Unión Internacional de Telecomunicaciones

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

X.1542

(09/2016)

SERIE X: REDES DE DATOS, COMUNICACIONES DE SISTEMAS ABIERTOS Y SEGURIDAD

Intercambio de información de ciberseguridad – Intercambio de estados/vulnerabilidad

Formato de intercambio de mensajes sobre información de sesión

Recomendación UIT-T X.1542

RECOMENDACIONES UIT-T DE LA SERIE X

REDES DE DATOS, COMUNICACIONES DE SISTEMAS ABIERTOS Y SEGURIDAD

REDES PÚBLICAS DE DATOS	X.1–X.199
INTERCONEXIÓN DE SISTEMAS ABIERTOS	X.200–X.299
INTERFUNCIONAMIENTO ENTRE REDES	X.300–X.399
SISTEMAS DE TRATAMIENTO DE MENSAJES	X.400–X.499
DIRECTORIO	X.500–X.599
GESTIÓN DE REDES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS Y ASPECTOS DE SISTEMAS	X.600–X.699
GESTIÓN DE INTERCONEXIÓN DE SISTEMAS ABIERTOS	X.700–X.799
SEGURIDAD	X.800–X.849
APLICACIONES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS	X.850–X.899
PROCESAMIENTO DISTRIBUIDO ABIERTO	X.900–X.999
SEGURIDAD DE LA INFORMACIÓN Y DE LAS REDES	
Aspectos generales de la seguridad	X.1000–X.1029
Seguridad de las redes	X.1030–X.1049
Gestión de la seguridad	X.1050–X.1069
Telebiometría	X.1080–X.1099
APLICACIONES Y SERVICIOS CON SEGURIDAD	
Seguridad en la multidifusión	X.1100–X.1109
Seguridad en la red residencial	X.1110–X.1119
Seguridad en las redes móviles	X.1120–X.1139
Seguridad en la web	X.1140–X.1149
Protocolos de seguridad	X.1150–X.1159
Seguridad en las comunicaciones punto a punto	X.1160–X.1169
Seguridad de la identidad en las redes	X.1170–X.1179
Seguridad en la TVIP	X.1180–X.1199
SEGURIDAD EN EL CIBERESPACIO	
Ciberseguridad	X.1200–X.1229
Lucha contra el correo basura	X.1230–X.1249
Gestión de identidades	X.1250–X.1279
APLICACIONES Y SERVICIOS CON SEGURIDAD	
Comunicaciones de emergencia	X.1300–X.1309
Seguridad en las redes de sensores ubicuos	X.1310–X.1339
Recomendaciones relacionadas con la PKI	
INTERCAMBIO DE INFORMACIÓN DE CIBERSEGURIDAD	
Aspectos generales de la ciberseguridad	X.1500–X.1519
Intercambio de estados/vulnerabilidad	X.1520–X.1539
Intercambio de eventos/incidentes/eurística	**X.1540–X.1549**
Intercambio de políticas	X.1550–X.1559
Petición de eurística e información	X.1560–X.1569
Identificación y descubrimiento	X.1570–X.1579
Intercambio asegurado	X.1580–X.1589
SEGURIDAD DE LA COMPUTACIÓN EN NUBE	
Visión general de la seguridad de la computación en nube	X.1600–X.1601
Diseño de la seguridad de la computación en nube	X.1602–X.1639
Prácticas óptimas y directrices en materia de seguridad de la computación en nube	X.1640–X.1659
Aplicación práctica de la seguridad de la computación en nube	X.1660–X.1679
Otras cuestiones de seguridad de la computación en nube	X.1680–X.1699

Para más información, véase la Lista de Recomendaciones del UIT-T.

Recomendación UIT-T X.1542

Formato de intercambio de mensajes sobre información de sesión

Resumen

En el actual entorno de red, las redes de computadoras son vulnerables a las amenazas procedentes de dentro y fuera de la organización. Los sistemas cortafuegos registran información de sesión sobre determinadas conexiones de protocolo de control de transmisión/protocolo Internet (TCP/IP) entrantes y salientes.

Sin embargo, estos sistemas no suelen ser compatibles entre sí, porque cada uno tiene su propia funcionalidad, sus propios mecanismos de control y sus propios formatos de registro de sesión.

Hoy en día, la mayoría de los gestores de seguridad necesitan mantener un formato de intercambio de mensajes sobre información de sesión coherente en distintos sistemas cortafuegos e incluso en distintas infraestructuras.

En la Recomendación UIT-T X.1542 se describe el modelo de información del formato de intercambio de mensajes sobre información de sesión (SIMEF) y se facilita un modelo de datos conexo especificado con un esquema de lenguaje extensible de marcado (XML). El SIMEF define una representación de modelo de datos para compartir la información de registro de sesión de la capa de transporte vinculada a la gestión centralizada de la seguridad de la red y el sistema de intercambio de información de seguridad. La especificación de cualquier protocolo de transporte queda fuera del alcance de la presente Recomendación.

Historia

Edición	Recomendación	Aprobación	Comisión de Estudio	ID único*
1.0	UIT-T X.1542	2016-09-07	17	11.1002/1000/12852

Términos

Modelo de datos, intercambio de mensajes, seguridad de la red, información de sesión.

* Para acceder a la Recomendación, sírvase digitar el URL http://handle.itu.int/ en el campo de dirección del navegador, seguido por el identificador único de la Recomendación. Por ejemplo, http://handle.itu.int/11.1002/1000/11830-en.

PREFACIO

La Unión Internacional de Telecomunicaciones (UIT) es el organismo especializado de las Naciones Unidas en el campo de las telecomunicaciones y de las tecnologías de la información y la comunicación. El Sector de Normalización de las Telecomunicaciones de la UIT (UIT-T) es un órgano permanente de la UIT. Este órgano estudia los aspectos técnicos, de explotación y tarifarios y publica Recomendaciones sobre los mismos, con miras a la normalización de las telecomunicaciones en el plano mundial.

La Asamblea Mundial de Normalización de las Telecomunicaciones (AMNT), que se celebra cada cuatro años, establece los temas que han de estudiar las Comisiones de Estudio del UIT-T, que a su vez producen Recomendaciones sobre dichos temas.

La aprobación de Recomendaciones por los Miembros del UIT-T es el objeto del procedimiento establecido en la Resolución 1 de la AMNT.

En ciertos sectores de la tecnología de la información que corresponden a la esfera de competencia del UIT-T, se preparan las normas necesarias en colaboración con la ISO y la CEI.

NOTA

En esta Recomendación, la expresión "Administración" se utiliza para designar, en forma abreviada, tanto una administración de telecomunicaciones como una empresa de explotación reconocida de telecomunicaciones.

La observancia de esta Recomendación es voluntaria. Ahora bien, la Recomendación puede contener ciertas disposiciones obligatorias (para asegurar, por ejemplo, la aplicabilidad o la interoperabilidad), por lo que la observancia se consigue con el cumplimiento exacto y puntual de todas las disposiciones obligatorias. La obligatoriedad de un elemento preceptivo o requisito se expresa mediante las frases "tener que, haber de, hay que + infinitivo" o el verbo principal en tiempo futuro simple de mandato, en modo afirmativo o negativo. El hecho de que se utilice esta formulación no entraña que la observancia se imponga a ninguna de las partes.

PROPIEDAD INTELECTUAL

La UIT señala a la atención la posibilidad de que la utilización o aplicación de la presente Recomendación suponga el empleo de un derecho de propiedad intelectual reivindicado. La UIT no adopta ninguna posición en cuanto a la demostración, validez o aplicabilidad de los derechos de propiedad intelectual reivindicados, ya sea por los miembros de la UIT o por terceros ajenos al proceso de elaboración de Recomendaciones.

En la fecha de aprobación de la presente Recomendación, la UIT ha recibido notificación de propiedad intelectual, protegida por patente, que puede ser necesaria para aplicar esta Recomendación. Sin embargo, debe señalarse a los usuarios que puede que esta información no se encuentre totalmente actualizada al respecto, por lo que se les insta encarecidamente a consultar la base de datos sobre patentes de la TSB en la dirección http://www.itu.int/ITU-T/ipr/.

ÍNDICE

Página

1 Alcance 1
2 Referencias 1
3 Definiciones 1
3.1 Términos definidos en otros documentos 1
3.2 Términos definidos en esta Recomendación 1
4 Siglas y acrónimos 1
5 Convenios 2
6 Resumen 2
7 Representación y definición 3
7.1 Documentos XML en SIMEF 3
7.2 Tipos de datos en SIMEF 4
8 El modelo de datos SIMEF 6
8.1 Resumen del modelo de datos 6
8.2 Clases de los mensajes 7
9 Consideraciones relativas a la seguridad 29
Apéndice I – Esquemas y ejemplos de SIMEF 30
I.1 Esquema de SIMEF 30
I.2 Ejemplos de SIMEF 31
Bibliografía 34

Recomendación UIT-T X.1542

Formato de intercambio de mensajes sobre información de sesión

1 Alcance

En la presente Recomendación se describe el formato de intercambio de mensajes sobre información de sesión (SIMEF), a saber, un modelo de datos que permite representar la información de sesión exportada por un sistema de seguridad (por ejemplo, un cortafuego), y se justifica la utilización de dicho modelo. Además, se presenta un modelo de datos conexo en lenguaje extensible de marcado (XML), se facilita una definición de tipo de documento (DTD) XML y se proporcionan ejemplos.

2 Referencias

Ninguna.

3 Definiciones

3.1 Términos definidos en otros documentos

Ninguno.

3.2 Términos definidos en esta Recomendación

En la presente Recomendación se definen los siguientes términos:

3.2.1 Analizador: Un analizador es un sistema de seguridad de red que detecta ataques mediante el análisis de la información de sesión entrante y saliente. También genera un registro de sesión y lo envía a los sistemas de gestión de la seguridad.

3.2.2 Información de sesión: Información que contiene la sesión del protocolo de control de transmisión/protocolo de datagrama de usuario (TCP/IP), las entidades del servicio de aplicación y de sesión tal como son vistas por los proveedores de la información de sesión. Una sesión se define como el conjunto de tráfico que se gestiona como una unidad de traducción. Las sesiones TCP/UDP se identifican inequívocamente por la tupla dirección IP de origen, puerto TCP/UDP de origen, dirección IP de destino y puerto TCP/UDP de destino.

NOTA – Esta definición está basada en [b-IETF RFC 2663].

4 Siglas y acrónimos

En la presente Recomendación se utilizan las siguientes siglas y acrónimos:

BSD Distribución de software Berkeley
CGI Interfaz de pasarela común
DTD Definición de tipo de documento
FTP Protocolo de transferencia de ficheros
HTTP Protocolo de transferencia de hipertexto
IP Protocolo Internet
LAN Red de área local
MAC Control de acceso a los medios
NAT Traducción de direcciones de red
NTP Protocolo de tiempo de red

POSIX	Interfaz de sistema operativo portable
SIMEF	Formato de intercambio de mensajes sobre información de sesión
SNA	Arquitectura de red compartida
SNMP	Protocolo simple de gestión de red
TCP	Protocolo de control de transmisión
UDP	Protocolo de datagramas de usuario
UML	Lenguaje de modelización unificado
URL	Localizador uniforme de recursos
UTF	Formato de transformación del conjunto universal de caracteres
VPN	Red privada virtual
XML	Lenguaje extensible de marcado

5 Convenios

UNIX ® es una marca registrada de Open Group.

POSIX ® es una marca registrada del IEEE.

