International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.1542

(09/2016)

SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

Cybersecurity information exchange –
Event/incident/heuristics exchange

Session information message exchange format

Recommendation ITU-T X.1542

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

PUBLIC DATA NETWORKS	X.1–X.199
OPEN SYSTEMS INTERCONNECTION	X.200–X.299
INTERWORKING BETWEEN NETWORKS	X.300–X.399
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	X.600–X.699
OSI MANAGEMENT	X.700–X.799
SECURITY	X.800–X.849
OSI APPLICATIONS	X.850–X.899
OPEN DISTRIBUTED PROCESSING	X.900–X.999
INFORMATION AND NETWORK SECURITY	
General security aspects	X.1000–X.1029
Network security	X.1030–X.1049
Security management	X.1050–X.1069
Telebiometrics	X.1080–X.1099
SECURE APPLICATIONS AND SERVICES	
Multicast security	X.1100–X.1109
Home network security	X.1110–X.1119
Mobile security	X.1120–X.1139
Web security	X.1140–X.1149
Security protocols	X.1150–X.1159
Peer-to-peer security	X.1160–X.1169
Networked ID security	X.1170–X.1179
IPTV security	X.1180–X.1199
CYBERSPACE SECURITY	
Cybersecurity	X.1200–X.1229
Countering spam	X.1230–X.1249
Identity management	X.1250–X.1279
SECURE APPLICATIONS AND SERVICES	
Emergency communications	X.1300–X.1309
Ubiquitous sensor network security	X.1310–X.1339
PKI related Recommendations	X.1340–X.1349
CYBERSECURITY INFORMATION EXCHANGE	
Overview of cybersecurity	X.1500–X.1519
Vulnerability/state exchange	X.1520–X.1539
Event/incident/heuristics exchange	**X.1540–X.1549**
Exchange of policies	X.1550–X.1559
Heuristics and information request	X.1560–X.1569
Identification and discovery	X.1570–X.1579
Assured exchange	X.1580–X.1589
CLOUD COMPUTING SECURITY	
Overview of cloud computing security	X.1600–X.1601
Cloud computing security design	X.1602–X.1639
Cloud computing security best practices and guidelines	X.1640–X.1659
Cloud computing security implementation	X.1660–X.1679
Other cloud computing security	X.1680–X.1699

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T X.1542

Session information message exchange format

Summary

In today's environment, computer networks are vulnerable to threats from both inside and outside an organization. Firewall systems log session information about selected incoming and outgoing transmission control protocol/Internet protocol (TCP/IP) connections.

However, those systems that are currently available are not generally interoperable because each system has its own special functionality, control mechanisms and session log formats.

The need most security administrators face today is the maintenance of a consistent session information exchange format across diverse firewall systems and even varied infrastructures.

Recommendation ITU-T X.1542 describes an information model for the session information message exchange format (SIMEF) and provides an associated data model specified with an extensible markup language (XML) schema. The SIMEF defines a data model representation for sharing transport layer session log information about centralized network security management and the security information exchange system. The specification of any transport protocol is beyond the scope of this Recommendation.

History

Edition	Recommendation	Approval	Study Group	Unique ID[*]
1.0	ITU-T X.1542	2016-09-07	17	11.1002/1000/12852

Keywords

Data model, message exchange, network security, session information.

[*] To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Recommendation ITU-T X.1542

Session information message exchange format

1 Scope

This Recommendation describes the session information message exchange format (SIMEF), a data model to represent session information exported by security systems such as firewalls, and explains the rationale for using this model. An implementation of the data model in the extensible markup language (XML) is presented, an XML document type definition (DTD) is developed, and examples are provided.

2 References

None.

3 Definitions

3.1 Terms defined elsewhere

None.

3.2 Terms defined in this Recommendation

This Recommendation defines the following term:

3.2.1 analyser: A network security system that detects attacks by analysing incoming and outgoing session information. It also generates session log and sends to the security management systems.

3.2.2 session information: Information containing the transmission control protocol/user datagram protocol (TCP/UDP) session, application service and session entities as viewed by session information providers. A session is defined as the set of traffic that is managed as a unit for translation. TCP/UDP sessions are uniquely identified by the tuple of (source IP address, source TCP/UDP port, target IP address, target TCP/UDP port).

NOTE – This definition is based on [b-IETF RFC 2663].

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

BSD	Berkeley Software Distribution
CGI	Common Gateway Interface
DTD	Document Type Definition
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
LAN	Local Area Network
MAC	Media Access Control
NAT	Network Address Translation
NTP	Network Time Protocol
POSIX	Portable Operating System Interface

SIMEF	Session Information Message Exchange Format
SNA	Shared Network Architecture
SNMP	Simple Network Management Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UML	Unified Modelling Language
URL	Uniform Resource Locator
UTF	Universal character set Transformation Format
VPN	Virtual Private Network
XML	extensible Markup Language

5 Conventions

UNIX ® is a registered trademark of The Open Group.

POSIX ® is a registered trademark of the IEEE.

6 Overview

In today's network environment, computer networks are vulnerable to threats from both inside and outside an organization. Therefore, most network security research has been devoted to the development of integrated network security management systems and network monitoring utilities that allow an organization to capture TCP/IP packets that pass through its network devices, and view the captured data as sequences of conversations between clients and servers. For example, firewall systems log session information about selected incoming and outgoing TCP/IP connections.

The concept of SIMEF is shown in Figure 1. The session information can be collected from firewall systems, network address translation (NAT) devices, and so on. SIMEF specifies the data model that covers client/server network connection, end user device and application service. The SIMEF defines a data model and related message classes for sharing the transport layer session information of interest to security management systems and information sharing systems. It can be applied to the intrusion information exchange system.

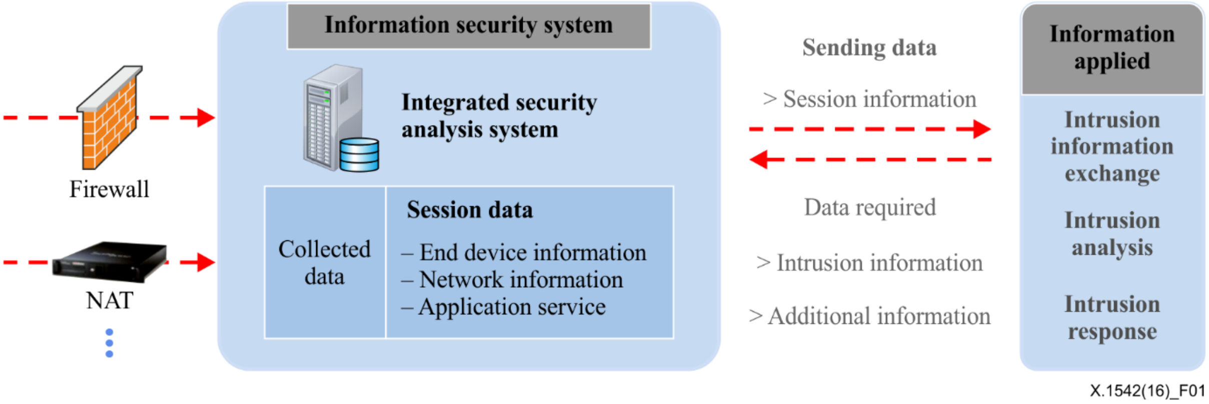

Figure 1 – The concept of SIMEF

7 Representation and definition

This Recommendation uses three notations: unified modelling language (UML) to describe the data model, XML to describe the markup used in SIMEF documents and SIMEF markup to represent the documents themselves.

7.1 SIMEF XML document

This clause describes SIMEF XML document formatting rules. Most of these rules are "inherited" from those for formatting XML documents. The format of an SIMEF XML document prolog is described in clauses 7.1.1 to 7.1.2.

7.1.1 XML declaration

SIMEF documents being exchanged between SIMEF-compliant applications shall begin with an XML declaration and shall specify the XML version in use. Specification of the encoding in use is recommended.

