CybOX™ Version 2.1.1. Part 28: HTTP Session Object

Committee Specification Draft 01 / Public Review Draft 01

20 June 2016

Specification URIs

This version:

http://docs.oasis-open.org/cti/cybox/v2.1.1/csprd01/part28-http-session/cybox-v2.1.1-csprd01-part28-http-session.docx (Authoritative)
http://docs.oasis-open.org/cti/cybox/v2.1.1/csprd01/part28-http-session/cybox-v2.1.1-csprd01-part28-http-session.html
http://docs.oasis-open.org/cti/cybox/v2.1.1/csprd01/part28-http-session/cybox-v2.1.1-csprd01-part28-http-session.pdf

Previous version:

N/A

Latest version:

http://docs.oasis-open.org/cti/cybox/v2.1.1/part28-http-session/cybox-v2.1.1-part28-http-session.docx (Authoritative)
http://docs.oasis-open.org/cti/cybox/v2.1.1/part28-http-session/cybox-v2.1.1-part28-http-session.html
http://docs.oasis-open.org/cti/cybox/v2.1.1/part28-http-session/cybox-v2.1.1-part28-http-session.pdf

Technical Committee:

OASIS Cyber Threat Intelligence (CTI) TC

Chair:

Richard Struse (Richard.Struse@HQ.DHS.GOV), DHS Office of Cybersecurity and Communications (CS&C)

Editors:

Desiree Beck (dbeck@mitre.org), MITRE Corporation
Trey Darley (trey@kingfisherops.com), Individual member
Ivan Kirillov (ikirillov@mitre.org), MITRE Corporation
Rich Piazza (rpiazza@mitre.org), MITRE Corporation

Additional artifacts:

This prose specification is one component of a Work Product whose components are listed in http://docs.oasis-open.org/cti/cybox/v2.1.1/csprd01/cybox-v2.1.1-csprd01-additional-artifacts.html.

Related work:

This specification is related to:

- *STIX™ Version 1.2.1*. Edited by Sean Barnum, Desiree Beck, Aharon Chernin, and Rich Piazza. 05 May 2016. OASIS Committee Specification 01. http://docs.oasis-open.org/cti/stix/v1.2.1/cs01/part1-overview/stix-v1.2.1-cs01-part1-overview.html.

Abstract:

The Cyber Observable Expression (CybOX™) is a standardized language for encoding and communicating high-fidelity information about cyber observables, whether dynamic events or stateful measures that are observable in the operational cyber domain. By specifying a common structured schematic mechanism for these cyber observables, the intent is to enable the potential for detailed automatable sharing, mapping, detection, and analysis heuristics. This specification document defines the HTTP Session Object data model, which is one of the Object data models for CybOX content.

Status:

This document was last revised or approved by the OASIS Cyber Threat Intelligence (CTI) TC on the above date. The level of approval is also listed above. Check the "Latest version" location noted above for possible later revisions of this document. Any other numbered Versions and other technical work produced by the Technical Committee (TC) are listed at https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=cti#technical.

TC members should send comments on this specification to the TC's email list. Others should send comments to the TC's public comment list, after subscribing to it by following the instructions at the "Send A Comment" button on the TC's web page at https://www.oasis-open.org/committees/cti/.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the TC's web page (https://www.oasis-open.org/committees/cti/ipr.php).

Citation format:

When referencing this specification the following citation format should be used:

[CybOX-v2.1.1-http-session]

CybOX™ Version 2.1.1. Part 28: HTTP Session Object. Edited by Desiree Beck, Trey Darley, Ivan Kirillov, and Rich Piazza. 20 June 2016. OASIS Committee Specification Draft 01 / Public Review Draft 01. http://docs.oasis-open.org/cti/cybox/v2.1.1/csprd01/part28-http-session/cybox-v2.1.1-csprd01-part28-http-session.html. Latest version: http://docs.oasis-open.org/cti/cybox/v2.1.1/part28-http-session/cybox-v2.1.1-part28-http-session.html.

Notices

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see https://www.oasis-open.org/policies-guidelines/trademark for above guidance.

STIX™, TAXII™, AND CybOX™ (STANDARD OR STANDARDS) AND THEIR COMPONENT PARTS ARE PROVIDED “AS IS” WITHOUT ANY WARRANTY OF ANY KIND, EITHER EXPRESSED, IMPLIED, OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, ANY WARRANTY THAT THESE STANDARDS OR ANY OF THEIR COMPONENT PARTS WILL CONFORM TO SPECIFICATIONS, ANY IMPLIED

WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR FREEDOM FROM INFRINGEMENT, ANY WARRANTY THAT THE STANDARDS OR THEIR COMPONENT PARTS WILL BE ERROR FREE, OR ANY WARRANTY THAT THE DOCUMENTATION, IF PROVIDED, WILL CONFORM TO THE STANDARDS OR THEIR COMPONENT PARTS. IN NO EVENT SHALL THE UNITED STATES GOVERNMENT OR ITS CONTRACTORS OR SUBCONTRACTORS BE LIABLE FOR ANY DAMAGES, INCLUDING, BUT NOT LIMITED TO, DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES, ARISING OUT OF, RESULTING FROM, OR IN ANY WAY CONNECTED WITH THESE STANDARDS OR THEIR COMPONENT PARTS OR ANY PROVIDED DOCUMENTATION, WHETHER OR NOT BASED UPON WARRANTY, CONTRACT, TORT, OR OTHERWISE, WHETHER OR NOT INJURY WAS SUSTAINED BY PERSONS OR PROPERTY OR OTHERWISE, AND WHETHER OR NOT LOSS WAS SUSTAINED FROM, OR AROSE OUT OF THE RESULTS OF, OR USE OF, THE STANDARDS, THEIR COMPONENT PARTS, AND ANY PROVIDED DOCUMENTATION. THE UNITED STATES GOVERNMENT DISCLAIMS ALL WARRANTIES AND LIABILITIES REGARDING THE STANDARDS OR THEIR COMPONENT PARTS ATTRIBUTABLE TO ANY THIRD PARTY, IF PRESENT IN THE STANDARDS OR THEIR COMPONENT PARTS AND DISTRIBUTES IT OR THEM “AS IS.”

Table of Contents

1 Introduction

[All text is normative unless otherwise labeled.]

The Cyber Observable Expression (CybOX^{TM}) provides a common structure for representing cyber observables across and among the operational areas of enterprise cyber security. CybOX improves the consistency, efficiency, and interoperability of deployed tools and processes, and it increases overall situational awareness by enabling the potential for detailed automatable sharing, mapping, detection, and analysis heuristics.