6 Resumen

En el actual entorno de red, las redes de computadoras son vulnerables a las amenazas procedentes de dentro y fuera de la organización. Por consiguiente, la mayor parte de los trabajos de investigación atinentes a la seguridad de la red han girado en torno a la creación de sistemas de gestión integrada de la seguridad de la red, así como de herramientas de supervisión de la red, que permitan a una organización capturar los paquetes TCP/IP que pasan a través de sus dispositivos de red de y ver los datos capturados como secuencias de conversaciones entre clientes y servidores. Por ejemplo, los sistemas cortafuegos registran información de sesión sobre determinadas conexiones TCP/IP entrantes y salientes.

El concepto de SIMEF se representa en la Figura 1. La información de sesión puede recopilarse a través de sistemas cortafuegos, dispositivos de traducción de direcciones de red (NAT) y otros. SIMEF especifica el modelo de datos que abarca la conexión de red del cliente/servidor, el dispositivo de usuario final y el servicio de aplicación. SIMEF define asimismo un modelo de datos y un conjunto de clases de mensaje conexo para compartir la información de sesión de la capa de transporte que reviste un interés particular para los sistemas de gestión de la seguridad e intercambio de información. Todo ello puede aplicarse al sistema de intercambio de información sobre intrusiones.

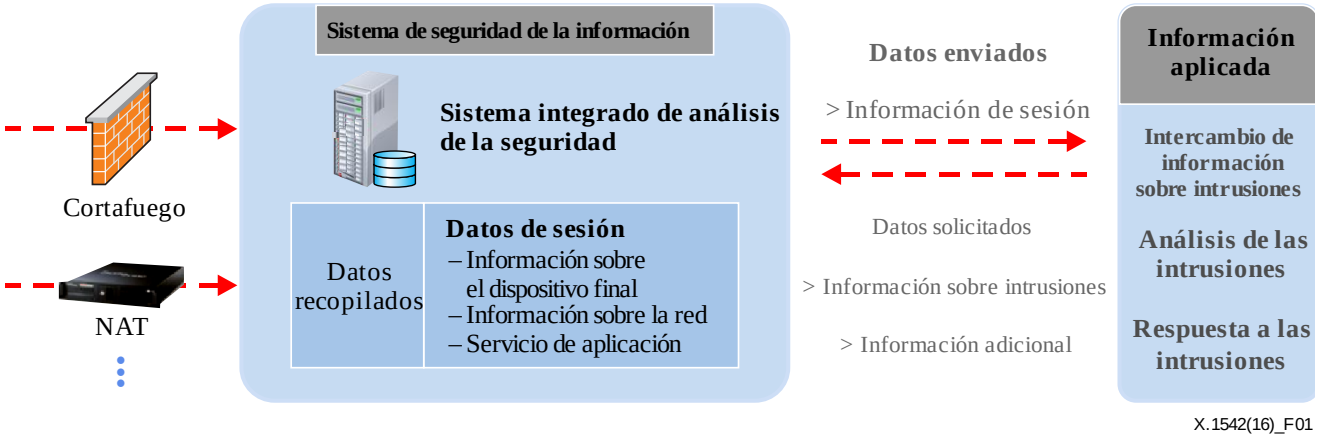

Figura 1 – El concepto de SIMEF

7 Representación y definición

En la presente Recomendación se utilizan tres notaciones, a saber: el lenguaje de modelización unificado (UML) para describir el modelo de datos, XML para describir el marcado utilizado en los documentos en SIMEF, y el marcado de SIMEF para representar los propios documentos.

7.1 Documentos XML en SIMEF

En este apartado se describen las normas aplicables al formateado de documentos XML en SIMEF. La mayoría de estas normas tienen su origen en las relativas al formateado de documentos XML. El formato del prólogo de un documento XML en SIMEF se detalla en las secciones 7.1.1 a 7.1.2.

7.1.1 Declaración XML

Todos los documentos en SIMEF que intercambian las aplicaciones compatibles con dicho formato comenzarán con una declaración XML y especificarán la versión de XML utilizada. También se recomienda especificar la codificación.

Por tanto, un mensaje en SIMEF debe comenzar como sigue:

```
<?xml version="1.0" encoding="UTF-8"?>
<simef: SIMEF-Message version="1.2" xmlns:simef="http://iana.org/simef"/>
```

Las aplicaciones compatibles con SIMEF pueden optar por omitir la declaración XML en el plano interno para ahorrar espacio y añadirla únicamente cuando envíen el mensaje a otro destinatario (por ejemplo, un navegador web). Esta práctica no es recomendable a menos que pueda llevarse a cabo sin menoscabar la información relativa a la codificación y la versión de cada mensaje.

En consecuencia, los implementadores pueden brindar a los analizadores y gestores la posibilidad de acordar fuera de banda la definición de tipo de documento (DTD) que van a utilizar para el intercambio de mensajes (ya sea una normalizada como la contemplada en la presente Recomendación o una con extensiones) y, a continuación, omitir la DTD en los mensajes en SIMEF. El método utilizado para la negociación de dicho acuerdo queda fuera del alcance de esta Recomendación.

7.1.2 Procesamiento de datos de caracteres en SIMEF

Por motivos de portabilidad, las aplicaciones compatibles con SIMEF solo deberían utilizar los formatos de codificación de caracteres UTF-8 y UTF-16, es decir, los mismos que debería emplearse para la codificación de los mensajes en SIMEF. De conformidad con la norma sobre XML, si no se especifica el formato de codificación de un mensaje en SIMEF, se asume UTF-8.

7.1.2.1 Referencias de entidad de caracteres

Se recomienda que las aplicaciones compatibles con SIMEF utilicen las formas de referencia de entidad (véase el apartado 3.2.3.1) de los caracteres «&», «<», «>», «"», y «'» (comillas simples) cuando escriban estos caracteres en datos, para evitar toda interpretación errónea.

7.1.2.2 Procesamiento de espacios en blanco

Todos los elementos de SIMEF deberán soportar el atributo "xml:space".

7.1.2.3 Lenguajes en SIMEF

Las aplicaciones compatibles con SIMEF especificarán el lenguaje en que están codificados sus contenidos. A tal efecto, se puede especificar el atributo "xml:lang" del elemento de nivel superior y permitir que todos los demás elementos "hereden" esa definición.

7.2 Tipos de datos en SIMEF

En los mensajes XML en SIMEF, todos los datos se expresarán en forma de texto, puesto que XML es un lenguaje de formateado de texto. Este proporciona información relativa a la transcripción de los atributos de las clases presentes en el modelo de datos. Cada tipo de datos del modelo posee requisitos de formateado específicos en el marco de un mensaje XML en SIMEF, los cuales se describen en la presente sección.

7.2.1 Enteros

Los atributos enteros se representan mediante el tipo de datos INTEGER. Los datos enteros se codificarán en base 10 o en base 16. El formato de codificación de enteros en base 10 utiliza los dígitos de '0' a '9' y un signo opcional ('+' o '–'). Por ejemplo, "123", "–456". El formato de codificación de enteros en base 16 utiliza los dígitos de '0' a '9' y de 'a' a 'f' (o sus equivalentes en mayúsculas), y viene precedido de los caracteres "0x". Por ejemplo, "0x1a2b".

7.2.2 Números reales

Los atributos reales (coma flotante) se representan mediante el tipo de datos REAL. Los datos reales se codificarán en base 10. El formato de codificación real es el de la función de biblioteca "strtod" de la interfaz de sistema operativo portable (POSIX) 1003.1 [b-IEEE 1003.1]: un signo opcional ('+' o '–') seguido de una cadena no vacía de dígitos decimales, que puede contener un carácter base acompañado de una parte exponente opcional. Una parte exponente se compone de una 'e' o 'E', seguida de un signo opcional, seguido de uno o más dígitos decimales. Por ejemplo, "123,45e02", "–567,89e–03". Las aplicaciones compatibles con SIMEF deberán soportar los caracteres de base '.' y ','.

7.2.3 Caracteres y cadenas

Los atributos de un sólo carácter se representan mediante el tipo de datos CHARACTER. Los atributos de múltiples caracteres de una longitud determinada están representados por el tipo de datos STRING. El único requisito de formateado particular que poseen los datos relativos a los caracteres y las cadenas es la necesidad de utilizar ocasionalmente referencias de caracteres para representar caracteres especiales.

7.2.3.1 Referencias de entidad de caracteres

En los documentos XML, ciertos caracteres adquieren un significado especial en determinados contextos. Para incluir el carácter real en uno de estos contextos se utilizará una secuencia de escape especial, denominada referencia de entidad.

Los caracteres que a veces requieren una secuencia de escape y sus referencias de entidad son los siguientes:

Carácter	Referencia de entidad
&	&amp;
<	&lt;
>	&gt;
"	&quot;
'	&apos;

7.2.3.2 Referencias de código de caracteres

Todos los caracteres definidos por las normas Unicode y [b-ISO/CEI 10646] pueden incluirse en un documento XML mediante el uso de una referencia de carácter. Una referencia de carácter comienza

con los caracteres '&' y '#', y termina con el carácter ';'. El código de carácter en cuestión se inserta entre ambos caracteres.

Si el código de carácter viene precedido por una 'x', se interpreta en hexadecimal (base 16); de lo contrario, se interpreta en decimal (base 10). Por ejemplo, el signo & se codifica como & o & y el signo menor que (<) se codifica como < o <. Todos los caracteres de uno, dos o cuatro bytes especificados en las normas ISO/CEI 10646 y Unicode pueden incluirse en un documento utilizando esta técnica.

7.2.4 Bytes

Los datos binarios se representan mediante el tipo de datos BYTE (y BYTE[]). Todos los datos binarios se codificarán utilizando una base 64.

7.2.5 Tipos enumerados

Los tipos enumerados se representan mediante el tipo de datos ENUM, y consisten en una lista ordenada de valores aceptables.

7.2.6 Cadenas fecha-hora

Las cadenas fecha-hora se representan mediante el tipo de datos DATETIME. Cada cadena fecha-hora identifica un momento determinado en el tiempo; no se admiten gamas. Las cadenas fecha-hora se formatean de conformidad con un subconjunto de [b-ISO 8601:2004], según se muestra a continuación. Las referencias de sección situadas entre paréntesis aluden a ciertos apartados de [b-ISO 8601:2004].

7.2.7 Sellos temporales de NTP

Los sellos temporales de protocolo de señales horarias de red (NTP) se representan mediante el tipo de datos NTPSTAMP y se detallan en [b-IETF RFC 1305] y [b-IETF RFC 5905]. Un sello temporal de NTP es un número de 64 bits en coma fija y sin signo. La parte entera se halla en los primeros 32 bits, y la parte fraccionaria en los segundos 32 bits. En los mensajes en SIMEF, los sellos temporales de NTP se codificarán como dos valores hexadecimales de 32 bits, separados por un punto ('.'). Por ejemplo, "0x12345678.0x87654321".

7.2.8 Listas de puertos

Las listas de puertos se representan mediante el tipo de datos PORTLIST y consisten en una lista separada por comas de números (enteros individuales) y gamas (N-M significa puertos de N a M, ambos inclusive). Todas las combinaciones de números y gamas pueden incluirse en una sola lista. Por ejemplo, "5-25,37,42,43,53,69-119,123-514".

7.2.9 Identificadores únicos

En la presente Recomendación se utilizan dos tipos de identificador único, que se representan mediante el tipo de datos STRING. Estos identificadores se aplican como atributos de los elementos XML correspondientes, y tendrán los siguientes valores únicos:

1 El atributo "*deviceid*" (identificador de dispositivo) (véase el apartado 8.2.3.2) de clase *Device* (dispositivo), si se especifica, tendrá un valor único en todos los analizadores presentes en el entorno de detección de intrusiones.