An SIMEF message should therefore start with:

```
<?xml version="1.0" encoding="UTF-8"?>

<simef: SIMEF-Message version="1.2" xmlns:simef="http://iana.org/simef"/>
```

SIMEF-compliant applications may choose to omit the XML declaration internally to conserve space, adding it only when the message is sent to another destination (e.g., a web browser). This practice is not recommended unless it can be accomplished without loss of each message's version and encoding information.

Implementers may decide, therefore, to have analysers and managers agree out-of-band on the particular document type definition (DTD) they will be using to exchange messages (the standard one as defined here or one with extensions), and then omit the DTD from SIMEF messages. The method for negotiating this agreement is outside the scope of this Recommendation.

7.1.2 Character data processing in SIMEF

For portability reasons, SIMEF-compliant applications should not use, and SIMEF messages should not be encoded in, character encodings other than UTF-8 and UTF-16. Consistent with the XML standard, if no encoding is specified for an SIMEF message, UTF-8 is assumed.

7.1.2.1 Character entity references

It is recommended that SIMEF-compliant applications use the entity reference form of the characters '&','<', '>', '"', and ''' (single quote) whenever writing these characters in data, to avoid any possibility of misinterpretation.

7.1.2.2 White space processing

All SIMEF elements shall support the "xml:space" attribute.

7.1.2.3 Languages in SIMEF

SIMEF-compliant applications shall specify the language in which their contents are encoded; in general, this can be done by specifying the "xml:lang" attribute for the top-level element and letting all other elements "inherit" that definition.

7.2 SIMEF data types

Within an XML SIMEF message, all data shall be expressed as text, since XML is a text-formatting language. It provides typing information for the attributes of the classes in the data model. Each data type in the model has specific formatting requirements in an XML SIMEF message; these requirements are set forth in this clause.

7.2.1 Integers

Integer attributes are represented by the INTEGER data type. Integer data shall be encoded in Base 10 or Base 16. Base 10 integer encoding uses the digits '0' to '9' and an optional sign ('+' or '−'). For example, "123", "−456". Base 16 integer encoding uses the digits '0' to '9' and 'a' to 'f' (or their uppercase equivalents), and is preceded by the characters "0x". For example, "0x1a2b".

7.2.2 Real numbers

Real (floating-point) attributes are represented by the REAL data type. Real data shall be encoded in Base 10. Real encoding is that of the Portable Operating System Interface (POSIX) 1003.1 [b-IEEE 1003.1] "strtod" library function: an optional sign ('+' or '−') followed by a non-empty string of decimal digits, optionally containing a radix character, then an optional exponent part. An exponent part consists of an 'e' or 'E', followed by an optional sign, followed by one or more decimal digits. For example, "123.45e02", "−567, 89e−03". SIMEF-compliant applications shall support both the '.' and ',' radix characters.

7.2.3 Characters and strings

Single character attributes are represented by the CHARACTER data type. Multi-character attributes of known length are represented by the STRING data type. Character and string data have no special formatting requirements, other than the need to occasionally use character references to represent special characters.

7.2.3.1 Character entity references

Within XML documents, certain characters have special meanings in some contexts. To include the actual character itself in one of these contexts, a special escape sequence, called an entity reference, shall be used.

The characters that sometimes need to be escaped and their entity referencesare:

Character	Entity reference
&	&amp;
<	&lt;
>	&gt;
"	&quot;
'	&apos;

7.2.3.2 Character code references

Any character defined by the [b-ISO/IEC 10646] and Unicode standards may be included in an XML document by the use of a character reference. A character reference is started with the characters '&' and '#', and ended with the character ';'. Between these characters, the character code for the character is inserted.

If the character code is preceded by an 'x' it is interpreted in hexadecimal (base 16); otherwise, it is interpreted in decimal (base 10). For instance, the ampersand (&) is encoded as &#38; or &#x0026; and the less-than sign (<) is encoded as &#60; or &#x003C;. Any 1, 2, or 4 byte character specified in ISO/IEC 10646 and Unicode standards can be included in a document using this technique.

7.2.4 Bytes

Binary data is represented by the BYTE (and BYTE[]) data type. Binary data shall be encoded in its entirety using base64.

7.2.5 Enumerated types

Enumerated types are represented by the ENUM data type, and consist of an ordered list of acceptable values.

7.2.6 Date-time strings

Date-time strings are represented by the DATETIME data type. Each date-time string identifies a particular instant in time; ranges are not supported. Date-time strings are formatted according to a subset of [b-ISO 8601:2004], as show below. Section references in parentheses refer to clauses of [b-ISO 8601:2004].

7.2.7 NTP timestamps

Network time protocol (NTP) timestamps are represented by the NTPSTAMP data type and are described in detail in [b-IETF RFC 1305] and [b-IETF RFC 5905]. An NTP timestamp is a 64-bit unsigned fixed-point number. The integer part is in the first 32 bits, and the fraction part is in the second 32 bits. Within SIMEF messages, NTP timestamps shall be encoded as two 32 bit hexadecimal values, separated by a period ('.'). For example, "0x12345678.0x87654321".

7.2.8 Port lists

Port lists are represented by the PORTLIST data type and consist of a comma-separated list of numbers (individual integers) and ranges (N–M means ports N to M, inclusive). Any combination of numbers and ranges may be used in a single list. For example,

"5-25,37,42,43,53,69–119,123-514".

7.2.9 Unique identifiers

There are two types of unique identifier used in this Recommendation. Both are represented by STRING data types. These identifiers are implemented as attributes on the relevant XML elements, and they shall have unique values as follows.

1 The Device class' (clause 8.2.3.2) "deviceid" attribute, if specified, shall have a value that is unique across all analysers in the intrusion detection environment.

2 The default value is "0", which indicates that the analyser cannot generate unique identifiers.

The "ident" attribute, if specified, of several classes shall have a value that is unique across all messages sent by the individual analyser. The "ident" attribute value shall be unique for each particular combination of data identifying an object, not for each object. Objects may have more than one "ident" value associated with them. For example, an identification of a host by name would have one value, while an identification of that host by address would have a second, and an identification of that host by both name and address would have yet another.

The default value is "0", which indicates that the analyser cannot generate unique identifiers.

The specification of methods for creating the unique values contained in these attributes is outside the scope of this Recommendation.

8 The SIMEF data model

In this clause, the individual components of the SIMEF data model are explained in detail. UML diagrams of the model are provided to show how the components are related to each other.

8.1 Data model overview

The relationship between the principal components of the data model is shown in Figure 2. The top-level class is SIMEF-Message; each type of message is a subclass of this top-level class. There are two types of messages defined: Connects and Heartbeats. Within each message, subclasses of the message class are used to provide the detailed information carried in the message. The Connect message class has several subclasses such as devices, policy, source, target, and additional data.

