mitre.org/.

[19] Mitre. http://www.mitre.org/.

[20] Mysql. http://www.mysql.com.

[21] Nist. http://nvd.nist.gov/cpe.cfm.

[22] Oval. http://oval.mitre.org/.

[23] Rest. http://predic8.com/rest-webservices.htm.

[24] Rest framework. http://www.django-rest-framework.org/.

[25] Scap. http://scap.nist.gov/.

[26] Spf. http://vim.spf13.com/.

[27] Vim. http://www.vim.org/.

[28] Xml. http://www.w3.org/XML/.

[29] Yeti. https://github.com/TAXIIProject/yeti.

[30] Inette Furey Damir Rajnovic Robert Martin Takeshi Takahashi Anthony Rutkowski, Youki Kadobayashi. *CYBEX - The Cybersecurity Information Exchange Framework*. 2010.

[31] Mitre Corporation. *Active Defense Strategy for Cyber*. Mitre Corporation, 2012.

[32] Mitre Corporation. *Cyber Information-Sharing Models: An Overview*. Mitre Corporation, 2012.

[33] Mitre Corporation. *A New Cyber Defense Playbook*. Mitre Corporation, 2012.

[34] Mitre Corporation. *STIX Whitepaper*. Mitre Corporation, 2012.

[35] Mitre Corporation. *The TAXII HTTP Protocol Binding Specification*. Mitre Corporation, 2013.

[36] Mitre Corporation. *The TAXII HTTP Protocol Binding Specification*. Mitre Corporation, 2013.

[37] Mitre Corporation. *The TAXII Services Specification.* Mitre Corporation, 2013.

[38] Mitre Corporation. *The TAXII XML Message Binding Specification.* Mitre Corporation, 2013.

[39] Antonio Montes Cristine Hoepers, Nandamudi L. Vijaykumar. *HIDEF: a Data Exchange Format for Information Collected in Honeypots and Honeynets.* InfoComp - VOLUME 7 - NUMBER 1, 2008.

[40] M.J. Cloppert E.M. Hutchins and R.M Amin PH.D. *Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains.* 2005.

[41] D. Curry H. Debar and B. Feinstein. *RFC 4765 - IDMEF.* IETF, 2007.

[42] M. Richard J. Connolly, M. Davidson and C. Skorupka. *TAXII Whitepaper.* Mitre Corporation, 2012.

[43] D. Rolsky D. Chamberlain J. Vincent, R. Spier and R. Foley. *RT Essentials.* A. Randal and T. Apandy, 2005.

[44] Sven Henkel Marko Jahnke, Michael Bussmann. *Components for Cooperative Intrusion Detection in Dynamic Coalition Enviroments.* RTO IST Symposium on Adaptive Defence in Unclassified Networks, 2004.

[45] Sven Henkel Marko Jahnke, Michael Bussmann. *Components for Cooperative Intrusion Detection in Dynamic Coalition Enviroments.* RTO IST Symposium on Adaptive Defence in Unclassified Networks, 2004.

[46] K. Moriarty and B. Trammell. *RFC 6546 - RID Messages.* IETF, 2010.

[47] J. Meijer R. Danyliw and Y. Demchenko. *RFC 5070 - IODEF.* IETF, 2007.

[48] Julio Saráchaga. *Estado del arte para advanced threats, information sharing and collaboration.* 2013.

[49] B. Trammell. *RFC 6546 - Transport of RID messages over HTTP/TLS.* IETF, 2012.

Índice de figuras

Capítulo 8

Anexo

8.1. Implementación

8.1.1. Servicios TAXII Implementados

```
def inbox(request, inbox):
"Handles TAXII Inbox Service requests."

def poll(request):
"Handles TAXII Poll Service requests."

def feed(request):
"Handles TAXII Feed Managment Service requests."

def subscription(request):
"Handles TAXII Subscription Service requests."
```

8.1.2. API REST implementada

```
class UserViewSet(viewsets.ModelViewSet):
#Gets, lists, creates or updates users.

class ProtocolBindingIdViewSet(viewsets.ModelViewSet):
#Gets, lists, creates or updates ProtocolBindingIds

class ContentBindingIdViewSet(viewsets.ModelViewSet):
#Gets, lists, creates or updates ContentBindingIds

class MessageBindingIdViewSet(viewsets.ModelViewSet):
#Gets, lists, creates or updates MessageBindingIds

class DataFeedPushMethodViewSet(viewsets.ModelViewSet):
#Gets, lists, creates or updates DataFeedPushMethods

class DataFeedPollInformationViewSet(viewsets.ModelViewSet):
#Gets, lists, creates or updates DataFeedPollInformations

class RemoteDataFeedPollInformationViewSet(viewsets.ModelViewSet):
#Gets, lists, creates or updates RemoteDataFeedPollInformations

class DataFeedSubscriptionMethodViewSet(viewsets.ModelViewSet):
#Gets, lists, creates or updates DataFeedSubscriptionMethods

class ContentBlockViewSet(viewsets.ModelViewSet):
#Gets, lists, creates or updates ContentBlocks

class DataFeedViewSet(viewsets.ModelViewSet):
#Gets, lists, creates or updates DataFeeds

class RemoteDataFeedViewSet(viewsets.ModelViewSet):
#Gets, lists, creates or updates RemoteDataFeeds

class DataFeedSubscriptionViewSet(viewsets.ModelViewSet):
#Gets, lists, creates or updates SubscriptionFeeds
```

```
class InboxViewSet(viewsets.ModelViewSet):
#Gets, lists, creates or updates Inboxes

class RemoteInboxViewSet(viewsets.ModelViewSet):
#Gets, lists, creates or updates RemoteInboxes

class ContentBlockRTIRViewSet(viewsets.ModelViewSet):
#Gets, lists, creates or updates ContentBlockRTIRs

class TAXIIServicesViewSet(viewsets.ModelViewSet):
#Gets, lists, creates or updates TAXIIServices

class FeedManagmentServicesViewSet(viewsets.ModelViewSet):
#Gets, lists, creates or updates FeedManagementServices

def get(request):
#Given the id of a TAXII Service we make a FeedInformation request to
that service address.
#The response is a list of the feed names of the TAXII client and
a list of all protocol bindings,
content binding and message binding.

def register(request):
#Given the id of a TAXII service we get the data feeds of the TAXII Client and
copy them to the current system.

def create(request):
#When in GET method return all the Content Blocks.
#When in POST method, given a content binding id, a title, description and content
we create a Content Block.

def send→ (request) :
#Given the id of a DataFeed Subscription we get the Data Feeds and send it to
send it to the inbox service of the organization of the subscription service.

def poll(request):
#Given the id of a remote data feed,
```

we get the poll service instances and for each make a poll request.

def get(request):
#We get all the date feed subsctiptions and
return the id, adress and data feed name of each.

def subscription→ (*request*) :
#Given the id of a TAXII Service and the id of a Data Feed and
the service id we make a Manage Feed Subscription request for that Data Feed.

8.1.3. Entidades desarrolladas utilizadas

```
class ProtocolBindingId():
”
Represents a protocol binding id, used to establish the exchange protocol
supported by a TAXII implementation.
Ex:
HTTP protocol binding id : ürn:taxii.mitre.org:protocol:http:1.0”
HTTPS protocol binding id : ürn:taxii.mitre.org:protocol:https:1.0”
”
title = CharField(blank=True)
description = TextField(blank=True)
bindingid = CharField(maxlength=MAXIDLEN)
datecreated = DateTimeField(autonowadd=True)
dateupdated = DateTimeField(autonow=True)

class ContentBindingId():
”
Represents a content binding id, used to establish the supported content
types for a given TAXII exchange (e.g., Poll, Inbox, etc.).
Ex:
STIX v1.0 content binding id : ürn:stix.mitre.org:xml:1.0”
”
title = CharField(blank=True)
description = TextField(blank=True)
bindingid = CharField(maxlength=MAXIDLEN)
datecreated = DateTimeField(autonowadd=True)
dateupdated = DateTimeField(autonow=True)

class MessageBindingId():
”
Represents a message binding id, used to establish the supported syntax
for a given TAXII exchange, .e.g., XML”.
Ex:
XML message binding id : ürn:taxii.mitre.org:message:xml:1.0”
”
title = CharField(blank=True)
description = TextField(blank=True)
```

```
bindingid = CharField(maxlength=MAXIDLEN)
datecreated = DateTimeField(autonowadd=True)
dateupdated = DateTimeField(autonow=True)

class DataFeedPushMethod():
"
Used to establish the protocols that can be used to push content via
a subscription. This appears in a Feed Information Response message,
as defined by the TAXII Services Specification.
"
title = CharField(blank=True)
description = TextField(blank=True)
protocolbinding = ForeignKey(ProtocolBindingId)
messagebinding = ForeignKey(MessageBindingId)
datecreated = DateTimeField(autonowadd=True)
dateupdated = DateTimeField(autonow=True)

class DataFeedPollInformation():
"
Used to establish the supported protocols and address of a Data Feed.
This appears in a Feed Information Response message, as defined by the
TAXII Services Specification.
"
title = CharField(blank=True)
description = TextField(blank=True)
address = URLField()
protocolbinding = ForeignKey(ProtocolBindingId)
messagebindings = ManyToManyField(MessageBindingId)
datecreated = DateTimeField(autonowadd=True)
dateupdated = DateTimeField(autonow=True)

class DataFeedSubscriptionMethod():
"
Used to identify the protocol and address of the TAXII daemon hosting
the Feed Management Service that can process subscriptions for a TAXII
Data Feed. This appears in a Feed Information Response message, as defined
by the TAXII Services Specification.
"
```

```
title = CharField(blank=True)
description = TextField(blank=True)
address = URLField()
protocolbinding = ForeignKey(ProtocolBindingId)
messagebindings = ManyToManyField(MessageBindingId)
datecreated = DateTimeField(autonowadd=True)
dateupdated = DateTimeField(autonow=True)

class ContentBlock():
Represents the content block of a TAXII Poll Response or Inbox message."
title = CharField(blank=True)  not required by TAXII
description = TextField(blank=True)  not required by TAXII
timestamplabel = DateTimeField(default=lambda:datetime.datetime.now(tzutc()))
submittedby = ForeignKey(User, blank=True, null=True)
messageid = CharField(blank=True)
contentbinding = ForeignKey(ContentBindingId)
content = TextField()
padding = TextField(blank=True)
datecreated = DateTimeField(autonowadd=True)
dateupdated = DateTimeField(autonow=True)

class DataFeed():
Represents a TAXII Data Feed"
name = CharField()
description = TextField(blank=True)
users = ManyToManyField(User, blank=True, null=True)
supportedcontentbindings = ManyToManyField(ContentBindingId)
pushmethods = ManyToManyField(DataFeedPushMethod)
pollserviceinstances = ManyToManyField(DataFeedPollInformation)
subscriptionmethods = ManyToManyField(DataFeedSubscriptionMethod, blank=True, null=True)
contentblocks = ManyToManyField(ContentBlock, blank=True, null=True)
datecreated = DateTimeField(autonowadd=True)
dateupdated = DateTimeField(autonow=True)

class DataFeedSubscription():
Represents a Data Feed Subscription. This is not used by TAXII at the moment."
subscriptionid = CharField(unique=True)  uri formatted subscription id
user = ForeignKey(User)
```

```
datafeed = ForeignKey(DataFeed)
datafeedmethod = ForeignKey(DataFeedSubscriptionMethod)
active = BooleanField(default=True)
expires = DateTimeField()
datecreated = DateTimeField(autonowadd=True)
dateupdated = DateTimeField(autonow=True)

class Inbox():
"
Characterizes a TAXII Inbox. Inboxes are the mechanism by which TAXII consumers
receive data from TAXII publishers. This Inbox implementation allows an Inbox
to be "bound"to zero or more Data Feeds, meaning that data received by an Inbox
can populate Data Feeds if a user configures it as such.
"
name = CharField(unique=True)
description = TextField(blank=True)
supportedcontentbindings = ManyToManyField(ContentBindingId)
supportedmessagebindings = ManyToManyField(MessageBindingId)
contentblocks = ManyToManyField(ContentBlock, blank=True, null=True)
supportedprotocolbinding = ForeignKey(ProtocolBindingId)
datafeeds = ManyToManyField(DataFeed, blank=True, null=True)
users = ManyToManyField(User, blank=True, null=True)
datecreated = DateTimeField(autonowadd=True)
dateupdated = DateTimeField(autonow=True)

class ContentBlockRTIR():
Represents the nexus between the RTIR ticket and the content block."
rtirid = IntegerField(unique=True)
contentblock = ForeignKey(ContentBlock)

class RemoteDataFeedPollInformation():
Represents DataFeed Poll Information of remote TAXII clients "
title = CharField(blank=True)
description = TextField(blank=True)
address = URLField()
protocolbinding = ForeignKey(ProtocolBindingId)
messagebindings = ManyToManyField(MessageBindingId)
datecreated = DateTimeField(autonowadd=True)
```

```
dateupdated = DateTimeField(autonow=True)

class RemoteDataFeed():
Represents DataFeeds of remote TAXII clients "
name = CharField(maxlength=MAXTITLELEN)
description = TextField(blank=True)
supportedcontentbindings = ManyToManyField(ContentBindingId)
pushmethods = ManyToManyField(DataFeedPushMethod)
pollserviceinstances = ManyToManyField(RemoteDataFeedPollInformation, null=True)
subscriptionmethods = ManyToManyField(DataFeedSubscriptionMethod, blank=True, null=True)
contentblocks = ManyToManyField(ContentBlock, blank=True, null=True)
datecreated = DateTimeField(autonowadd=True)
dateupdated = DateTimeField(autonow=True)

class RemoteInbox():
Represents Inboxes of remote TAXII clients "
name = CharField(unique=True)  this will become part of the URL where it can be accessed at
description = TextField(blank=True)
supportedcontentbindings = ManyToManyField(ContentBindingId)
supportedmessagebindings = ManyToManyField(MessageBindingId)
supportedprotocolbinding = ForeignKey(ProtocolBindingId)
datafeed = ForeignKey(DataFeed, blank=True)
address = URLField()
datecreated = DateTimeField(autonowadd=True)
dateupdated = DateTimeField(autonow=True)

class TAXIIServices():
Represents all TAXII Services available "
name = CharField(unique=True)
description = TextField(blank=True)
inbox = URLField()
poll = URLField()
feedmanagment = URLField()
subscription = URLField()
```