This document serves as the specification for the CybOX HTTP Session Object Version 2.1.1 data model, which is one of eighty-eight CybOX Object data models.

In Section **1.1** we discuss additional specification documents, in Section **1.2** we provide document conventions, and in Section **1.3** we provide terminology. References are given in Section **1.4**. In Section **2**, we give background information necessary to fully understand the HTTP Session Object data model. We present the HTTP Session Object data model specification details in Section **3** and conformance information in Section **4**.

1.1 CybOX^{TM} Specification Documents

The CybOX specification consists of a formal UML model and a set of textual specification documents that explain the UML model. Specification documents have been written for each of the individual data models that compose the full CybOX UML model.

CybOX has a modular design comprising two fundamental data models and a collection of Object data models. The fundamental data models – CybOX Core and CybOX Common – provide essential CybOX structure and functionality. The CybOX Objects, defined in individual data models, are precise characterizations of particular types of observable cyber entities (e.g., HTTP session, Windows registry key, DNS query).

Use of the CybOX Core and Common data models is required; however, use of the CybOX Object data models is purely optional: users select and use only those Objects and corresponding data models that are needed. Importing the entire CybOX suite of data models is not necessary.

The *CybOX™ Version 2.1.1 Part 1: Overview* document provides a comprehensive overview of the full set of CybOX data models, which in addition to the Core, Common, and numerous Object data models, includes various extension data models and a vocabularies data model, which contains a set of default controlled vocabularies. *CybOX™ Version 2.1.1 Part 1: Overview* also summarizes the relationship of CybOX to other languages, and outlines general CybOX data model conventions.

1.2 Document Conventions

The following conventions are used in this document.

1.2.1 Fonts

The following font and font style conventions are used in the document:

- Capitalization is used for CybOX high-level concepts, which are defined in *CybOX™ Version 2.1.1 Part 1: Overview*.

Examples: Action, Object, Event, Property

- The `Courier New` font is used for writing UML objects.

  Examples: `ActionType`, `cyboxCommon:BaseObjectPropertyType`

  Note that all high-level concepts have a corresponding UML object. For example, the Action high-level concept is associated with a UML class named, `ActionType`.

- The '*italic*' font (with single quotes) is used for noting actual, explicit values for CybOX Language properties. The *italic* font (without quotes) is used for noting example values.

  Example: *'HashNameVocab-1.0,' high, medium, low*

1.2.2 UML Package References

Each CybOX data model is captured in a different UML package (e.g., Core package) where the packages together compose the full CybOX UML model. To refer to a particular class of a specific package, we use the format `package_prefix:class`, where `package_prefix` corresponds to the appropriate UML package.

The package_prefix for the HTTP Session data model is `HTTOSessionObj`. Note that in this specification document, we do not explicitly specify the package prefix for any classes that originate from the HTTP Session Object data model.

1.2.3 UML Diagrams

This specification makes use of UML diagrams to visually depict relationships between CybOX Language constructs. Note that the diagrams have been extracted directly from the full UML model for CybOX; they have not been constructed purely for inclusion in the specification documents. Typically, diagrams are included for the primary class of a data model, and for any other class where the visualization of its relationships between other classes would be useful. This implies that there will be very few diagrams for classes whose only properties are either a data type or a class from the CybOX Common data model. Other diagrams that are included correspond to classes that specialize a superclass and abstract or generalized classes that are extended by one or more subclasses.

In UML diagrams, classes are often presented with their attributes elided, to avoid clutter. The fully described class can usually be found in a related diagram. A class presented with an empty section at the bottom of the icon indicates that there are no attributes other than those that are visualized using associations.

1.2.3.1 Class Properties

Generally, a class property can be shown in a UML diagram as either an attribute or an association (i.e., the distinction between attributes and associations is somewhat subjective). In order to make the size of UML diagrams in the specifications manageable, we have chosen to capture most properties as attributes and to capture only higher-level properties as associations, especially in the main top-level component diagrams. In particular, we will always capture properties of UML data types as attributes.

1.2.3.2 Diagram Icons and Arrow Types

Diagram icons are used in a UML diagram to indicate whether a shape is a class, enumeration, or a data type, and decorative icons are used to indicate whether an element is an attribute of a class or an enumeration literal. In addition, two different arrow styles indicate either a directed association relationship (regular arrowhead) or a generalization relationship (triangle-shaped arrowhead). The icons and arrow styles we use are shown and described in **Table 1-1**.

Table 1-1. UML diagram icons

Icon	Description
▤	This diagram icon indicates a class. If the name is in italics, it is an abstract class.
‹E›	This diagram icon indicates an enumeration.
‹D›	This diagram icon indicates a data type.
●	This decorator icon indicates an attribute of a class. The green circle means its visibility is public. If the circle is red or yellow, it means its visibility is private or protected.
▬	This decorator icon indicates an enumeration literal.
⟶	This arrow type indicates a directed association relationship.
—▷	This arrow type indicates a generalization relationship.

1.2.4 Property Table Notation

Throughout Section 3, tables are used to describe the properties of each data model class. Each property table consists of a column of names to identify the property, a type column to reflect the datatype of the property, a multiplicity column to reflect the allowed number of occurrences of the property, and a description column that describes the property. Package prefixes are provided for classes outside of the HTTP Session Object data model (see Section 1.2.2).

Note that if a class is a specialization of a superclass, only the properties that constitute the specialization are shown in the property table (i.e., properties of the superclass will not be shown). However, details of the superclass may be shown in the UML diagram.

1.2.5 Property and Class Descriptions

Each class and property defined in CybOX is described using the format, “The X property verb Y.” For example, in the specification for the CybOX Core data model, we write, “The `id` property specifies a globally unique identifier for the Action.” In fact, the verb “specifies” could have been replaced by any number of alternatives: “defines,” “describes,” “contains,” “references,” etc.

However, we thought that using a wide variety of verb phrases might confuse a reader of a specification document because the meaning of each verb could be interpreted slightly differently. On the other hand, we didn’t want to use a single, generic verb, such as “describes,” because although the different verb choices may or may not be meaningful from an implementation standpoint, a distinction could be useful to those interested in the modeling aspect of CybOX.

Consequently, we have preferred to use the three verbs, defined as follows, in class and property descriptions:

Verb	CybOX Definition
captures	Used to record and preserve information without implying anything about the structure of a class or property. Often used for properties that encompass general content. This is the least precise of the three verbs.
	Examples: The `Observable_Source` property characterizes the source of the Observable information. Examples of details captured include identifying characteristics, time-related attributes, and a list of the tools used to collect the information. The `Description` property captures a textual description of the Action.
characterizes	Describes the distinctive nature or features of a class or property. Often used to describe classes and properties that themselves comprise one or more other properties.
	Examples: The `Action` property characterizes a cyber observable Action. The `Obfuscation_Technique` property characterizes a technique an attacker could potentially leverage to obfuscate the Observable.
specifies	Used to clearly and precisely identify particular instances or values associated with a property. Often used for properties that are defined by a controlled vocabulary or enumeration; typically used for properties that take on only a single value.
	Example: The `cybox_major_version` property specifies the major version of the CybOX Language used for the set of Observables.