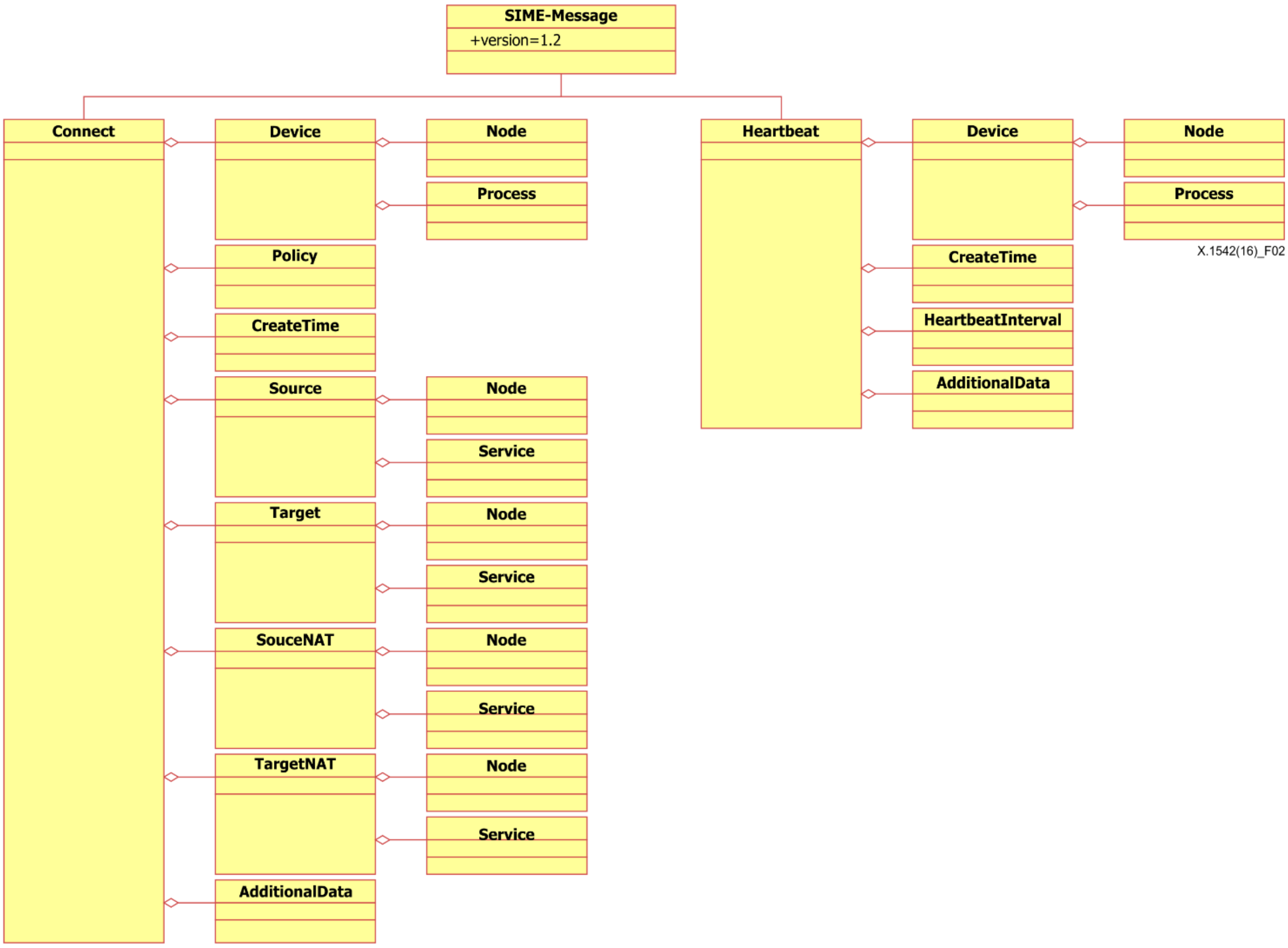

Figure 2 – The SIMEF data model

8.1.1 SIMEF classes

All SIMEF messages are instances of the SIMEF-Message class; Connect and Heartbeat. The individual classes are described in this clause. See Figure 3, Table 1 and Table 2.

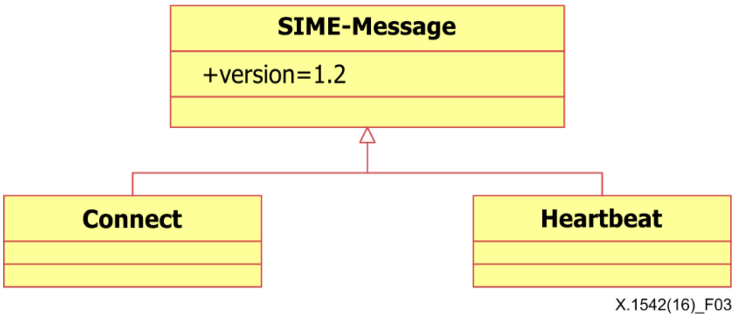

Figure 3 – The top-level class of the SIMEF data model

Table 1 – Attributes of SIMEF classes

Attribute	Usage	Data type	Description
Version	Required	STRING	SIMEF Version Information, Default Value: 1.2

Table 2 – Components of SIMEF classes

Classes	Aggregation	Data type	Description
Connect	Exactly one		Session Information Class
Heartbeat	Zero or one		System Status Information Class, Optional Providing

8.2 The message classes

The individual classes are described in clauses 8.2.1 to 8.2.4.

8.2.1 Connect class

The connect class is for containing the session information. It expresses the type of log generated by the connection in the firewall, it also shows all information about the connection attempts to the interior as well as the outside. See Table 3. The value allowed for the criticality attributes with the Connect class is shown in Table 4. A connect class is composed of several aggregate classes, as shown in Figure 4. The aggregate classes themselves are described in Table 5.

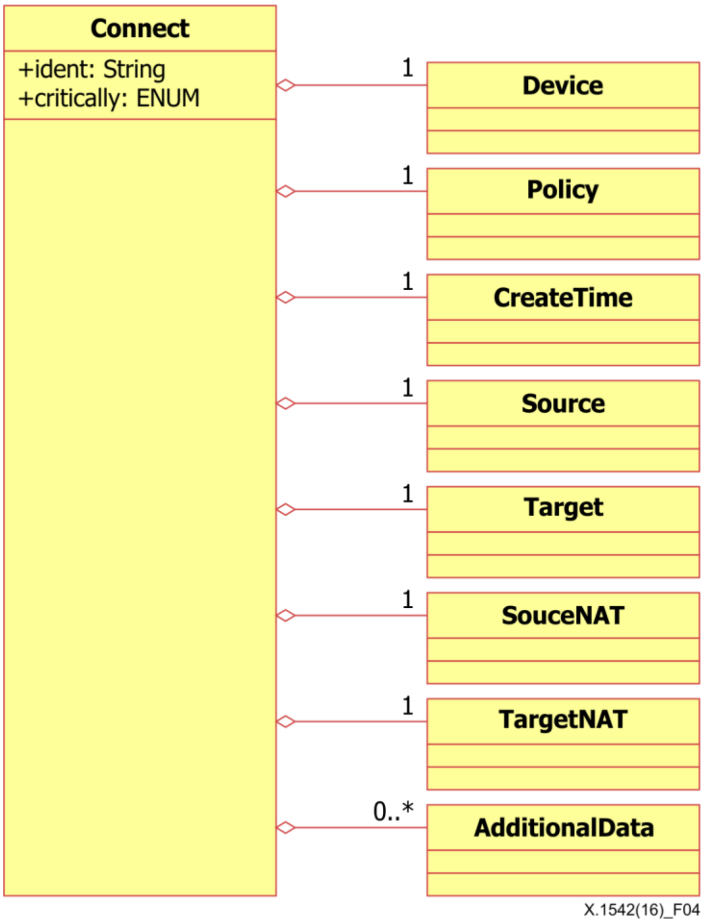

Figure 4 – The aggregate classes of the Connect class

Table 3 – Attributes of the Connect class

Attribute	Usage	Data type	Description
ident	Optional	STRING	Unique ID for the access information
criticality	Optional	ENUM	Classification according to the assessment of the event that is generated by the connection, Default Value: Unknown

Table 4 – Value of criticality attribute

Value	Keyword	Definition
0	unknown	When the effect of the event is unknown, or cannot be determined
1	normal	If the normal connection
2	suspicious	If the suspicious connection
3	warning	If the connection can be an alarm
4	critical	If the connection is sensitive to the action

Table 5 – Components of the Connect class

Class	Aggregation	Data type	Description
Device	Exactly one		Information of the analyser generating a log
Policy	Exactly one		Information carried in the analyser for the connection
CreateTime	Exactly one	DATETIME	Time for the log creation
Source	Exactly one		The source for the event causing the connection
Target	Exactly one		Destination information of an event that causes a connection
SourceNAT	Exactly one		NAT source information in the event that causes the connection
TargetNAT	Exactly one		NAT destination information of an event that causes a connection
AdditionalData	Zero or more		Additional information generated by the detector that is not in the other class

8.2.1.1 Policy class

The Policy class provides the action information to indicate is how to deal with a session in the analyser. See Figure 5.

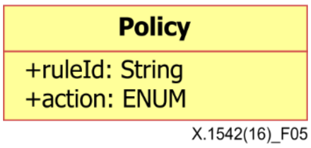

Figure 5 – The policy class

The values allowed for action attributes of the Policy class (see Table 6) are listed in Table 7.