2 El valor por defecto es "0", lo cual indica que el analizador no puede generar identificadores únicos.

El atributo "*ident*", si se especifica, de varias clases tendrá un valor único en todos los mensajes enviados por el analizador individual. El valor del atributo "*ident*" será único para cada combinación particular de datos de identificación de un objeto, en lugar de ser único para cada objeto. Los objetos pueden tener más de un valor "*ident*" asociado. Por ejemplo, la identificación de un anfitrión por nombre tendría un valor, la identificación de ese anfitrión

por dirección tendría un segundo valor, y la identificación de ese mismo anfitrión por nombre y dirección tendría otro valor.

El valor por defecto es "0", lo cual indica que el analizador no puede generar identificadores únicos.

La especificación de los métodos de creación de los valores únicos contenidos en estos atributos queda fuera del alcance de la presente Recomendación.

8 El modelo de datos SIMEF

En este apartado se detallan los componentes individuales del modelo de datos SIMEF. Se proporcionan diagramas UML del modelo para mostrar las relaciones que existen entre los componentes.

8.1 Resumen del modelo de datos

En la Figura 2 se muestran las relaciones que existen entre los componentes principales del modelo de datos. La clase de nivel superior se denomina *SIMEF-Message* (SIMEF-Mensaje); cada tipo de mensaje integra una subclase de esta clase de nivel superior. Existen dos tipos de mensajes definidos: *Connects* (conexiones) y *Heartbeats* (latidos). En cada mensaje, las subclases de la clase de mensaje se utilizan para proporcionar la información detallada que contiene el mensaje. La clase de mensaje *Connect* tiene varias subclases, por ejemplo, *Devices* (dispositivos), *Policy* (política), *Source* (origen), *Target* (destino) y *AdditionalData* (datos adicionales).

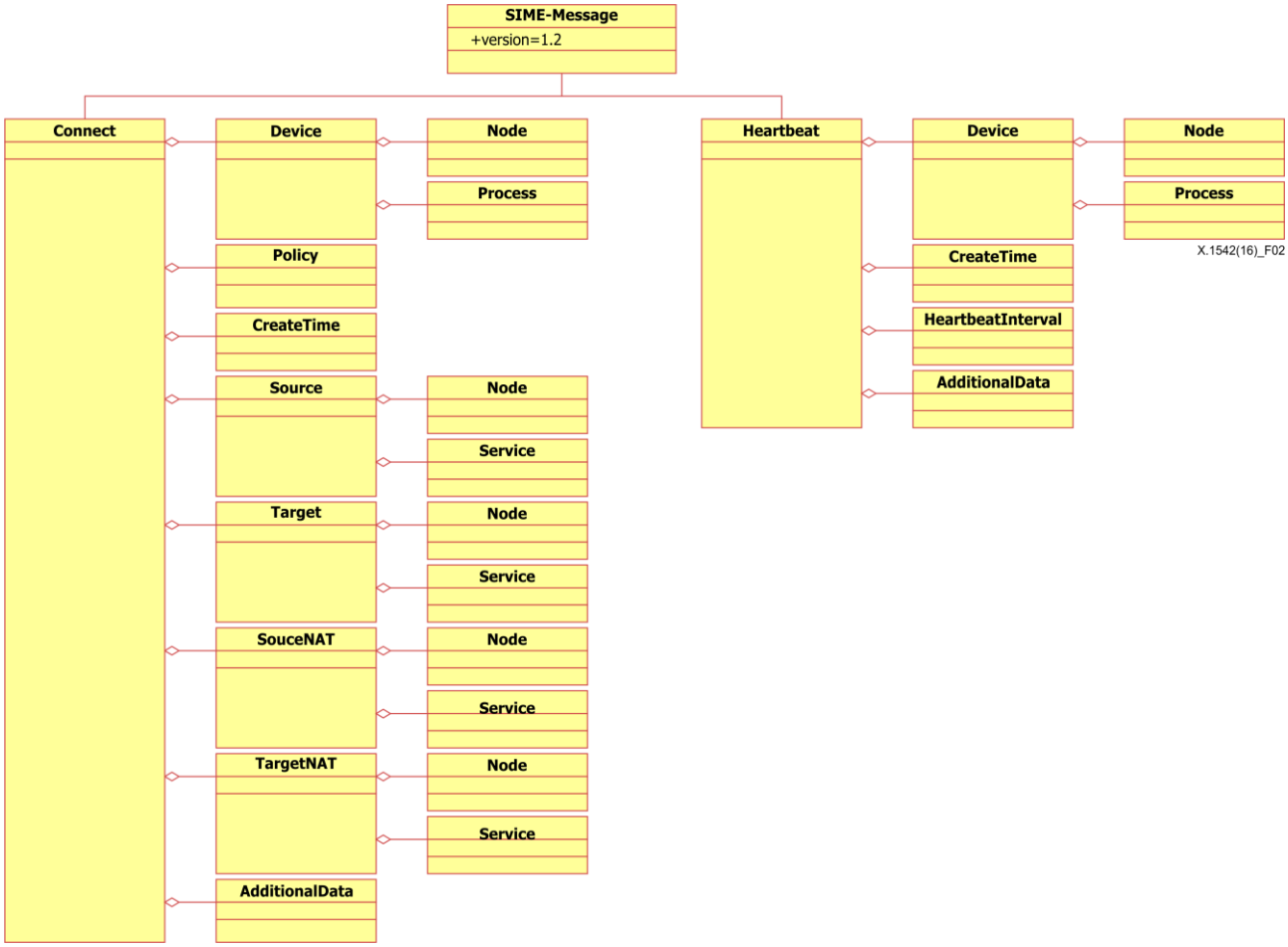

Figura 2 – El modelo de datos SIMEF

8.1.1 Clases de mensajes SIMEF

Todos los mensajes SIMEF derivan de la clase *SIMEF-Message*: *Connect* y *Heartbeat*. Las clases individuales se describen en esta sección. Véanse la Figura 3 y los Cuadros 1 y 2.

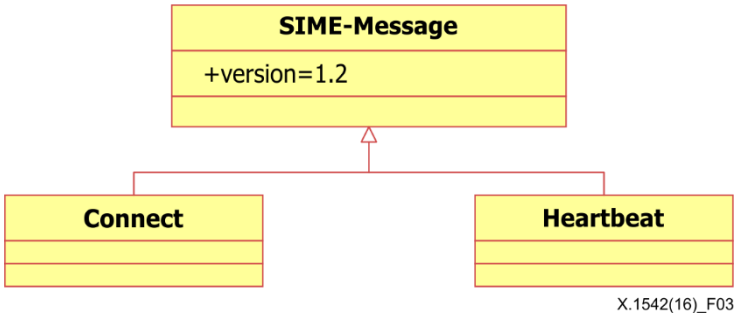

Figura 3 – Clase de nivel superior del modelo de datos SIMEF

Cuadro 1 – Atributos de las clases de mensajes SIMEF

Atributo	Utilización	Tipo de datos	Descripción
Version	Requerida	STRING	Información sobre la versión de SIMEF; valor por defecto: 1.2

Cuadro 2 – Componentes de las clases de mensajes SIMEF

Clase	Agregación	Tipo de datos	Descripción
Connect	Exactamente uno		Clase de información de sesión
Heartbeat	Cero o ninguno		Clase de información de situación del sistema (opcional)

8.2 Clases de los mensajes

En los apartados 8.2.1 a 8.2.4 se describen las clases individuales.

8.2.1 Clase *Connect*

La clase *Connect* (conexión) contiene la información de sesión. Expresa el tipo de registro generado por la conexión en el cortafuego y muestra toda la información disponible acerca de los intentos de conexión en los planos interno y externo. Véase el Cuadro 3. En el Cuadro 4 se muestra los valores permitidos del atributo *criticality* (criticidad) de la clase *Connect*. La clase *Connect* se compone de varias clases agregadas, tal como se muestra en la Figura 4. Las clases agregadas se describen en el Cuadro 5.

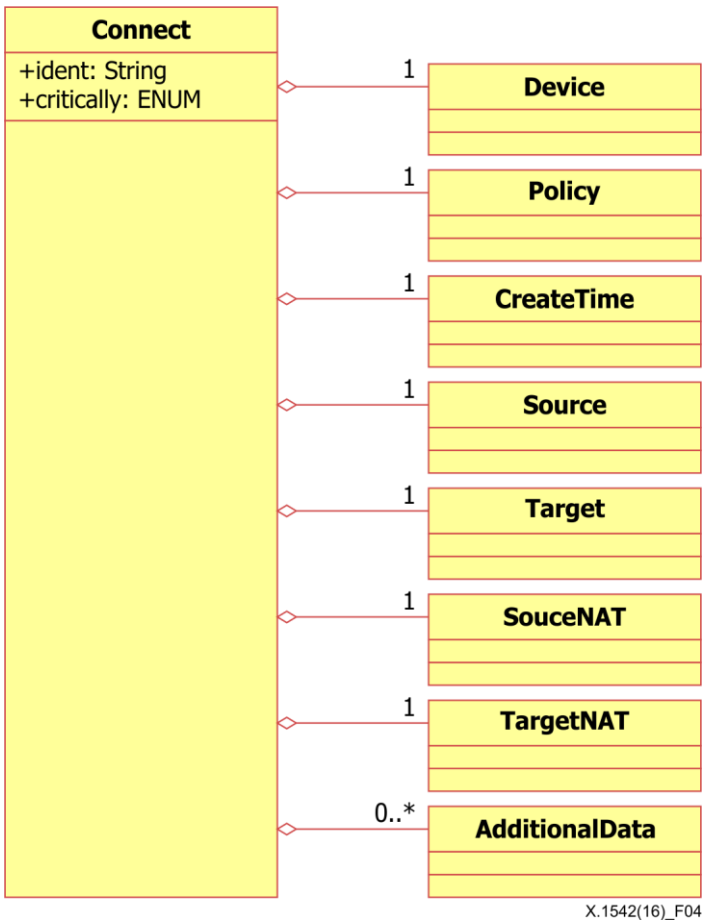

Figura 4 – Clases agregadas de la clase *Connect*

Cuadro 3 – Atributos de la clase *Connect*

Atributo	Utilización	Tipo de datos	Descripción
ident	Opcional	STRING	Identificador único para el acceso a la información
criticality	Opcional	ENUM	Clasificación conforme a la evaluación del evento que origina la conexión Valor por defecto: Unknown (desconocido)

Cuadro 4 – Valores del atributo *criticality*

Valor	Palabra clave	Definición
0	*unknown*	Cuando el efecto del evento se desconoce o no puede determinarse
1	*normal*	Cuando la conexión es normal
2	*suspicious*	Cuando la conexión es sospechosa
3	*warning*	Cuando la conexión puede ser alarmante
4	*critical*	Cuando la conexión puede requerir la adopción de medidas

Cuadro 5 – Componentes de la clase *Connect*

Clase	Agregación	Tipo de datos	Descripción
Device	Exactamente uno		Información del analizador que genera el registro
Policy	Exactamente uno		Información transportada en el analizador para la conexión
CreateTime	Exactamente uno	DATETIME	Tiempo para la creación del registro
Source	Exactamente uno		Origen del evento que causa la conexión
Target	Exactamente uno		Información de destino del evento que causa una conexión
SourceNAT	Exactamente uno		Información de origen de NAT del evento que causa la conexión
TargetNAT	Exactamente uno		Información de destino de NAT del evento que causa la conexión
AdditionalData	Cero o más		Información adicional generada por el detector que no figura en las otras clases

8.2.1.1 Clase *Policy*

La clase *Policy* (política) proporciona la información necesaria para indicar el modo en que se ha de gestionar una sesión en el analizador. Véase la Figura 5.