1.3 Terminology

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in **[RFC2119]**.

1.4 Normative References

[RFC2119] Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels”, BCP 14, RFC 2119, March 1997. http://www.ietf.org/rfc/rfc2119.txt.

[RFC 2822] Resnick, P., “Internet Message Format”, RFC2822, April 2001. https://www.ietf.org/rfc/rfc2822.txt

[RFC 5988] Nottingham, M., “Web Linking”, RCF5988, October 2010. https://tools.ietf.org/html/rfc5988

2 Background Information

In this section, we provide high-level information about the HTTP Session Object data model that is necessary to fully understand the specification details given in Section 3.

2.1 Cyber Observables

A cyber observable is a dynamic event or a stateful property that occurs, or may occur, in the operational cyber domain. Examples of stateful properties include the value of a registry key, the MD5 hash of a file, and an IP address. Examples of events include the deletion of a file, the receipt of an HTTP GET request, and the creation of a remote thread.

A cyber observable is different than a cyber indicator. A cyber observable is a statement of fact, capturing what was observed or could be observed in the cyber operational domain. Cyber indicators are cyber observable patterns, such as a registry key value associated with a known bad actor or a spoofed email address used on a particular date.

2.2 Objects

Objects in CybOX are individual data models for characterizing a particular cyber entity, such as a Windows registry key, or an Email Message. Accordingly, each release of the CybOX Language includes a particular set of Objects that are part of the release. The data model for each of these Objects is defined by its own specification that describes the context-specific classes and properties that compose the Object.

3 Data Model

3.1 HTTPSessionObjectType Class

The HTTPSessionObjectType class is intended to capture the details of an HTTP session. The UML diagram corresponding to the HTTPSessionObjectType class is shown in **Figure 3-1**.

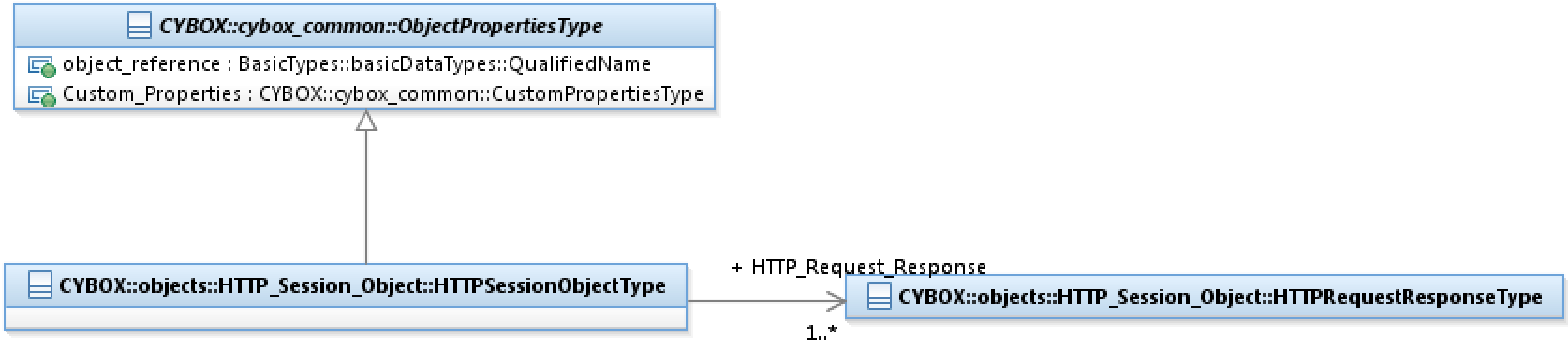
CYBOX::cybox_common::ObjectPropertiesType
object_reference : BasicTypes::basicDataTypes::QualifiedName
Custom_Properties : CYBOX::cybox_common::CustomPropertiesType
CYBOX::objects::HTTP_Session_Object::HTTPSessionObjectType
+ HTTP_Request_Response
1..*
CYBOX::objects::HTTP_Session_Object::HTTPRequestResponseType

Figure 3-1. UML diagram of the HTTPSessionObjectType class

The property table of the HTTPSessionObjectType class is given in **Table 3-1**.

Table 3-1. Properties of the HTTPSessionObjectType class

Name	Type	Multiplicity	Description
HTTP_Request_Response	HTTPRequestResponseType	1..*	The HTTP_Request_Response property specifies a single HTTP Request/Response pair.

3.2 HTTPRequestResponseType Class

The `HTTPRequestResponseType` class captures a single HTTP request/response pair. The UML diagram corresponding to the `HTTPRequestResponseType` class is shown in Figure 3-2.