Table 6 – Attributes of Policy class

Attribute	Usage	Data type	Description
ruleId	Optional	STRING	The unique identifier of the firewall policy to be generated by the connection
action	Optional	ENUM	Classification according to the firewall of the operation caused by the connection, Default Value: Unknown

Table 7 – Value of action attribute

Value	Keyword	Definition
0	unknown	If the unknown behaviour
1	pass	If allowing the connection
2	block	If denying the connection
3	protect	If encrypting the packet transmitted or inserting an integrity check code [virtual private network (VPN) log]
4	reject	If rejecting the connection. However, providing the error messages when the access denied

8.2.2 Heartbeat class

Analysers use Heartbeat messages to indicate their current status to managers. Heartbeats are intended to be sent in a regular period, say, every 10 min or every hour. The receipt of a Heartbeat message from an analyser indicates to the manager that the analyser is up and running; lack of a Heartbeat message (or more likely, lack of some number of consecutive Heartbeat messages) indicates that the analyser or its network connection has failed.

All managers shall support the receipt of Heartbeat messages; however, the use of these messages by analysers is optional. Developers of manager software should permit the software to be configured on a per-analyser basis to use/not use Heartbeat messages. A Heartbeat message is composed of several aggregate classes, as shown in Figure 6.

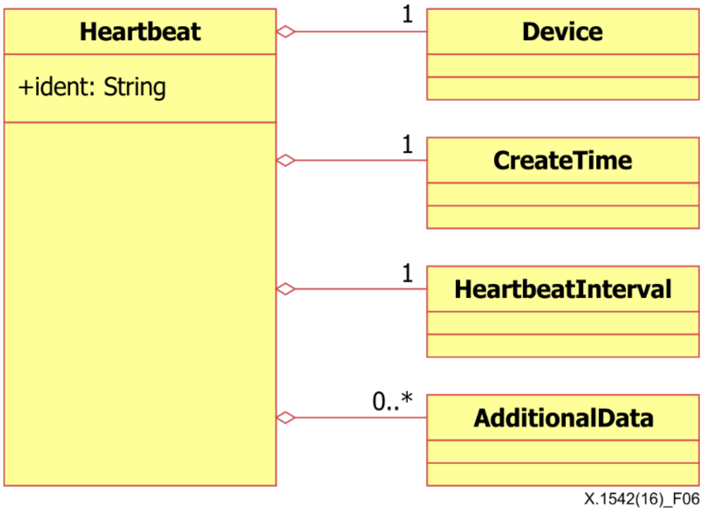

Figure 6 – The aggregate classes of the heartbeat class

Information about the attribute and components of the Heartbeat class is shown in Table 8 and Table 9, respectively.

Table 8 – Attribute of the Heartbeat class

Attribute	Usage	Data type	Description
ident	Optional	STRING	Unique identifier for the Heartbeat.

Table 9 – Components of the Heartbeat class

Classes	Aggregation	Data type	Description
Device	exactly one		Identification information for the analyser that originated the heartbeat
CreateTime	exactly one	DATETIME	The time the heartbeat was created
HeartbeatInterval	exactly one	INTEGER	The interval in seconds at which heartbeats are generated.
AdditionalData	zero or more		Information included by the analyser that does not fit into the data model

8.2.3 Core classes

The core classes (Device, CreateTime, Source, Target, SourceNAT, TargetNAT, and AdditionalData) are the main parts of Connect and Heartbeat classes, as shown in Figure 7. The individual classes are described in this clause.

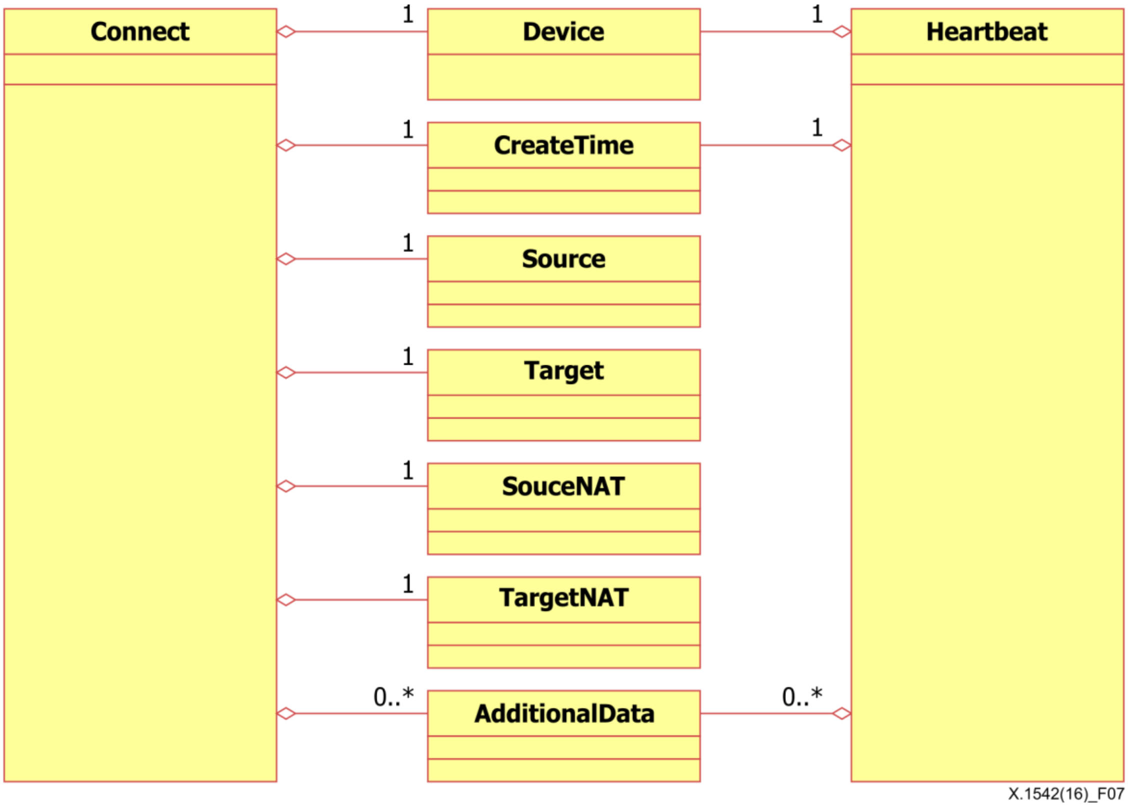

Figure 7 – Core classes

8.2.3.1 Device class

The Device class identifies the analyser from which the Connect or Heartbeat message originates. Only one device may be encoded for each connect or heartbeat, and that shall be the device at which the connect or heartbeat originated.

The Device class is composed of three aggregate classes, as shown in Figure 8.

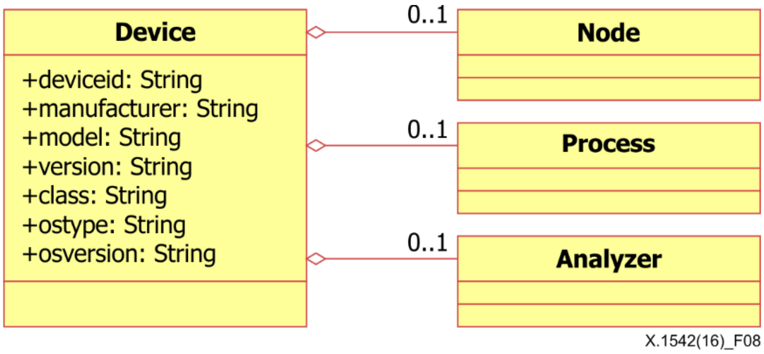

Figure 8 – The aggregate classes of the Device class

The Device class has seven attributes, as shown in Table 10.