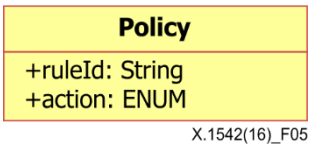

Figura 5 – Clase *Policy*

En el Cuadro 7 figuran los valores permitidos del atributo *action* de la clase *Policy* (véase el Cuadro 6).

Cuadro 6 – Atributos de la clase *Policy*

Atributo	Utilización	Tipo de datos	Descripción
ruleId	Opcional	STRING	Identificador único de la política de cortafuego que origina la conexión
action	Opcional	ENUM	Clasificación conforme al cortafuego de la operación que origina la conexión Valor por defecto: Unknown (desconocido)

Cuadro 7 – Valores del atributo *action*

Valor	Palabra clave	Definición
0	*unknown*	Cuando se desconoce el comportamiento
1	*pass*	Cuando se permite la conexión
2	*block*	Cuando se deniega la conexión
3	*protect*	Cuando se encripta el paquete transmitido o se inserta un código de verificación de la integridad [registro de red privada virtual (VPN)]
4	*reject*	Cuando se rechaza de la conexión; en este caso, se transmiten mensajes de error cuando se deniega el acceso

8.2.2 Clase *Heartbeat*

Los analizadores utilizan mensajes *Heartbeat* (latido) para indicar su estado actual a los gestores. Estos deben enviarse periódicamente, por ejemplo, una vez cada 10 minutos o cada hora. La recepción de uno de estos mensajes de un analizador indica al gestor que el analizador en cuestión está en marcha y funciona correctamente. La ausencia de dichos mensajes (o, más probablemente, la falta de un número determinado de mensajes consecutivos) indica un fallo en el analizador o su conexión de red.

Todos los gestores deberán soportar la recepción de mensajes tipo *Heartbeat*, no obstante, la utilización de dichos mensajes por los analizadores es opcional. Los desarrolladores de software de gestión deberían permitir que el software en cuestión se configurase de manera individualizada, con objeto de que cada analizador utilice u omita estos mensajes en función de sus necesidades. Un mensaje *Heartbeat* se compone de varias clases agregadas, tal como se muestra en la Figura 6.

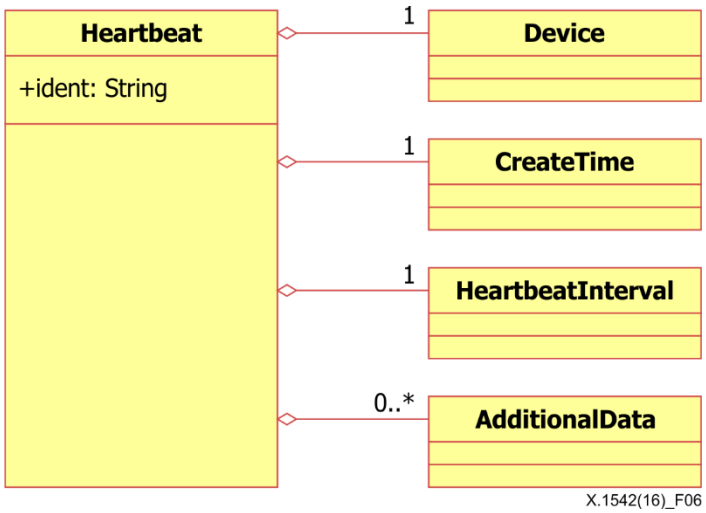

Figura 6 – Clases agregadas de la clase *Heartbeat*

La información sobre el atributo y componentes de la clase *Heartbeat* figuran en el Cuadro 8 y el Cuadro 9 respectivamente.

Cuadro 8 – Atributo de la clase *Heartbeat*

Atributo	Utilización	Tipo de datos	Descripción
ident	Opcional	STRING	Identificador único del latido

Cuadro 9 – Componentes de la clase *Heartbeat*

Clase	Agregación	Tipo de datos	Descripción
Device	Exactamente uno		Información de identificación del analizador que originó el latido
CreateTime	Exactamente uno	DATETIME	Momento en que se originó el latido
HeartbeatInterval	Exactamente uno	INTEGER	El intervalo (en segundos) en que se generan los latidos
AdditionalData	Cero o más		Información incluida por el analizador que no se ajusta al modelo de datos

8.2.3 Clases fundamentales

Las clases fundamentales (*Device*, *CreateTime*, *Source*, *Target*, *SourceNAT*, *TargetNAT* y *AdditionalData*) integran las partes principales de las clases *Connect* y *Heartbeat*, tal como se muestra en la Figura 7. Las clases individuales se describen en este apartado.

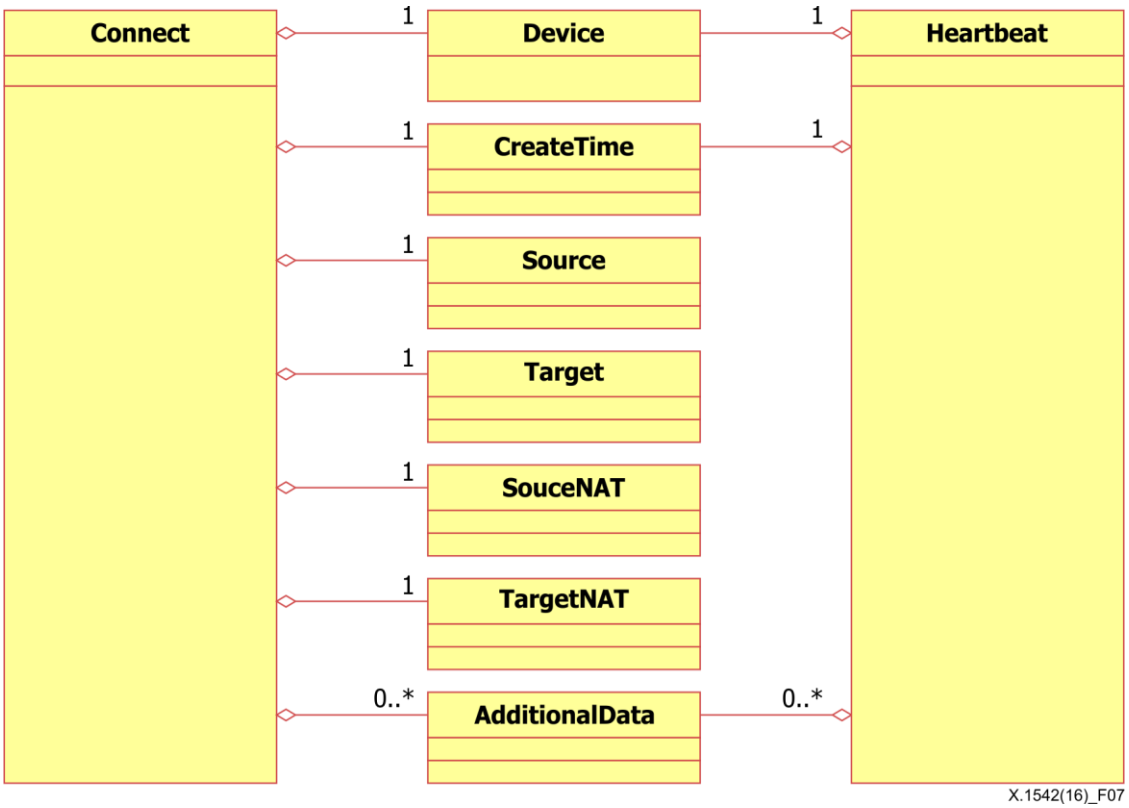

Figura 7 – Clases fundamentales

8.2.3.1 Clase *Device*

La clase *Device* (dispositivo) identifica el analizador desde el que se originan los mensajes de *Connect* o *Heartbeat*. Sólo puede codificarse un dispositivo para cada conexión o latido, y dicho dispositivo será aquel en que se origine la conexión o el latido.

La clase *Device* se compone de tres clases agregadas, tal como se muestra en la Figura 8.

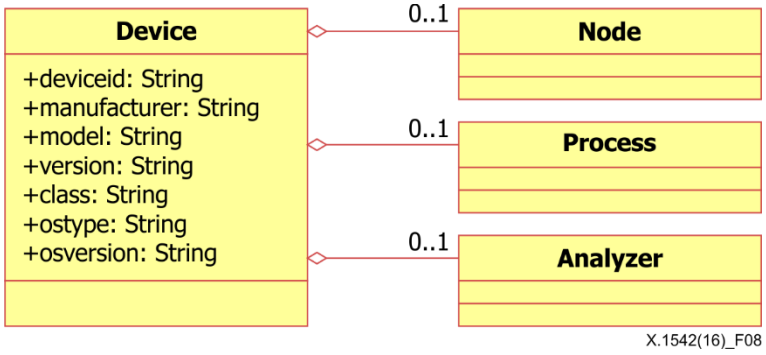

Figura 8 – Clases agregadas de la clase *Device*

La clase *Device* tiene siete atributos, tal como se muestra en el Cuadro 10.

Cuadro 10 – Atributos de la clase Device

Atributo	Utilización	Tipo de datos	Descripción
deviceid	Opcional	STRING	Identificador único para el dispositivo; si el dispositivo utiliza los atributos "ident" en otras clases para proporcionar identificadores únicos a esos objetos, deberá asimismo facilitar un atributo "deviceid" válido
Manufacturer	Opcional	STRING	Fabricante del software o hardware del dispositivo
Model	Opcional	STRING	Nombre/número del modelo del software o hardware del dispositivo
Version	Opcional	STRING	Número de versión del software o hardware del dispositivo
Class	Opcional	STRING	Clase de software o hardware del dispositivo
Ostype	Opcional	STRING	Nombre del sistema operativo
osversion	Opcional	STRING	Versión del sistema operativo

En sistemas compatibles con POSIX 1003.1, el atributo *ostype* (tipo de sistema operativo) es el valor devuelto en utsname.sysname por la llamada al sistema uname(), o el resultado de la instrucción "uname –s".

En sistemas compatibles con POSIX 1003.1, el atributo *osversion* (versión del sistema operativo) es el valor devuelto en utsname.release por la llamada al sistema uname(), o el resultado de la instrucción "uname –r".

El contenido de los atributos *manufacturer* (fabricante), *model* (modelo), *version* (versión) *y class* (clase) son específicos del proveedor, no obstante, pueden utilizarse en conjunto para identificar diferentes tipos de analizadores.

Las clases agregadas que conforman la clase *Device* se describen en el Cuadro 11.

Cuadro 11 – Componentes de la clase *Device*

Clase	Agregación	Tipo de datos	Descripción
Node	Cero o ninguno		Información sobre el anfitrión o dispositivo en que reside el analizador (dirección de la red, nombre de la red, etc.)
Process	Cero o ninguno		Información sobre el proceso en que se está ejecutando el analizador
Analyser	Cero o ninguno		Información sobre el analizador a través del cual puede haber pasado el mensaje

8.2.3.2 Clase *CreateTime*

La clase *CreateTime* (crear tiempo) se utiliza para indicar la fecha y la hora actuales en el dispositivo. Si esta diferencia debiera aplicarse para ajustar los tiempos en los elementos <CreateTime> y < NTP timestamps >, cabría ajustar también los sellos temporales de NTP.

