STIX™ Version 2.0. Part 3: Cyber Observable Core Concepts

Committee Specification 01

19 July 2017

Specification URIs

This version:

http://docs.oasis-open.org/cti/stix/v2.0/cs01/part3-cyber-observable-core/stix-v2.0-cs01-part3-cyber-observable-core.docx (Authoritative)
http://docs.oasis-open.org/cti/stix/v2.0/cs01/part3-cyber-observable-core/stix-v2.0-cs01-part3-cyber-observable-core.html
http://docs.oasis-open.org/cti/stix/v2.0/cs01/part3-cyber-observable-core/stix-v2.0-cs01-part3-cyber-observable-core.pdf

Previous version:

http://docs.oasis-open.org/cti/stix/v2.0/csprd02/part3-cyber-observable-core/stix-v2.0-csprd02-part3-cyber-observable-core.docx (Authoritative)
http://docs.oasis-open.org/cti/stix/v2.0/csprd02/part3-cyber-observable-core/stix-v2.0-csprd02-part3-cyber-observable-core.html
http://docs.oasis-open.org/cti/stix/v2.0/csprd02/part3-cyber-observable-core/stix-v2.0-csprd02-part3-cyber-observable-core.pdf

Latest version:

http://docs.oasis-open.org/cti/stix/v2.0/stix-v2.0-part3-cyber-observable-core.docx (Authoritative)
http://docs.oasis-open.org/cti/stix/v2.0/stix-v2.0-part3-cyber-observable-core.html
http://docs.oasis-open.org/cti/stix/v2.0/stix-v2.0-part3-cyber-observable-core.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:

Trey Darley (trey@kingfisherops.com), Kingfisher Operations, sprl
Ivan Kirillov (ikirillov@mitre.org), MITRE Corporation

Additional artifacts:

This prose specification is one component of a Work Product that also includes:

- *STIX™ Version 2.0. Part 1: STIX Core Concepts*. http://docs.oasis-open.org/cti/stix/v2.0/cs01/part1-stix-core/stix-v2.0-cs01-part1-stix-core.html.
- *STIX™ Version 2.0. Part 2: STIX Objects.* http://docs.oasis-open.org/cti/stix/v2.0/cs01/part2-stix-objects/stix-v2.0-cs01-part2-stix-objects.html.
- (this document) *STIX™ Version 2.0. Part 3: Cyber Observable Core Concepts.* http://docs.oasis-open.org/cti/stix/v2.0/cs01/part3-cyber-observable-core/stix-v2.0-cs01-part3-cyber-observable-core.html.
- *STIX™ Version 2.0. Part 4: Cyber Observable Objects.* http://docs.oasis-open.org/cti/stix/v2.0/cs01/part4-cyber-observable-objects/stix-v2.0-cs01-part4-cyber-observable-objects.html.

 Copyright © OASIS Open 2017. All Rights Reserved.

- *STIX™ Version 2.0. Part 5: STIX Patterning.* http://docs.oasis-open.org/cti/stix/v2.0/cs01/part5-stix-patterning/stix-v2.0-cs01-part5-stix-patterning.html.

Related work:

This specification replaces or supersedes:

- *STIX™ Version 1.2.1. Part 1: Overview.* Edited by Sean Barnum, Desiree Beck, Aharon Chernin, and Rich Piazza. Latest version: http://docs.oasis-open.org/cti/stix/v1.2.1/stix-v1.2.1-part1-overview.html.
- *CybOX™ Version 2.1.1. Part 01: Overview.* Edited by Trey Darley, Ivan Kirillov, Rich Piazza, and Desiree Beck. Latest version: http://docs.oasis-open.org/cti/cybox/v2.1.1/cybox-v2.1.1-part01-overview.html.

This specification is related to:

- *TAXII™ Version 2.0.* Edited by John Wunder, Mark Davidson, and Bret Jordan. Latest version: http://docs.oasis-open.org/cti/taxii/v2.0/taxii-v2.0.html.