Table 10 – Attributes of device class

Attribute	Usage	Data type	Description
deviceid	Optional	STRING	A unique identifier for the device. If the device makes use of the "ident" attributes on other classes to provide unique identifiers for those objects, then it shall also provide a valid "deviceid" attribute.
Manufacturer	Optional	STRING	The manufacturer of the device software or hardware.
Model	Optional	STRING	The model name/number of the device software or hardware.
Version	Optional	STRING	The version number of the device software or hardware.
Class	Optional	STRING	The class of device software or hardware.
Ostype	Optional	STRING	Operating system name.
osversion	Optional	STRING	Operating system version.

For the ostype attribute on POSIX 1003.1 compliant systems, this is the value returned in utsname.sysname by the uname() system call, or the output of the "uname –s" command.

For the osversion attribute on POSIX 1003.1 compliant systems, this is the value returned in utsname.release by the uname() system call, or the output of the "uname –r" command.

The contents of "manufacturer", "model", "version", and "class" attributes are vendor-specific, but may be used together to identify different types of analysers.

The aggregate classes that make up a Device class are explained in Table 11.

Table 11 – Components of device class

Classes	Aggregation	Data type	Description
Node	Zero or one		Information about the host or device on which the analyser resides (network address, network name, etc.)
Process	Zero or one		Information about the process in which the analyser is executing
Analyser	Zero or one		Information about the analyser through which the message may have gone.

8.2.3.2 CreateTime class

The CreateTime class is used to indicate the current date and time on the device. If this difference should then be used to adjust the times in the <CreateTime> and < NTP timestamps > elements, then NTP timestamps should also be adjusted.

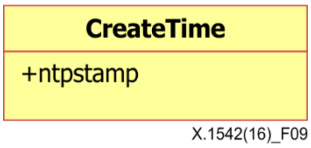

Figure 9 – The CreateTime class

The attribute of the CreateTime class is shown in Table 12.

Table 12 – Attribute of the CreateTime class

Attribute	Usage	Data type	Description
ntpstamp	Required	ntpstamp	Information about the current time on the device.

8.2.3.3 Source class

The Source class contains information about the possible source(s) of the event(s) that generated a session. An event may have more than one source (e.g., in a distributed denial-of-service attack).

The Source class is composed of three aggregate classes, as shown in Figure 10.

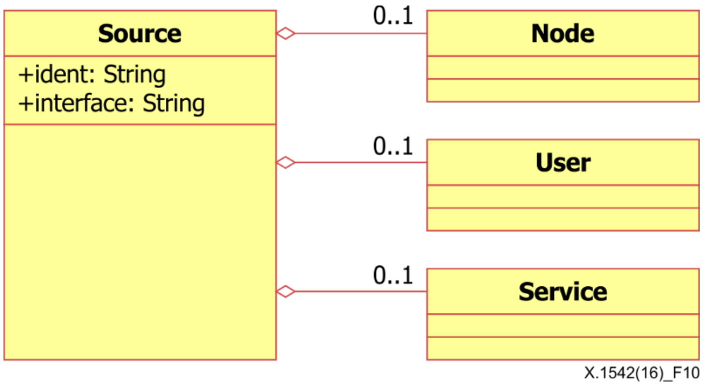

Figure 10 – The aggregate classes of the Source class

The Source class has two attributes shown in Table 13.

Table 13 – Attributes of the Source class

Attribute	Usage	Data type	Description
ident	Optional	STRING	A unique identifier for this source.
Interface	Optional	STRING	May be used by a network-based device with multiple interfaces to indicate which interface this source was seen on.

The aggregate classes that make up Source class are explained in Table 14.

Table 14 – Components of the Source class

Classes	Aggregation	Data type	Description
Node	Zero or one		Information about the host or device that appears to be causing the events (network address, network name, etc.).
User	Zero or one		Information about the user that appears to be causing the event(s).
Service	Zero or one		Information about the network service involved in the event(s).

8.2.3.4 Target class

The Target class contains information about the possible target(s) of the event(s) that generated a session. An event may have more than one target (e.g., in the case of a port sweep).

The Target class is composed of three aggregate classes, as shown in Figure 11.

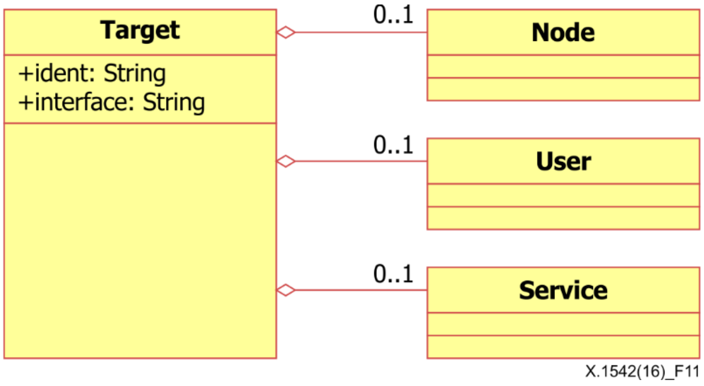

Figure 11 – The aggregate classes of the Target class

The Target class has two attributes, as shown in Table 15.

Table 15 – Attributes of the Target class

Attribute	Usage	Data type	Description
ident	Optional	STRING	A unique identifier for this target.
Interface	Optional	STRING	May be used by a network-based device with multiple interfaces to indicate which interface this target was seen on.

The aggregate classes that make up the Target class are explained in Table 16.

Table 16 – Components of Target class

Classes	Aggregation	Data type	Description
Node	Zero or one		Information about the host or device at which the event(s) (network address, network name, etc.) is being directed.
User	Zero or one		Information about the user at which the event(s) is being directed.
Service	Zero or one		Information about the network service involved in the event(s).

8.2.3.5 SourceNAT class

The SourceNAT class contains information about the possible source(s) of the NAT event(s) that generated a session. An event may have more than one source transformed by NAT.

The SourceNAT class is composed of three aggregate classes, as shown in Figure 12.

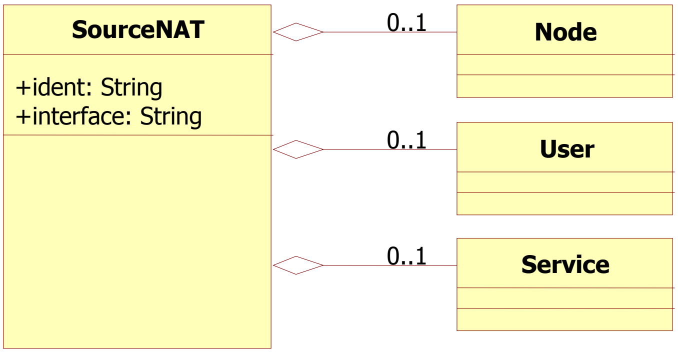

Figure 12 – The aggregate classes of the SourceNAT class

The Source class has two attributes, as shown in Table 17.

Table 17 – Attributes of the SourceNAT class

Attribute	Usage	Data type	Description
ident	Optional	STRING	A unique identifier for this source transformed by NAT.
interface	Optional	STRING	May be used by a network-based device with multiple interfaces to indicate which interface this source transformed by NAT was seen on.

The aggregate classes that make up the SourceNAT class are explained in Table 18.

Table 18 – Components of the SourceNAT class

Classes	Aggregation	Data type	Description
Node	Zero or one		Information about the host or device that appears to be causing the events (network address, network name, etc.).
User	Zero or one		Information about the user that appears to be causing the event(s).
Service	Zero or one		Information about the network service involved in the event(s).

8.2.3.6 TargetNAT class

The TargetNAT class contains information about the possible target(s) of the NAT event(s) that generated a session. An event may have more than one target transformed by NAT.

The TargetNAT class is composed of three aggregate classes, as shown in Figure 13.

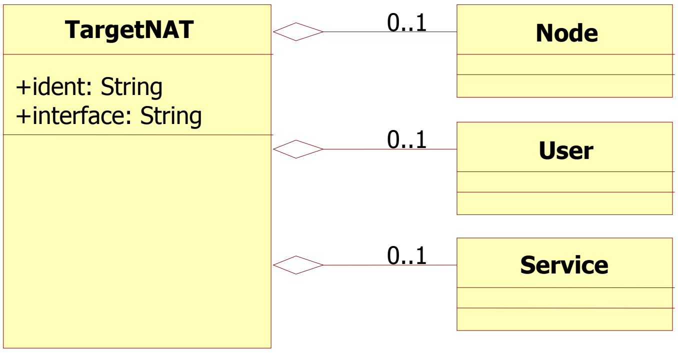

Figure 13 – The aggregate classes of the TargetNAT class

The TargetNAT class has two attributes, as shown in Table 19.