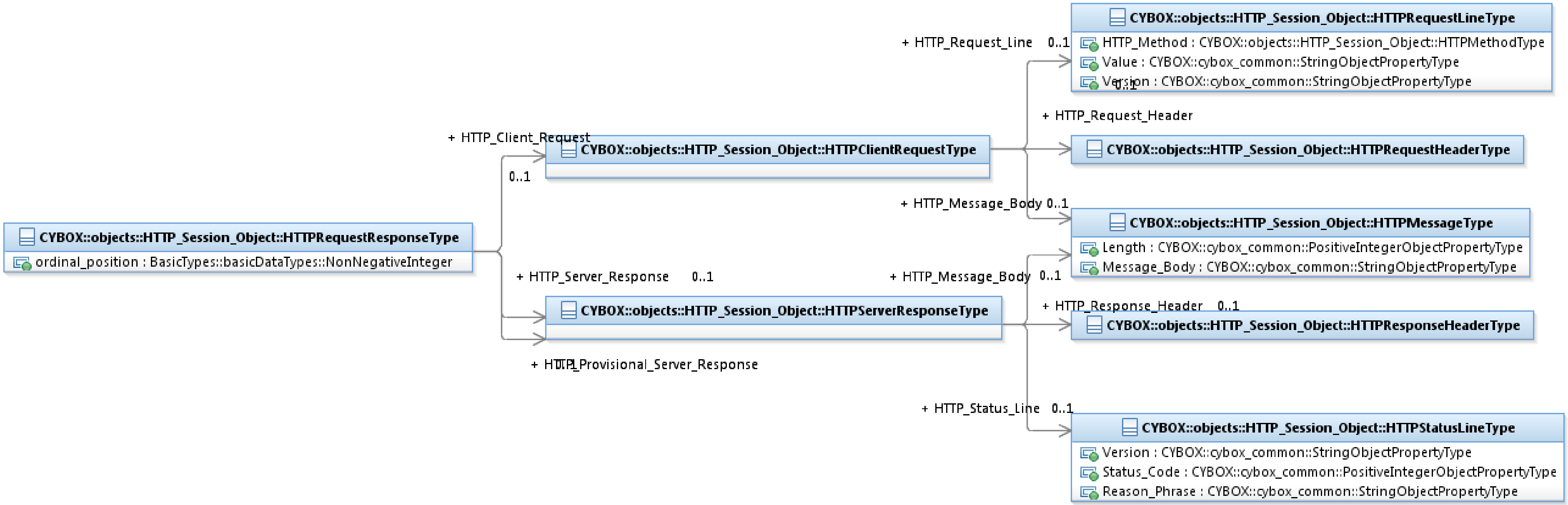
CYBOX::objects::HTTP_Session_Object::HTTPRequestResponseType
ordinal_position : BasicTypes::basicDataTypes::NonNegativeInteger
+ HTTP_Client_Request
0..1
CYBOX::objects::HTTP_Session_Object::HTTPClientRequestType
+ HTTP_Server_Response
0..1
CYBOX::objects::HTTP_Session_Object::HTTPServerResponseType
+ HTTP_Provisional_Server_Response
0..1
+ HTTP_Request_Line
0..1
CYBOX::objects::HTTP_Session_Object::HTTPRequestLineType
HTTP_Method : CYBOX::objects::HTTP_Session_Object::HTTPMethodType
Value : CYBOX::cybox_common::StringObjectPropertyType
Version : CYBOX::cybox_common::StringObjectPropertyType
0..1
+ HTTP_Request_Header
CYBOX::objects::HTTP_Session_Object::HTTPRequestHeaderType
+ HTTP_Message_Body 0..1
CYBOX::objects::HTTP_Session_Object::HTTPMessageType
Length : CYBOX::cybox_common::PositiveIntegerObjectPropertyType
Message_Body : CYBOX::cybox_common::StringObjectPropertyType
+ HTTP_Message_Body 0..1
+ HTTP_Response_Header 0..1
CYBOX::objects::HTTP_Session_Object::HTTPResponseHeaderType
+ HTTP_Status_Line 0..1
CYBOX::objects::HTTP_Session_Object::HTTPStatusLineType
Version : CYBOX::cybox_common::StringObjectPropertyType
Status_Code : CYBOX::cybox_common::PositiveIntegerObjectPropertyType
Reason_Phrase : CYBOX::cybox_common::StringObjectPropertyType

Figure 3-2. UML diagram for the `HTTPRequestResponseType` class

The property table of the `HTTPRequestResponseType` class is given in **Table 3-2**.

Table 3-2. Properties of the `HTTPRequestResponseType` class

Name	Type	Multiplicity	Description

ordinal_position	`basicDataTypes:`
	`NonNegativeInteger`
		0..1	The `ordinal_position` property specifies the ordinal positioning of the HTTP request/response pair in the context of the HTTP session. This may be useful in certain cases for preserving observed HTTP request/response ordering.

HTTP_Client_Request	`HTTPClientRequestType`	0..1	The `HTTP_Client_Request` property specifies the HTTP client request portion of a single HTTP request/response pair.
HTTP_Provisional_ Server_Response	`HTTPServerResponseType`	0..1	The `HTTP_Provisional_Server_Response` property specifies an HTTP provisional server response that was sent before the regular HTTP response (captured in the `HTTP_Server_Response` property).
HTTP_Server_Response	`HTTPServerResponseType`	0..1	The `HTTP_Server_Response` property specifies the HTTP server response portion of a single HTTP request/response pair.

3.3 HTTPClientRequestType Class

The `HTTPClientRequestType` class captures the details of an HTTP client request.

The property table of the `HTTPClientRequestType` class is given in **Table 3-3**.

Table 3-3. Properties of the `HTTPClientRequestType` class

Name	Type	Multiplicity	Description
HTTP_Request_Line	`HTTPRequestLineType`	0..1	The `HTTP_Request_Line` property specifies the HTTP request line of the HTTP client request.
HTTP_Request_Header	`HTTPRequestHeaderType`	0..1	The `HTTP_Request_Header` property specifies all of the HTTP header fields that may be found in the HTTP client request.
HTTP_Message_Body	`HTTPMessageType`	0..1	The `HTTP_Message_Body` property specifies the optional message body that may be included in the HTTP client request.

3.4 HTTPServerResponseType Class

The HTTPServerResponseType class captures the details of an HTTP server response.

The property table of the HTTPServerResponseType class is given in **Table 3-4**.

Table 3-4. Properties of the HTTPServerResponseType class

Name	Type	Multiplicity	Description
HTTP_Status_Line	HTTPStatusLineType	0..1	The HTTP_Status_Line property captures the status line returned as part of the HTTP server response.
HTTP_Response_Header	HTTPResponseHeaderType	0..1	The HTTP_Response_Header property captures the details of the HTTP Header returned as part of the HTTP server response.
HTTP_Message_Body	HTTPMessageType	0..1	The HTTP_Message_Body property captures the HTTP message body returned as part of the HTTP server response.

3.5 HTTPRequestLineType Class

The HTTPRequestLineType class captures a single HTTP request line.

The property table of the HTTPRequestLineType class is given in **Table 3-5**.

Table 3-5. Properties of the HTTPRequestLineType class

Name	Type	Multiplicity	Description
HTTP_Method	HTTPMethodType	0..1	The HTTP_Method property captures the HTTP method portion of the HTTP request line.
Value	cyboxCommon:	0..1	The Value property captures the value (typically a resource path) portion

	StringObjectPropertyType		of the HTTP request line.
Version	cyboxCommon: StringObjectPropertyType	0..1	The Version property captures the HTTP version portion of the HTTP request line.

3.6 HTTPRequestHeaderType Class

The HTTPRequestHeaderType class captures the raw and/or parsed header of an HTTP request.

The property table of the HTTPRequestHeaderType class is given in **Table 3-6**.

Table 3-6. Properties of the HTTPRequestHeaderType class

Name	Type	Multiplicity	Description
Raw_Header	cyboxCommon: StringObjectPropertyType	0..1	The Raw_Header property captures the HTTP request header as a raw, unparsed string.
Parsed_Header	HTTPRequestHeaderFieldsType	0..1	The Parsed_Header property captures the HTTP request header as a set of parsed HTTP header fields.

