CybOXTM Version 2.1.1 Part 28: HTTP Session Object

Working Draft 01

15 December 2015

Technical Committee:

[OASIS Cyber Threat Intelligence (CTI) TC](https://www.oasis-open.org/committees/cti)

Chair:

Richard Struse ([Richard.Struse@HQ.DHS.GOV](mailto:Richard.Struse@HQ.DHS.GOV)), [DHS Office of Cybersecurity and Communications (CS&C)](http://www.dhs.gov/office-cybersecurity-and-communications)

Editors:

Desiree Beck ([dbeck@mitre.org](mailto:ikirillov@mitre.org)), [MITRE Corporation](http://www.mitre.org/)

Trey Darley ([trey@soltra.com](mailto:trey@soltra.com)), [Soltra](http://www.soltra.com/)

Ivan Kirillov ([ikirillov@mitre.org](mailto:ikirillov@mitre.org)), [MITRE Corporation](http://www.mitre.org/)

Rich Piazza ([rpiazza@mitre.org](mailto:ikirillov@mitre.org)), [MITRE Corporation](http://www.mitre.org/)

Additional artifacts:

This prose specification is one component of a Work Product which consists of:

* *CybOX™ Version 2.1.1 Part 01: Overview*. [URI]
* *CybOX™ Version 2.1.1 Part 02: Common*. [URI]
* *CybOX™ Version 2.1.1 Part 03: Core*. [URI]
* *CybOX™ Version 2.1.1 Part 04: Default Extensions*. [URI]
* *CybOX™ Version 2.1.1 Part 05: Default Vocabularies*. [URI]
* *CybOX™ Version 2.1.1 Part 06: UML Model*. [URI]
* *CybOX™ Version 2.1.1 Part 07: API Object*. [URI]
* *CybOX™ Version 2.1.1 Part 08: ARP Cache Object*. [URI]
* *CybOX™ Version 2.1.1 Part 09: AS Object*. [URI]
* *CybOXTM Version 2.1.1 Part 10: Account Object*. [URI]
* *CybOXTM Version 2.1.1 Part 11: Address Object*. [URI]
* *CybOXTM Version 2.1.1 Part 12: Archive File Object*. [URI]
* *CybOXTM Version 2.1.1 Part 13: Artifact Object*. [URI]
* *CybOXTM Version 2.1.1 Part 14: Code Object*. [URI]
* *CybOXTM Version 2.1.1 Part 15: Custom Object*. [URI]
* *CybOXTM Version 2.1.1 Part 16: DNS Cache Object*. [URI]
* *CybOXTM Version 2.1.1 Part 17: DNS Query Object*. [URI]
* *CybOXTM Version 2.1.1 Part 18: DNS Record Object*. [URI]
* *CybOXTM Version 2.1.1 Part 19: Device Object*. [URI]
* *CybOXTM Version 2.1.1 Part 20: Disk Object*. [URI]
* *CybOXTM Version 2.1.1 Part 21: Disk Partition Object*. [URI]
* *CybOXTM Version 2.1.1 Part 22: Domain Name Object*. [URI]
* *CybOXTM Version 2.1.1 Part 23: Email Message Object*. [URI]
* *CybOXTM Version 2.1.1 Part 24: File Object*. [URI]
* *CybOXTM Version 2.1.1 Part 25: GUI Dialogbox Object*. [URI]
* *CybOXTM Version 2.1.1 Part 26: GUI Object*. [URI]
* *CybOXTM Version 2.1.1 Part 27: GUI Window Object*. [URI]
* *CybOXTM Version 2.1.1 Part 28: HTTP Object*. (this document)
* *CybOXTM Version 2.1.1 Part 29: Hostname Session Object*. [URI]
* *CybOXTM Version 2.1.1 Part 30: Image File Object*. [URI]
* *CybOXTM Version 2.1.1 Part 31: Library File Object*. [URI]
* *CybOXTM Version 2.1.1 Part 32: Link Object*. [URI]
* *CybOXTM Version 2.1.1 Part 33: Linux Package Object*. [URI]
* *CybOXTM Version 2.1.1 Part 34: Memory Object*. [URI]
* *CybOXTM Version 2.1.1 Part 35: Mutex Object*. [URI]
* *CybOXTM Version 2.1.1 Part 36: Network Connection Object*. [URI]
* *CybOXTM Version 2.1.1 Part 37: Network Flow Object*. [URI]
* *CybOXTM Version 2.1.1 Part 38: Network Packet Object*. [URI]
* *CybOXTM Version 2.1.1 Part 39: Network Route Entry Object*. [URI]
* *CybOXTM Version 2.1.1 Part 40: Network Route Object*. [URI]
* *CybOXTM Version 2.1.1 Part 41: Network Socket Object*. [URI]
* *CybOXTM Version 2.1.1 Part 42: Network Subnet Object*. [URI]
* *CybOXTM Version 2.1.1 Part 43: PDF File Object*. [URI]
* *CybOXTM Version 2.1.1 Part 44: Pipe Object*. [URI]
* *CybOXTM Version 2.1.1 Part 45: Port Object*. [URI]
* *CybOXTM Version 2.1.1 Part 46: Process Object*. [URI]
* *CybOXTM Version 2.1.1 Part 47: Product Object*. [URI]