Table 19 – Attributes of the TargetNAT class

Attribute	Usage	Data type	Description
ident	Optional	STRING	A unique identifier for this target transformed by NAT.
interface	Optional	STRING	May be used by a network-based device with multiple interfaces to indicate which interface this target transformed by NAT was seen on.

The aggregate classes that make up Target class are explained in Table 20.

Table 20 – Components of the TargetNAT class

Classes	Aggregation	Data type	Description
Node	Zero or one		Information about the host or device at which the event(s) (network address, network name, etc.) is being directed.
User	Zero or one		Information about the user at which the event(s) is being directed.
Service	Zero or one		Information about the network service involved in the event(s).

8.2.3.7 AdditionalData class

The AdditionalData class is used to provide information that cannot be represented by the SIMEF data model. AdditionalData can be used to provide atomic data (integers, strings, etc.) in cases where only small amounts of additional information need to be sent; it can also be used to extend the data model and the DTD to support the transmission of complex data (such as packet headers).

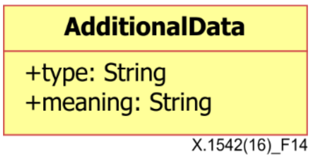

Figure 14 – The AdditionalData class

The AdditionalData class has two attributes, as shown in Table 21.

Table 21 – Attributes of the AdditionalData class

Attribute	Usage	Data type	Description
type	Required	ENUM	A data type describing the meaning of the element content. Default Value : string
meaning	Optional	STRING	A string describing the meaning of the element content.

The types of AdditionalData class are represented and the permitted values for this attribute are shown in Table 22.

Table 22 – Value of Type attribute

Value	Keyword	Definition
0	boolean	The element contains a boolean value, i.e., the strings "true" or "false"
1	byte	The element content is a single 8 bit byte
2	character	The element content is a single character
3	date-time	The element content is a date-time string
4	integer	The element content is an integer
5	ntpstamp	The element content is an NTP timestamp
6	portlist	The element content is a list of ports
7	real	The element content is a real number
8	string	The element content is a string
9	Byte-string	The element is a byte[]
10	xml	The element content is XML-tagged data

These values of the AdditionalData class are vendor/implementation dependent; the method for ensuring that managers understand the strings sent by analysers is outside the scope of this Recommendation.

8.2.4 Support classes

The support classes make up the major parts of the core classes and are shared between them.

8.2.4.1 Node class

The Node class is used to identify hosts and other network devices (routers, switches, etc.).

The Node class is composed of three aggregate classes, as shown in Figure 15. The attributes, value of type attribute and components of the Node class are shown in Table 23, Table 24 and Table 25, respectively.

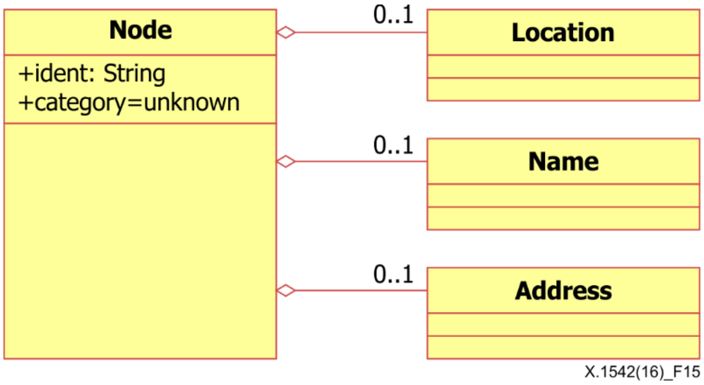

Figure 15 – The aggregate classes of the Node class

Table 23 – Attributes of the Node class

Attribute	Usage	Data type	Description
ident	Optional	STRING	A unique identifier for the node; see clause 7.2.9.
category	Optional	ENUM	The "domain" from which the name information was obtained. Default value = unknown

Table 24 – Value of Type attribute

Value	Keyword	Definition
0	Unknown	Domain unknown or not relevant
1	ads	Windows 2000 Advanced Directory Services
2	afs	Andrew File System (Transarc)
3	coda	Coda Distributed File System
4	dfs	Distributed File System (IBM)
5	dns	Domain Name System
6	hosts	Local hosts file
7	kerberos	Kerberos realm
8	nds	Novell Directory Services
9	nis	Network Information Services (Sun)
10	nisplus	Network Information Services Plus (Sun)
11	nt	Windows NT domain
12	wfw	Windows for Workgroups

Table 25 – Components of the Node class

Classes	Aggregation	Data type	Description
Location	Zero or one	STRING	The location of the equipment
Name	Zero or one	STRING	The name of the equipment. This information shall be provided if no Address information is given.
Address	Zero or more		The network or hardware address of the equipment. Unless a name (above) is provided, at least one address shall be specified.

8.2.4.2 The Address class

The Address class is used to represent network, hardware, and application addresses.

The Address class is composed of two aggregate classes, as shown in Figure 16.

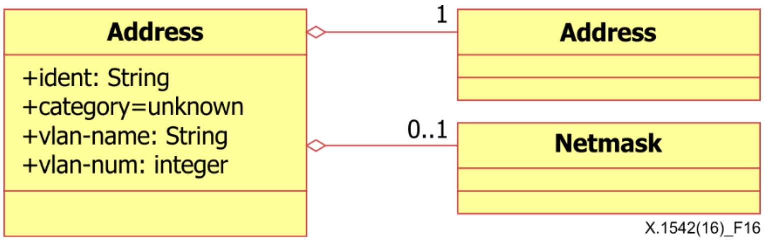

Figure 16 – The aggregate classes of the Address class

The attributes, value of type attribute and components of the Address class are listed in Table 26, Table 27 and Table 28, respectively.

Table 26 – Attributes of the Address class

Attribute	Usage	Data type	Description
ident	Optional	STRING	A unique identifier for the address; see clause 7.2.9.
category	Optional	ENUM	The type of address represented. The permitted values for this attribute are shown below. Default value: unknown.
vlan-name	Optional	STRING	The name of the local area network (LAN) (virtual LAN) to which the address belongs.
Vlan-num	Optional	INTEGER	The number of the LAN (virtual LAN) to which the address belongs.

Table 27 – Value of Type attribute

Value	Keyword	Definition
0	unknown	Address type unknown
1	atm	Asynchronous Transfer Mode network address
2	e-mail	Electronic mail address ([b-IETF RFC 2822])
3	lotus-notes	Lotus Notes e-mail address
4	Mac	Media access control (MAC) address
5	Sna	IBM Shared Network Architecture (SNA) address
6	Vm	IBM VM ("PROFS") e-mail address
7	ipv4-addr	IPv4 host address in dotted-decimal notation (a.b.c.d)
8	ipv4-addr-hex	IPv4 host address in hexadecimal notation
9	ipv4-net	IPv4 network address in dotted-decimal notation, slash, significant bits (a.b.c.d/nn)
10	ipv4-net-mask	IPv4 network address in dotted-decimal notation, slash, network mask in dotted-decimal notation (a.b.c.d./w.x.y.z)
11	ipv6-addr	IPv6 host address
12	ipv6-addr-hex	IPv6 host address in hexadecimal notation
13	ipv6-net	IPv6 network address, slash, significant bits
14	ipv6-net-mask	IPv6 network address, slash, network mask

Table 28 – Components of the Address class

Classes	Aggregation	Data type	Description
Address	Exactly one	STRING	The address information. The format of this data is governed by the category attribute.
Netmask	Zero or one	STRING	The network mask for the address, if appropriate.

8.2.4.3 The User class

The User class is used to describe users. It is primarily used as a "container" class for the UserId aggregate class, as shown in Figure 17.