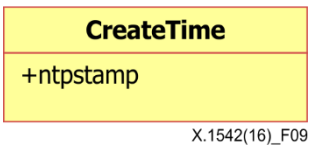

Figura 9 – Clase *CreateTime*

Los atributos de la clase *CreateTime* figuran en el Cuadro 12.

Cuadro 12 – Atributos de la clase *CreateTime*

Atributo	Utilización	Tipo de datos	Descripción
ntpstamp	Requerido	ntpstamp	Información sobre la hora actual del dispositivo

8.2.3.3 Clase *Source*

La clase *Source* (origen) contiene información acerca del o los posibles orígenes del o los eventos que han generado una sesión. Un evento puede tener más de un origen (por ejemplo, en el caso de un ataque de denegación de servicio distribuido).

La clase *Source* se compone de tres clases agregadas, tal como se muestra en la Figura 10.

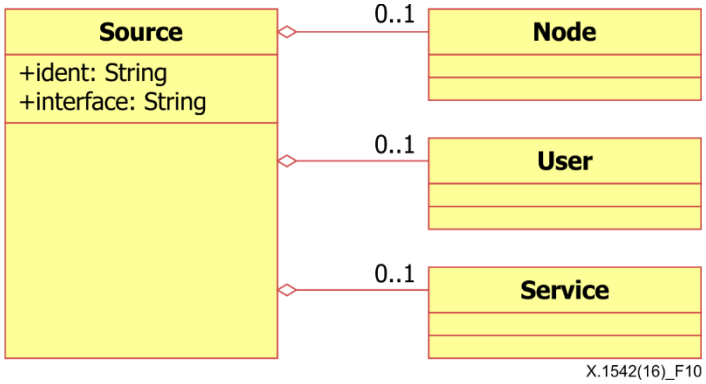

Figura 10 – Clases agregadas de la clase *Source*

La clase *Source* tiene dos atributos que figuran en el Cuadro 13.

Cuadro 13 – Atributos de la clase *Source*

Atributo	Utilización	Tipo de datos	Descripción
ident	Opcional	STRING	Identificador único para este origen
Interface	Opcional	STRING	Un dispositivo de red con múltiples interfaces puede utilizarlo para indicar en qué interfaz se ha visto este origen

Las clases agregadas que integran la clase *Source* se describen en el Cuadro 14.

Cuadro 14 – Componentes de la clase *Source*

Clase	Agregación	Tipo de datos	Descripción
Node	Cero o ninguno		Información sobre el anfitrión o dispositivo que parece originar el o los eventos (dirección de la red, el nombre de la red, etc.)
User	Cero o ninguno		Información sobre el usuario que parece originar el o los eventos
Service	Cero o ninguno		Información sobre el servicio de red que participa en el o los eventos

8.2.3.4 Clase *Target*

La clase *Target* (destino) contiene información sobre el o los posibles destinos del o los eventos que han generado una sesión. Un evento puede tener más de un destino (por ejemplo, en el caso de un barrido de puerto).

La clase *Target* se compone de tres clases agregadas, tal como se muestra en la Figura 11.

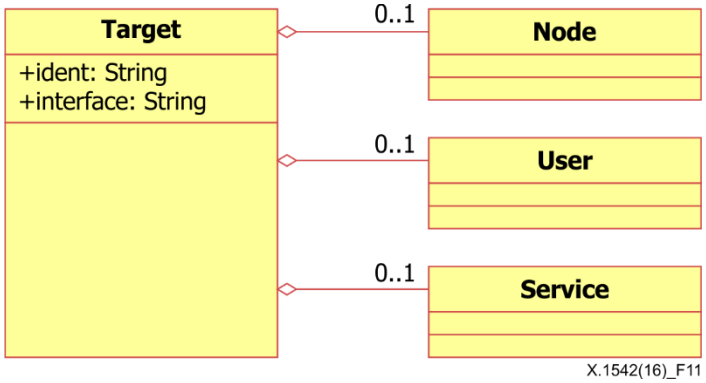

Figura 11 – Clases agregadas de la clase *Target*

La clase *Target* tiene dos atributos que figuran en el Cuadro 15.

Cuadro 15 – Atributos de la clase *Target*

Atributo	Utilización	Tipo de datos	Descripción
ident	Opcional	STRING	Identificador único para este destino
Interface	Opcional	STRING	Un dispositivo de red con múltiples interfaces puede utilizarlo para indicar en qué interfaz se ha visto este destino

Las clases agregadas que integran la clase *Target* se describen en el Cuadro 16.

Cuadro 16 – Componentes de la clase *Target*

Clase	Agregación	Tipo de datos	Descripción
Node	Cero o ninguno		Información sobre el anfitrión o dispositivo hacia el que se dirigen el o los eventos (dirección de la red, el nombre de la red, etc.)
User	Cero o ninguno		Información sobre el usuario hacia el que se dirigen el o los eventos
Service	Cero o ninguno		Información sobre el servicio de red que participa en el o los eventos

8.2.3.5 Clase *SourceNAT*

La clase *SourceNAT* (NAT de origen) contiene información acerca del o los posibles orígenes del o los eventos NAT que han generado una sesión. Un evento puede tener más de un origen transformado por el NAT.

La clase *SourceNAT* se compone de tres clases agregadas, tal como se muestra en la Figura 12.

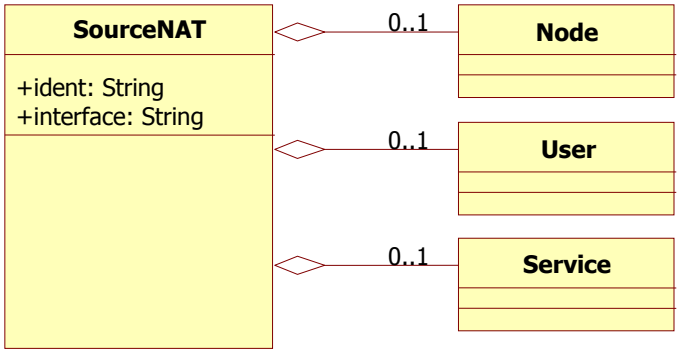

Figura 12 – Clases agregadas de la clase *SourceNAT*

La clase *SourceNAT* tiene dos atributos que figuran en el Cuadro 17.

Cuadro 17 – Atributos de la clase *SourceNAT*

Atributo	Utilización	Tipo de datos	Descripción
ident	Opcional	STRING	Identificador único para este origen transformado por el NAT
interface	Opcional	STRING	Un dispositivo de red con múltiples interfaces puede utilizarlo para indicar en qué interfaz se ha visto este origen transformado por el NAT

Las clases agregadas que integran la clase SourceNAT se describen en el Cuadro 18.

Cuadro 18 – Componentes de la clase *SourceNAT*

Clase	Agregación	Tipo de datos	Descripción
Node	Cero o ninguno		Información sobre el anfitrión o dispositivo que parece originar el o los eventos (dirección de la red, el nombre de la red, etc.)
User	Cero o ninguno		Información sobre el usuario que parece originar el o los eventos
Service	Cero o ninguno		Información sobre el servicio de red que participa en el o los eventos

8.2.3.6 Clase *TargetNAT*

La clase *TargetNAT* (NAT de destino) contiene información acerca del o los posibles destinos del o los eventos NAT que han generado una sesión. Un evento puede tener más de un destino transformado por el NAT.

La clase *TargetNAT* se compone de tres clases agregadas, tal como se muestra en la Figura 13.

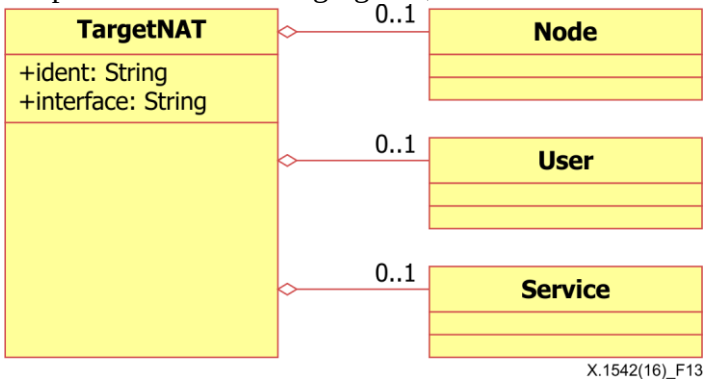

Figura 13 – Clases agregadas de la clase *TargetNAT*

La clase *TargetNAT* tiene dos atributos que figuran en el Cuadro 19.

Cuadro 19 – Atributos de la clase *TargetNAT*

Atributo	Utilización	Tipo de datos	Descripción
ident	Opcional	STRING	Identificador único para este destino transformado por el NAT
interface	Opcional	STRING	Un dispositivo de red con múltiples interfaces puede utilizarlo para indicar en qué interfaz se ha visto este destino transformado por el NAT

Las clases agregadas que integran la clase *TargetNAT* se describen en el Cuadro 20.

Cuadro 20 – Componentes de la clase *TargetNAT*

Clase	Agregación	Tipo de datos	Descripción
Node	Cero o ninguno		Información sobre el anfitrión o dispositivo hacia el que se dirigen el o los eventos (dirección de la red, el nombre de la red, etc.)
User	Cero o ninguno		Información sobre el usuario hacia el que se dirigen el o los eventos
Service	Cero o ninguno		Información sobre el servicio de red que participa en el o los eventos

8.2.3.7 Clase *AdditionalData*

La clase *AdditionalData* (datos adicionales) se utiliza para proporcionar información que el modelo de datos de SIMEF no puede representar. Esta clase puede utilizarse para facilitar datos atómicos (enteros, cadenas, etc.) en los casos en que sólo es necesario enviar pequeñas cantidades de información adicional. También pueden utilizarse para ampliar el modelo de datos y la DTD, a fin de permitir la transmisión de datos complejos (véanse los encabezamientos de paquetes).

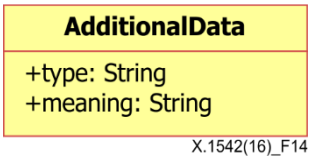

Figura 14 – Clase *AdditionalData*

La clase *AdditionalData* tiene dos atributos que figuran en el Cuadro 21.

Cuadro 21 – Atributos de la clase *AdditionalData*

Atributo	Utilización	Tipo de datos	Descripción
type	Requerida	ENUM	Tipo de datos que describe el significado del contenido del elemento Valor por defecto: string (cadena)
meaning	Opcional	STRING	Cadena que describe el significado del contenido del elemento

En el Cuadro 22 figuran los tipos de la clase *AdditionalData* así como los valores permitidos para este atributo.

Cuadro 22 – Valores del atributo tipo

Valor	Palabra clave	Definición
0	*boolean*	El elemento contiene un valor booleano, a saber, las cadenas "verdadero" o "falso"
1	*byte*	El elemento contiene un único byte de 8 bits
2	*character*	El elemento contiene un único carácter
3	*date-time*	El elemento contiene una cadena fecha-hora
4	*integer*	El elemento contiene un entero
5	*ntpstamp*	El elemento contiene un sello temporal de NTP
6	*portlist*	El elemento contiene una lista de puertos
7	*real*	El elemento contiene un número real
8	*string*	El elemento contiene una cadena
9	*Byte-string*	El elemento contiene un byte[]
10	*xml*	El elemento contiene datos con etiquetas XML

Estos valores de la clase *AdditionalData* dependen del proveedor o de su aplicación. El método para garantizar que los gestores comprendan las cadenas enviadas por los analizadores queda fuera del alcance de la presente Recomendación.

8.2.4 Clases *Support*

Las clases *Support* (soporte) integran las partes principales de las clases fundamentales y se comparten entre sí.