Abstract:

Structured Threat Information Expression (STIX™) is a language for expressing cyber threat and observable information. STIX Cyber Observables are defined in two documents. This document defines concepts that apply across all of STIX Cyber Observables.

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/.

This Committee Specification is provided under the Non-Assertion Mode of the OASIS IPR Policy, the mode chosen when the Technical Committee was established. 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).

Note that any machine-readable content (Computer Language Definitions) declared Normative for this Work Product is provided in separate plain text files. In the event of a discrepancy between any such plain text file and display content in the Work Product's prose narrative document(s), the content in the separate plain text file prevails.

Citation format:

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

[STIX-v2.0-Pt3-Cyb-Core]

STIX™ Version 2.0. Part 3: Cyber Observable Core Concepts. Edited by Trey Darley and Ivan Kirillov. 19 July 2017. OASIS Committee Specification 01. http://docs.oasis-open.org/cti/stix/v2.0/cs01/part3-cyber-observable-core/stix-v2.0-cs01-part3-cyber-observable-core.html. Latest version: http://docs.oasis-open.org/cti/stix/v2.0/stix-v2.0-part3-cyber-observable-core.html.

 Copyright © OASIS Open 2017. All Rights Reserved.

Notices

Copyright © OASIS Open 2017. 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 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.

Portions copyright © United States Government 2012-2017. All Rights Reserved.

STIX™, CYBOX™, AND TAXII™ (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

 Copyright © OASIS Open 2017. All Rights Reserved.

 Copyright © OASIS Open 2017. All Rights Reserved.

1 Introduction

The STIX 2.0 specification defines structured representations for observable objects and their properties in the cyber domain. These can be used to describe data in many different functional domains, including but not limited to:

- Malware characterization
- Intrusion detection
- Incident response & management
- Digital forensics

STIX Cyber Observables document the facts concerning **what** happened on a network or host, but not necessarily the who or when, and never the why. For example, information about a file that existed, a process that was observed running, or that network traffic occurred between two IPs can all be captured as Cyber Observable data.

STIX Cyber Observables are used by various STIX Domain Objects (SDOs) to provide additional context to the data that they characterize. The Observed Data SDO, for example, indicates that the raw data was observed at a particular time and by a particular party.

The Cyber Observable Objects chosen for inclusion in STIX 2.0 represent a minimally viable product (MVP) that fulfills basic consumer and producer requirements. Objects and properties not included in STIX 2.0, but deemed necessary by the community, will be included in future releases.

This document (*STIX™ Version 2.0. Part 3: Cyber Observable Core Concepts*) in the STIX specification describes Cyber Observable Core Concepts. *STIX™ Version 2.0. Part 4: Cyber Observable Objects* contains the definitions for the Cyber Observable Objects.

1.0 IPR Policy

This Committee Specification is provided under the Non-Assertion Mode of the OASIS IPR Policy, the mode chosen when the Technical Committee was established. 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).

1.1 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].

All text is normative except for examples, the overview (section 1.4), and any text marked non-normative.

1.2 Normative References

[Character Sets] "N. Freed and M. Dürst, "Character Sets", IANA, December 2013, [Online]. Available**:** http://www.iana.org/assignments/character-sets/character-sets.xhtml

 Copyright © OASIS Open 2017. All Rights Reserved.

[IEEE 754-2008] "IEEE Standard for Floating-Point Arithmetic", IEEE 754-2008, August 2008. [Online] Available: http://ieeexplore.ieee.org/document/4610935/

[ISO10118] **"**ISO/IEC 10118-3:2004 Information technology -- Security techniques -- Hash-functions -- Part 3: Dedicated hash-functions", 2004. [Online]. Available: http://www.iso.org/iso/catalogue_detail.htm?csnumber=39876

[FIPS81] "DES MODES OF OPERATION", FIPS PUB 81, December 1980, National Institute of Standards and Technology (NIST). [Online]. Available: http://csrc.nist.gov/publications/fips/fips81/fips81.htm

[FIPS186-4] "Digital Signature Standard (DSS)", FIPS PUB 186-4, July 2013, Information Technology Laboratory, National Institute of Standards and Technology (NIST). [Online]. Available: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf.