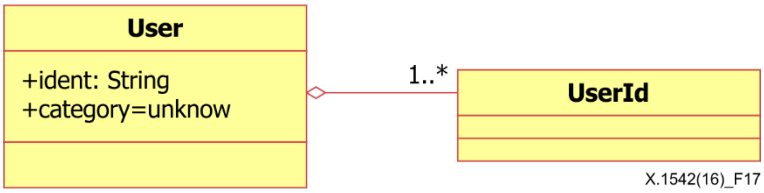

Figure 17 – The aggregate classes of the User class

The attributes, value of type attribute and components of the User class are listed in Table 29, Table 30 and Table 31, respectively.

Table 29 – Attributes of the User class

Attribute	Usage	Data type	Description
ident	Optional	STRING	A unique identifier for the user; see clause 7.2.9.
category	Optional	ENUM	The type of user represented. The permitted values for this attribute are shown below. Default value = unknown.

Table 30 – Value of Type attribute

Value	Keyword	Definition
0	unknown	User type unknown
1	application	An application user
2	os-device	An operating system or device user

Table 31 – Components of the User class

Classes	Aggregation	Data type	Description
UserId	One or more		Identification of a user, as indicated by its type attribute

8.2.4.3.1 The UserId class

The UserId class provides specific information about a user. More than one UserId can be used within the User class to indicate attempts to transition from one user to another, or to provide complete information about privileges of a user (or process).

The UserId class is composed of two aggregate classes, as shown in Figure 18.

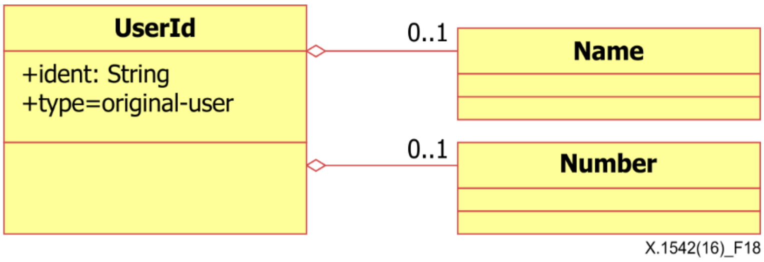

Figure 18 – The aggregate classes of the UserId class

The attributes and value of type attribute of the UserId class are listed in Table 32 and Table 33, respectively.

Table 32 – Attributes of the UserId class

Attribute	Usage	Data type	Description
ident	Optional	STRING	A unique identifier for the user id, see clause 7.2.9.
type	Optional	ENUM	The type of user information represented. The permitted values for this attribute are shown below. Default value = original-user

Table 33 – Value of Type attribute

Value	Keyword	Definition
0	current-user	The current user id being used by the user or process.
1	original-user	The actual identity of the user or process being reported on. On those systems that (a) do some type of auditing and (b) support extracting a user id from the "audit id" token, that value should be used. On those systems that do not support this, and where the user has logged into the system, the "login id" should be used.
2	target-user	The user id the user or process is attempting to become. This would apply, on Unix systems for example, when the user attempts to use "su", "rlogin", "telnet", etc.
3	user-privs	Another user id the user or process has the ability to use, or a user id associated with a file permission. Multiple UserId elements of this type may be used to specify a list of privileges.
4	current-group	The current group id (if applicable) being used by the user or process.
5	group-privs	Another group id the group or process has the ability to use, or a group id associated with a file permission. For example, On Berkeley Software Distribution (BSD)-derived Unix systems, multiple UserId elements of this type would be used to include all the group ids on the "group list".
6	other-privs	Not used in a user, group, or process context, only used in the file context. The file permissions assigned to users who do not match either the user or group permissions on the file.

The aggregate classes that make up the UserId are listed in Table 34.

Table 34 – Components of the UserId class

Classes	Aggregation	Data type	Description
Name	Zero or one	STRING	A user or group name.
Num	Zero or one	INTERGER	A user or group number.

8.2.4.4 The Process class

The Process class is used to describe processes being executed on sources, targets and analysers.

The Process class is composed of five aggregate classes, as shown in Figure 19.

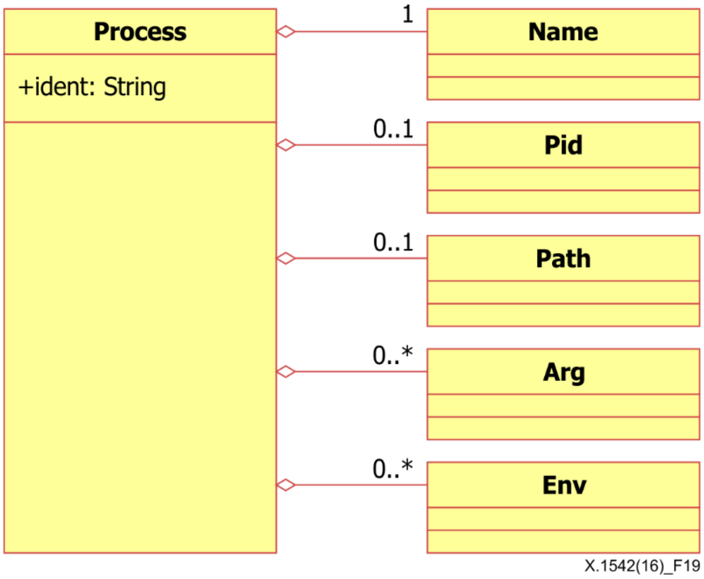

Figure 19 – The aggregate classes of the Process class

The Process class has one attribute (see Table 35).

Table 35 – Attribute of the Process class

Attribute	Usage	Data type	Description
ident	Optional	STRING	A unique identifier for the process; see clause 7.2.9.

The aggregate classes that make up Process are listed in Table 36.

Table 36 – Components of the Process class

Classes	Aggregation	Data type	Description
Name	Exactly one	STRING	The name of the program being executed.
Pid	Zero or one	INTEGER	The process identifier of the process.
Path	Zero or one	STRING	The full path of the program being executed.
Arg	Zero or more	STRING	A command-line argument to the program.
Env	Zero or more	STRING	An environment string associated with the process; generally of the format "VARIABLE=value".

In the Process class, the name class is a short name and Multiple arguments may be specified with multiple uses of arg. Multiple environment strings may be specified with multiple uses of env.

8.2.4.5 The Service class

The Service class describes network services on sources and targets. It can identify services by name, port, portlist and protocol. When Service occurs as an aggregate class of Source, it is understood that the service is one from which activity of interest is originating; and that the service is "attached" to the Node, Process, and User information also contained in Source. Likewise, when Service occurs as an aggregate class of Target, it is understood that the service is one to which activity of interest is being directed; and that the service is "attached" to the Node, Process, and User information also contained in Target. If Service occurs in both Source and Target, then information in both locations should be the same. If information is the same in both locations and implementers wish to carry it in only one location, they should specify it as an aggregate of the Target class.

The Service class is composed of four aggregate classes, as shown in Figure 20.

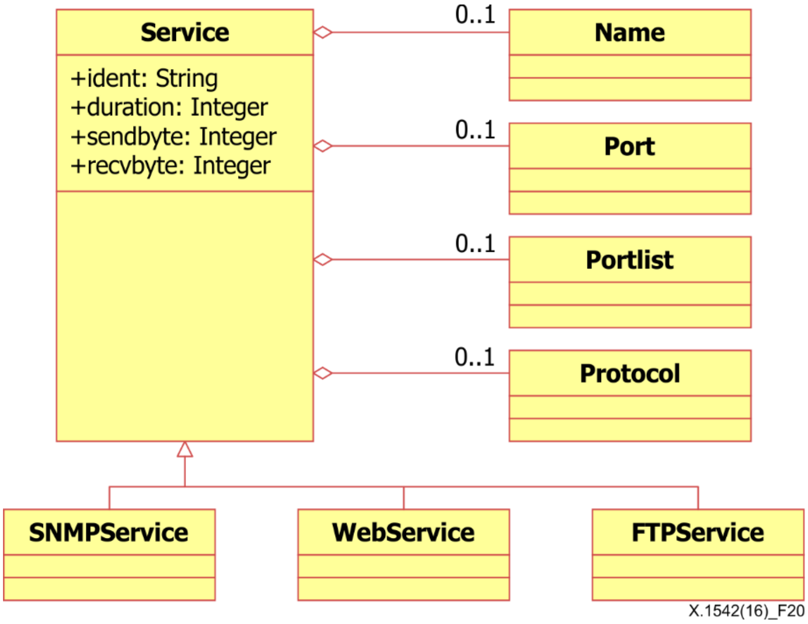

Figure 20 – The aggregate classes of the Service class

The Service class has four attributes, listed in Table 37.