8.2.4.1 Clase *Node*

La clase *Node* (nodo) se utiliza para identificar anfitriones y otros dispositivos de red (encaminadores, conmutadores, etc.).

La clase *Node* se compone de tres clases agregadas, tal como se muestra en la Figura 15. Los atributos, el valor de los atributos tipo y los componentes de la clase *Node* figuran en los Cuadro 23, 24 y 25 respectivamente.

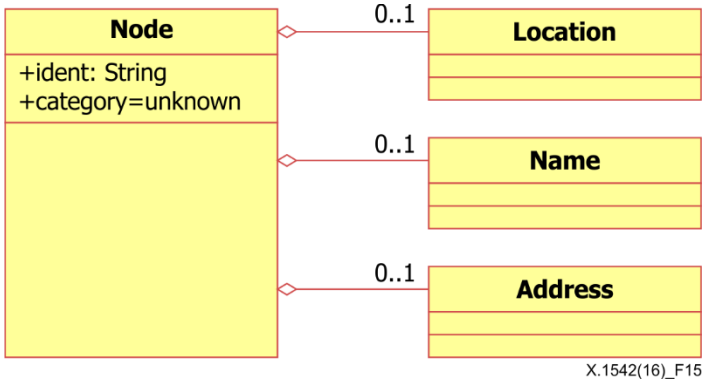

Figura 15 – Clases agregadas de la clase *Node*

Cuadro 23 – Atributos de la clase *Node*

Atributo	Utilización	Tipo de datos	Descripción
ident	Opcional	STRING	Identificador único para el nodo; véase el apartado 7.2.9
category	Opcional	ENUM	"Dominio" del que se obtiene la información sobre el nombre El valor por defecto = unknown (desconocido)

Cuadro 24 – Valores del atributo tipo

Valor	Palabra clave	Definición
0	*Unknown*	Dominio desconocido o irrelevante
1	*ads*	Servicios de directorio avanzados de Windows 2000
2	*afs*	Sistema de archivos Andrew (Transarc)
3	*coda*	Sistema de archivos distribuido Coda
4	*dfs*	Sistema de archivos distribuido (IBM)
5	*dns*	Sistema de Nombres de Dominio
6	*hosts*	Archivo de anfitrión local
7	*kerberos*	Dominio Kerberos
8	*nds*	Servicios de directorio de Novell
9	*nis*	Servicios de Información de Red (Sun)
10	*nisplus*	Servicios de Información de Red Plus (Sun)
11	*nt*	Dominio NT de windows
12	*wfw*	Windows para Grupos de Trabajo

Cuadro 25 – Componentes de la clase *Node*

Clase	Agregación	Tipo de datos	Descripción
Location	Cero o ninguno	STRING	Ubicación del equipo
Name	Cero o ninguno	STRING	Nombre del equipo; esta información se facilitará si no se proporciona información sobre la dirección
Address	Cero o ninguno		Dirección de la red o el hardware del equipo A menos que se proporcione un nombre (véase supra), se especificará al menos una dirección

8.2.4.2 Clase *Address*

La clase *Address* (dirección) se utiliza para representar direcciones de red, hardware y aplicaciones.

La clase *Address* se compone de dos clases agregadas, tal como se muestra en la Figura 16.

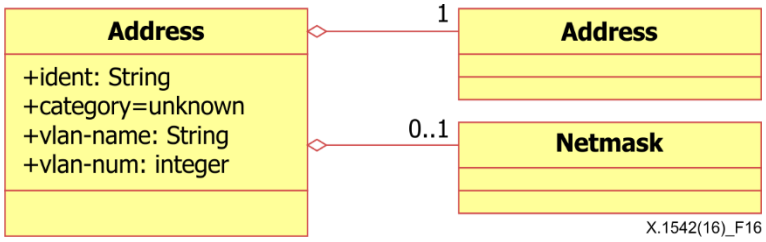

Figura 16 – Clases agregadas de la clase *Address*

Los atributos, el valor de los atributos tipo y los componentes que integran la clase *Address* se describen en los Cuadros 26, 27 y 28 respectivamente.

Cuadro 26 – Atributos de la clase *Address*

Atributo	Utilización	Tipo de datos	Descripción
ident	Opcional	STRING	Identificador único para la dirección; véase el apartado 7.2.9
category	Opcional	ENUM	Tipo de dirección representado; los valores permitidos para este atributo se muestran a continuación Valor por defecto: unknown (desconocido)
vlan-name	Opcional	STRING	Nombre de la red de área local (LAN) (LAN virtual) a la que pertenece la dirección
vlan-num	Opcional	INTEGER	Número de la LAN (LAN virtual) a la que pertenece la dirección

Cuadro 27 – Valores del atributo tipo

Valor	Palabra clave	Definición
0	*unknown*	Tipo de dirección desconocido
1	*atm*	Dirección de red en modo de transferencia asíncrono
2	*e-mail*	Dirección de correo electrónico ([b-IETF RFC 2822])
3	*lotus-notes*	Dirección de correo-e de Lotus Notes
4	*Mac*	Dirección de control de acceso a los medios (MAC)
5	*Sna*	Dirección de arquitectura de red compartida (SNA) de IBM
6	*Vm*	Dirección de correo-e de IBM VM ("PROFS")
7	*ipv4-addr*	Dirección de anfitrión IPv4 en notación decimal de puntos (a.b.c.d)
8	*ipv4-addr-hex*	Dirección de anfitrión IPv4 en notación hexadecimal
9	*ipv4-net*	Dirección de red IPv4 en notación decimal de puntos, barra oblicua y bits significativos (a.b.c.d/nn)
10	*ipv4-net-mask*	Dirección de red IPv4 en notación decimal de puntos, barra oblicua y máscara de red en notación decimal de puntos (a.b.c.d./w.x.y.z)
11	*ipv6-addr*	Dirección de anfitrión IPv6
12	*ipv6-addr-hex*	Dirección de anfitrión IPv6 en notación hexadecimal
13	*ipv6-net*	Dirección de red IPv6, barra oblicua y bits significativos
14	*ipv6-net-mask*	Dirección de red IPv6, barra oblicua y máscara de red

Cuadro 28 – Componentes de la clase *Address*

Clase	Agregación	Tipo de datos	Descripción
Address	Exactamente uno	STRING	Información sobre la dirección; el formato de estos datos se rige por el atributo category
Netmask	Cero o ninguno	STRING	Máscara de red para la dirección, si procede

8.2.4.3 Clase *User*

La clase *User* (usuario) se utiliza para describir a los usuarios y se emplea como una clase "contenedor" de la clase agregada *UserId*, como se muestra en la Figura 17.

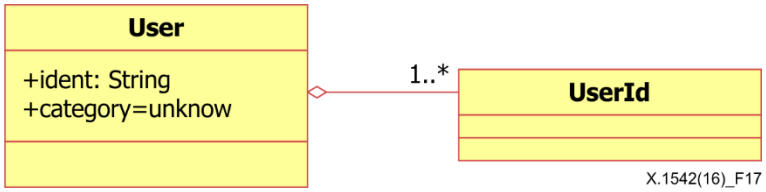

Figura 17 – Clases agregadas de la clase *User*

Los atributos, el valor de los atributos tipo y los componentes que integran la clase *Users* se describen en los Cuadros 29, 30 y 31 respectivamente.

Cuadro 29 – Atributos de la clase *User*

Atributo	Utilización	Tipo de datos	Descripción
ident	Opcional	STRING	Identificador único para el usuario; véase el apartado 7.2.9
category	Opcional	ENUM	Tipo de usuario representado; los valores permitidos para este atributo se muestran a continuación El valor por defecto = unknown (desconocido)

Cuadro 30 – Valores del atributo tipo

Valor	Palabra clave	Definición
0	*unknown*	Tipo de usuario desconocido
1	*application*	Usuario de aplicación
2	*os-device*	Usuario de dispositivo o sistema operativo

Cuadro 31 – Componentes de clase *User*

Clase	Agregación	Tipo de datos	Descripción
UserId	Uno o más		Identificación de un usuario, tal como indica su atributo type

8.2.4.3.1 Clase *UserId*

La clase *UserId* (identificador de usuario) proporciona información específica sobre un usuario. Dentro de la clase *User* pueden utilizarse diversos *UserId*, a fin de indicar intentos de transición de un usuario a otro o de proporcionar información detallada acerca de los privilegios de un usuario (o proceso).

La clase *UserId* se compone de dos clases agregadas, tal como se muestra en la Figura 18.

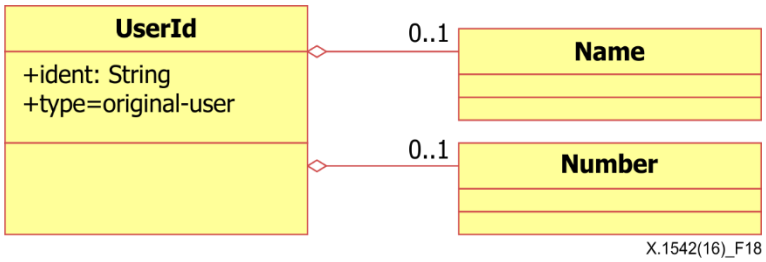

Figura 18 – Clases agregadas de la clase *UserId*

Los atributos y el valor de los atributos tipo que integran la clase *UserId* se describen en los Cuadros 32 y 33 respectivamente.

Cuadro 32 – Atributos de la clase *UserId*

Atributo	Utilización	Tipo de datos	Descripción
ident	Opcional	STRING	Identificador único para el identificador de usuario; véase el apartado 7.2.9
type	Opcional	ENUM	Tipo de información de usuario representado; los valores permitidos para este atributo se muestran a continuación Valor por defecto = original-user (original-usuario)

Cuadro 33 – Valores del atributo tipo

Valor	Palabra clave	Definición
0	*current-user*	Identificador de usuario actual que utiliza el usuario o proceso
1	*original-user*	Identidad real del usuario o proceso objeto de informe; ese valor debería utilizarse en aquellos sistemas que (a) realizan algún tipo de auditoría y (b) permiten la extracción de un identificador de usuario del testigo de "audit id" (identificador de auditoría) En los sistemas que no permiten lo antedicho y en los que se registra el propio usuario, debería utilizarse el "login id" (identificador de registro)
2	*target-user*	Identificador de usuario que el usuario o proceso trata de alcanzar; esto se aplicaría, por ejemplo, en los sistemas Unix cuando el usuario trata de utilizar "su", "rlogin", "telnet", etc.
3	*user-privs*	Otro identificador de usuario que el usuario o proceso tiene la capacidad de utilizar, o un identificador de usuario vinculado a un permiso de fichero Con objeto de especificar una lista de privilegios, pueden utilizarse múltiples elementos de UserID de este tipo
4	*current-group*	Identificador de grupo actual (si procede) que utiliza el usuario o proceso
5	*group-privs*	Otro identificador de grupo que el grupo o proceso tiene la capacidad de utilizar, o un identificador de grupo vinculado a un permiso de fichero Por ejemplo, en los sistemas Unix que derivan de la distribución de software Berkeley (BSD), podrían utilizarse múltiples elementos de UserID de este tipo para incluir todos los identificadores de grupo en la "group list" (lista de grupo)
6	*other-privs*	Permisos de fichero asignados a los usuarios que no se ajustan a los permisos de usuario o grupo con respecto al fichero; no se utilizan en contextos relacionados con usuarios, grupos o procesos, sino únicamente en aquellos referentes a los ficheros

Las clases agregadas que integran la clase *UserId* se describen en el Cuadro 34.