[FIPS202] "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions", FIPS PUB 202, August 2015, Information Technology Laboratory, National Institute of Standards and Technology (NIST). [Online]. Available: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf

[MD6] Rivest, R. et. al, "The MD6 hash function - A proposal to NIST for SHA-3", October 2008. [Online]. Available: http://groups.csail.mit.edu/cis/md6/submitted-2008-10-27/Supporting_Documentation/md6_report.pdf

[NIST 800-38A] M. Dworkin, "Recommendation for Block Cipher Modes of Operation Methods and Techniques", NIST Special Publication 800-38A, 2001. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf

[NIST 800-38D] M. Dworkin, "Recommendation for Block Cipher Modes of Operation:Galois/Counter Mode (GCM) and GMAC", NIST Special Publication 800-38D, November 2007. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf

[NIST 800-38E] M. Dworkin, "Recommendation for Block Cipher Modes of Operation: The XTS-AES Mode for Confidentiality on Storage Devices", NIST Special Publication 800-38E, January 2010. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38e.pdf

[NIST 800-67] W. Barker and E. Barker, "Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher", NIST Special Publication 800-67, January 2012. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-67r1.pdf

[RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, DOI 10.17487/RFC1321, April 1992, http://www.rfc-editor.org/info/rfc1321.

 Copyright © OASIS Open 2017. All Rights Reserved.

[RFC2119] Bradner, S., ""Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, http://www.rfc-editor.org/info/rfc2119

[RFC2144] Adams, C., "The CAST-128 Encryption Algorithm", RFC 2144, DOI 10.17487/RFC2144, May 1997, http://www.rfc-editor.org/info/rfc2144.

[RFC2612] Adams, C. and J. Gilchrist, "The CAST-256 Encryption Algorithm", RFC 2612, DOI 10.17487/RFC2612, June 1999, http://www.rfc-editor.org/info/rfc2612.

[RFC3174] Eastlake 3rd, D. and P. Jones, "US Secure Hash Algorithm 1 (SHA1)", RFC 3174, DOI 10.17487/RFC3174, September 2001, http://www.rfc-editor.org/info/rfc3174.

[RFC6234] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, DOI 10.17487/RFC6234, May 2011, http://www.rfc-editor.org/info/rfc6234.

[RFC7539] Nir, Y. and A. Langley, "ChaCha20 and Poly1305 for IETF Protocols", RFC 7539, DOI 10.17487/RFC7539, May 2015, http://www.rfc-editor.org/info/rfc7539.

[RFC8017] Moriarty, K., Ed., Kaliski, B., Jonsson, J., and A. Rusch, "PKCS #1: RSA Cryptography Specifications Version 2.2", RFC 8017, DOI 10.17487/RFC8017, November 2016, http://www.rfc-editor.org/info/rfc8017.

[RIPEND-160] H. Dobbertin, A. Bosselaers, and B. Preneel, "RIPEMD-160:A Strengthened Version of RIPEMD", April 1996, [Online]. Available: http://homes.esat.kuleuven.be/~bosselae/ripemd160/pdf/AB-9601/AB-9601.pdf

[Salsa20] D. Bernstein, "Salsa20 specification" (n.d.). [Online]. Available: https://cr.yp.to/snuffle/spec.pdf

[Salsa20/8 20/12] D. Bernstein, "Salsa20/8 and Salsa20/12" (n.d.). [Online]. Available: https://cr.yp.to/snuffle/812.pdf

[SSDEEP] J. Kornblum, "Identifying Almost Identical Files Using Context Triggered Piecewise Hashing", Proceedings of The Digital Forensic Research Conference (DFRWS) 2006. [Online]. Available: http://dfrws.org/sites/default/files/session-files/paper-identifying_almost_identical_files_using_context_triggered_piecewise_hashing.pdf

1.3 Non-Normative References

[RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014. http://www.rfc-editor.org/info/rfc7159.txt.