Table 37 – Attributes of the Service class

Attribute	Usage	Data type	Description
ident	Optional	STRING	A unique identifier for the service; see clause 7.2.9.
duration	Optional	INTEGER	Time for connection
sendbyte	Optional	INTEGER	byte size by sending after connection
recvByte	Optional	INTEGER	byte size by receiving after connection

The aggregate classes that make up the Service class are listed in Table 38.

Table 38 – Components of the Service class

Classes	Aggregation	Data type	Description
Name	Zero or one	STRING	The name of the service. Whenever possible, the name from the Internet Assigned Numbers Authority (IANA) list of well-known ports should be used.
Port	Zero or one	INTEGER	The port number being used.
Portlist	Zero or one	PORTLIST	A list of port numbers being used; see clause 7.2.8 for formatting rules.
Protocol	Zero or one	STRING	Additional information about the protocol being used.

8.2.4.5.1 The WebService class

The WebService class carries additional information related to web traffic.

The WebService class is composed of four aggregate classes, as shown in Figure 21.

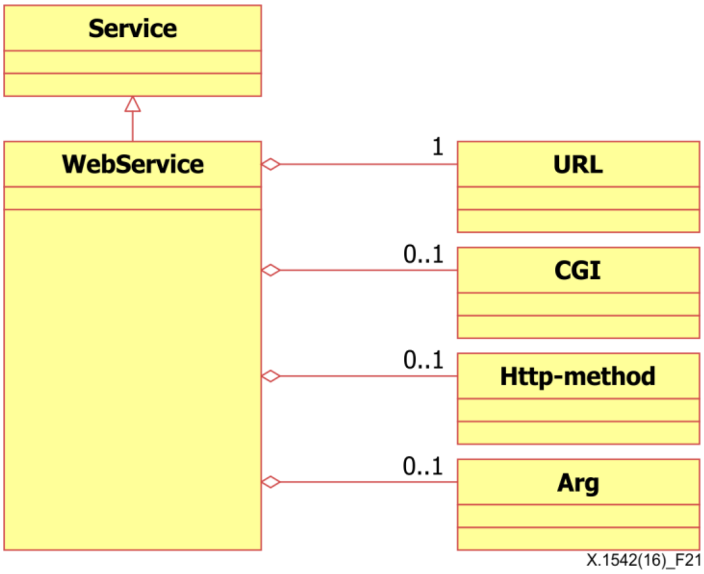

Figure 21 – The aggregate classes of the WebService class

The aggregate classes that make up the WebService class are listed in Table 39.

Table 39 – Components of the WebService class

Classes	Aggregation	Data type	Description
URL	Exactly one	STRING	The uniform resource locator (URL) in the request.
CGI	Zero or one	STRING	The common gateway interface (CGI) script in the request, without arguments.
Http-method	Zero or one	STRING	The hypertext transfer protocol (HTTP) method (PUT, GET) used in the request.
Arg	Zero or one	STRING	The arguments to the CGI script.

8.2.4.5.2 The SNMPService class

The SNMPService class carries additional information related to simple network management protocol (SNMP) traffic.

The SNMPService class is composed of eight aggregate classes, as shown in Figure 22.

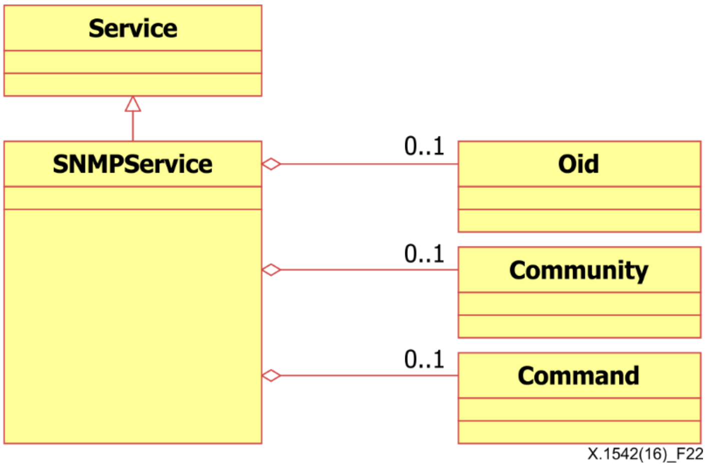

Figure 22 – The aggregate classes of the SNMPService class

The aggregate classes that make up SNMPService are listed in Table 40.

Table 40 – Components of the SNMPService class

Classes	Aggregation	Data type	Description
Oid	Zero or one	STRING	The object identifier in the request.
Community	Zero or one	STRING	Object's community string
Command	Zero or one	STRING	The command sent to the SNMP server (GET, SET, etc.).

8.2.4.5.3 The FTPService class

The FTPService class carries additional information related to file transfer protocol (FTP) traffic.

The FTPService class is composed of two aggregate classes, as shown in Figure 23.

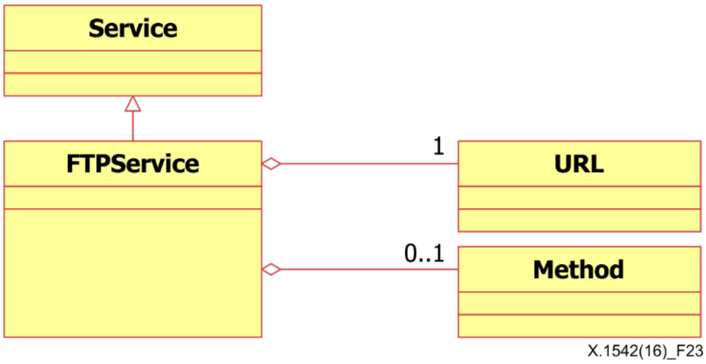

Figure 23 – The aggregate classes of the FTPService class

The aggregate classes that make up the FTPService class are listed in Table 41.

Table 41 – Components of the FTPService class

Classes	Aggregation	Data type	Description
URL	Exactly one	STRING	The URL in the request.
Method	Zero or one	STRING	The FTP method (PUT, GET) used in the request.

9 Security consideration

This clause discusses some of the security considerations that shall be taken into account by implementers of the SIMEF.

This Recommendation describes the information model for the session information message exchange format (SIMEF) and provides an associated data model specified with XML schema. The SIMEF defines a data model representation for sharing the transport layer session log information regarding the centralized network security management and security information exchange system.

Although there are no security concerns directly applicable to the format of this data, the data itself may contain security-sensitive information whose confidentiality, integrity or availability may need to be protected.

This Recommendation suggests that the systems used to collect, transmit, process and store this data should be protected against unauthorized use, and that the data itself should be protected against unauthorized access. The means for achieving this protection are outside the scope of this Recommendation.

Appendix I

SIMEF example and schema

(This appendix does not form an integral part of this Recommendation.)

This appendix gives an example of an XML schema for the SIMEF model. The following examples are XML schema, SYSLOG schema in order to encode the session information into the SIMEF model.

I.1 SIMEF Schema

I.1.1 XML Schema

```
  <?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="0xaaaaaaaaaaaaaaaa">
                   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 SYSLOG Schema

```
 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 SIMEF examples

I.2.1 Firewall permission

```
<?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="0xaaaaaaaaaaaaaaaaaa">
                    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 VPN log

```
<?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"></Policy>
              <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 NAT log

```
<?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>
```

Bibliography

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

[b-ISO/IEC 10646] ISO/IEC 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 OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	**Data networks, open system communications and security**
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2016