8.1.4. Cyber Obsevables utilizados en el caso de estudio

```
<cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance" xmlns:cybox="http://cybox.mitre.org/cybox-2" xmlns:
    cyboxCommon="http://cybox.mitre.org/common-2" xmlns:
    ArtifactObj="http://cybox.mitre.org/objects#ArtifactObject-2"
     xmlns:example="http://example.com/" xsi:schemaLocation="http
    ://cybox.mitre.org/cybox-2 http://cybox.mitre.org/XMLSchema/
    core/2.1/cybox_core.xsd      http://cybox.mitre.org/objects#
    ArtifactObject-2 http://cybox.mitre.org/XMLSchema/objects/
    Artifact/2.1/Artifact_Object.xsd      " cybox_major_version
    ="2" cybox_minor_version="1" cybox_update_version="0">
    <cybox:Observable id="example:Observable-6dde4f72-cca0-44f0
        -8206-f381accf6b87">
        <cybox:Description>
            This Observable specifies an instance of an Artifact
                object, specifically some
            network traffic that was captured in a PCAP file and
                then base64 encoded for transport.
        </cybox:Description>
        <cybox:Object id="example:Object-2a1f9a17-b799-4a17-b4ca
            -f3cb018ad89f">
            <cybox:Properties xsi:type="ArtifactObj:
                ArtifactObjectType" type="Network Traffic">
                <ArtifactObj:Packaging>
                    <ArtifactObj:Encoding algorithm="Base64"/>
                </ArtifactObj:Packaging>
                <ArtifactObj:Raw_Artifact>1MOyoQIABAAAAAAAAAAAAP
                    //
                    AAABAAAAsmdKQq6RBwBGAAAARgAAADAnzJBjADgGLEMrQgARQAAOAAAQABAEW
                    +
                    wAAIARmT7AqKoUwKiqCAA1gBsAQMclEDKBgAABAAEAAAAABmdvb2dsZQNjb20AABA
                    /
                    HBABGAAAARgAAADAnzJBjADgGLEMrQgARQAAOAAAQABAEWVHwKiqCMCoqhSAGw
                    /Z0pCn+
                    YGAEYAAABGAAAAAOAYsQytAMCfMkGMCABFAAA4zM0AAIARmHnAqKoUwKiqCAA
                    +
                    CtZu7gYAAAQABAAAAAAMxMDQBOQMxOTICNjYHaW4tYWRkcgRhcnBhAAAMAAH
                    /
                    dwoASgAAAEoAAAAAwJ8yQYwA4BixDK0IAEUAADwAAEAAQBFlQ8CoqgjAqKoUgBsA
                    +
                    QAAgBGVOcCoqhTAqKoIADWAGwA4oxd1wIGAAAEAAQAAAAADd3d3Bm5ldGJzZANvc
                    /+
                    UpprW2hKQrD8BwBKAAAASgAAADAnzJBjADgGLEMrQgARQAAPAAAQABAEWVDwK
                    +
                    BwBmAAAAZgAAADgGLEMrQDAnzJBjAgARQAAWNRPAACAEZDXwKiqFMCoqggANYA
                    +AAEAAcC4IH//
```

```
                        lKaa2RoSkKSOgsASgAAAEoAAAAAwJ8yQYwA4BixDK0IAEUAADwAAEAAQBFlQ8CoqgjA
                        +1
                        TkAAIARkAfAqKoUwKiqCAA1gBsAKryJ3KKBgAABAAAAAAAAAA3d3dwFsBmdvb2dsZG
                        +
                        BgAABAAAAAAAAA3d3dwdleGFtcGxlA2NvbQAAHAABomhKQhCDDABPAAAATwAAAAD
                        +
                        wKiqCMCoqhSAGwA1AC1EKCZtAQAAAQAAAAAAAAN3d3cHZXhhbXBsZQdub3RnaW5c
                        /AqKoUwKiqCAA1gBsALb+
                        kJm2FgwABAAAAAAAAA3d3dwdleGFtcGxlB25vdGdpbmgAABwAAcFoSkIsFQoARwA
                        +4
                        wEAAAEAAAAAAAADd3d3A2lzYwNvcmcAAP8AAcFoSkLIMAsAcwAAAHMAAAAA4BixD
                        +44
                        GAAAEAAgAAAAADd3d3A2lzYwNvcmcAAP8AAcAMABwAAQAAAlgAECABBPgAAAACAAA
                        /
                        CwBSAAAAUgAAADAnzJBjADgGLEMrQgARQAARAAAQABAEWU7wKiqCMCoqhSAHAA
                        /
                        kwWlOFgAABAAEAAAAAATEBMAEwAzEyNwdpbi1hZGRyBGFycGEAAAwAAcAMAAwA
                        +
                        HPZDQQYwKiqOAA1BqsAX7VsMm6FgwABAAAAAAAABV9sZGFwBF90Y3AXRGVmY
                        +6
                        BQBTAAAAUwAAABgCEXkVQASqQAyIwgARQAARQAAQAA6Efih2Q0EGMCoqjgANQ
                        </ArtifactObj:Raw_Artifact>
                    </cybox:Properties>
                </cybox:Object>
            </cybox:Observable>
        </cybox:Observables>

        <cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-
            instance" xmlns:cybox="http://cybox.mitre.org/cybox-2" xmlns:
            cyboxCommon="http://cybox.mitre.org/common-2" xmlns:
            ArtifactObj="http://cybox.mitre.org/objects#ArtifactObject-2"
             xmlns:example="http://example.com/" xsi:schemaLocation="http
            ://cybox.mitre.org/cybox-2 http://cybox.mitre.org/XMLSchema/
            core/2.1/cybox_core.xsd      http://cybox.mitre.org/objects#
            ArtifactObject-2 http://cybox.mitre.org/XMLSchema/objects/
            Artifact/2.1/Artifact_Object.xsd      " cybox_major_version
            ="2" cybox_minor_version="1" cybox_update_version="0">
            <cybox:Observable id="example:Observable-cdcf44b5-317b-4ae5
                -99e0-b601fe90fe3d">
                <cybox:Description>
                    This Observable specifies an example pattern against
                        an Artifact object,
                    testing the contents of the artifact for a
                        particular byte string that
                    represents a URL captured in a PCAP.
                </cybox:Description>
                <cybox:Object id="example:Object-2c7bef9b-9eb7-4d3b-833f
                    -49b70b3c4956">
                    <cybox:Properties xsi:type="ArtifactObj:
```

```
                ArtifactObjectType" type="Network Traffic">
                <ArtifactObj:Raw_Artifact condition="Contains
                    ">777777076578616D706C6503636F6D</ArtifactObj
                    :Raw_Artifact>
            </cybox:Properties>
        </cybox:Object>
    </cybox:Observable>
</cybox:Observables>

<cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance" xmlns:cybox="http://cybox.mitre.org/cybox-2" xmlns:
    cyboxCommon="http://cybox.mitre.org/common-2" xmlns:FileObj="
    http://cybox.mitre.org/objects#FileObject-2" xmlns:
    cyboxVocabs="http://cybox.mitre.org/default_vocabularies-2"
    xmlns:example="http://example.com" xsi:schemaLocation="http
    ://cybox.mitre.org/cybox-2 http://cybox.mitre.org/XMLSchema/
    core/2.1/cybox_core.xsd      http://cybox.mitre.org/objects#
    FileObject-2 http://cybox.mitre.org/XMLSchema/objects/File
    /2.1/File_Object.xsd      http://cybox.mitre.org/
    default_vocabularies-2 http://cybox.mitre.org/XMLSchema/
    default_vocabularies/2.1/cybox_default_vocabularies.xsd      "
     cybox_major_version="2" cybox_minor_version="1"
    cybox_update_version="0">
    <cybox:Observable id="example:Observable-a727a717-1852-4c79
        -9a16-2f3a8b4632c2">
        <cybox:Event id="example:Event-44578866-b0c5-4551-84dd-0
            f1f02f8210f">
            <cybox:Actions>
                <cybox:Action id="example:Action-a18a058c-effa
                    -4060-b8be-25e1b1ade75f" action_status="
                    Success" context="Host" timestamp="2013-04-08
                    T09:22:00.0Z">
                    <cybox:Type xsi:type="cyboxVocabs:
                        ActionTypeVocab-1.0">Create</cybox:Type>
                    <cybox:Name xsi:type="cyboxVocabs:
                        ActionNameVocab-1.0">Create File</cybox:
                        Name>
                    <cybox:Associated_Objects>
                        <cybox:Associated_Object id="example:
                            Object-5ec92e95-a31f-470b-97c4-
                            aa9046189fbb">
                            <cybox:Properties xsi:type="FileObj:
                                FileObjectType">
                                <FileObj:File_Name>foobar.dll</
                                    FileObj:File_Name>
                                <FileObj:File_Path>C:\Windows\
                                    system32</FileObj:File_Path>
                                <FileObj:Hashes>
                                    <cyboxCommon:Hash>
```

```
                                                    <cyboxCommon:Type>MD5</
                                                        cyboxCommon:Type>
                                                    <cyboxCommon:
                                                        Simple_Hash_Value
                                                        datatype="hexBinary
                                                        ">6
                                                        E48C348D742A931EC2CE90ABD7DAC6A
                                                        </cyboxCommon:
                                                        Simple_Hash_Value>
                                                </cyboxCommon:Hash>
                                            </FileObj:Hashes>
                                        </cybox:Properties>
                                        <cybox:Association_Type xsi:type="
                                            cyboxVocabs:
                                            ActionObjectAssociationTypeVocab
                                            -1.0">Affected</cybox:
                                            Association_Type>
                                    </cybox:Associated_Object>
                                </cybox:Associated_Objects>
                            </cybox:Action>
                        </cybox:Actions>
                    </cybox:Event>
                </cybox:Observable>
            </cybox:Observables>

            <cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-
                instance" xmlns:cybox="http://cybox.mitre.org/cybox-2" xmlns:
                cyboxCommon="http://cybox.mitre.org/common-2" xmlns:URIObject
                ="http://cybox.mitre.org/objects#URIObject-2" xmlns:example="
                http://example.com/" xsi:schemaLocation="http://cybox.mitre.
                org/cybox-2 http://cybox.mitre.org/XMLSchema/core/2.1/
                cybox_core.xsd http://cybox.mitre.org/objects#URIObject-2
                 http://cybox.mitre.org/XMLSchema/objects/URI/2.1/URI_Object.
                xsd " cybox_major_version="2" cybox_minor_version="1"
                cybox_update_version="0">
                <cybox:Observable id="example:Observable-0b9af310-0d5a-4c44-
                    bdd7-aea3d99f13b6">
                    <cybox:Object id="example:Object-15be6630-b2df-4bf9
                        -8750-3f45ca9e19cf">
                        <cybox:Properties xsi:type="URIObject:URIObjectType"
                            type="Domain Name">
                            <URIObject:Value>example.com</URIObject:Value>
                        </cybox:Properties>
                    </cybox:Object>
                </cybox:Observable>
            </cybox:Observables>

            <cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-
                instance" xmlns:cybox="http://cybox.mitre.org/cybox-2" xmlns:
```

```
    cyboxCommon="http://cybox.mitre.org/common-2" xmlns:URIObject
    ="http://cybox.mitre.org/objects#URIObject-2" xmlns:example="
    http://example.com/" xsi:schemaLocation="http://cybox.mitre.
    org/cybox-2 http://cybox.mitre.org/XMLSchema/core/2.1/
    cybox_core.xsd      http://cybox.mitre.org/objects#URIObject-2
     http://cybox.mitre.org/XMLSchema/objects/URI/2.1/URI_Object.
    xsd      " cybox_major_version="2" cybox_minor_version="1"
    cybox_update_version="0">
     <cybox:Observable id="example:Observable-1c9af310-0d5a-4c44-
         bdd7-aea3d99f13b6">
         <cybox:Object id="example:Object-26be6630-b2df-4bf9
             -8750-3f45ca9e19cf">
             <cybox:Properties xsi:type="URIObject:URIObjectType"
                  type="Domain Name">
                 <URIObject:Value condition="StartsWith">mega</
                     URIObject:Value>
             </cybox:Properties>
         </cybox:Object>
     </cybox:Observable>
</cybox:Observables>

<cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance" xmlns:cybox="http://cybox.mitre.org/cybox-2" xmlns:
    cyboxCommon="http://cybox.mitre.org/common-2" xmlns:
    AddressObj="http://cybox.mitre.org/objects#AddressObject-2"
    xmlns:example="http://example.com/" xsi:schemaLocation="http
    ://cybox.mitre.org/cybox-2 http://cybox.mitre.org/XMLSchema/
    core/2.1/cybox_core.xsd      http://cybox.mitre.org/objects#
    AddressObject-2 http://cybox.mitre.org/XMLSchema/objects/
    Address/2.1/Address_Object.xsd      " cybox_major_version="2"
    cybox_minor_version="1" cybox_update_version="0">
     <cybox:Observable id="example:Observable-0b9af309-0d5a-4c44-
         bdd7-aea3d99f13b6">
         <cybox:Object id="example:Object-15be6630-c2df-4bf9
             -8750-3f45ca9e19cf">
             <cybox:Properties xsi:type="AddressObj:
                 AddressObjectType" category="ipv4-addr">
                 <AddressObj:Address_Value>192.168.0.5</
                     AddressObj:Address_Value>
             </cybox:Properties>
         </cybox:Object>
     </cybox:Observable>
</cybox:Observables>

<cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance" xmlns:cybox="http://cybox.mitre.org/cybox-2" xmlns:
    cyboxCommon="http://cybox.mitre.org/common-2" xmlns:
    AddressObj="http://cybox.mitre.org/objects#AddressObject-2"
    xmlns:example="http://example.com/" xsi:schemaLocation="http
```

```
    ://cybox.mitre.org/cybox-2 http://cybox.mitre.org/XMLSchema/
    core/2.1/cybox_core.xsd      http://cybox.mitre.org/objects#
    AddressObject-2 http://cybox.mitre.org/XMLSchema/objects/
    Address/2.1/Address_Object.xsd      " cybox_major_version="2"
    cybox_minor_version="1" cybox_update_version="0">
     <cybox:Observable id="example:Observable-0b9af309-0d5a-4c44-
         bdd7-aea3d99f13b6">
      <cybox:Object id="example:Object-15be6630-c2df-4bf9
          -8750-3f45ca9e19cf">
       <cybox:Properties xsi:type="AddressObj:
           AddressObjectType" category="ipv4-addr">
        <AddressObj:Address_Value pattern_type="Regex"
            apply_condition="ANY">192\.168\.1\.(0|1|2)</
            AddressObj:Address_Value>
       </cybox:Properties>
      </cybox:Object>
     </cybox:Observable>
</cybox:Observables>

<cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance" xmlns:cybox="http://cybox.mitre.org/cybox-2" xmlns:
    cyboxCommon="http://cybox.mitre.org/common-2" xmlns:
    AddressObj="http://cybox.mitre.org/objects#AddressObject-2"
    xmlns:example="http://example.com/" xsi:schemaLocation="http
    ://cybox.mitre.org/cybox-2 http://cybox.mitre.org/XMLSchema/
    core/2.1/cybox_core.xsd      http://cybox.mitre.org/objects#
    AddressObject-2 http://cybox.mitre.org/XMLSchema/objects/
    Address/2.1/Address_Object.xsd      " cybox_major_version="2"
    cybox_minor_version="1" cybox_update_version="0">
     <cybox:Observable id="example:Observable-0b9af309-0d5a-4c44-
         bdd7-aea3d99f13b6">
      <cybox:Object id="example:Object-15be6630-c2df-4bf9
          -8750-3f45ca9e19cf">
       <cybox:Properties xsi:type="AddressObj:
           AddressObjectType" category="ipv6-addr">
        <AddressObj:Address_Value>2607:f0d0:1002:51::4</
            AddressObj:Address_Value>
       </cybox:Properties>
      </cybox:Object>
     </cybox:Observable>
</cybox:Observables>

<cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance" xmlns:cybox="http://cybox.mitre.org/cybox-2" xmlns:
    cyboxCommon="http://cybox.mitre.org/common-2" xmlns:
    AddressObj="http://cybox.mitre.org/objects#AddressObject-2"
    xmlns:example="http://example.com/" xsi:schemaLocation="http
    ://cybox.mitre.org/cybox-2 http://cybox.mitre.org/XMLSchema/
    core/2.1/cybox_core.xsd      http://cybox.mitre.org/objects#
```

```
    AddressObject-2 http://cybox.mitre.org/XMLSchema/objects/
    Address/2.1/Address_Object.xsd " cybox_major_version="2"
    cybox_minor_version="1" cybox_update_version="0">
    <cybox:Observable id="example:Observable-0b9af309-0d5a-4c44-
        bdd7-aea3d99f13b6">
        <cybox:Object id="example:Object-15be6630-c2df-4bf9
            -8750-3f45ca9e19cf">
            <cybox:Properties xsi:type="AddressObj:
                AddressObjectType" category="ipv6-addr">
                <AddressObj:Address_Value pattern_type="Regex"
                    apply_condition="ANY">2001:0db8
                    :0000:0000:0000:ff00:0042:832[0-9]</
                    AddressObj:Address_Value>
            </cybox:Properties>
        </cybox:Object>
    </cybox:Observable>
</cybox:Observables>

<cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance" xmlns:cybox="http://cybox.mitre.org/cybox-2" xmlns:
    cyboxCommon="http://cybox.mitre.org/common-2" xmlns:AddrObj="
    http://cybox.mitre.org/objects#AddressObject-2" xmlns:URIObj
    ="http://cybox.mitre.org/objects#URIObject-2" xmlns:FileObj="
    http://cybox.mitre.org/objects#FileObject-2" xmlns:
    EmailMessageObj="http://cybox.mitre.org/objects#
    EmailMessageObject-2" xmlns:cyboxVocabs="http://cybox.mitre.
    org/default_vocabularies-2" xmlns:example="http://example.com
    /" xsi:schemaLocation="http://cybox.mitre.org/cybox-2 http://
    cybox.mitre.org/XMLSchema/core/2.1/cybox_core.xsd http://
    cybox.mitre.org/objects#URIObject-2 http://cybox.mitre.org/
    XMLSchema/objects/URI/2.1/URI_Object.xsd http://cybox.
    mitre.org/objects#FileObject-2 http://cybox.mitre.org/
    XMLSchema/objects/File/2.1/File_Object.xsd http://cybox.
    mitre.org/objects#EmailMessageObject-2 http://cybox.mitre.org
    /XMLSchema/objects/Email_Message/2.1/Email_Message_Object.xsd
        http://cybox.mitre.org/default_vocabularies-2 http://
    cybox.mitre.org/XMLSchema/default_vocabularies/2.1/
    cybox_default_vocabularies.xsd " cybox_major_version="2"
    cybox_minor_version="1" cybox_update_version="0">
    <!-- This collection of observables were observed as part of
        the widespread "Iran-Oil" (among many other names used)
        attack campaign in March 2012 -->
    <cybox:Observable id="example:Observable-1a937ec2-90ab-4e0e-
        a37c-db9b2e66a58e">
        <!-- Receive "Iran-Oil" attack campaign email message
            -->
        <cybox:Event>
            <cybox:Type xsi:type="cyboxVocabs:EventTypeVocab
                -1.0.1">Email Ops</cybox:Type>
```

```
<cybox:Description>Receive "Iran-Oil" attack
    campaign email message.</cybox:Description>
<cybox:Actions>
    <cybox:Action>
        <cybox:Type xsi:type="cyboxVocabs:
            ActionTypeVocab-1.0">Receive</cybox:Type>
        <cybox:Associated_Objects>
            <cybox:Associated_Object id="example:
                Object-51359587-f201-4383-b032-5
                a64522fcd7d">
                <cybox:Properties xsi:type="
                    EmailMessageObj:
                    EmailMessageObjectType">
                    <EmailMessageObj:Header>
                        <EmailMessageObj:To>
                            <EmailMessageObj:
                                Recipient category="e
                                -mail">
                                <AddrObj:
                                    Address_Value>
                                    william.
                                    abnett@gmail.com
                                    </AddrObj:
                                    Address_Value>
                            </EmailMessageObj:
                                Recipient>
                        </EmailMessageObj:To>
                        <EmailMessageObj:From
                            category="e-mail">
                            <AddrObj:Address_Value>
                                wmorrison89@gmail.com
                                </AddrObj:
                                Address_Value>
                        </EmailMessageObj:From>
                        <EmailMessageObj:Subject>
                            Iran's Oil and Nuclear
                            Situation</
                            EmailMessageObj:Subject>
                        <EmailMessageObj:Date
                            datatype="dateTime
                            ">2012-03-02T07:42:24Z</
                            EmailMessageObj:Date>
                    </EmailMessageObj:Header>
                    <EmailMessageObj:Raw_Header
                        datatype="string">
                                Return-Path: &lt
                                    ;
                                    wmorrison89@gmail
                                    .com&gt;
```

```
Received-SPF: pass (google.com: domain of wmorrison89@gmail.com
    designates
10.236.185.4 as permitted sender) client-ip=10.236.185.4;
Authentication-Results: mr.google.com; spf=pass (google.com:
    domain of
wmorrison89@gmail.com designates 10.236.185.4 as permitted
    sender)
smtp.mail=wmorrison89@gmail.com; dkim=pass header.i=
    wmorrison89@gmail.com
Received: from mr.google.com ([10.236.185.4]) by 10.236.185.4
    with SMTP
id t4mr5301660yhm.129.1330692273662 (num_hops = 1); Fri, 02 Mar
    2012
04:44:33 -0800 (PST)
MIME-Version: 1.0
Received: by 10.236.185.4 with SMTP id t4mr4236541yhm
    .129.1330692265380;
Fri,
02 Mar 2012 04:44:25 -0800 (PST)
Received: by 10.147.35.14 with HTTP; Fri, 2 Mar 2012 04:44:24
    -0800 (PST)
In-Reply-To:
&lt;CADY6HTa-jmaqmtVyyT-nLz6reztNjcs-617wL4bt9YBOGu+h4w@mail.
    gmail.com&gt;
References:
&lt;CADY6HTa-jmaqmtVyyT-nLz6reztNjcs-617wL4bt9YBOGu+h4w@mail.
    gmail.com&gt;
Date: Fri, 2 Mar 2012 07:44:24 -0500
Message-ID:
&lt;CADY6HTZ6oopY5v6WkYU81YcSQw3X124CK_Fx4jhnhUiU3Y9z6A@mail.
    gmail.com&gt;
Subject: Iran's Oil and Nuclear Situation
From: william abnett &lt;wmorrison89@gmail.com&gt;
To: william.abnett &lt;william.abnett@gmail.com&gt;
Content-Type: multipart/mixed; boundary="20
    cf303f67fac8928804ba41efd5"
                                        </EmailMessageObj:
                                            Raw_Header>
                                    <EmailMessageObj:Attachments>
                                        <EmailMessageObj:File
                                            object_reference="cybox:
                                            guid-49d31c13-8d7b-4528-
                                            b8d6-ce8ed0d43ad7"/>
                                    </EmailMessageObj:Attachments>
                                </cybox:Properties>
                                <cybox:Association_Type xsi:type="
                                    cyboxVocabs:
                                    ActionObjectAssociationTypeVocab
                                    -1.0">Returned</cybox:
```

```
                                Association_Type>
                            </cybox:Associated_Object>
                        </cybox:Associated_Objects>
                    </cybox:Action>
                </cybox:Actions>
            </cybox:Event>
        </cybox:Observable>
        <cybox:Observable id="example:Obervable-35f04c28-5fd2-4d72-8
            aae-2ad04ee1811f">
            <!-- Open Iran-Oil corrupted .doc file -->
            <cybox:Event>
                <cybox:Type xsi:type="cyboxVocabs:EventTypeVocab
                    -1.0.1">File Ops (CRUD)</cybox:Type>
                <cybox:Description>Open Iran-Oil corrupted .doc file
                    .</cybox:Description>
                <cybox:Actions>
                    <cybox:Action>
                        <cybox:Type xsi:type="cyboxVocabs:
                            ActionTypeVocab-1.0">Open</cybox:Type>
                        <cybox:Associated_Objects>
                            <cybox:Associated_Object id="example:
                                Object-49d31c13-8d7b-4528-b8d6-
                                ce8ed0d43ad7">
                                <cybox:Description>
                                    The word document contains flash
                                        , which downloads a
                                    corrupted mp4 file. The mp4 file
                                        itself is not anything
                                    special
                                    but an 0C filled (22kb) mp4 file
                                        with a valid mp4
                                    header.
                                </cybox:Description>
                                <cybox:Properties xsi:type="FileObj:
                                    FileObjectType">
                                    <FileObj:File_Name>Iran's Oil
                                        and Nuclear Situation.doc</
                                        FileObj:File_Name>
                                    <FileObj:Size_In_Bytes>106604</
                                        FileObj:Size_In_Bytes>
                                    <FileObj:Hashes>
                                        <cyboxCommon:Hash>
                                            <cyboxCommon:Type>MD5</
                                                cyboxCommon:Type>
                                            <cyboxCommon:
                                                Simple_Hash_Value
                                                condition="Equals">
                                                E92A4FC283EB2802AD6D0E24C7FCC857
                                                </cyboxCommon:
```

```
                                                    Simple_Hash_Value>
                                            </cyboxCommon:Hash>
                                        </FileObj:Hashes>
                                    </cybox:Properties>
                                    <cybox:Association_Type xsi:type="
                                        cyboxVocabs:
                                        ActionObjectAssociationTypeVocab
                                        -1.0">Affected</cybox:
                                        Association_Type>
                                </cybox:Associated_Object>
                            </cybox:Associated_Objects>
                        </cybox:Action>
                    </cybox:Actions>
                </cybox:Event>
            </cybox:Observable>
            <cybox:Observable id="example:Observable-f005fbc6-7427-43ea
                -8e1e-9a341836f76b">
                <!-- Download Iran-Oil invalid .mp4 downloader file -->
                <cybox:Event>
                    <cybox:Type xsi:type="cyboxVocabs:EventTypeVocab
                        -1.0.1">File Ops (CRUD)</cybox:Type>
                    <cybox:Description>Download Iran-Oil invalid .mp4
                        downloader file.</cybox:Description>
                    <cybox:Actions>
                        <cybox:Action>
                            <cybox:Type xsi:type="cyboxVocabs:
                                ActionTypeVocab-1.0">Download</cybox:Type
                                >
                            <cybox:Associated_Objects>
                                <cybox:Associated_Object idref="example:
                                    Object-49d31c13-8d7b-4528-b8d6-
                                    ce8ed0d43ad7">
                                    <cybox:Association_Type xsi:type="
                                        cyboxVocabs:
                                        ActionObjectAssociationTypeVocab
                                        -1.0">Initiating</cybox:
                                        Association_Type>
                                </cybox:Associated_Object>
                                <cybox:Associated_Object id="example:
                                    Object-8b463e0d-cc16-4036-950e-5
                                    eeb09bc51aa">
                                    <!-- Iran-Oil invalid .mp4
                                        downloader file -->
                                    <cybox:Description>
                                        This mp4 file causes memory
                                            corruption and code
                                        execution via heap-spraying code
                                            injection.
                                    </cybox:Description>
```

```
                <cybox:Properties xsi:type="FileObj:
                    FileObjectType">
                    <FileObj:File_Name>test.mp4</
                        FileObj:File_Name>
                    <FileObj:Size_In_Bytes>22384</
                        FileObj:Size_In_Bytes>
                    <FileObj:Hashes>
                        <cyboxCommon:Hash>
                            <cyboxCommon:Type>MD5</