* *CybOXTM Version 2.1.1 Part 48: SMS Message Object*. [URI]
* *CybOXTM Version 2.1.1 Part 49: Semaphore Object*. [URI]
* *CybOXTM Version 2.1.1 Part 50: Socket Address Object*. [URI]
* *CybOXTM Version 2.1.1 Part 51: System Object*. [URI]
* *CybOXTM Version 2.1.1 Part 52: URI Object*. [URI]
* *CybOXTM Version 2.1.1 Part 53: URL History Object*. [URI]
* *CybOXTM Version 2.1.1 Part 54: Unix File Object*. [URI]
* *CybOXTM Version 2.1.1 Part 55: Unix Network Route Entry Object*. [URI]
* *CybOXTM Version 2.1.1 Part 56: Unix Pipe Object*. [URI]
* *CybOXTM Version 2.1.1 Part 57: Unix Process Object*. [URI]
* *CybOXTM Version 2.1.1 Part 58: Unix User Account Object*. [URI]
* *CybOXTM Version 2.1.1 Part 59: Unix Volume Object*. [URI]
* *CybOXTM Version 2.1.1 Part 60: User Account Object*. [URI]
* *CybOXTM Version 2.1.1 Part 61: User Session Object*. [URI]
* *CybOXTM Version 2.1.1 Part 62: Volume Object*. [URI]
* *CybOXTM Version 2.1.1 Part 63: Whois Object*. [URI]
* *CybOXTM Version 2.1.1 Part 64: Win Computer Account Object*. [URI]
* *CybOXTM Version 2.1.1 Part 65: Win Critical Section Object*. [URI]
* *CybOXTM Version 2.1.1 Part 66: Win Driver Object*. [URI]
* *CybOXTM Version 2.1.1 Part 67: Win Event Log Object*. [URI]
* *CybOXTM Version 2.1.1 Part 68: Win Event Object*. [URI]
* *CybOXTM Version 2.1.1 Part 69: Win Executable File Object*. [URI]
* *CybOXTM Version 2.1.1 Part 70: Win File Object*. [URI]
* *CybOXTM Version 2.1.1 Part 71: Win Filemapping Object*. [URI]
* *CybOXTM Version 2.1.1 Part 72: Win Handle Object*. [URI]
* *CybOXTM Version 2.1.1 Part 73: Win Hook Object*. [URI]
* *CybOXTM Version 2.1.1 Part 74: Win Kernel Hook Object*. [URI]
* *CybOXTM Version 2.1.1 Part 75: Win Kernel Object*. [URI]
* *CybOXTM Version 2.1.1 Part 76: Win Mailslot Object*. [URI]
* *CybOXTM Version 2.1.1 Part 77: Win Memory Page Region Object*. [URI]
* *CybOXTM Version 2.1.1 Part 78: Win Mutex Object*. [URI]
* *CybOXTM Version 2.1.1 Part 79: Win Network Route Entry Object*. [URI]
* *CybOXTM Version 2.1.1 Part 80: Win Network Share Object*. [URI]
* *CybOXTM Version 2.1.1 Part 81: Win Pipe Object*. [URI]
* *CybOXTM Version 2.1.1 Part 82: Win Prefetch Object*. [URI]
* *CybOXTM Version 2.1.1 Part 83: Win Process Object*. [URI]
* *CybOXTM Version 2.1.1 Part 84: Win Registry Key Object*. [URI]
* *CybOXTM Version 2.1.1 Part 85: Win Semaphore Object*. [URI]
* *CybOXTM Version 2.1.1 Part 86: Win Service Object*. [URI]
* *CybOXTM Version 2.1.1 Part 87: Win System Object*. [URI]
* *CybOXTM Version 2.1.1 Part 88: Win System Restore Object*. [URI]
* *CybOXTM Version 2.1.1 Part 89: Win Task Object*. [URI]
* *CybOXTM Version 2.1.1 Part 90: Win Thread Object*. [URI]
* *CybOXTM Version 2.1.1 Part 91: Win User Account Object*. [URI]
* *CybOXTM Version 2.1.1 Part 92: Win Volume Object*. [URI]
* *CybOXTM Version 2.1.1 Part 93: Win Waitable Timer Object*. [URI]
* *CybOXTM Version 2.1.1 Part 94: X509 Certificate Object*. [URI]

Related work:

This specification is related to:

* *STIXTM Version 1.2.1 (placeholder)*

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 [Working Draft](https://www.oasis-open.org/policies-guidelines/tc-process#dWorkingDraft) (WD) has been produced by one or more TC Members; it has not yet been voted on by the TC or [approved](https://www.oasis-open.org/policies-guidelines/tc-process#committeeDraft) as a Committee Draft (Committee Specification Draft or a Committee Note Draft). The OASIS document [Approval Process](https://www.oasis-open.org/policies-guidelines/tc-process#standApprovProcess) begins officially with a TC vote to approve a WD as a Committee Draft. A TC may approve a Working Draft, revise it, and re-approve it any number of times as a Committee Draft.

