CybOX™ Version 2.1.1. Part 06: UML Model

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/part06-uml-model/cybox-v2.1.1-csprd01-part06-uml-model.docx (Authoritative)
http://docs.oasis-open.org/cti/cybox/v2.1.1/csprd01/part06-uml-model/cybox-v2.1.1-csprd01-part06-uml-model.html
http://docs.oasis-open.org/cti/cybox/v2.1.1/csprd01/part06-uml-model/cybox-v2.1.1-csprd01-part06-uml-model.pdf

Previous version:

N/A

Latest version:

http://docs.oasis-open.org/cti/cybox/v2.1.1/part06-uml-model/cybox-v2.1.1-part06-uml-model.docx (Authoritative)
http://docs.oasis-open.org/cti/cybox/v2.1.1/part06-uml-model/cybox-v2.1.1-part06-uml-model.html
http://docs.oasis-open.org/cti/cybox/v2.1.1/part06-uml-model/cybox-v2.1.1-part06-uml-model.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 document describes the use of UML to create a data model for CybOX.

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-uml-model]

CybOX™ Version 2.1.1. Part 06: UML Model. 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/part06-uml-model/cybox-v2.1.1-csprd01-part06-uml-model.html. Latest version: http://docs.oasis-open.org/cti/cybox/v2.1.1/part06-uml-model/cybox-v2.1.1-part06-uml-model.html.

Notices

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 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.

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

Table of Contents

1 Introduction

[All text is normative unless otherwise labeled.]

The Cyber Observable Expression (CybOX™) Language 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 specification document provides brief summary information on the form and use of the CybOX Language UML model. In addition to this textual specification document, *CybOX Version 2.2.1 Part 6: UML Model* consists of an actual digital serialization of the UML model and a set of relevant UML diagrams extracted from the UML model and used throughout the CybOX Language specification.

In Section **1.1** we discuss the additional specification documents, in Section **1.2** we provide document conventions, and in Section **0** we provide terminology. References are given in Sections **1.4** and **1.5**. In Section **2**, we give summary information on the form of the digitally serialized UML model artifact, and in Section **3** we provide general information and conventions for how the UML model is used to define the individual data models. Conformance information is provided in Section **4**.

1.1 CybOX™ 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 the eighty-eight Object data models, includes a set of default controlled vocabularies. *CybOX Version 2.1.1 Part 1: Overview* also summarizes the relationship of CybOX to other externally defined data models, 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*

1.3 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.

1.5 Non-Normative References

[GitHub-IO] CybOX – Cyber Observable eXpression | CybOX Project Documentation. (n.d.). The MITRE Corporation. [Online]. Available: http://cyboxproject.github.io/. Accessed Dec 15, 2015.

[UML-2.4.1] Documents associated with Unified Modeling Language (UML), V2.4.1. (Aug. 2011). The Object Management Group (OMG). [Online]. Available: http://www.omg.org/spec/UML/2.4.1/.

[XMI] Documents associated with XMI Version 2.1. (September 2005). The Object Management Group (OMG). [Online]. Available: http://www.omg.org/spec/XMI/2.1/.

[PNG] Portable Network Graphics (PNG) Specification (November 2003). The World Wide Web Consortium (W3C). [Online]. Available: http://www.w3.org/TR/PNG/.

2 UML Model Artifact

The CybOX UML model is formally represented in the form of a digital serialization using the XML Metadata Interchange (XMI) language. The XMI language is intended to be an open standardized form supporting the expression of UML models in a non-proprietary manner. In reality, many UML modeling tools tend to include some proprietary elements in their XMI output. The CybOX UML model was produced using Rational Software Architect (RSA) version 9.1, a product of the IBM Corporation. Effort has been made to minimize the level of proprietary content (from the RSA tool) in the XMI serialization, but it should be noted that some portion may still remain.