                                cyboxCommon:Type>
                            <cyboxCommon:
                                Simple_Hash_Value
                                condition="Equals
                                ">8933598
                                C8B1FA5E493497B11C48DA4F2
                                </cyboxCommon:
                                Simple_Hash_Value>
                        </cyboxCommon:Hash>
                    </FileObj:Hashes>
                </cybox:Properties>
                <cybox:Related_Objects>
                    <cybox:Related_Object idref="
                        example:Object-49d31c13-8d7b
                        -4528-b8d6-ce8ed0d43ad7">
                        <cybox:Relationship xsi:type
                            ="cyboxVocabs:
                            ObjectRelationshipVocab
                            -1.0">Downloaded_By</
                            cybox:Relationship>
                    </cybox:Related_Object>
                    <cybox:Related_Object idref="
                        example:Object-61041b8b-0c15
                        -48a0-ac5f-b49488788010">
                        <cybox:Relationship xsi:type
                            ="cyboxVocabs:
                            ObjectRelationshipVocab
                            -1.0">Downloaded_From</
                            cybox:Relationship>
                    </cybox:Related_Object>
                </cybox:Related_Objects>
                <cybox:Association_Type xsi:type="
                    cyboxVocabs:
                    ActionObjectAssociationTypeVocab
                    -1.0">Affected</cybox:
                    Association_Type>
            </cybox:Associated_Object>
            <cybox:Associated_Object id="example:
                Object-61041b8b-0c15-48a0-ac5f-
                b49488788010">
```

```
                        <!-- URL from which malicious .mp4
                            file was downloaded-->
                        <cybox:Properties xsi:type="URIObj:
                            URIObjectType" type="URL">
                            <URIObj:Value condition="Equals
                                ">http://208.115.230.76/test.
                                mp4</URIObj:Value>
                        </cybox:Properties>
                        <cybox:Association_Type xsi:type="
                            cyboxVocabs:
                            ActionObjectAssociationTypeVocab
                            -1.0">Utilized</cybox:
                            Association_Type>
                    </cybox:Associated_Object>
                </cybox:Associated_Objects>
            </cybox:Action>
        </cybox:Actions>
    </cybox:Event>
</cybox:Observable>
<cybox:Observable id="example:Observable-210f18f3-3874-4f9a
    -861d-71b328be90c6">
    <!-- Create Iran-Oil.exe Trojan file-->
    <cybox:Event>
        <cybox:Type xsi:type="cyboxVocabs:EventTypeVocab
            -1.0.1">File Ops (CRUD)</cybox:Type>
        <cybox:Description>Create Iran-Oil.exe Trojan file
            .</cybox:Description>
        <cybox:Actions>
            <cybox:Action>
                <cybox:Type xsi:type="cyboxVocabs:
                    ActionTypeVocab-1.0">Create</cybox:Type>
                <cybox:Associated_Objects>
                    <cybox:Associated_Object idref="example:
                        Object-8b463e0d-cc16-4036-950e-5
                        eeb09bc51aa">
                        <cybox:Association_Type xsi:type="
                            cyboxVocabs:
                            ActionObjectAssociationTypeVocab
                            -1.0">Initiating</cybox:
                            Association_Type>
                    </cybox:Associated_Object>
                    <cybox:Associated_Object id="example:
                        Object-b7e0bc39-f519-4878-8fb0
                        -5902554efe1c">
                        <cybox:Description>
                            The file (us.exe MD5:
                                FD1BE09E499E8E380424B3835FC973A8

                            4861440 bytes) is created in the
```

```
        logged in user %Temp%
    directory. The size of the
        embedded file is 22.5 KB
        (23040
    bytes) and the size of the
        created us.exe is 4.63MB. It
        is an
    odd discrepancy until you look
        at the file and it looks like
         the
    code is repeated over and over -
         211 times. The file resource
    section indicates the file is
        meant to look like a java
        updater,
    which is always larger than 22.5
       KB and that would explain all
    this padding, which is done at
        the time when the file is
        being
    written to the disk.
</cybox:Description>
<cybox:Properties xsi:type="FileObj:
    FileObjectType">
    <FileObj:File_Name>us.exe</
        FileObj:File_Name>
    <FileObj:File_Path>%Temp%</
        FileObj:File_Path>
    <FileObj:Size_In_Bytes>4861440</
        FileObj:Size_In_Bytes>
    <FileObj:Hashes>
        <cyboxCommon:Hash>
            <cyboxCommon:Type>MD5</
                cyboxCommon:Type>
            <cyboxCommon:
                Simple_Hash_Value
                condition="Equals">
                FD1BE09E499E8E380424B3835FC973A8
                </cyboxCommon:
                Simple_Hash_Value>
        </cyboxCommon:Hash>
    </FileObj:Hashes>
</cybox:Properties>
<cybox:Related_Objects>
    <cybox:Related_Object idref="
        example:Object-8b463e0d-cc16
        -4036-950e-5eeb09bc51aa">
        <cybox:Relationship xsi:type
            ="cyboxVocabs:
```

```
            ObjectRelationshipVocab
            -1.0">Created_By</cybox:
            Relationship>
    </cybox:Related_Object>
    <!-- The trojan connects to the
        following set of URLs/IPs for
         C&C -->
    <cybox:Related_Object idref="
        example:Object-41b220d8-4c45
        -48de-9d08-30d661b2dc8e">
         <cybox:Relationship xsi:type
            ="cyboxVocabs:
            ObjectRelationshipVocab
            -1.0">Connected_To</cybox
            :Relationship>
    </cybox:Related_Object>
    <cybox:Related_Object idref="
        example:Object-61aa225b-90ef
        -415c-8bbd-a17282e457c9">
         <cybox:Relationship xsi:type
            ="cyboxVocabs:
            ObjectRelationshipVocab
            -1.0">Connected_To</cybox
            :Relationship>
    </cybox:Related_Object>
    <cybox:Related_Object idref="
        example:Object-568db11e-39ee
        -43d7-83d8-032bdec3801a">
         <cybox:Relationship xsi:type
            ="cyboxVocabs:
            ObjectRelationshipVocab
            -1.0">Connected_To</cybox
            :Relationship>
    </cybox:Related_Object>
    <cybox:Related_Object idref="
        example:Object-80bea4d1-0e70
        -4a03-a54f-e40373bf94f1">
         <cybox:Relationship xsi:type
            ="cyboxVocabs:
            ObjectRelationshipVocab
            -1.0">Connected_To</cybox
            :Relationship>
    </cybox:Related_Object>
    <cybox:Related_Object idref="
        example:Object-af7cb3b6-d70b
        -4b3b-b24f-7cfad739710f">
         <cybox:Relationship xsi:type
            ="cyboxVocabs:
            ObjectRelationshipVocab
```

```
                                        -1.0">Connected_To</cybox
                                        :Relationship>
                                    </cybox:Related_Object>
                                    <cybox:Related_Object idref="
                                        example:guid-5ceb9d54-24e2
                                        -4627-948d-6b92ac81962a">
                                        <cybox:Relationship xsi:type
                                            ="cyboxVocabs:
                                            ObjectRelationshipVocab
                                            -1.0">Connected_To</cybox
                                            :Relationship>
                                    </cybox:Related_Object>
                                </cybox:Related_Objects>
                                <cybox:Association_Type xsi:type="
                                    cyboxVocabs:
                                    ActionObjectAssociationTypeVocab
                                    -1.0">Affected</cybox:
                                    Association_Type>
                            </cybox:Associated_Object>
                        </cybox:Associated_Objects>
                    </cybox:Action>
                </cybox:Actions>
            </cybox:Event>
        </cybox:Observable>

        <cybox:Observable id="example:Observable-b650c988-aac7-45ff
            -967d-9f1e5fc66161">
            <!-- Execute Iran-Oil .exe Trojan file -->
            <cybox:Event>
                <cybox:Type xsi:type="cyboxVocabs:EventTypeVocab
                    -1.0.1">File Ops (CRUD)</cybox:Type>
                <cybox:Description>Execute Iran-Oil .exe Trojan file
                    .</cybox:Description>
                <cybox:Actions>
                    <cybox:Action>
                        <cybox:Type xsi:type="cyboxVocabs:
                            ActionTypeVocab-1.0">Execute</cybox:Type>
                        <cybox:Associated_Objects>
                            <cybox:Associated_Object idref="example:
                                Object-8b463e0d-cc16-4036-950e-5
                                eeb09bc51aa">
                                <cybox:Association_Type xsi:type="
                                    cyboxVocabs:
                                    ActionObjectAssociationTypeVocab
                                    -1.0">Initiating</cybox:
                                    Association_Type>
                            </cybox:Associated_Object>
                            <cybox:Associated_Object idref="example:
                                Object-b7e0bc39-f519-4878-8fb0
```

```
                                -5902554efe1c">
                                <cybox:Association_Type xsi:type="
                                    cyboxVocabs:
                                    ActionObjectAssociationTypeVocab
                                    -1.0">Affected</cybox:
                                    Association_Type>
                            </cybox:Associated_Object>
                        </cybox:Associated_Objects>
                    </cybox:Action>
                </cybox:Actions>
            </cybox:Event>
        </cybox:Observable>
        <cybox:Observable id="example:Observable-dee72b3e-82fb-4319-
            bfcc-007e3cf930e8">
            <!-- Iran-Oil core embedded .exe Trojan file -->
            <cybox:Object id="example:Object-bed1ff22-08e8-4e04-b7ac
                -908b5271176f">
                <cybox:Properties xsi:type="FileObj:FileObjectType">
                    <FileObj:File_Name>us-embedded.exe</FileObj:
                        File_Name>
                    <FileObj:Size_In_Bytes>23040</FileObj:
                        Size_In_Bytes>
                    <FileObj:Hashes>
                        <cyboxCommon:Hash>
                            <cyboxCommon:Type>MD5</cyboxCommon:Type>
                            <cyboxCommon:Simple_Hash_Value condition
                                ="Equals">
                                CB3DCDE34FD9FF0E19381D99B02F9692</
                                cyboxCommon:Simple_Hash_Value>
                        </cyboxCommon:Hash>
                    </FileObj:Hashes>
                </cybox:Properties>
                <cybox:Related_Objects>
                    <cybox:Related_Object idref="example:Object-
                        b7e0bc39-f519-4878-8fb0-5902554efe1c">
                        <cybox:Relationship xsi:type="cyboxVocabs:
                            ObjectRelationshipVocab-1.0">
                            Contained_Within</cybox:Relationship>
                    </cybox:Related_Object>
                </cybox:Related_Objects>
            </cybox:Object>
        </cybox:Observable>
        <cybox:Observable id="example:Observable-a24ff8bc-b534
            -4616-838b-8bbe260a8e8f">
            <!-- Trojan .exe file connects out to C&C URLs/IPs -->
            <cybox:Event>
                <cybox:Type xsi:type="cyboxVocabs:EventTypeVocab
                    -1.0.1">App Layer Traffic</cybox:Type>
                <cybox:Description>Trojan .exe file connects out to
```

```
    C2 URLs/IPs.</cybox:Description>
<cybox:Actions>
    <cybox:Action>
        <cybox:Type xsi:type="cyboxVocabs:
            ActionTypeVocab-1.0">Connect</cybox:Type>
        <cybox:Associated_Objects>
            <cybox:Associated_Object idref="example:
                Object-b7e0bc39-f519-4878-8fb0
                -5902554efe1c">
                <cybox:Association_Type xsi:type="
                    cyboxVocabs:
                    ActionObjectAssociationTypeVocab
                    -1.0">Initiating</cybox:
                    Association_Type>
            </cybox:Associated_Object>
            <cybox:Associated_Object idref="example:
                Object-41b220d8-4c45-48de-9d08-30
                d661b2dc8e">
                <cybox:Association_Type xsi:type="
                    cyboxVocabs:
                    ActionObjectAssociationTypeVocab
                    -1.0">Utilized</cybox:
                    Association_Type>
            </cybox:Associated_Object>
            <cybox:Associated_Object idref="example:
                Object-61aa225b-90ef-415c-8bbd-
                a17282e457c9">
                <cybox:Association_Type xsi:type="
                    cyboxVocabs:
                    ActionObjectAssociationTypeVocab
                    -1.0">Utilized</cybox:
                    Association_Type>
            </cybox:Associated_Object>
            <cybox:Associated_Object idref="example:
                Object-568db11e-39ee-43d7-83d8-032
                bdec3801a">
                <cybox:Association_Type xsi:type="
                    cyboxVocabs:
                    ActionObjectAssociationTypeVocab
                    -1.0">Utilized</cybox:
                    Association_Type>