URI patterns:

Initial publication URI:  
http://docs.oasis-open.org/cti/stix/v1.2.1/csd01/part1-overview/stix-v1.2.1-csd01-part1-overview.docx

Permanent “Latest version” URI:  
http://docs.oasis-open.org/cti/stix/v1.2.1/stix-v1.2.1-part1-overview.docx

(Managed by OASIS TC Administration; please don’t modify.)

Copyright © OASIS Open 2015. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full [Policy](https://www.oasis-open.org/policies-guidelines/ipr) may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

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.

Portions copyright © United States Government 2012-2015. All Rights Reserved.  
  
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 6](#_Toc437955258)

[1.1 CybOXTM Specification Documents 6](#_Toc437955259)

[1.2 Document Conventions 6](#_Toc437955260)

[1.2.1 Fonts 6](#_Toc437955261)

[1.2.2 UML Package References 7](#_Toc437955262)

[1.2.3 UML Diagrams 7](#_Toc437955263)

[1.2.3.1 Class Properties 7](#_Toc437955264)

[1.2.3.2 Diagram Icons and Arrow Types 7](#_Toc437955265)

[1.2.4 Property Table Notation 8](#_Toc437955266)

[1.2.5 Property and Class Descriptions 8](#_Toc437955267)

[1.3 Terminology 9](#_Toc437955268)

[1.4 Normative References 9](#_Toc437955269)

[2 Background Information 10](#_Toc437955270)

[2.1 Cyber Observables 10](#_Toc437955271)

[2.2 Objects 10](#_Toc437955272)

[3 Data Model 11](#_Toc437955273)

[3.1 HTTPSessionObjectType Class 11](#_Toc437955274)

[3.2 HTTPRequestResponseType Class 12](#_Toc437955275)

[3.3 HTTPClientRequestType Class 13](#_Toc437955276)

[3.4 HTTPServerResponseType Class 13](#_Toc437955277)

[3.5 HTTPRequestLineType Class 14](#_Toc437955278)

[3.6 HTTPRequestHeaderType Class 14](#_Toc437955279)

[3.7 HTTPRequestHeaderFieldsType Class 15](#_Toc437955280)

[3.8 HTTPResponseHeaderType Class 19](#_Toc437955281)

[3.9 HTTPResponseHeaderFieldsType Class 19](#_Toc437955282)

[3.10 HTTPMessageType Class 24](#_Toc437955283)

[3.11 HTTPStatusLineType Class 25](#_Toc437955284)

[3.12 HostFieldType Class 25](#_Toc437955285)

[3.13 HTTPMethodType Class 26](#_Toc437955286)

[3.14 HTTPMethodEnum Enumeration 26](#_Toc437955287)

[4 Conformance 27](#_Toc437955288)

[Acknowledgments 28](#_Toc437955289)

[Revision History 29](#_Toc437955290)

# Introduction

[All text is normative unless otherwise labeled.]

The Cyber Observable Expression (CybOXTM) 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**.

## CybOXTM 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*](#AdditionalArtifacts) 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*](#AdditionalArtifacts) also summarizes the relationship of CybOX to other languages, and outlines general CybOX data model conventions.

## Document Conventions

The following conventions are used in this document.

### 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*](#AdditionalArtifacts).

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 (withsingle 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*

### 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.

### 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.

#### 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.

#### 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. |
|  | This diagram icon indicates an enumeration. |
|  | 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. |

### 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.

### Property and Class Descriptions

Each class and property defined in CybOX is described using the format, “The X property verbY.” 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. |