3.7 HTTPRequestHeaderFieldsType Class

The HTTPRequestHeaderFieldsType class captures parsed HTTP request header fields.

The property table of the HTTPRequestHeaderFieldsType class is given in **Table 3-7**.

Table 3-7. Properties of the HTTPRequestHeaderFieldsType class

Name	Type	Multiplicity	Description

Accept	cyboxCommon: StringObjectPropertyType	0..1	The Accept property specifies the HTTP Request Accept header field, which defines the Content-Types that are acceptable.
Accept_Charset	cyboxCommon: StringObjectPropertyType	0..1	The Accept_Charset property specifies the HTTP Request Accept-Charset header field, which defines the character sets that are acceptable.
Accept_Language	cyboxCommon: StringObjectPropertyType	0..1	The Accept_Language property specifies the HTTP Request Accept-Language header field, which defines the acceptable languages for response.
Accept_Datetime	cyboxCommon: StringObjectPropertyType	0..1	The Accept_Datetime property specifies the HTTP Request Accept-Datetime header field, which defines the acceptable version time.
Accept_Encoding	cyboxCommon: StringObjectPropertyType	0..1	The Accept_Encoding property specifies the HTTP Request Accept-Encoding header field, which defines the acceptable encodings.
Authorization	cyboxCommon: StringObjectPropertyType	0..1	The Authorization property specifies the HTTP Request Authorization header field, which defines the authentication credentials for use in HTTP authentication.
Cache_Control	cyboxCommon: StringObjectPropertyType	0..1	The Cache_Control property specifies the HTTP Request Cache-Control header field, which defines the directives that MUST be obeyed by all caching mechanisms along the request/response chain.
Connection	cyboxCommon: StringObjectPropertyType	0..1	The Connection property specifies the HTTP Request Connection header field, which defines the type of connection that the user-agent would prefer.
Cookie	cyboxCommon:	0..1	The Cookie property specifies the HTTP Request Cookie header

	StringObjectPropertyType		field, which defines the HTTP cookie previously sent by the server.
Content_Length	cyboxCommon: IntegerObjectPropertyType	0..1	The Content_Length property specifies the HTTP Request Content-Length header field, which defines the length of the request body in octets.
Content_MD5	cyboxCommon: StringObjectPropertyType	0..1	The Content_MD5 property specifies the HTTP Request Content-MD5 header field, which defines a Base64 encoded binary MD5 sum of the content of the request body.
Content_Type	cyboxCommon: StringObjectPropertyType	0..1	The Content_Type property specifies the HTTP Request Content-Type header field, which defines the MIME type of the body of the request (used with POST and PUT requests).
Date	cyboxCommon: DateTimeObjectPropertyType	0..1	The Date property specifies the HTTP Request Date header field, which defines the date and time that the message was sent.
Expect	cyboxCommon: StringObjectPropertyType	0..1	The Expect property specifies the HTTP Request Expect header field, which defines the particular server behaviors that are required by the client.
From	AddressObj: AddressObjectType	0..1	The From property specifies the HTTP Request From header field, which defines the email address of the user making the request.
Host	HostFieldType	0..1	The Host property specifies the HTTP Request Host header field, which specifies the domain name of the server and the TCP port number on which the server is listening.
If_Match	cyboxCommon: StringObjectPropertyType	0..1	The If_Match property specifies the HTTP Request If-Match header field, which allows the action to be performed if the client supplied entity matches the same entity on the server.

If_Modified_Since	`cyboxCommon:DateTimeObjectPropertyType`	0..1	The `If_Modified_Since` property specifies the HTTP Request If-Modified-Since header field, which allows a 304 Not Modified response to be returned if content is unchanged since the input date/time.
If_None_Match	`cyboxCommon:StringObjectPropertyType`	0..1	The `If_None_Match` property specifies the HTTP Request If-None-Match header field, which allows the action to be performed only if the client supplied entity does not match the same entity on the server.
If_Range	`cyboxCommon:StringObjectPropertyType`	0..1	The `If_Range` property specifies the HTTP Request If-Range header field, which allows the client to request the part(s) of the entity that they are missing, or otherwise the new entity.
If_Unmodified_Since	`cyboxCommon:DateTimeObjectPropertyType`	0..1	The `If_Unmodified_Since` property specifies the HTTP Request If-Unmodified-Since header field, which allows a response to be sent only if the entity has not been modified since a specific date/time.
Max_Forwards	`cyboxCommon:IntegerObjectPropertyType`	0..1	The `Max_Forwards` property specifies the HTTP Request Max-Forwards header field, which defines the maximum number of times the message can be forwarded through proxies or gateways.
Pragma	`cyboxCommon:StringObjectPropertyType`	0..1	The `Pragma` property specifies the HTTP Request Pragma header field, which defines any implementation-specific values that may apply to any recipient across the request-response chain.
Proxy_Authorization	`cyboxCommon:StringObjectPropertyType`	0..1	The `Proxy_Authorization` property specifies the HTTP Request Proxy-Authorization header field, which defines the authorization credentials for connecting to a proxy.
Range	`cyboxCommon:StringObjectPropertyType`	0..1	The `Range` property specifies the HTTP Request Range header field, which defines the range, in bytes, for requesting only part of an entity (bytes are numbered from 0).

Referer	`URIObj:URIObjectType`	0..1	The `Referer` property specifies the HTTP Request Range Referer field, which defines the address of the previous web page from which a link to the currently requested page was followed.
TE	`cyboxCommon:StringObjectPropertyType`	0..1	The `TE` property specifies the HTTP Request TE field, which defines the transfer encodings the user agent is willing to accept.
User_Agent	`cyboxCommon:StringObjectPropertyType`	0..1	The `User_Agent` property specifies the HTTP Request User-Agent field, which defines the user agent string of the user agent.
Via	`cyboxCommon:StringObjectPropertyType`	0..1	The `Via` property specifies the HTTP Request Via field, which defines any proxies through which the request was sent.
Warning	`cyboxCommon:StringObjectPropertyType`	0..1	The `Warning` property specifies the HTTP Request Warning field, which defines any general warnings about possible problems with the entity body.
DNT	`cyboxCommon:StringObjectPropertyType`	0..1	The `DNT` property specifies the non-standard HTTP Request DNT field, which is typically used to request that a web application disable their tracking of a user.
X_Requested_With	`cyboxCommon:StringObjectPropertyType`	0..1	The `X_Requested_With` property specifies the non-standard HTTP Request X-Requested-With field, which is typically used to identify Ajax requests.
X_Forwarded_For	`cyboxCommon:StringObjectPropertyType`	0..1	The `X_Forwarded_For` property specifies the non-standard HTTP Request X-Forwarded-For field, which is typically used to identify the originating IP address of a client connecting to a web server through an HTTP proxy or load balancer.
X_Forwarded_Proto	`cyboxCommon:StringObjectPropertyType`	0..1	The `X_Forwarded_Proto` property specifies the non-standard HTTP Response X-Forwarded-Proto field, which identifies the

			originating protocol of an HTTP request.
X_ATT_DeviceId	cyboxCommon:StringObjectPropertyType	0..1	The X_ATT_DeviceId property specifies the non-standard HTTP Request X-ATT-DeviceId field, which is typically used to identify the make, model, and firmware of AT&T devices.
X_Wap_Profile	URIObj:URIObjectType	0..1	The X_Wap_Profile property specifies the non-standard HTTP Request X-Wap-Profile field, which is typically used to link to an XML file on the Internet with a full description and details about the device currently connecting.