            </cybox:Associated_Object>
            <cybox:Associated_Object idref="example:
                Object-80bea4d1-0e70-4a03-a54f-
                e40373bf94f1">
                <cybox:Association_Type xsi:type="
                    cyboxVocabs:
                    ActionObjectAssociationTypeVocab
                    -1.0">Utilized</cybox:
```

```
                                Association_Type>
                        </cybox:Associated_Object>
                        <cybox:Associated_Object idref="example:
                            Object-af7cb3b6-d70b-4b3b-b24f-7
                            cfad739710f">
                            <cybox:Association_Type xsi:type="
                                cyboxVocabs:
                                ActionObjectAssociationTypeVocab
                                -1.0">Utilized</cybox:
                                Association_Type>
                        </cybox:Associated_Object>
                        <cybox:Associated_Object idref="example:
                            Object-5ceb9d54-24e2-4627-948d-6
                            b92ac81962a">
                            <cybox:Association_Type xsi:type="
                                cyboxVocabs:
                                ActionObjectAssociationTypeVocab
                                -1.0">Utilized</cybox:
                                Association_Type>
                        </cybox:Associated_Object>
                    </cybox:Associated_Objects>
                </cybox:Action>
            </cybox:Actions>
        </cybox:Event>
    </cybox:Observable>
    <!-- The next six Observables represent the 3 different URL/
        IP pairs of C&C servers that the trojan communicates with
        -->
    <cybox:Observable id="example:Observable-066cef51-c886-432e
        -9a22-a17f57f3f3f2">
        <!-- One of three Command and Control URLs-->
        <cybox:Object id="example:Object-41b220d8-4c45-48de-9d08
            -30d661b2dc8e">
            <cybox:Properties xsi:type="URIObj:URIObjectType"
                type="URL">
                <URIObj:Value condition="Equals">www.documents.
                    myPicture.info</URIObj:Value>
            </cybox:Properties>
            <cybox:Related_Objects>
                <cybox:Related_Object idref="example:Object-61
                    aa225b-90ef-415c-8bbd-a17282e457c9">
                    <cybox:Relationship xsi:type="cyboxVocabs:
                        ObjectRelationshipVocab-1.0">Resolved_To
                        </cybox:Relationship>
                </cybox:Related_Object>
            </cybox:Related_Objects>
        </cybox:Object>
    </cybox:Observable>
    <cybox:Observable id="example:Observable-4e05804c-f552-44e1
```

```
    -9793-ff4bb7f88f9c">
    <!-- One of three Command and Control IPs-->
    <cybox:Object id="example:Object-61aa225b-90ef-415c-8bbd
        -a17282e457c9">
        <cybox:Properties xsi:type="AddrObj:
            AddressObjectType" category="ipv4-addr">
            <AddrObj:Address_Value condition="Equals
                ">199.192.156.134</AddrObj:Address_Value>
        </cybox:Properties>
    </cybox:Object>
</cybox:Observable>
<cybox:Observable id="example:Observable-75ce59ad-1f01-4eae
    -9ecc-0b22c4c24ce7">
    <!-- One of three Command and Control URLs-->
    <cybox:Object id="example:Object-568db11e-39ee-43d7-83d8
        -032bdec3801a">
        <cybox:Properties xsi:type="URIObj:URIObjectType"
            type="URL">
            <URIObj:Value condition="Equals">documents.
                myPicture.info</URIObj:Value>
        </cybox:Properties>
        <cybox:Related_Objects>
            <cybox:Related_Object idref="example:Object-80
                bea4d1-0e70-4a03-a54f-e40373bf94f1">
                <cybox:Relationship xsi:type="cyboxVocabs:
                    ObjectRelationshipVocab-1.0">Resolved_To
                    </cybox:Relationship>
            </cybox:Related_Object>
        </cybox:Related_Objects>
    </cybox:Object>

</cybox:Observable>
<cybox:Observable id="example:Observable-1ea53b14-8fe9-467b-
    a298-62d9684e797d">
    <!-- One of three Command and Control IPs-->
    <cybox:Object id="example:Object-80bea4d1-0e70-4a03-a54f
        -e40373bf94f1">
        <cybox:Properties xsi:type="AddrObj:
            AddressObjectType" category="ipv4-addr">
            <AddrObj:Address_Value condition="Equals
                ">199.192.156.134</AddrObj:Address_Value>
        </cybox:Properties>
    </cybox:Object>
</cybox:Observable>
<cybox:Observable id="example:Observable-f6c8ee75-ee7e-4490-
    bd5d-0661d0db7264">
    <!-- One of three Command and Control URLs-->
    <cybox:Object id="example:Object-af7cb3b6-d70b-4b3b-b24f
        -7cfad739710f">
```

```
        <cybox:Properties xsi:type="URIObj:URIObjectType"
            type="URL">
            <URIObj:Value condition="Equals">ftp.documents.
                myPicture.info</URIObj:Value>
        </cybox:Properties>
        <cybox:Related_Objects>
            <cybox:Related_Object idref="example:Object-5
                ceb9d54-24e2-4627-948d-6b92ac81962a">
                <cybox:Relationship xsi:type="cyboxVocabs:
                    ObjectRelationshipVocab-1.0">Resolved_To
                    </cybox:Relationship>
            </cybox:Related_Object>
        </cybox:Related_Objects>
    </cybox:Object>
</cybox:Observable>


<cybox:Observable id="example:Observable-c78c0a83-6d14-45f8
    -827f-f758f0cd11ea">
    <!-- One of three Command and Control IPs-->
    <cybox:Object id="example:Object-5ceb9d54-24e2-4627-948d
        -6b92ac81962a">
        <cybox:Properties xsi:type="AddrObj:
            AddressObjectType" category="ipv4-addr">
            <AddrObj:Address_Value condition="Equals
                ">199.192.156.134</AddrObj:Address_Value>
        </cybox:Properties>
    </cybox:Object>
</cybox:Observable>


<cybox:Observable id="example:Observable-47d6a950-884d-46b5
    -9938-ac5555065a81">
    <!-- This composed observable defines a pattern that is
        true if the receive email event occurs AND the create
         malicious .doc file event occurs AND the download
        the downloader .mp4 file event occurs AND the create
        trojan .exe file event occurs AND the execute trojan
        .exe file event occurs AND the connect to all three
        of the C&C URLs/IPs event occurs-->
    <!-- This yields a very tight filter that will have very
         low false positives but could miss almost any
        variation of the attack elements-->
    <cybox:Observable_Composition operator="AND">
        <!-- Receive "Iran-Oil" attack campaign email
            message -->
        <cybox:Observable idref="example:Observable-1a937ec2
            -90ab-4e0e-a37c-db9b2e66a58e"/>
        <!-- Open Iran-Oil corrupted .doc file-->
```

```
        <cybox:Observable idref="example:Observable-35f04c28
            -5fd2-4d72-8aae-2ad04ee1811f"/>
        <!-- Download Iran-Oil invalid .mp4 downloader file
            -->
        <cybox:Observable idref="example:Observable-f005fbc6
            -7427-43ea-8e1e-9a341836f76b"/>
        <!-- Create Iran-Oil .exe Trojan file -->
        <cybox:Observable idref="example:Observable-210f18f3
            -3874-4f9a-861d-71b328be90c6"/>
        <!-- Execute Iran-Oil .exe Trojan file -->
        <cybox:Observable idref="example:Observable-b650c988
            -aac7-45ff-967d-9f1e5fc66161"/>
        <!-- Trojan .exe file connects out to C&C URLs/IPs
            -->
        <cybox:Observable idref="example:Observable-a24ff8bc
            -b534-4616-838b-8bbe260a8e8f"/>
    </cybox:Observable_Composition>
</cybox:Observable>

<cybox:Observable id="example:Observable
    -80594430-7567-4402-88a4-05d556b21884">
    <!-- This composed observable defines a pattern that is
        true if the receive email event occurs OR the create
        malicious .doc file event occurs OR the download the
        downloader .mp4 file event occurs OR the create
        trojan .exe file event occurs OR the execute trojan .
        exe file event occurs OR the connect to all three of
        the C&C URLs/IPs event occurs -->
    <!-- This yields a very loose filter that could have
        false positives but could catch numerous potential
        variations of the attack elements -->
    <cybox:Observable_Composition operator="OR">
        <!-- Receive "Iran-Oil" attack campaign email
            message -->
        <cybox:Observable idref="example:Observable-1a937ec2
            -90ab-4e0e-a37c-db9b2e66a58e"/>
        <!-- Open Iran-Oil corrupted .doc file -->
        <cybox:Observable idref="example:Observable-35f04c28
            -5fd2-4d72-8aae-2ad04ee1811f"/>
        <!-- Download Iran-Oil invalid .mp4 downloader file
            -->
        <cybox:Observable idref="example:guid-f005fbc6
            -7427-43ea-8e1e-9a341836f76b"/>
        <!-- Create Iran-Oil .exe Trojan file -->
        <cybox:Observable idref="example:Observable-210f18f3
            -3874-4f9a-861d-71b328be90c6"/>
        <!-- Execute Iran-Oil .exe Trojan file -->
        <cybox:Observable idref="example:Observable-b650c988
            -aac7-45ff-967d-9f1e5fc66161"/>
```

```
        <!-- Trojan .exe file connects out to C&C URLs/IPs
            -->
        <cybox:Observable idref="example:Observable-a24ff8bc
            -b534-4616-838b-8bbe260a8e8f"/>
    </cybox:Observable_Composition>
</cybox:Observable>

<cybox:Observable id="example:Observable-7d932074-fded
    -4056-870e-dd51980501d4">
    <!-- This composed observable defines a pattern that is
        true if (the receive email event occurs AND the
        create malicious .doc file event occurs) OR (the
        download the downloader .mp4 file event occurs AND
        the create trojan .exe file event occurs AND the
        execute trojan .exe file event occurs) OR the connect
         to all three of the C&C URLs/IPs event occurs-->
    <cybox:Observable_Composition operator="OR">
        <cybox:Observable>
            <cybox:Observable_Composition operator="AND">
                <!-- Receive "Iran-Oil" attack campaign
                    email message -->
                <cybox:Observable idref="example:Observable
                    -1a937ec2-90ab-4e0e-a37c-db9b2e66a58e"/>
                <!-- Open Iran-Oil corrupted .doc file-->
                <cybox:Observable idref="example:Observable
                    -35f04c28-5fd2-4d72-8aae-2ad04ee1811f"/>
            </cybox:Observable_Composition>
        </cybox:Observable>
        <cybox:Observable>
            <cybox:Observable_Composition operator="AND">
                <!-- Download Iran-Oil invalid .mp4
                    downloader file-->
                <cybox:Observable idref="example:Observable-
                    f005fbc6-7427-43ea-8e1e-9a341836f76b"/>
                <!-- Create Iran-Oil .exe Trojan file-->
                <cybox:Observable idref="example:Observable
                    -210f18f3-3874-4f9a-861d-71b328be90c6"/>
                <!-- Execute Iran-Oil .exe Trojan file-->
                <cybox:Observable idref="example:Observable-
                    b650c988-aac7-45ff-967d-9f1e5fc66161"/>
            </cybox:Observable_Composition>
        </cybox:Observable>
        <!-- Trojan .exe file connects out to C&C URLs/IPs
            -->
        <cybox:Observable idref="example:Observable-a24ff8bc
            -b534-4616-838b-8bbe260a8e8f"/>
    </cybox:Observable_Composition>
</cybox:Observable>
```

```
    <!-- CybOX enables a wide myriad of other potential
        observable pattern variations at the logical composition
        level or utilizing patterns at the Object attribute level
         including Regex all of which allow the user to define an
         almost infinitely variable set of patterns and filters
        -->
</cybox:Observables>

<cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance" xmlns:cybox="http://cybox.mitre.org/cybox-2" xmlns:
    cyboxCommon="http://cybox.mitre.org/common-2" xmlns:
    AddressObj="http://cybox.mitre.org/objects#AddressObject-2"
    xmlns:PortObj="http://cybox.mitre.org/objects#PortObject-2"
    xmlns:SocketAddressObj="http://cybox.mitre.org/objects#
    SocketAddressObject-1" xmlns:NetworkConnectionObj="http://
    cybox.mitre.org/objects#NetworkConnectionObject-2" xmlns:
    HTTPSessionObj="http://cybox.mitre.org/objects#
    HTTPSessionObject-2" xmlns:example="http://example.com/" xsi:
    schemaLocation="http://cybox.mitre.org/cybox-2 http://cybox.
    mitre.org/XMLSchema/core/2.1/cybox_core.xsd      http://cybox.
    mitre.org/objects#NetworkConnectionObject-2 http://cybox.
    mitre.org/XMLSchema/objects/Network_Connection/2.1/
    Network_Connection_Object.xsd      " cybox_major_version="2"
    cybox_minor_version="1" cybox_update_version="0">
    <cybox:Observable id="example:Observable-1b427720-98d7-4735-
        b125-754c7e08f285">
        <cybox:Description>
            This Observable specifies an example instance of a
                Network Connection Object with an HTTP Session.
        </cybox:Description>
        <cybox:Object id="example:Object-d1fdd983-530b-489f-9ab8
            -ed3cb5212c35">
            <cybox:Properties xsi:type="NetworkConnectionObj:
                NetworkConnectionObjectType">
                <NetworkConnectionObj:Layer3_Protocol datatype="
                    string">IPv4</NetworkConnectionObj:
                    Layer3_Protocol>
                <NetworkConnectionObj:Layer4_Protocol datatype="
                    string">TCP</NetworkConnectionObj:
                    Layer4_Protocol>
                <NetworkConnectionObj:Layer7_Protocol datatype="
                    string">HTTP</NetworkConnectionObj:
                    Layer7_Protocol>
                <NetworkConnectionObj:Source_Socket_Address>
                    <SocketAddressObj:IP_Address>
                        <AddressObj:Address_Value>192.168.1.15</
                            AddressObj:Address_Value>
                    </SocketAddressObj:IP_Address>
                    <SocketAddressObj:Port>
```

```
            <PortObj:Port_Value>5525</PortObj:
                Port_Value>