For the broadest possible interoperability between UML tools the model is provided as an XMI serialization using UML2.2/XMI2.1 [XMI] containing only the model and not the diagrams. A set of relevant UML diagrams, extracted from the UML model and leveraged throughout the CybOX Language specification documents, is also provided in a rastered (portable network graphics [PNG]) form.

In addition, for those with tools that can import the more complete RSA tool native .EMX format, the model with embedded diagrams is also provided in this form.

3 Data Model Conventions

The following general information and conventions are used to define the individual data models in UML.

3.1 UML Packages

Each CybOX data model is captured in a different UML package (e.g., Core package, FileObj package, etc.). 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. **Table 3-1** lists some of the key packages used throughout the CybOX data model specification documents, along with the prefix notation and an example. Each of the eighty-eight CybOX Objects are defined within their own UML package, to support modularity. They are too numerous to mentioned here, but are described in each of the separate specifications documents, parts 7 through 94.

Table 3-1. Package prefixes used by the CybOX Language

Package	CybOX Core
Prefix	**cybox**
Description	The CybOX Core data model defines the main classes of the CybOX data model, such as ActionType, EventType, ObservableType, and ObjectType.
Example	`cybox:ObservableType`
Package	**CybOX Common**
Prefix	**cyboxCommon**
Description	The CybOX Common data model defines classes that are shared across the various CybOX data models.
Example	`cyboxCommon:ConfidenceType`
Package	**CybOX Default Vocabularies**
Prefix	**cyboxVocabs**
Description	The CybOX default vocabularies define the classes for default controlled vocabularies used within CybOX.
Example	`cyboxVocabs:ActionTypeVocab`
Package	**CybOX Basic Data Types**
Prefix	**basicDataTypes**
Description	The CybOX Basic Data Types data model defines the types used within CybOX.
Example	`basicDataTypes:URI`

3.2 Naming Conventions

The UML classes, enumerations, and properties defined in CybOX follow the particular naming conventions outlined in **Table 3-2**.

Table 3-2. Naming formats of different object types

Object Type	Format	Example
Class	CamelCase ending with “Type”	ActionType
Property (simple)	Lowercase with underscores between words	scale
Property (complex)	Capitalized with underscores between words	Discovery_Method
Enumeration	CamelCase ending with “Enum” or “Type	DateTimePrecisionEnum; EffectTypeEnum
Enumeration value	*varies*	Flash drive; Public Disclosure; Externally-Located
Data type	CamelCase, or if the words are acronyms, all capitalized with underscores between words	PositiveInteger; URI

3.3 UML Stereotypes

Certain UML classes are associated with the UML stereotype `<<choice>>`. The `<<choice>>` stereotype specifies that only one of the available properties of the class can be populated at any time. The CybOX UML models utilize `Has_Choice` as the role/property name for associations to `<<choice>>` stereotyped classes. This property is a modeling convention rather than a native element of the underlying data model and acts as a placeholder for one of the available properties of the `<<choice>>` stereotyped class.

NOTE: Importing the UML models into a tool other than Rational Software Architect (RSA) version 9.1 (using the files with the uml file extensions) might not apply the stereotype correctly. If not, the classes that contain the word “`Choice`” are the ones that the stereotype should have been applied to.

3.4 UML Diagrams

This document indicates how UML diagrams are used to visually depict relationships between CybOX Language constructs in the rest of the specification. Note that the example diagrams have been extracted directly from the full UML model for CybOX; they have not been constructed purely for inclusion in this or the other specification documents. Typically, diagrams are included where the visualization of their relationships between classes is useful for illustration purposes. 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. All component data models include a top-level diagram (see **Figure 3-1**).