3.8 HTTPResponseHeaderType Class

The HTTPResponseHeaderType class captures the raw and/or parsed header of an HTTP response.

The property table of the HTTPResponseHeaderType class is given in **Table 3-8**.

Table 3-8. Properties of the HTTPResponseHeaderType class

Name	**Type**	**Multiplicity**	**Description**
Raw_Header	cyboxCommon: StringObjectPropertyType	0..1	The Raw_Header property captures the HTTP response header as a raw, unparsed string.
Parsed_Header	HTTPResponseHeaderFieldsType	0..1	The Parsed_Header property captures the HTTP response header as a set of parsed HTTP header fields.

3.9 HTTPResponseHeaderFieldsType Class

The HTTPResponseHeaderFieldsType class captures parsed HTTP request header fields.

The property table of the HTTPResponseHeaderFieldsType class is given in **Table 3-9**.

Table 3-9. Properties of the `HTTPResponseHeaderFieldsType` class

Name	Type	Multiplicity	Description
Access_Control_Allow_Origin	`cyboxCommon: StringObjectPropertyType`	0..1	The `Access_Control_Allow_Origin` property specifies the HTTP Response Access-Control-Allow-Origin header field, which defines which web sites can participate in cross-origin resource sharing.
Accept_Ranges	`cyboxCommon: StringObjectPropertyType`	0..1	The `Accept_Ranges` property specifies the HTTP Response Accept-Ranges header field, which defines the partial content range types this server supports.
Age	`cyboxCommon: IntegerObjectPropertyType`	0..1	The `Age` property specifies the HTTP Response Authorization header field, which defines the age the object has been in a proxy cache, in seconds.
Cache_Control	`cyboxCommon: StringObjectPropertyType`	0..1	The `Cache_Control` property specifies the HTTP Response Cache-Control header field, which tells all caching mechanisms from server to client whether they may cache this object.
Connection	`cyboxCommon: StringObjectPropertyType`	0..1	The `Connection` property specifies the HTTP Response Connection header field, which specifies the options that are desired for the connection.
Content_Encoding	`cyboxCommon: StringObjectPropertyType`	0..1	The `Content_Encoding` property specifies the HTTP Response Content-Encoding header field, which defines the type of encoding used on the data.
Content_Language	`cyboxCommon: StringObjectPropertyType`	0..1	The `Content_Language` property specifies the HTTP Response Content-Language header field, which defines the language the content is in.

Content_Length	`cyboxCommon:`
	`IntegerObjectPropertyType`
		0..1	The `Content_Length` property specifies the HTTP Response Content-Length header field, which defines the length of the request body in octets.

Content_Location	`cyboxCommon:`
	`StringObjectPropertyType`
		0..1	The `Content_Location` property specifies the HTTP Response Content-Location header field, which defines an alternate location for the returned data.

Content_MD5	`cyboxCommon:`
	`StringObjectPropertyType`
		0..1	The `Content_MD5` property specifies the HTTP Response Content-MD5 header field, which defines the base64-encoded binary MD5 sum of the content of the response.

Content_Disposition	`cyboxCommon:`
	`StringObjectPropertyType`
		0..1	The `Content_Disposition` property specifies the HTTP Response Content-Disposition header field, which provides a means for the origin server to suggest a default filename if the user requests that the content is saved to a file.

Content_Range	`cyboxCommon:`
	`StringObjectPropertyType`
		0..1	The `Content_Range` property specifies the HTTP Response Content-Range header field, which defines where in a full body message the partial message belongs.

Content_Type	`cyboxCommon:`
	`StringObjectPropertyType`
		0..1	The `Content_Type` property specifies the HTTP Response Content-Type header field, which defines the MIME type of the content.

Date	`cyboxCommon:`
	`DateTimeObjectPropertyType`
		0..1	The `Date` property specifies the HTTP Request Date header field, which defines the date and time that the message was sent.

ETag	`cyboxCommon:`	0..1	The `ETag` property specifies the HTTP Response ETag header field, which defines an identifier for a specific

	StringObjectPropertyType		version of a resource, often a message digest.
Expires	cyboxCommon:DateTimeObjectPropertyType	0..1	The Expires property specifies the HTTP Response Expires header field, which defines the date/time after which the response is considered stale.
Last_Modified	cyboxCommon:DateTimeObjectPropertyType	0..1	The Last_Modified property specifies the HTTP Response Last-Modified header field, which defines the date/time for the requested object, in **[RFC 2822]** format.
Link	cyboxCommon:StringObjectPropertyType	0..1	The Link property specifies the HTTP Response Link header field, which defines a typed relationship with another resource, where the relation type is defined by **[RFC 5988].**
Location	URIObj:URIObjectType	0..1	The Location property specifies the HTTP Response Location header field, which defines the location used in redirection, or when a new resource has been created.
P3P	cyboxCommon:StringObjectPropertyType	0..1	The P3P property specifies the HTTP Response P3P header field, which sets P3P policy to be used by the browser.
Pragma	cyboxCommon:StringObjectPropertyType	0..1	The Pragma property specifies the HTTP Response Pragma header field, which defines any implementation-specific values that may apply to any recipient across the request-response chain.
Proxy_Authenticate	cyboxCommon:StringObjectPropertyType	0..1	The Proxy_Authenticate property specifies the HTTP Response Proxy-Authenticate header field, which defines the type of authentication necessary to access the proxy.