        </SocketAddressObj:Port>
    </NetworkConnectionObj:Source_Socket_Address>
    <NetworkConnectionObj:Destination_Socket_Address
        >
        <SocketAddressObj:IP_Address>
            <AddressObj:Address_Value
                >198.49.123.10</AddressObj:
                Address_Value>
        </SocketAddressObj:IP_Address>
        <SocketAddressObj:Port>
            <PortObj:Port_Value>80</PortObj:
                Port_Value>
        </SocketAddressObj:Port>
    </NetworkConnectionObj:
        Destination_Socket_Address>
    <NetworkConnectionObj:Layer7_Connections>
        <NetworkConnectionObj:HTTP_Session>
            <HTTPSessionObj:HTTP_Request_Response>
                <HTTPSessionObj:HTTP_Client_Request>
                    <HTTPSessionObj:
                        HTTP_Request_Line>
                        <HTTPSessionObj:HTTP_Method
                            datatype="string">GET</
                            HTTPSessionObj:
                            HTTP_Method>
                        <HTTPSessionObj:Version>HTTP
                            /1.1</HTTPSessionObj:
                            Version>
                    </HTTPSessionObj:
                        HTTP_Request_Line>
                    <HTTPSessionObj:
                        HTTP_Request_Header>
                        <HTTPSessionObj:
                            Parsed_Header>
                            <HTTPSessionObj:
                                Accept_Encoding>gzip
                                </HTTPSessionObj:
                                Accept_Encoding>
                            <HTTPSessionObj:
                                Connection>close</
                                HTTPSessionObj:
                                Connection>
                        </HTTPSessionObj:
                            Parsed_Header>
                    </HTTPSessionObj:
                        HTTP_Request_Header>
                </HTTPSessionObj:HTTP_Client_Request
```

```
                                        >
                                    <HTTPSessionObj:HTTP_Server_Response
                                        >
                                        <HTTPSessionObj:HTTP_Status_Line
                                            >
                                            <HTTPSessionObj:Version>HTTP
                                                /1.1</HTTPSessionObj:
                                                Version>
                                            <HTTPSessionObj:Status_Code
                                                >200</HTTPSessionObj:
                                                Status_Code>
                                            <HTTPSessionObj:
                                                Reason_Phrase>OK</
                                                HTTPSessionObj:
                                                Reason_Phrase>
                                        </HTTPSessionObj:
                                            HTTP_Status_Line>
                                        <HTTPSessionObj:
                                            HTTP_Response_Header>
                                            <HTTPSessionObj:
                                                Parsed_Header>
                                                <HTTPSessionObj:Server>
                                                    Apache</
                                                    HTTPSessionObj:Server
                                                    >
                                                <HTTPSessionObj:
                                                    Transfer_Encoding>
                                                    chunked</
                                                    HTTPSessionObj:
                                                    Transfer_Encoding>
                                            </HTTPSessionObj:
                                                Parsed_Header>
                                        </HTTPSessionObj:
                                            HTTP_Response_Header>
                                    </HTTPSessionObj:
                                        HTTP_Server_Response>
                                </HTTPSessionObj:HTTP_Request_Response>
                            </NetworkConnectionObj:HTTP_Session>
                        </NetworkConnectionObj:Layer7_Connections>
                    </cybox:Properties>
                </cybox:Object>
            </cybox:Observable>
        </cybox:Observables>

        <cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-
            instance" xmlns:cybox="http://cybox.mitre.org/cybox-2" xmlns:
            cyboxCommon="http://cybox.mitre.org/common-2" xmlns:
            AddressObj="http://cybox.mitre.org/objects#AddressObject-2"
            xmlns:PortObj="http://cybox.mitre.org/objects#PortObject-2"
```

```
xmlns:SocketAddressObj="http://cybox.mitre.org/objects#
SocketAddressObject-1" xmlns:NetworkConnectionObj="http://
cybox.mitre.org/objects#NetworkConnectionObject-2" xmlns:
HTTPSessionObj="http://cybox.mitre.org/objects#
HTTPSessionObject-2" xmlns:example="http://example.com/" xsi:
schemaLocation="http://cybox.mitre.org/cybox-2 http://cybox.
mitre.org/XMLSchema/core/2.1/cybox_core.xsd      http://cybox.
mitre.org/objects#NetworkConnectionObject-2 http://cybox.
mitre.org/XMLSchema/objects/Network_Connection/2.1/
Network_Connection_Object.xsd      " cybox_major_version="2"
cybox_minor_version="1" cybox_update_version="0">
 <cybox:Observable id="example:Observable-1b427720-98d7-4735-
     b125-754c7e08f285">
      <cybox:Description>
           This Observable specifies an example pattern written
                against a Network Connection Object
           with an HTTP Session, specifically the Network
               Protocols, Destination Socket IP Address and Port
               ,
           and HTTP Request Method and Value.
      </cybox:Description>
      <cybox:Object id="example:Object-d1fdd983-530b-489f-9ab8
          -ed3cb5212c35">
           <cybox:Properties xsi:type="NetworkConnectionObj:
               NetworkConnectionObjectType">
                <NetworkConnectionObj:Layer3_Protocol datatype="
                    string" condition="Equals">IPv4</
                    NetworkConnectionObj:Layer3_Protocol>
                <NetworkConnectionObj:Layer4_Protocol datatype="
                    string" condition="Equals">TCP</
                    NetworkConnectionObj:Layer4_Protocol>
                <NetworkConnectionObj:Layer7_Protocol datatype="
                    string" condition="Equals">HTTP</
                    NetworkConnectionObj:Layer7_Protocol>
                <NetworkConnectionObj:Destination_Socket_Address
                    >
                     <SocketAddressObj:IP_Address>
                          <AddressObj:Address_Value datatype="
                              string" condition="StartsWith
                              ">198.49</AddressObj:Address_Value>
                     </SocketAddressObj:IP_Address>
                     <SocketAddressObj:Port>
                          <PortObj:Port_Value condition="Equals
                              ">80</PortObj:Port_Value>
                     </SocketAddressObj:Port>
                </NetworkConnectionObj:
                    Destination_Socket_Address>
                <NetworkConnectionObj:Layer7_Connections>
                     <NetworkConnectionObj:HTTP_Session>
```

```
                                <HTTPSessionObj:HTTP_Request_Response>
                                    <HTTPSessionObj:HTTP_Client_Request>
                                        <HTTPSessionObj:
                                            HTTP_Request_Line>
                                            <HTTPSessionObj:HTTP_Method
                                                datatype="string"
                                                condition="Equals">GET</
                                                HTTPSessionObj:
                                                HTTP_Method>
                                            <HTTPSessionObj:Value
                                                condition="Contains">.asp
                                                </HTTPSessionObj:Value>
                                        </HTTPSessionObj:
                                            HTTP_Request_Line>
                                    </HTTPSessionObj:HTTP_Client_Request
                                        >
                                </HTTPSessionObj:HTTP_Request_Response>
                            </NetworkConnectionObj:HTTP_Session>
                        </NetworkConnectionObj:Layer7_Connections>
                    </cybox:Properties>
                </cybox:Object>
            </cybox:Observable>
        </cybox:Observables>

        <cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-
            instance" xmlns:cybox="http://cybox.mitre.org/cybox-2" xmlns:
            cyboxCommon="http://cybox.mitre.org/common-2" xmlns:
            AddressObj="http://cybox.mitre.org/objects#AddressObject-2"
            xmlns:PortObj="http://cybox.mitre.org/objects#PortObject-2"
            xmlns:SocketAddressObj="http://cybox.mitre.org/objects#
            SocketAddressObject-1" xmlns:NetworkConnectionObj="http://
            cybox.mitre.org/objects#NetworkConnectionObject-2" xmlns:
            example="http://example.com/" xsi:schemaLocation="http://
            cybox.mitre.org/cybox-2 http://cybox.mitre.org/XMLSchema/core
            /2.1/cybox_core.xsd http://cybox.mitre.org/objects#
            NetworkConnectionObject-2 http://cybox.mitre.org/XMLSchema/
            objects/Network_Connection/2.1/Network_Connection_Object.xsd
                " cybox_major_version="2" cybox_minor_version="1"
            cybox_update_version="0">
            <cybox:Observable id="example:Observable-1aec7752-29e1
                -4018-806c-7a9a21ddb20e">
                <cybox:Description>
                    This Observable specifies an example instance of a
                        Network Connection Object.
                </cybox:Description>
                <cybox:Object id="example:Object-54400c36-5038-478b-bffe
                    -808c40b2f04e">
                    <cybox:Properties xsi:type="NetworkConnectionObj:
                        NetworkConnectionObjectType">
```

```
                    <NetworkConnectionObj:Layer3_Protocol datatype="
                        string">IPv4</NetworkConnectionObj:
                        Layer3_Protocol>
                    <NetworkConnectionObj:Layer4_Protocol datatype="
                        string">TCP</NetworkConnectionObj:
                        Layer4_Protocol>
                    <NetworkConnectionObj:Source_Socket_Address>
                        <SocketAddressObj:IP_Address>
                            <AddressObj:Address_Value>192.168.1.15</
                                AddressObj:Address_Value>
                        </SocketAddressObj:IP_Address>
                        <SocketAddressObj:Port>
                            <PortObj:Port_Value>5525</PortObj:
                                Port_Value>
                        </SocketAddressObj:Port>
                    </NetworkConnectionObj:Source_Socket_Address>
                    <NetworkConnectionObj:Destination_Socket_Address
                        >
                        <SocketAddressObj:IP_Address>
                            <AddressObj:Address_Value
                                >198.49.123.10</AddressObj:
                                Address_Value>
                        </SocketAddressObj:IP_Address>
                        <SocketAddressObj:Port>
                            <PortObj:Port_Value>80</PortObj:
                                Port_Value>
                        </SocketAddressObj:Port>
                    </NetworkConnectionObj:
                        Destination_Socket_Address>
                </cybox:Properties>
            </cybox:Object>
        </cybox:Observable>
    </cybox:Observables>

    <cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-
        instance" xmlns:cybox="http://cybox.mitre.org/cybox-2" xmlns:
        cyboxCommon="http://cybox.mitre.org/common-2" xmlns:
        AddressObj="http://cybox.mitre.org/objects#AddressObject-2"
        xmlns:PortObj="http://cybox.mitre.org/objects#PortObject-2"
        xmlns:SocketAddressObj="http://cybox.mitre.org/objects#
        SocketAddressObject-1" xmlns:NetworkConnectionObj="http://
        cybox.mitre.org/objects#NetworkConnectionObject-2" xmlns:
        example="http://example.com/" xsi:schemaLocation="http://
        cybox.mitre.org/cybox-2 http://cybox.mitre.org/XMLSchema/core
        /2.1/cybox_core.xsd http://cybox.mitre.org/objects#
        NetworkConnectionObject-2 http://cybox.mitre.org/XMLSchema/
        objects/Network_Connection/2.1/Network_Connection_Object.xsd
            " cybox_major_version="2" cybox_minor_version="1"
        cybox_update_version="0">
```

```
    <cybox:Observable id="example:Observable-1aec7752-29e1
        -4018-806c-7a9a21ddb20e">
        <cybox:Description>
            This Observable specifies an example pattern written
                against a Network Connection Object,
            specifically the Layer 3 and 4 Protocols and
                Destination Socket IP Address and Port.
        </cybox:Description>
        <cybox:Object id="example:Object-54400c36-5038-478b-bffe
            -808c40b2f04e">
            <cybox:Properties xsi:type="NetworkConnectionObj:
                NetworkConnectionObjectType">
                <NetworkConnectionObj:Layer3_Protocol datatype="
                    string" condition="Equals">IPv4</
                    NetworkConnectionObj:Layer3_Protocol>
                <NetworkConnectionObj:Layer4_Protocol datatype="
                    string" condition="Equals">TCP</
                    NetworkConnectionObj:Layer4_Protocol>
                <NetworkConnectionObj:Destination_Socket_Address
                    >
                    <SocketAddressObj:IP_Address>