 Copyright © OASIS Open 2017. All Rights Reserved.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, http://www.rfc-editor.org/info/rfc4648.

1.4 Overview

1.4.1 Cyber Observable Objects

STIX 2.0 defines a set of Cyber Observable Objects for characterizing host-based, network, and related entities. Each of these objects correspond to a data point commonly represented in CTI and digital forensics. Using the building blocks of Cyber Observable Objects, in conjunction with relationships between these objects, individuals can create, document, and share comprehensive information about computer systems and their state.

Throughout this document, Cyber Observable Objects are referred to simply as "Observable Objects". These should not be confused with STIX Domain Objects (SDOs), as defined in *STIX™ Version 2.0. Part 1: STIX Core Concepts* and *STIX™ Version 2.0. Part 2: STIX Objects*.

1.4.2 Cyber Observable Relationships

A Cyber Observable Relationship is a reference linking two (or more) related Cyber Observable Objects. Cyber Observable Relationships are only resolvable within the same `observable-objects` container. References are a property on Cyber Observable Objects that contain the ID of a different Cyber Observable Object.

Throughout this document, Cyber Observable Relationships are referred to simply as "Relationships". These should not be confused with STIX Relationship Objects (SROs), as defined in *STIX™ Version 2.0. Part 1: STIX Core Concepts* and *STIX™ Version 2.0. Part 2: STIX Objects*.

1.4.3 Cyber Observable Extensions

Each Observable Object defines a set of base properties that are generally applicable across any instance of the Object. However, there is also a need to encode additional data beyond the base definition of the Object data models. To enable this, STIX permits the specification of such additional properties through the set of Predefined Cyber Observable Object Extensions. Where applicable, Predefined Object Extensions are included in the definitions of Objects. For example, the File Object includes Predefined Object Extensions for characterizing PDF files, raster image files, archive files, NTFS files, and Windows PE binary files.

Producers may also define and include their own Custom Object Extensions. For further information, refer to section 5 (Customizing Cyber Observable Objects.)

1.4.4 Vocabularies & Enumerations

Many Cyber Observable Objects contain properties whose values are constrained by a predefined enumeration or open vocabulary. In the case of enumerations, this is a requirement that producers must use the values in the enumeration and cannot use any outside values. In the case of open vocabularies, this is a suggestion for producers that permits the use of values outside of the suggested vocabulary. If used consistently, vocabularies make it less likely that, for example, one entity refers to the md5 hashing algorithm as "MD5" and another as "md-5-hash", thereby making comparison and correlation easier.

 Copyright © OASIS Open 2017. All Rights Reserved.

1.5 Naming Requirements

1.5.1 Property Names and String Literals

In the JSON serialization all property names and string literals **MUST** be exactly the same, including case, as the names listed in the property tables in this specification. For example, the SDO common property **created_by_ref** must result in the JSON key name "created_by_ref". Properties marked required in the property tables **MUST** be present in the JSON serialization.

1.5.2 Reserved Names

Reserved property names are marked with a type called `RESERVED` and a description text of “RESERVED FOR FUTURE USE”. Any property name that is marked as `RESERVED` **MUST NOT** be present in STIX content conforming to this version of the specification.

1.6 Document Conventions

1.6.1 Naming Conventions

All type names, property names, and literals are in lowercase, except when referencing canonical names defined in another standard (e.g., literal values from an IANA registry). Words in property names are separated with an underscore(_), while words in type names and string enumerations are separated with a hyphen (-). All type names, property names, object names, and vocabulary terms are between three and 250 characters long.