Cuadro 34 – Componentes de la clase *UserId*

Clase	Agregación	Tipo de datos	Descripción
Name	Cero o ninguno	STRING	Nombre de usuario o de grupo
Num	Cero o ninguno	INTEGER	Número de grupo o de usuario

8.2.4.4 Clase *Process*

La clase *Process* (proceso) se utiliza para describir los procesos que se ejecutan en los orígenes, destinos y analizadores.

La clase *Process* se compone de cinco clases agregadas, tal como se muestra en la Figura 19.

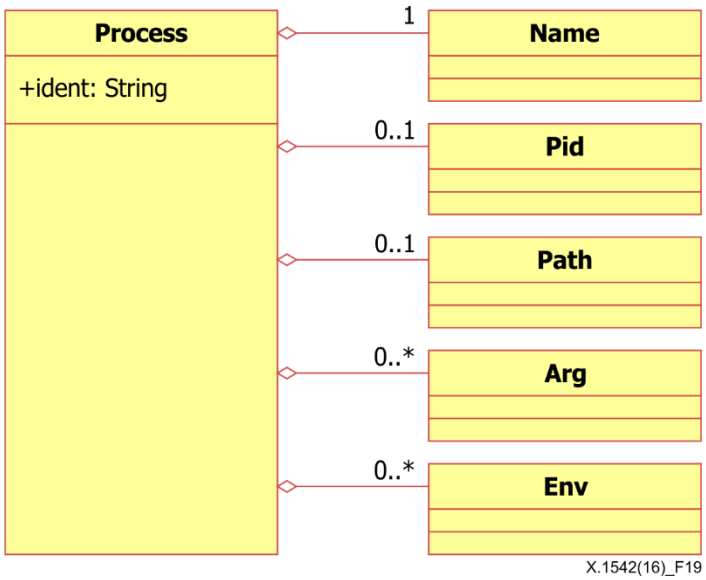

Figura 19 – Clases agregadas de la clase *Process*

La clase *Process* se compone de un atributo (véase el Cuadro 35).

Cuadro 35 – Atributo de la clase *Process*

Atributo	Utilización	Tipo de datos	Descripción
ident	Opcional	STRING	Identificador único para el proceso; véase el apartado 7.2.9

Las clases agregadas que integran la clase *Process* se describen en el Cuadro 36.

Cuadro 36 – Componentes de la clase *Process*

Clase	Agregación	Tipo de datos	Descripción
Name	Exactamente uno	STRING	Nombre del programa en ejecución
Pid	Cero o ninguno	INTEGER	Identificador de proceso del proceso
Path	Cero o ninguno	STRING	Ruta completa del programa en ejecución
Arg	Cero o ninguno	STRING	Argumento de línea de instrucciones para el programa
Env	Cero o ninguno	STRING	Cadena de entorno vinculada al proceso; generalmente del formato "VARIABLE=value"

En la clase *Process*, la clase *Name* es un nombre corto y pueden especificarse múltiples argumentos con múltiples usos de la clase *Arg*. También pueden especificarse múltiples cadenas de entorno con múltiples usos de la clase *Env*.

8.2.4.5 Clase *Service*

La clase *Service* (servicio) describe los servicios de red en orígenes y destinos, y puede identificar servicios por nombre, puerto, lista de puertos y protocolo. Cuando el servicio tiene lugar como clase agregada de *Source*, se entiende que su actividad de interés es originaria y que el propio servicio se halla "anexado" a la información relativa al nodo, el proceso y/o el usuario que también engloba la

clase *Source*. Del mismo modo, cuando el servicio tiene lugar como clase agregada de *Target*, se entiende que su actividad de interés es direccional y que el propio servicio se halla "anexado" a la información relativa al nodo, el proceso y/o el usuario que también engloba la clase *Target*. Si el servicio tiene lugar tanto en *Source* como en *Target*, la información presente en ambas ubicaciones debería ser la misma. Si la información es la misma en ambas ubicaciones y los implementadores desean limitarla a una de ellas, deben especificarla como un agregado de la clase *Target*.

La clase *Service* se compone de cuatro clases agregadas, tal como se muestra en la Figura 20.

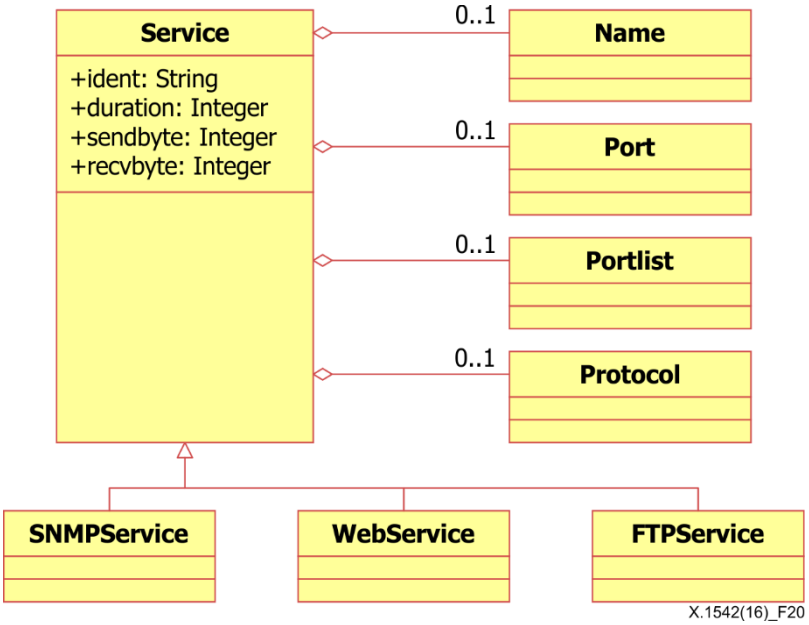

Figura 20 – Clases agregadas de la clase *Service*

La clase *Service* se compone de los cuatro atributos que figuran en el Cuadro 37.

Cuadro 37 – Atributos de la clase *Service*

Atributo	Utilización	Tipo de datos	Descripción
ident	Opcional	STRING	Identificador único para el servicio; véase el apartado 7.2.9
duration	Opcional	INTEGER	Tiempo para la conexión
sendbyte	Opcional	INTEGER	Tamaño de byte enviado después de la conexión
recvByte	Opcional	INTEGER	Tamaño de byte recibido después de la conexión

Las clases agregadas que integran la clase *Service* se describen en el Cuadro 38.

Cuadro 38 – Componentes de la clase *Service*

Clase	Agregación	Tipo de datos	Descripción
Name	Cero o ninguno	STRING	Nombre del servicio; siempre que sea posible, debe utilizarse el nombre de la lista puertos conocidos de la Autoridad de asignación de números Internet (IANA)
Port	Cero o ninguno	INTEGER	Número de puerto en uso
Portlist	Cero o ninguno	PORTLIST	Lista de los números de puerto en uso; para obtener información sobre las normas de formateado, véase el apartado 7.2.8
Protocol	Cero o ninguno	STRING	Información adicional sobre el protocolo en uso

8.2.4.5.1 Clase *WebService*

La clase *WebService* (servicio web) contiene información adicional relacionada con el tráfico web.

La clase *WebService* se compone de cuatro clases agregadas, tal como se muestra en la Figura 21.

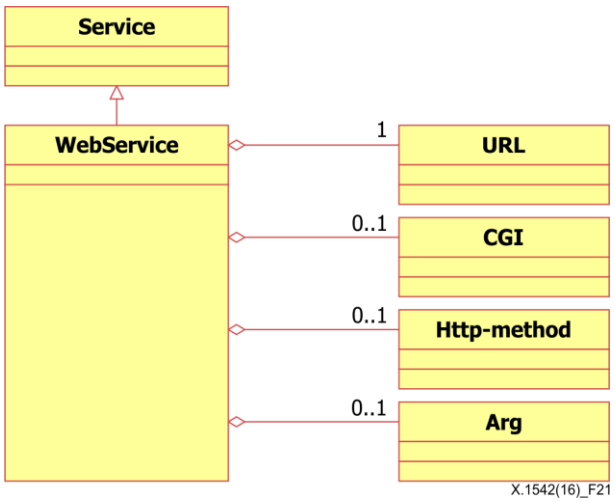

Figura 21 – Clases agregadas de la clase *WebService*

Las clases agregadas que integran la clase *WebService* se describen en el Cuadro 39.

Cuadro 39 – Componentes de la clase *WebService*

Clase	Agregación	Tipo de datos	Descripción
URL	Exactamente uno	STRING	Localizador uniforme de recursos (URL) utilizado en la solicitud
CGI	Cero o ninguno	STRING	Guion de interfaz común de pasarela (CGI) utilizado en la solicitud, sin argumentos
Http-method	Cero o ninguno	STRING	Método del protocolo de transferencia de hipertexto (HTTP) (PUT, GET) utilizado en la solicitud
Arg	Cero o ninguno	STRING	Argumentos del guion de CGI

8.2.4.5.2 Clase *SNMPService*

La clase *SNMPService* (servicio SNMP) contiene información adicional relacionada con el tráfico del protocolo simple de gestión de red (SNMP).

La clase *SNMPService* se compone de ocho clases agregadas, tal como se muestra en la Figura 22.

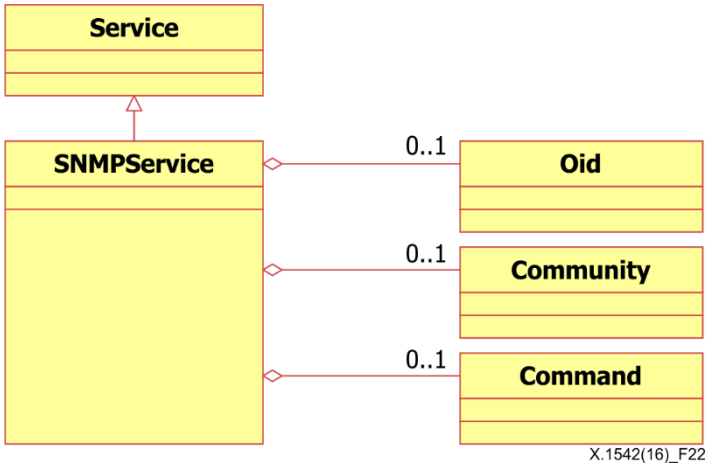

Figura 22 – Clases agregadas de la clase *SNMPService*

Las clases agregadas que integran la clase *SNMPService* se describen en el Cuadro 40.

Cuadro 40 – Componentes de la clase *SNMPService*

Clase	Agregación	Tipo de datos	Descripción
Oid	Cero o ninguno	STRING	Identificador de objeto utilizado en la solicitud
Community	Cero o ninguno	STRING	Cadena comunitaria del objeto
Command	Cero o ninguno	STRING	Instrucción enviada al servidor SNMP (GET, SET, etc.)

8.2.4.5.3 Clase *FTPService*

La clase *FTPService* (servicio FTP) contiene información adicional relacionada con el tráfico del protocolo de transferencia de ficheros (FTP).

La clase *FTPService* se compone de dos clases agregadas, tal como se muestra en la Figura 23.