Refresh	`cyboxCommon:StringObjectPropertyType`	0..1	The `Refresh` property specifies the HTTP Response Refresh header field, which specifies a given interval, in seconds, after which the current page should be refreshed.
Retry_After	`cyboxCommon:IntegerObjectPropertyType`	0..1	The `Retry_After` property specifies the HTTP Response Retry-After header field, which defines the period, in seconds, after which the client should try again if an entity is temporarily unavailable.
Server	`cyboxCommon:StringObjectPropertyType`	0..1	The `Server` property specifies the HTTP Response Server field, which defines a name for the responding server.
Set_Cookie	`cyboxCommon:StringObjectPropertyType`	0..1	The `Set_Cookie` property specifies the HTTP Response Set-Cookie field, which defines an HTTP cookie.
Strict_Transport_Security	`cyboxCommon:StringObjectPropertyType`	0..1	The `Strict_Transport_Security` property specifies the HTTP response Strict-Transport-Security field, which defines the HSTS Policy informing the HTTP client how long to cache the HTTPS-only policy and whether this applies to subdomains.
Trailer	`cyboxCommon:StringObjectPropertyType`	0..1	The `Trailer` property specifies the HTTP Response Trailer field, which indicates that the given set of header fields is present in the trailer of a message encoded with chunked transfer-coding.
Transfer_Encoding	`cyboxCommon:StringObjectPropertyType`	0..1	The `Transfer_Encoding` property specifies the HTTP Response Transfer-Encoding field, which defines the form of encoding used to safely transfer the entity to the user.

Vary	cyboxCommon: StringObjectPropertyType	0..1	The Vary property specifies the HTTP Response Vary field, which informs downstream proxies on how to match future request headers to decide whether the cached response can be used rather than requesting a fresh one from the origin server.
Via	cyboxCommon: StringObjectPropertyType	0..1	The Via property specifies the HTTP Response Via field, which informs the client of proxies through which the response was sent.
Warning	cyboxCommon: StringObjectPropertyType	0..1	The Warning property specifies the HTTP Response Warning field, which defines any general warnings about possible problems with the entity body.
WWW_Authenticate	cyboxCommon: StringObjectPropertyType	0..1	The WWW_Authenticate property specifies the HTTP Response WWW-Authenticate field, which defines the authentication scheme that should be used to access the requested entity.
X_Frame_Options	cyboxCommon: StringObjectPropertyType	0..1	The X_Frame_Options property specifies the non-standard HTTP Response X-Frame-Options field, which is used as a form of clickjacking protection, supporting no rendering within a frame and no rendering if origin mismatch.
X_XSS_Protection	cyboxCommon: StringObjectPropertyType	0..1	The X_XSS_Protection property specifies the non-standard HTTP Response X-XSS-Protection field, which is used as a cross-site scripting (XSS) filter.
X_Content_Type_Options	cyboxCommon: StringObjectPropertyType	0..1	The X_Content_Type_Options property specifies the non-standard HTTP Response X-Content-Type-Options field, which supports the 'nosniff' parameter to prevent the MIME-sniffing of a response away from the declared content type.

X_Powered_By	`cyboxCommon:`
	`StringObjectPropertyType`
		0..1	The `X_Powered_By` property specifies the non-standard HTTP Response X-Powered-By field, which specifies the technology supporting the web application running on the server.

X_UA_Compatible	`cyboxCommon:`
	`StringObjectPropertyType`
		0..1	The `X_UA_Compatible` property specifies the non-standard HTTP Response X-UA-Compatible field, which is used to recommend the preferred rendering engine to use to display the content.

3.10 HTTPMessageType Class

The `HTTPMessageType` class captures a single HTTP message body and its length.

The property table of the `HTTPMessageType` class is given in **Table 3-10**.

Table 3-10. Properties of the `HTTPMessageType` class

Name	Type	Multiplicity	Description

Length	`cyboxCommon:`
	`PositiveIntegerObjectPropertyType`
		0..1	The `Length` property captures the length of the HTTP message body, in bytes.

Message_Body	`cyboxCommon:`
	`StringObjectPropertyType`
		0..1	The `Message_Body` property captures the data contained in the HTTP message body.

3.11 HTTPStatusLineType Class

The `HTTPStatusLineType` class captures a single HTTP response status line.

The property table of the `HTTPStatusLineType` class is given in **Table 3-11**.

Table 3-11. Properties of the `HTTPStatusLineType` class

Name	Type	Multiplicity	Description

Version	`cyboxCommon:`
	`StringObjectPropertyType`
		0..1	The `Version` property captures the HTTP version portion of the HTTP status line.

Status_Code	`cyboxCommon:`
	`PositiveIntegerObjectPropertyType`
		0..1	The `Status_Code` property captures the HTTP status code portion of the HTTP status line.

Reason_Phrase	`cyboxCommon:`
	`StringObjectPropertyType`
		0..1	The `Reason_Phrase` property captures the HTTP reason phrase portion of the HTTP status line.

3.12 HostFieldType Class

The `HostFieldType` class captures the details of the HTTP request Host header field.

The property table of the `HostFieldType` class is given in **Table 3-12**.

Table 3-12. Properties of the `HostFieldType` class

Name	Type	Multiplicity	Description
Domain_Name	`URIObj:URIObjectType`	0..1	The `Domain_Name` property specifies the domain name of the server.
Port	`PortObj:PortObjectType`	0..1	The `Port` property specifies the TCP port number on which the server is listening.

3.13 HTTPMethodType Data Type

The HTTPMethodType data type specifies the HTTP method type. Its core value SHOULD be a literal found in the HTTPMethodEnum enumeration. Its base type is the BaseObjectPropertyType data type, in order to permit complex (i.e., regular-expression based) specifications.

3.14 HTTPMethodEnum Enumeration

The literals of the HTTPMethodEnum enumeration are given in **Table 3-13**.

Table 3-13. Literals of the HTTPMethodEnum enumeration

Enumeration Literal	Description
GET	
POST	
HEAD	
PUT	

4 Conformance

Implementations have discretion over which parts (components, properties, extensions, controlled vocabularies, etc.) of CybOX they implement (e.g., Observable/Object).

[1] Conformant implementations must conform to all normative structural specifications of the UML model or additional normative statements within this document that apply to the portions of CybOX they implement (e.g., implementers of the entire Observable class must conform to all normative structural specifications of the UML model regarding the Observable class or additional normative statements contained in the document that describes the Observable class).

[2] Conformant implementations are free to ignore normative structural specifications of the UML model or additional normative statements within this document that do not apply to the portions of CybOX they implement (e.g., non-implementers of any particular properties of the Observable class are free to ignore all normative structural specifications of the UML model regarding those properties of the Observable class or additional normative statements contained in the document that describes the Observable class).

The conformance section of this document is intentionally broad and attempts to reiterate what already exists in this document.

Appendix A. Acknowledgments

The following individuals have participated in the creation of this specification and are gratefully acknowledged.