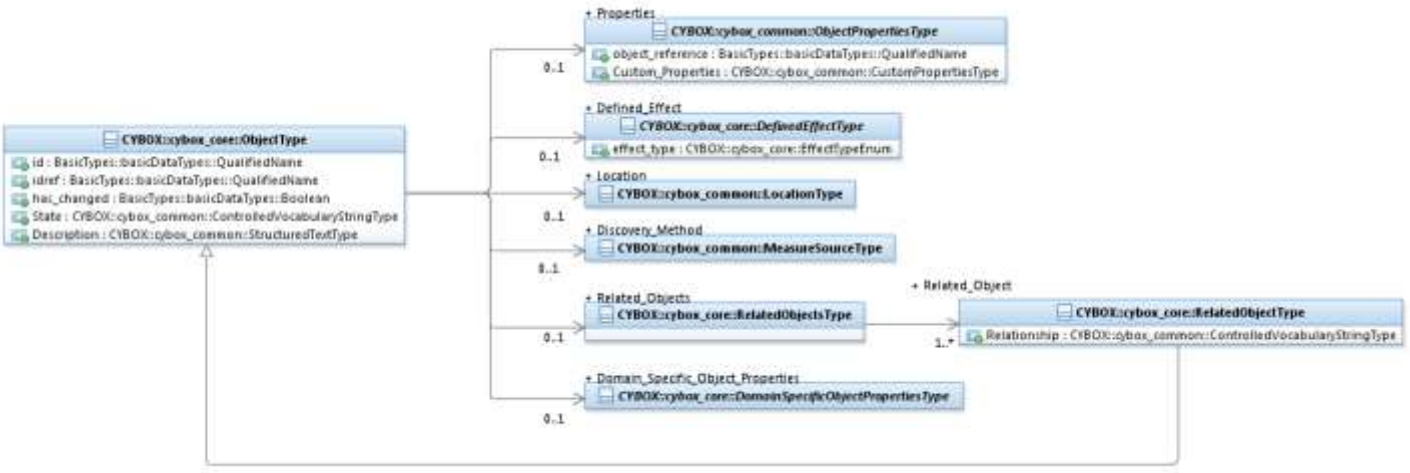

Figure 3-1. Top-level package diagram (ObjectType data model)

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 (see **Figure 3-2**).

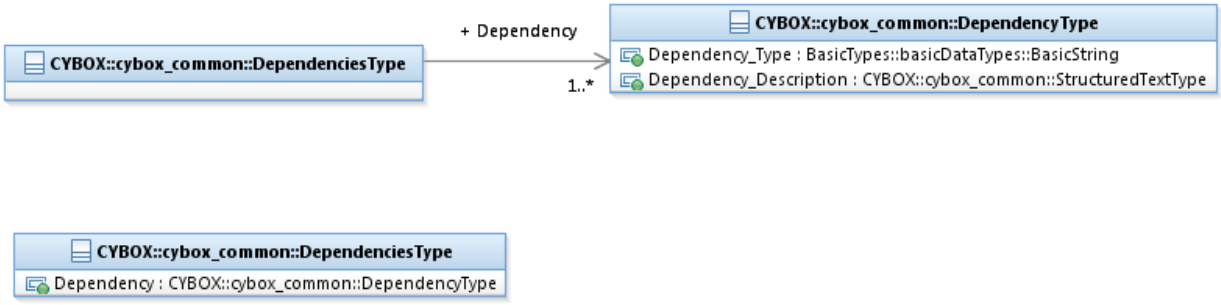

Figure 3-2. Different presentations of class attributes

3.4.1 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. For example, properties of a class that are identifiers, titles, and timestamps will be represented as attributes.

3.4.2 Diagram Icons and Arrow Types

Diagram icons are used in a UML diagram to indicate whether a shape is a class, enumeration, or 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 3-3**.

Table 3-3. 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.

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 and to 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 and any 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 individuals listed below 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

Fujitsu Limited

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

Andrei Sîrghi

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
Emmanuelle Vargas-Gonzalez
John Wunder
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

National Security Agency

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
Michael Pepin
Natalie Suarez
David Waters
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
United Kingdom Cabinet Office

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

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