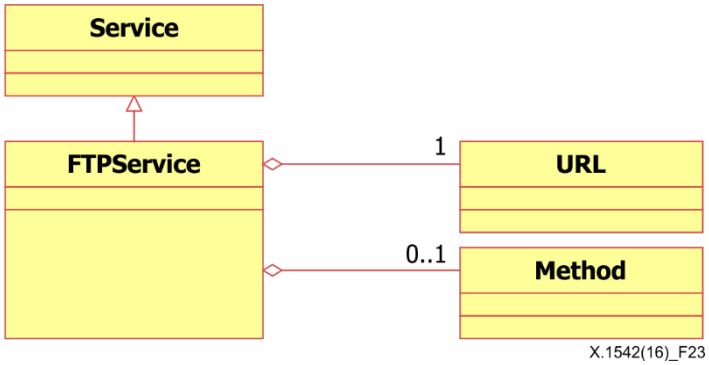

Figura 23 – Clases agregadas de la clase *FTPService*

Las clases agregadas que integran la clase *FTPService* se describen en el Cuadro 41.

Cuadro 41 – Componentes de la clase *FTPService*

Clase	Agregación	Tipo de datos	Descripción
URL	Exactamente uno	STRING	URL utilizado en la solicitud
Method	Cero o ninguno	STRING	Método de FTP (PUT, GET) utilizado en la solicitud

9 Consideraciones relativas a la seguridad

En este apartado se describen ciertas consideraciones especiales de seguridad que los implementadores de SIMEF deberán tener en cuenta.

En la presente Recomendación se describe el modelo de información del formato de intercambio de mensajes sobre información de sesión (SIMEF) y se facilita un modelo de datos conexo especificado con un esquema XML. El SIMEF define una representación de modelo de datos para compartir la información de registro de sesión de la capa de transporte vinculada a la gestión centralizada de la seguridad de la red y el sistema de intercambio de información de seguridad.

Si bien el formato de estos datos no suscita problemas de seguridad específicos, los datos en sí pueden contener información delicada desde el punto de vista de la seguridad, cuya confidencialidad, integridad o disponibilidad podría requerir protección.

En esta Recomendación se sugiere que los sistemas utilizados para recopilar, transmitir, procesar y almacenar estos datos se protejan contra el uso y el acceso no autorizados, respectivamente. Los medios encaminados a la provisión de dicha protección quedan fuera del alcance de la presente Recomendación.

Apéndice I

Esquemas y ejemplos de SIMEF

(Este Apéndice no forma parte integrante de la presente Recomendación.)

El presente Apéndice contiene un ejemplo de esquema XML para el modelo de SIMEF. A continuación se facilitan sendos ejemplos de un esquema XML y un esquema SYSLOG para codificar la información de sesión en el modelo de SIMEF.

I.1 Esquema de SIMEF

I.1.1 Esquema XML

```
<?xml version="1.0" encoding="UTF-8"?>
<simef:SIMEF-Message version="1.2" xmlns:simef=http://iana.org/simef/>
     <Connect ident="1008380" criticality="normal">
        <Device Deviceid="TTA-FW" model="FW1000">
           <Node>
              <Address category="ipv4-addr">
                <address>1.1.1.1</address>
              </Address>
           </Node>
        </Device>
        <CreateTime ntpstamp="0xaaaaaaaaaaaaaaaaaa">
                  2010-08-18T15:41:28+00:00
           </CreateTime>
        <Policy Ruleid="45" action="pass"></Policy>
           <Source>
           <Node>
              <Address category="ipv4-addr">
                <address>2.2.2.2</address>
              </Address>
           </Node>
           <Service duration="9" size="144">
              <port>38168</port>
              <protocol>17</protocol>
           </Service>
        </Source>
        <Target>
           <Node>
              <Address category="ipv4-addr">
                <address>3.3.3.3</address>
              </Address>
           </Node>
           <Service duration="9" size="0">
              <name>dns</name>
              <port>53</port>
              <protocol>17</protocol>
           </Service>
        </Target>
        <Classification origin="vendor-specific">
           <name>45</name>
        </Classification>
     </Connect>
</simef:SIMEF-Message >
```

I.1.2 Esquema SYSLOG

```
  2014-03-18  15:41:28  Local0.Notice  1.1.1.1  TTA:  TTA-FW  device_id=  TTA
[Root]system-notification-00257(traffic):    start_time="2014-03-18    15:41:19"
duration=9 policy_id=45 service=dns proto=17 src zone=Untrust dst zone=Trust
action=Permit sent=144 rcvd=0 src=2.2.2.2 dst=3.3.3.3 src_port=38168 dst_port=53
src-xlated ip=2.2.2.2 port=38168 dst-xlated ip=3.3.3.3 port=53 session_id=1008380
reason=Close - AGE OUT<000>
```

I.2 Ejemplos de SIMEF

I.2.1 Permiso de cortafuego

```
<?xml version="1.0" encoding="UTF-8"?>
<SIMEF-Message version=1.2>
       <Connect ident="1008380" criticality="1">
              <Device Deviceid="TTA-FW" manufacturer="TTA" model="FW1000">
                     <Node>
                            <Address category="ipv4-addr">
                                   <address>1.1.1.1</address>
                            </Address>
                     </Node>
              </Device>
              <Policy Ruleid="45" action="1"></Policy>
              <CreateTime ntpstamp="0xaaaaaaaaaaaaaaaa">
                     2014-03-18T15:41:28+00:00
              </CreateTime>
              <Source>
                     <Node>
                            <Address category="ipv4-addr">
                                   <address>2.2.2.2</address>
                            </Address>
                     </Node>
                        <Service duration="9" size="144">
                            <port>38168</port>
                            <protocol>17</protocol>
                     </Service>
              </Source>
              <Target>
                     <Node>
                            <Address category="ipv4-addr">
                                   <address>3.3.3.3</address>
                            </Address>
                     </Node>
                     <Service duration="9" size="0">
                            <name>dns</name>
                            <port>53</port>
                            <protocol>17</protocol>
                     </Service>
              </Target>
              <Classification origin="2">
                     <name>45</name>
              </Classification>
       </Connect>
</SIMEF-Message>
```

I.2.2 Registro de VPN

```
<?xml version="1.0" encoding="UTF-8"?>
<SIMEF-Message version=1.2>
       <Connect ident="1008057" criticality="1">
              <Device Deviceid="TTA-VPN" manufacturer="TTA" model="VPN1000">
                     <Node>
                            <Address category="ipv4-addr">
                                   <address>1.1.1.1</address>
```

```
                        </Address>
                    </Node>
                </Device>
                <Policy ruleid="700" action="3"></Ploicy>
                <CreateTime ntpstamp="0xaaaaaaaaaaaaaaaaaaaa">
                    2014-03-19T12:51:22+00:00
                </CreateTime>
                <Source>
                    <Node>
                        <Address category="ipv4-addr">
                            <address>2.2.2.2</address>
                        </Address>
                    </Node>
                    <Service duration="41" size="16905">
                        <port>59078</port>
                        <protocol>TCP</protocol>
                    </Service>
                </Source>
                <Target>
                    <Node>
                        <Address category="ipv4-addr">
                            <address>3.3.3.3</address>
                        </Address>
                    </Node>
                    <Service duration="41" size="1448">
                        <name>junos-http</name>
                        <port>80</port>
                        <protocol>TCP</protocol>
                    </Service>
                </Target>
                <Classification origin="2">
                    <name>700</name>
                </Classification>
        </Connect>
</SIMEF-Message>
```

I.2.3 Registro de NAT

```
<?xml version="1.0" encoding="UTF-8"?>
<SIMEF-Message version=1.2>
        <Connect ident="1009632" criticality="1">
                <Device Deviceid="TTA-FW" manufacturer="TTA" model="FW1000">
                <Node>
                        <Address category="ipv4-addr">
                            <address>1.1.1.1</address>
                        </Address>
                    </Node>
                </Device>
                <Policy ruleid="57" action="1"></Policy>
                <CreateTime ntpstamp="0xaaaaaaaaaaaaaaaaaaaa">
                    2014-03-19T16:21:12+00:02
                </CreateTime>
                <Source>
                    <Node>
                        <Address ident="" category="ipv4-addr”>
                            <address>2.2.2.2</address>
                        </Address>
                    </Node>
                    <Service duration="41" size="16905">
                        <port>59078</port>
                        <protocol>TCP</protocol>
                    </Service>
                </Source>
                <Target>
```

```
                <Node>
                        <Address ident="" category="ipv4-addr">
                                <address>3.3.3.3</address>
                        </Address>
                </Node>
                <Service duration="41" size="1448">
                        <name>junos-http</name>
                        <port>80</port>
                        <protocol>TCP</protocol>
                </Service>
        </Target>
        <SourceNat>
                <Node>
                        <name>trust</name>
                        <Address category="ipv4-addr">
                                <address>4.4.4.4</address>
                        </Address>
                </Node>
                <Service>
                        <port>59078</port>
                </Service>
        </SourceNat>
        <TargetNat>
                <Node>
                        <Address category="ipv4-addr">
                                <address>5.5.5.5</address>
                        </Address>
                </Node>
                <Service>
                        <port>80</port>
                </Service>
        </TargetNat>
    </Connect>
</SIMEF-Message>
```

Bibliografía

[b-ISO 8601:2004] ISO 8601:2004, *Data elements and interchange formats – Information interchange – Representation of dates and times.*

[b-ISO/CEI 10646] ISO/CEI 10646:2012, *Information technology – Universal Coded Character Set (UCS).*

[b-IEEE Std 1003.1] IEEE Std 1003.1-2008 – *IEEE Standard for Information Technology – Portable Operating System Interface (POSIX(R)).*

[b-IETF RFC 1305] IETF RFC 1305 (1992), *Network Time Protocol (Version 3) Specification, Implementation*.

[b-IETF RFC 2663] IETF RFC 2663 (1999), *IP Network Address Translator (NAT) Terminology and Considerations*.

[b-IETF RFC 2822] IETF RFC 2822 (2001), *Internet Message Format*.

[b-IETF RFC 5905] IETF RFC 5905 (2010), *Network Time Protocol Version 4: Protocol and Algorithms Specification*.

SERIES DE RECOMENDACIONES DEL UIT-T

Serie A	Organización del trabajo del UIT-T
Serie D	Principios de tarificación y contabilidad y cuestiones económicas y políticas de las telecomunicaciones/TIC internacionales
Serie E	Explotación general de la red, servicio telefónico, explotación del servicio y factores humanos
Serie F	Servicios de telecomunicación no telefónicos
Serie G	Sistemas y medios de transmisión, sistemas y redes digitales
Serie H	Sistemas audiovisuales y multimedia
Serie I	Red digital de servicios integrados
Serie J	Redes de cable y transmisión de programas radiofónicos y televisivos, y de otras señales multimedia
Serie K	Protección contra las interferencias
Serie L	Medio ambiente y TIC, cambio climático, ciberdesechos, eficiencia energética, construcción, instalación y protección de los cables y de otros elementos de planta exterior
Serie M	Gestión de las telecomunicaciones, incluida la RGT y el mantenimiento de redes
Serie N	Mantenimiento: circuitos internacionales para transmisiones radiofónicas y de televisión
Serie O	Especificaciones de los aparatos de medida
Serie P	Terminales y métodos de evaluación subjetivos y objetivos
Serie Q	Conmutación y señalización
Serie R	Transmisión telegráfica
Serie S	Equipos terminales para servicios de telegrafía
Serie T	Terminales para servicios de telemática
Serie U	Conmutación telegráfica
Serie V	Comunicación de datos por la red telefónica
Serie X	**Redes de datos, comunicaciones de sistemas abiertos y seguridad**
Serie Y	Infraestructura mundial de la información, aspectos del protocolo Internet, redes de próxima generación, Internet de las cosas y ciudades inteligentes
Serie Z	Lenguajes y aspectos generales de soporte lógico para los sistemas de telecomunicación

Impreso en Suiza
Ginebra, 2017