                        <AddressObj:Address_Value datatype="
                            string" condition="StartsWith
                            ">198.49</AddressObj:Address_Value>
                    </SocketAddressObj:IP_Address>
                    <SocketAddressObj:Port>
                        <PortObj:Port_Value condition="Equals
                            ">80</PortObj:Port_Value>
                    </SocketAddressObj:Port>
                </NetworkConnectionObj:
                    Destination_Socket_Address>
            </cybox:Properties>
        </cybox:Object>
    </cybox:Observable>
</cybox:Observables>

<cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance" xmlns:cybox="http://cybox.mitre.org/cybox-2" xmlns:
    cyboxCommon="http://cybox.mitre.org/common-2" xmlns:FileObj="
    http://cybox.mitre.org/objects#FileObject-2" xmlns:PDFFileObj
    ="http://cybox.mitre.org/objects#PDFFileObject-1" xmlns:
    example="http://example.com/" xsi:schemaLocation="http://
    cybox.mitre.org/cybox-2 http://cybox.mitre.org/XMLSchema/core
    /2.1/cybox_core.xsd      http://cybox.mitre.org/objects#
    PDFFileObject-1 http://cybox.mitre.org/XMLSchema/objects/
    PDF_File/1.1/PDF_File_Object.xsd      " cybox_major_version
    ="2" cybox_minor_version="1" cybox_update_version="0">
    <cybox:Observable id="example:Observable-9c132c3c-fc65-41c9-
        a0f9-ea146dc61cd0">
```

```
<cybox:Description>
    This Observable specifies an example instance of a
       PDF File Object.
</cybox:Description>
<cybox:Object id="example:Object-3e6f7315-9591-4f84-ba51
    -7b8c44ce5d36">
    <cybox:Properties xsi:type="PDFFileObj:
        PDFFileObjectType">
        <FileObj:File_Name>test.pdf</FileObj:File_Name>
        <FileObj:Size_In_Bytes>25523</FileObj:
            Size_In_Bytes>
        <PDFFileObj:Version>1.4</PDFFileObj:Version>
        <PDFFileObj:Indirect_Objects>
            <PDFFileObj:Indirect_Object>
                <PDFFileObj:ID>
                    <PDFFileObj:Object_Number>1</
                        PDFFileObj:Object_Number>
                    <PDFFileObj:Generation_Number>0</
                        PDFFileObj:Generation_Number>
                </PDFFileObj:ID>
                <PDFFileObj:Contents>
                    <PDFFileObj:Non_Stream_Contents>

                        &lt;&lt; /Type /Catalog
                        /Outlines 2 0 R
                        &gt;&gt;

                    </PDFFileObj:Non_Stream_Contents>
                </PDFFileObj:Contents>
            </PDFFileObj:Indirect_Object>
            <PDFFileObj:Indirect_Object>
                <PDFFileObj:ID>
                    <PDFFileObj:Object_Number>2</
                        PDFFileObj:Object_Number>
                    <PDFFileObj:Generation_Number>0</
                        PDFFileObj:Generation_Number>
                </PDFFileObj:ID>
                <PDFFileObj:Contents>
                    <PDFFileObj:Non_Stream_Contents>

                        &lt;&lt; /Type Outlines
                        /Count 0
                        &gt;&gt;

                    </PDFFileObj:Non_Stream_Contents>
                </PDFFileObj:Contents>
            </PDFFileObj:Indirect_Object>
        </PDFFileObj:Indirect_Objects>
        <PDFFileObj:Cross_Reference_Tables>
```

```
<PDFFileObj:Cross_Reference_Table>
    <PDFFileObj:Subsections>
        <PDFFileObj:Subsection>
            <PDFFileObj:First_Object_Number
                >0</PDFFileObj:
                First_Object_Number>
            <PDFFileObj:Number_Of_Objects
                >3</PDFFileObj:
                Number_Of_Objects>
            <PDFFileObj:
                Cross_Reference_Entries>
                <PDFFileObj:
                    Cross_Reference_Entry>
                    <PDFFileObj:Byte_Offset
                        >0</PDFFileObj:
                        Byte_Offset>
                    <PDFFileObj:
                        Generation_Number
                        >65535</PDFFileObj:
                        Generation_Number>
                </PDFFileObj:
                    Cross_Reference_Entry>
                <PDFFileObj:
                    Cross_Reference_Entry>
                    <PDFFileObj:Byte_Offset
                        >9</PDFFileObj:
                        Byte_Offset>
                    <PDFFileObj:
                        Generation_Number>0</
                        PDFFileObj:
                        Generation_Number>
                </PDFFileObj:
                    Cross_Reference_Entry>
                <PDFFileObj:
                    Cross_Reference_Entry>
                    <PDFFileObj:Byte_Offset
                        >74</PDFFileObj:
                        Byte_Offset>
                    <PDFFileObj:
                        Generation_Number>0</
                        PDFFileObj:
                        Generation_Number>
                </PDFFileObj:
                    Cross_Reference_Entry>
            </PDFFileObj:
                Cross_Reference_Entries>
        </PDFFileObj:Subsection>
    </PDFFileObj:Subsections>
</PDFFileObj:Cross_Reference_Table>
```

```
                    </PDFFileObj:Cross_Reference_Tables>
                    <PDFFileObj:Trailers>
                        <PDFFileObj:Trailer>
                            <PDFFileObj:Size>3</PDFFileObj:Size>
                            <PDFFileObj:Root>
                                <PDFFileObj:Object_Number>1</
                                    PDFFileObj:Object_Number>
                                <PDFFileObj:Generation_Number>0</
                                    PDFFileObj:Generation_Number>
                            </PDFFileObj:Root>
                            <PDFFileObj:Last_Cross_Reference_Offset
                                >408</PDFFileObj:
                                Last_Cross_Reference_Offset>
                        </PDFFileObj:Trailer>
                    </PDFFileObj:Trailers>
                </cybox:Properties>
            </cybox:Object>
        </cybox:Observable>
</cybox:Observables>

<cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance" xmlns:cybox="http://cybox.mitre.org/cybox-2" xmlns:
    cyboxCommon="http://cybox.mitre.org/common-2" xmlns:
    PDFFileObj="http://cybox.mitre.org/objects#PDFFileObject-1"
    xmlns:example="http://example.com/" xsi:schemaLocation="http
    ://cybox.mitre.org/cybox-2 http://cybox.mitre.org/XMLSchema/
    core/2.1/cybox_core.xsd      http://cybox.mitre.org/objects#
    PDFFileObject-1 http://cybox.mitre.org/XMLSchema/objects/
    PDF_File/1.1/PDF_File_Object.xsd      " cybox_major_version
    ="2" cybox_minor_version="1" cybox_update_version="0">
    <cybox:Observable id="example:Observable-343961d3-8e62-4150-
        a701-4ade0afd1f69">
        <cybox:Description>
            This Observable specifies an example of a pattern
                against a PDF File Object, specifically
            the Version and JS_Count metadata entry.
        </cybox:Description>
        <cybox:Object id="example:Object-72d46b20-4048-427c-a077
            -d601c2874348">
            <cybox:Properties xsi:type="PDFFileObj:
                PDFFileObjectType">
                <PDFFileObj:Metadata>
                    <PDFFileObj:Keyword_Counts>
                        <PDFFileObj:JS_Count>
                            <PDFFileObj:Non_Obfuscated_Count
                                condition="GreaterThanOrEqual
                                ">3</PDFFileObj:
                                Non_Obfuscated_Count>
                        </PDFFileObj:JS_Count>
```

```
                    </PDFFileObj:Keyword_Counts>
                </PDFFileObj:Metadata>
                <PDFFileObj:Version condition="GreaterThan
                    ">1.2</PDFFileObj:Version>
            </cybox:Properties>
        </cybox:Object>
    </cybox:Observable>
</cybox:Observables>

<cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance" xmlns:cybox="http://cybox.mitre.org/cybox-2" xmlns:
    cyboxCommon="http://cybox.mitre.org/common-2" xmlns:AddrObj="
    http://cybox.mitre.org/objects#AddressObject-2" xmlns:URIObj
    ="http://cybox.mitre.org/objects#URIObject-2" xmlns:FileObj="
    http://cybox.mitre.org/objects#FileObject-2" xmlns:
    cyboxVocabs="http://cybox.mitre.org/default_vocabularies-2"
    xmlns:EmailMessageObj="http://cybox.mitre.org/objects#
    EmailMessageObject-2" xmlns:example="http://example.com/" xsi
    :schemaLocation="http://cybox.mitre.org/cybox-2 http://cybox.
    mitre.org/XMLSchema/core/2.1/cybox_core.xsd http://cybox.
    mitre.org/objects#FileObject-2 http://cybox.mitre.org/
    XMLSchema/objects/File/2.1/File_Object.xsd http://cybox.
    mitre.org/objects#EmailMessageObject-2 http://cybox.mitre.org
    /XMLSchema/objects/Email_Message/2.1/Email_Message_Object.xsd
        http://cybox.mitre.org/default_vocabularies-2 http://
    cybox.mitre.org/XMLSchema/default_vocabularies/2.1/
    cybox_default_vocabularies.xsd " cybox_major_version="2"
    cybox_minor_version="1" cybox_update_version="0">
    <cybox:Observable id="example:Obervable-298376a2-cf65
        -4778-9894-ed9a9bb5441d">
        <cybox:Object id="example:Object-f9769431-db6b-448f-b34e
            -72eb3c3e07d9">
            <cybox:Properties xsi:type="EmailMessageObj:
                EmailMessageObjectType">
                <EmailMessageObj:Header>
                    <EmailMessageObj:To>
                        <EmailMessageObj:Recipient category="e-
                            mail">
                            <AddrObj:Address_Value>
                                victim1@target.com</AddrObj:
                                Address_Value>
                        </EmailMessageObj:Recipient>
                        <EmailMessageObj:Recipient category="e-
                            mail">
                            <AddrObj:Address_Value>
                                victim2@target.com</AddrObj:
                                Address_Value>
                        </EmailMessageObj:Recipient>
                    </EmailMessageObj:To>
```

```
                    <EmailMessageObj:From category="e-mail">
                        <AddrObj:Address_Value>attacker@example.
                            com</AddrObj:Address_Value>
                    </EmailMessageObj:From>
                    <EmailMessageObj:Subject>New modifications
                        to the specification</EmailMessageObj:
                        Subject>
                </EmailMessageObj:Header>
            </cybox:Properties>
            <cybox:Related_Objects>
                <cybox:Related_Object idref="example:Object-8108
                    c0dc-bded-4b0c-b423-8b92ef1d6503">
                    <cybox:Relationship xsi:type="cyboxVocabs:
                        ObjectRelationshipVocab-1.0">
                        Received_From</cybox:Relationship>
                </cybox:Related_Object>
            </cybox:Related_Objects>
        </cybox:Object>
    </cybox:Observable>
    <cybox:Observable>
        <cybox:Object id="example:Object-8108c0dc-bded-4b0c-b423
            -8b92ef1d6503">
            <cybox:Properties xsi:type="AddrObj:
                AddressObjectType" category="ipv4-addr">
                <AddrObj:Address_Value>192.168.1.1</AddrObj:
                    Address_Value>
            </cybox:Properties>
        </cybox:Object>
    </cybox:Observable>
</cybox:Observables>

<cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance" xmlns:cybox="http://cybox.mitre.org/cybox-2" xmlns:
    cyboxCommon="http://cybox.mitre.org/common-2" xmlns:AddrObj="
    http://cybox.mitre.org/objects#AddressObject-2" xmlns:URIObj
    ="http://cybox.mitre.org/objects#URIObject-2" xmlns:FileObj="
    http://cybox.mitre.org/objects#FileObject-2" xmlns:
    cyboxVocabs="http://cybox.mitre.org/default_vocabularies-2"
    xmlns:EmailMessageObj="http://cybox.mitre.org/objects#
    EmailMessageObject-2" xmlns:example="http://example.com/" xsi
    :schemaLocation="http://cybox.mitre.org/cybox-2 http://cybox.
    mitre.org/XMLSchema/core/2.1/cybox_core.xsd http://cybox.
    mitre.org/objects#FileObject-2 http://cybox.mitre.org/
    XMLSchema/objects/File/2.1/File_Object.xsd http://cybox.
    mitre.org/objects#EmailMessageObject-2 http://cybox.mitre.org
    /XMLSchema/objects/Email_Message/2.1/Email_Message_Object.xsd
        http://cybox.mitre.org/default_vocabularies-2 http://
    cybox.mitre.org/XMLSchema/default_vocabularies/2.1/
    cybox_default_vocabularies.xsd " cybox_major_version="2"
```

```
    cybox_minor_version="1" cybox_update_version="0">
    <cybox:Observable id="example:Obervable-298376a2-cf65
        -4778-9894-ed9a95b5441d">
        <cybox:Object id="example:Object-f9769431-db6b-448f-b34e
            -72eb3c3e07d1">
            <cybox:Properties xsi:type="EmailMessageObj:
                EmailMessageObjectType">
                <EmailMessageObj:Header>
                    <EmailMessageObj:From category="e-mail">
                        <AddrObj:Address_Value condition="Equals
                            " apply_condition="ANY">
                            attacker@example.com##comma##
                            attacker1@example.com##comma##
                            attacker@bad.example.com</AddrObj:
                            Address_Value>
                    </EmailMessageObj:From>
                    <EmailMessageObj:Subject condition="Equals">
                        New modifications to the specification</
                        EmailMessageObj:Subject>
                </EmailMessageObj:Header>
            </cybox:Properties>
        </cybox:Object>
    </cybox:Observable>
</cybox:Observables>

<cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance" xmlns:cybox="http://cybox.mitre.org/cybox-2" xmlns:
    cyboxCommon="http://cybox.mitre.org/common-2" xmlns:FileObj="
    http://cybox.mitre.org/objects#FileObject-2" xmlns:
    cyboxVocabs="http://cybox.mitre.org/default_vocabularies-2"
    xmlns:example="http://example.com/" xsi:schemaLocation="http
    ://cybox.mitre.org/cybox-2 http://cybox.mitre.org/XMLSchema/
    core/2.1/cybox_core.xsd      http://cybox.mitre.org/objects#