## 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**]**.

## 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>

# 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**.

## 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.

## 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.

# Data Model

## 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**.

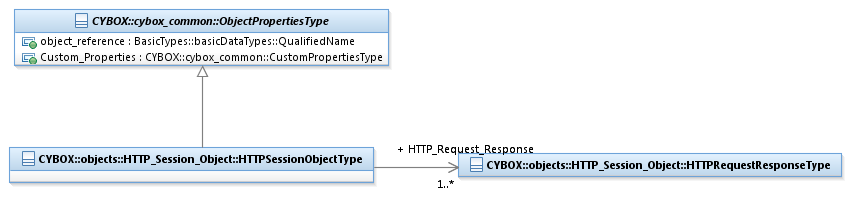


Figure 3‑1. UML diagram of the HTTPSessionObjectType class

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. |

## 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.

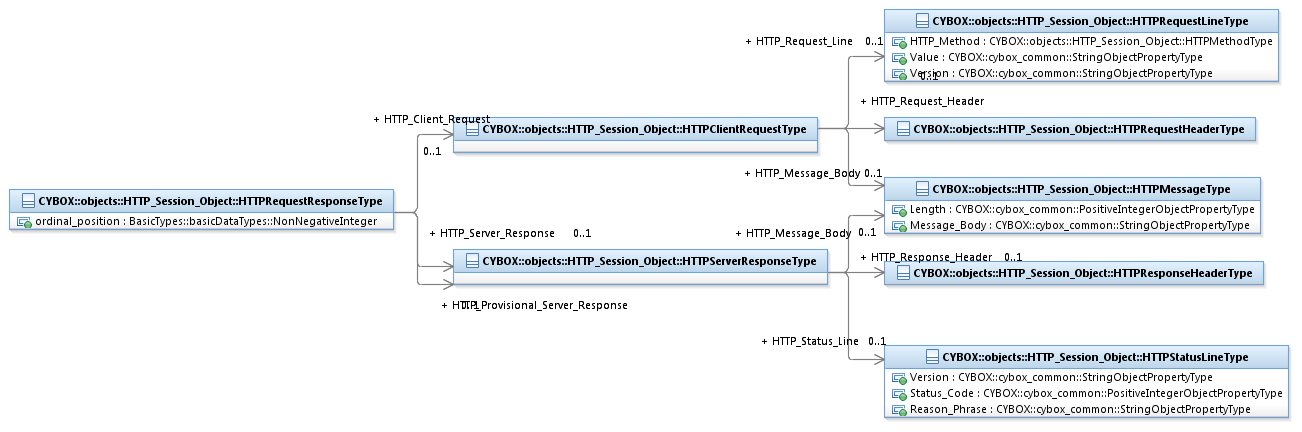


Figure 3‑2. UML diagram for the HTTPRequestResponseType class

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. |

## 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. |

## 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. |

## 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:  StringObjectPropertyType | 0..1 | The Value property captures the value (typically a resource path) portion of the HTTP request line. |
| **Version** | cyboxCommon:  StringObjectPropertyType | 0..1 | The Version property captures the HTTP version portion of the HTTP request line. |

## 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. |

## 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:  StringObjectPropertyType | 0..1 | The Cookie property specifies the HTTP Request Cookie header 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. |

## 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. |

## 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:  StringObjectPropertyType | 0..1 | The ETag property specifies the HTTP Response ETag header field, which defines an identifier for a specific 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]**](#rfc2822) 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].**](#rfc5988) |
| **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. |

## 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. |

## 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. |

## 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. |

## 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.

## 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** |  |

# 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.

Acknowledgments

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

Participants:

Dean Thompson, Australia and New Zealand Banking Group (ANZ Bank)

Bret Jordan, Blue Coat Systems, Inc.

Adnan Baykal, Center for Internet Security (CIS)

Liron Schiff, Comilion (mobile) Ltd.

Jane Ginn, Cyber Threat Intelligence Network, Inc. (CTIN)

Richard Struse, DHS Office of Cybersecurity and Communications (CS&C)

Ryusuke Masuoka, Fujitsu Limited

Eric Burger, Georgetown University

Jason Keirstead, IBM

Paul Martini, iboss, Inc.

Jerome Athias, Individual

Sanjiv Kalkar, Individual

Terry MacDonald, Individual

Alex Pinto, Individual

Patrick Maroney, Integrated Networking Technologies, Inc.

Wouter Bolsterlee, Intelworks BV

Joep Gommers, Intelworks BV

Sergey Polzunov, Intelworks BV

Rutger Prins, Intelworks BV

Andrei Sîrghi, Intelworks BV

Jonathan Baker, MITRE Corporation

Sean Barnum, MITRE Corporation

Mark Davidson, MITRE Corporation

Ivan Kirillov, MITRE Corporation

John Wunder, MITRE Corporation

Mike Boyle, National Security Agency

Jessica Fitzgerald-McKay, National Security Agency

Takahiro Kakumaru, NEC Corporation

John-Mark Gurney, New Context Services, Inc.

Christian Hunt, New Context Services, Inc.

Andrew Storms, New Context Services, Inc.

Igor Baikalov, Securonix

Bernd Grobauer, Siemens AG

John Anderson, Soltra

Trey Darley, Soltra

Paul Dion, Soltra

Brandon Hanes, Soltra

Ali Khan, Soltra

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

Revision History

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