Aetna

David Crawford

AIT Austrian Institute of Technology

Roman Fiedler

Florian Skopik

Australia and New Zealand Banking Group (ANZ Bank)

Dean Thompson

Blue Coat Systems, Inc.

Owen Johnson

Bret Jordan

Century Link

Cory Kennedy

CIRCL

Alexandre Dulaunoy

Andras Iklody

Raphaël Vinot

Citrix Systems

Joey Peloquin

Dell

Will Urbanski

Jeff Williams

DTCC

Dan Brown

Gordon Hundley

Chris Koutras

EMC

Robert Griffin

Jeff Odom

Ravi Sharda

Financial Services Information Sharing and Analysis Center (FS-ISAC)

David Eilken

Chris Ricard

Fortinet Inc.

Gavin Chow

Kenichi Terashita

Airbus Group SAS

Joerg Eschweiler

Marcos Orallo

Anomali

Ryan Clough

Wei Huang

Hugh Njemanze

Katie Pelusi

Aaron Shelmire

Jason Trost

Bank of America

Alexander Foley

Center for Internet Security (CIS)

Sarah Kelley

Check Point Software Technologies

Ron Davidson

Cisco Systems

Syam Appala

Ted Bedwell

David McGrew

Pavan Reddy

Omar Santos

Jyoti Verma

Cyber Threat Intelligence Network, Inc. (CTIN)

Doug DePeppe

Jane Ginn

Ben Othman

DHS Office of Cybersecurity and Communications (CS&C)

Richard Struse

Marlon Taylor

EclecticIQ

Marko Dragoljevic

Joep Gommers

Sergey Polzunov

Rutger Prins

Fujitsu Limited
Neil Edwards
Frederick Hirsch
Ryusuke Masuoka
Daisuke Murabayashi

Google Inc.
Mark Risher

Hitachi, Ltd.
Kazuo Noguchi
Akihito Sawada
Masato Terada

iboss, Inc.
Paul Martini

Individual
Jerome Athias
Peter Brown
Elysa Jones
Sanjiv Kalkar
Bar Lockwood
Terry MacDonald
Alex Pinto

Intel Corporation
Tim Casey
Kent Landfield

JPMorgan Chase Bank, N.A.
Terrence Driscoll
David Laurance

LookingGlass
Allan Thomson
Lee Vorthman

Mitre Corporation
Greg Back
Jonathan Baker
Sean Barnum
Desiree Beck
Nicole Gong
Jasen Jacobsen
Ivan Kirillov
Richard Piazza
Jon Salwen
Charles Schmidt
Andrei Sîrghi
Raymon van der Velde

eSentire, Inc.
Jacob Gajek

FireEye, Inc.
Phillip Boles
Pavan Gorakav
Anuj Kumar
Shyamal Pandya
Paul Patrick
Scott Shreve

Fox-IT
Sarah Brown

Georgetown University
Eric Burger

Hewlett Packard Enterprise (HPE)
Tomas Sander

IBM
Peter Allor
Eldan Ben-Haim
Sandra Hernandez
Jason Keirstead
John Morris
Laura Rusu
Ron Williams

IID
Chris Richardson

Integrated Networking Technologies, Inc.
Patrick Maroney

Johns Hopkins University Applied Physics Laboratory
Karin Marr
Julie Modlin
Mark Moss
Pamela Smith

Kaiser Permanente
Russell Culpepper
Beth Pumo

Lumeta Corporation
Brandon Hoffman

MTG Management Consultants, LLC.
James Cabral

Emmanuelle Vargas-Gonzalez
John Wunder

National Council of ISACs (NCI)
Scott Algeier
Denise Anderson
Josh Poster

NEC Corporation
Takahiro Kakumaru

North American Energy Standards Board
David Darnell

Object Management Group
Cory Casanave

Palo Alto Networks
Vishaal Hariprasad

Queralt, Inc.
John Tolbert

Resilient Systems, Inc.
Ted Julian

Securonix
Igor Baikalov

Siemens AG
Bernd Grobauer

Soltra
John Anderson
Aishwarya Asok Kumar
Peter Ayasse
Jeff Beekman
Michael Butt
Cynthia Camacho
Aharon Chernin
Mark Clancy
Brady Cotton
Trey Darley
Mark Davidson
Paul Dion
Daniel Dye
Robert Hutto
Raymond Keckler
Ali Khan
Chris Kiehl
Clayton Long

National Security Agency
Mike Boyle
Jessica Fitzgerald-McKay

New Context Services, Inc.
John-Mark Gurney
Christian Hunt
James Moler
Daniel Riedel
Andrew Storms

OASIS
James Bryce Clark
Robin Cover
Chet Ensign

Open Identity Exchange
Don Thibeau

PhishMe Inc.
Josh Larkins

Raytheon Company-SAS
Daniel Wyschogrod

Retail Cyber Intelligence Sharing Center (R-CISC)
Brian Engle

Semper Fortis Solutions
Joseph Brand

Splunk Inc.
Cedric LeRoux
Brian Luger
Kathy Wang

TELUS
Greg Reaume
Alan Steer

Threat Intelligence Pty Ltd
Tyron Miller
Andrew van der Stock

ThreatConnect, Inc.
Wade Baker
Cole Iliff
Andrew Pendergast
Ben Schmoker
Jason Spies

TruSTAR Technology
Chris Roblee

Michael Pepin
Natalie Suarez
David Waters
Benjamin Yates

Symantec Corp.
Curtis Kostrosky

The Boeing Company
Crystal Hayes

ThreatQuotient, Inc.
Ryan Trost

U.S. Bank
Mark Angel
Brad Butts
Brian Fay
Mona Magathan
Yevgen Sautin

US Department of Defense (DoD)
James Bohling
Eoghan Casey
Gary Katz
Jeffrey Mates

VeriSign
Robert Coderre
Kyle Maxwell
Eric Osterweil

United Kingdom Cabinet Office
Iain Brown
Adam Cooper
Mike McLellan
Chris O'Brien
James Penman
Howard Staple
Chris Taylor
Laurie Thomson
Alastair Treharne
Julian White
Bethany Yates

US Department of Homeland Security
Evette Maynard-Noel
Justin Stekervetz

ViaSat, Inc.
Lee Chieffalo
Wilson Figueroa
Andrew May

Yaana Technologies, LLC
Anthony Rutkowski

The authors would also like to thank the larger CybOX Community for its input and help in reviewing this document.

Appendix B. Revision History

Revision	Date	Editor	Changes Made
wd01	15 December 2015	Desiree Beck Trey Darley Ivan Kirillov Rich Piazza	Initial transfer to OASIS template