    FileObject-2 http://cybox.mitre.org/XMLSchema/objects/File
    /2.1/File_Object.xsd      http://cybox.mitre.org/
    default_vocabularies-2 http://cybox.mitre.org/XMLSchema/
    default_vocabularies/2.1/cybox_default_vocabularies.xsd      "
     cybox_major_version="2" cybox_minor_version="1"
    cybox_update_version="0">
    <cybox:Observable id="example:Observable-58115a77-e24a-42b5-
        bb29-7bd56fa9655f">
        <cybox:Description>This observable specifies a specific
            file observation.</cybox:Description>
        <cybox:Object id="example:Object-17e97e7c-d3e6-4138-891b
            -291576dc5d41">
            <cybox:Properties xsi:type="FileObj:FileObjectType">
                <FileObj:File_Name>bad_file24.exe</FileObj:
                    File_Name>
                <FileObj:File_Path>AppData\Mozilla</FileObj:
```

```
                File_Path>
            <FileObj:File_Extension>.exe</FileObj:
                File_Extension>
            <FileObj:Size_In_Bytes>3282</FileObj:
                Size_In_Bytes>
            <FileObj:Hashes>
                                <cyboxCommon:Hash>
                                    <cyboxCommon:
                                        Type xsi:type
                                        ="cyboxVocabs
                                        :
                                        HashNameVocab
                                        -1.0">MD5</
                                        cyboxCommon:
                                        Type>
                                    <cyboxCommon:
                                        Simple_Hash_Value
                                        >
                                        a7a0390e99406f8975a1895860f55f2
                                        </cyboxCommon
                                        :
                                        Simple_Hash_Value
                                        >
                                </cyboxCommon:Hash>
                            </FileObj:Hashes>
        </cybox:Properties>
    </cybox:Object>
</cybox:Observable>
</cybox:Observables>

<cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance" xmlns:cybox="http://cybox.mitre.org/cybox-2" xmlns:
    cyboxCommon="http://cybox.mitre.org/common-2" xmlns:FileObj="
    http://cybox.mitre.org/objects#FileObject-2" xmlns:
    cyboxVocabs="http://cybox.mitre.org/default_vocabularies-2"
    xmlns:example="http://example.com/" xsi:schemaLocation="http
    ://cybox.mitre.org/cybox-2 http://cybox.mitre.org/XMLSchema/
    core/2.1/cybox_core.xsd http://cybox.mitre.org/objects#
    FileObject-2 http://cybox.mitre.org/XMLSchema/objects/File
    /2.1/File_Object.xsd http://cybox.mitre.org/
    default_vocabularies-2 http://cybox.mitre.org/XMLSchema/
    default_vocabularies/2.1/cybox_default_vocabularies.xsd "
     cybox_major_version="2" cybox_minor_version="1"
    cybox_update_version="0">
    <cybox:Observable id="example:Observable-58115a77-e24a-42b5-
        bb29-7bd66fa9655f">
        <cybox:Description>This observable specifies a specific
            file observation.</cybox:Description>
        <cybox:Object id="example:Object-17e97e7c-d3e6-4139-891b
```

```
            -291576dc5d41">
            <cybox:Properties xsi:type="FileObj:FileObjectType">
                <FileObj:File_Name condition="Contains">Paris</
                    FileObj:File_Name>
                <FileObj:File_Extension>.exe</FileObj:
                    File_Extension>
            </cybox:Properties>
        </cybox:Object>
    </cybox:Observable>
</cybox:Observables>

<cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance" xmlns:cybox="http://cybox.mitre.org/cybox-2" xmlns:
    cyboxCommon="http://cybox.mitre.org/common-2" xmlns:FileObj="
    http://cybox.mitre.org/objects#FileObject-2" xmlns:example="
    http://example.com/" xsi:schemaLocation="http://cybox.mitre.
    org/cybox-2 http://cybox.mitre.org/XMLSchema/core/2.1/
    cybox_core.xsd      http://cybox.mitre.org/objects#FileObject
    -2 http://cybox.mitre.org/XMLSchema/objects/File/2.1/
    File_Object.xsd      " cybox_major_version="2"
    cybox_minor_version="1" cybox_update_version="0">
    <cybox:Observable id="example:Observable-9769042a-294d-4f2c
        -963b-579702df0472">
        <cybox:Description>
            This observables specifies a pattern for a file with
                a file name that fits a certain pattern.
            The file name starts with 'bad_file', ends with '.
                exe', and has
            between two and five numbers in it.
        </cybox:Description>
        <cybox:Object id="example:Object-dae8802e-b0df-4989-9ac3
            -d816b153842b">
            <cybox:Properties xsi:type="FileObj:FileObjectType">
                <FileObj:File_Name pattern_type="Regex">bad_file
                    [0-9]{2,5}\.exe</FileObj:File_Name>
            </cybox:Properties>
        </cybox:Object>
    </cybox:Observable>
</cybox:Observables>

<cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance" xmlns:cybox="http://cybox.mitre.org/cybox-2" xmlns:
    cyboxCommon="http://cybox.mitre.org/common-2" xmlns:URIObject
    ="http://cybox.mitre.org/objects#URIObject-2" xmlns:example="
    http://example.com/" xsi:schemaLocation="http://cybox.mitre.
    org/cybox-2 http://cybox.mitre.org/XMLSchema/core/2.1/
    cybox_core.xsd      http://cybox.mitre.org/objects#URIObject-2
     http://cybox.mitre.org/XMLSchema/objects/URI/2.1/URI_Object.
    xsd      " cybox_major_version="2" cybox_minor_version="1"
    cybox_update_version="0">
```

```
    <cybox:Observable id="example:Observable-2d9af310-0d5a-4c44-
        bdd7-aea3d99f13b6">
        <cybox:Object id="example:Object-37be6630-b2df-4bf9
            -8750-3f45ca9e19cf">
            <cybox:Properties xsi:type="URIObject:URIObjectType"
                type="URL">
                <URIObject:Value>http://example.com/index1.html
                    </URIObject:Value>
            </cybox:Properties>
        </cybox:Object>
    </cybox:Observable>
</cybox:Observables>

<cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance" xmlns:cybox="http://cybox.mitre.org/cybox-2" xmlns:
    cyboxCommon="http://cybox.mitre.org/common-2" xmlns:URIObject
    ="http://cybox.mitre.org/objects#URIObject-2" xmlns:example="
    http://example.com/" xsi:schemaLocation="http://cybox.mitre.
    org/cybox-2 http://cybox.mitre.org/XMLSchema/core/2.1/
    cybox_core.xsd http://cybox.mitre.org/objects#URIObject-2
     http://cybox.mitre.org/XMLSchema/objects/URI/2.1/URI_Object.
    xsd " cybox_major_version="2" cybox_minor_version="1"
    cybox_update_version="0">
    <cybox:Observable id="example:Observable-1c9af310-0d5a-4c44-
        bdd7-aea3d99f13b6">
        <cybox:Object id="example:Object-26be6630-b2df-4bf9
            -8750-3f45ca9e19cf">
            <cybox:Properties xsi:type="URIObject:URIObjectType"
                type="URL">
                <URIObject:Value condition="Equals">http://
                    example.com/index1.html</URIObject:Value>
            </cybox:Properties>
        </cybox:Object>
    </cybox:Observable>
</cybox:Observables>

<cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance" xmlns:cybox="http://cybox.mitre.org/cybox-2" xmlns:
    cyboxCommon="http://cybox.mitre.org/common-2" xmlns:
    X509CertificateObj="http://cybox.mitre.org/objects#
    X509CertificateObject-2" xmlns:example="http://example.com/"
    xsi:schemaLocation="http://cybox.mitre.org/cybox-2 http://
    cybox.mitre.org/XMLSchema/core/2.1/cybox_core.xsd http://
    cybox.mitre.org/objects#X509CertificateObject-2 http://cybox.
    mitre.org/XMLSchema/objects/X509_Certificate/2.1/
    X509_Certificate_Object.xsd " cybox_major_version="2"
    cybox_minor_version="1" cybox_update_version="0">
    <cybox:Observable id="example:Observable-342df9c7-b309-4297-
        b3ce-ab3954ccb9db">
```

```
<cybox:Description>
    This Observable specifies an example instance of a
        X509 Certificate Object.
</cybox:Description>
<cybox:Object id="example:Object-a677503f-b20b-44d0-a50f
    -fc055bb0e255">
    <cybox:Properties xsi:type="X509CertificateObj:
        X509CertificateObjectType">
        <X509CertificateObj:Certificate>
            <X509CertificateObj:Version>1</
                X509CertificateObj:Version>
            <X509CertificateObj:Serial_Number>1234</
                X509CertificateObj:Serial_Number>
            <X509CertificateObj:Signature_Algorithm>
                md5WithRSAEncryption</X509CertificateObj:
                Signature_Algorithm>
            <X509CertificateObj:Issuer>C=US, ST=
                California, O=www.example.com, OU=new, CN
                =new</X509CertificateObj:Issuer>
            <X509CertificateObj:Validity>
                <X509CertificateObj:Not_Before
                    >2013-04-08T12:00:00</
                    X509CertificateObj:Not_Before>
                <X509CertificateObj:Not_After>2014-04-07
                    T23:59:59</X509CertificateObj:
                    Not_After>
            </X509CertificateObj:Validity>
            <X509CertificateObj:Subject>C=US, ST=
                Maryland, L=Baltimore, O=John Doe, OU=
                ExampleCorp, CN=www.example.com/
                emailAddress=doe@example.com</
                X509CertificateObj:Subject>
            <X509CertificateObj:Subject_Public_Key>
                <X509CertificateObj:Public_Key_Algorithm
                    >rsaEncryption</X509CertificateObj:
                    Public_Key_Algorithm>
                <X509CertificateObj:RSA_Public_Key>
                    <X509CertificateObj:Modulus>
                        00:b2:21:98:0a:c4:bc:62:b5:25:ac
                            :ea:b0:c8:bb:
                        33:35:19:d5:0c:64:b9:99:41:b7
                            :96:fc:f3:31:e1:
                        17:34:d0:8e:56:12:44:ad:22:eb:e8
                            :1c:9c:5b:66:
                        70:33:52:45:c9:ec:4f
                            :12:51:70:39:dc:53:85:17:
                        16:67:6e:ee:f4:d5:6f:d5:ca:b3
                            :47:5e:1b:0c:8d:
                        c5:cc:2b:6b:c1:90:d7:77:31:0d:bf
```

```
                                        :7a:c7:47:12:
                                    8f:a0:21:c7:4c:d0:16:33:00:c1:0f
                                        :d7:b8:80:e3:
                                    d2:d3:6b:c1:ea:9e:5c:5c:ea:7d
                                        :88:a1:10:bc:b8:
                                    e8:35:1c:9e:27:52:7e:22:4d
                                </X509CertificateObj:Modulus>
                                <X509CertificateObj:Exponent>65537</
                                    X509CertificateObj:Exponent>
                            </X509CertificateObj:RSA_Public_Key>
                        </X509CertificateObj:Subject_Public_Key>
                    </X509CertificateObj:Certificate>
                    <X509CertificateObj:Certificate_Signature>
                        <X509CertificateObj:Signature_Algorithm>
                            md5WithRSAEncryption</X509CertificateObj:
                            Signature_Algorithm>
                        <X509CertificateObj:Signature>
                            22:fd:12:5f:c5:af:cc:0a:ab:87:6d:fb:24:5
                                f:b6:59:5d:9d:
                            92:2e:4a:1b:8b:ac:7d:99:17:5d:cd:19:f6:
                                ad:ef:63:2f:92:
                            ad:2f:4b:cf:0a:52:90:bc:2c:0e:28:03:be:
                                ac:15:8e:3c:15:
                            1d:a2:40:03:f7:ef:6a:15:09:79:a9:46:ed:
                                b7:16:1b:41:72:
                            0d:19:ab:33:d3:9a:df:ab:97:50:65:f2:5e
                                :33:a1:ee:33:1d:
                            5d:ac:11:bb:63:33:cb:cc:6d:5d:01:85:b5:6
                                d:c8:f3:d9:f7:
                            8f:0e:fc:ba:1f:34:e9:96:6e:6c:cf:f2:ef:3
                                c:bf:1a:b5:22:
                            68:5d
                        </X509CertificateObj:Signature>
                    </X509CertificateObj:Certificate_Signature>
                </cybox:Properties>
            </cybox:Object>
        </cybox:Observable>
    </cybox:Observables>

    <cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-
        instance" xmlns:cybox="http://cybox.mitre.org/cybox-2" xmlns:
        cyboxCommon="http://cybox.mitre.org/common-2" xmlns:
        X509CertificateObj="http://cybox.mitre.org/objects#
        X509CertificateObject-2" xmlns:example="http://example.com/"
        xsi:schemaLocation="http://cybox.mitre.org/cybox-2 http://
        cybox.mitre.org/XMLSchema/core/2.1/cybox_core.xsd http://
        cybox.mitre.org/objects#X509CertificateObject-2 http://cybox.
        mitre.org/XMLSchema/objects/X509_Certificate/2.1/
        X509_Certificate_Object.xsd " cybox_major_version="2"
```

```
    cybox_minor_version="1" cybox_update_version="0">
    <cybox:Observable id="example:Observable-72b7b0b3-a95f-4bb8-
        b42b-f5f86e1370b9">
        <cybox:Description>
            This Observable specifies an example pattern against
                an X509 Certificate Object,
            testing both the certificate issuer and subject.
        </cybox:Description>
        <cybox:Object id="example:Object-551e35af-6f73-4182-ae42
            -8803b56781ed">
            <cybox:Properties xsi:type="X509CertificateObj:
                X509CertificateObjectType">
                <X509CertificateObj:Certificate>
                    <X509CertificateObj:Issuer condition="
                        Contains">www.example.com</
                        X509CertificateObj:Issuer>
                    <X509CertificateObj:Subject condition="
                        FitsPattern" pattern_type="Regex">(\w+\s)
                        *doe|doe@example\.com</X509CertificateObj
                        :Subject>
                </X509CertificateObj:Certificate>
            </cybox:Properties>
        </cybox:Object>
    </cybox:Observable>
</cybox:Observables>
```