1.6.2 Font Colors and Style

The following color, font and font style conventions are used in this document:

- The `Consolas` font is used for all type names, property names and literals.
  - type names are in red with a light red background - `hashes`
  - property names are in bold style - **`protocols`**
  - literals (values) are in blue with a blue background - `SHA-256`
- In an object's property table, if a common property is being redefined in some way, then the background is dark gray.
- All examples in this document are expressed in JSON. They are in `Consolas` 9-point font, with straight quotes, black text and a `light grey background`, and 2-space indentation.
- Parts of the example may be omitted for conciseness and clarity. These omitted parts are denoted with the ellipses (...).
- The term “hyphen” is used throughout this document to refer to the ASCII hyphen or minus character, which in Unicode is “hyphen-minus”, U+002D.

Copyright © OASIS Open 2017. All Rights Reserved.

2 Cyber Observable Specific Data Types

The Cyber Observable specification within STIX makes use of many common types that are defined in section 2 of *STIX™ Version 2.0. Part 2: STIX Objects*. In addition, data types specific to` the representation of Cyber Observables are defined in this section. The table below lists common data types from STIX Core with a gray background and the Cyber Observable specific types with a white background.

Type	Description
`boolean`	A value of `true` or `false`.
`float`	An IEEE 754 [IEEE 754-2008] double-precision number.
`hashes`	One or more cryptographic hashes.
`integer`	A whole number.
`list`	An ordered sequence of values. The phrasing "`list` of type `<type>`" is used to indicate that all values within the list **MUST** conform to the specified type.
`open-vocab`	A value from a STIX open (`open-vocab`) or suggested vocabulary.
`string`	A series of Unicode characters.
`timestamp`	A time value (date and time).
`binary`	A sequence of bytes.
`hex`	An array of octets as hexadecimal.
`dictionary`	A set of key/value pairs.
`object-ref`	A local reference to a Cyber Observable Object.
`observable-objects`	One or more Cyber Observable Objects.

 Copyright © OASIS Open 2017. All Rights Reserved.

2.1 Binary

Type Name: `binary`

The `binary` data type represents a sequence of bytes. In order to allow pattern matching on custom objects, for all properties that use the binary type, the property name **MUST** end with '_bin'.

The JSON MTI serialization represents this as a base64--encoded string as specified in [RFC4648]. Other serializations **SHOULD** use a native binary type, if available.

2.2 Hexadecimal

Type Name: `hex`

The `hex` data type encodes an array of octets (8-bit bytes) as hexadecimal. The string **MUST** consist of an even number of hexadecimal characters, which are the digits '0' through '9' and the letters 'a' through 'f'. In order to allow pattern matching on custom objects, for all properties that use the `hex` type, the property name **MUST** end with '_hex'.

Examples

```
...
   "src_flags_hex": "00000002"
...
```

2.3 Dictionary

Type Name: `dictionary`

A `dictionary` captures an arbitrary set of key/value pairs. `dictionary` keys **MUST** be unique in each dictionary, **MUST** be in ASCII, and are limited to the characters a-z (lowercase ASCII), A-Z (uppercase ASCII), numerals 0-9, hyphen (-), and underscore (_). `dictionary` keys **SHOULD** be no longer than 30 ASCII characters in length, **MUST** have a minimum length of 3 ASCII characters, **MUST** be no longer than 256 ASCII characters in length, and **SHOULD** be lowercase.

`dictionary` values **MUST** be valid property base types.

2.4 Object Reference

Type Name: `object-ref`

The Object Reference data type specifies a local reference to an Observable Object, that is, one which **MUST** be valid within the local scope of the Observable Objects (`observable-objects`) container that holds both the source Observable Object and the Observable Object that it references.

Examples

The following example demonstrates how a Network Traffic Object specifies its destination via a reference to an IPv4 Address Object.

```
{
  "0": {
```


Copyright © OASIS Open 2017. All Rights Reserved.

```
    "type": "ipv4-addr",
    "value": "198.51.100.2"
  },
  "1": {
    "type": "network-traffic",
    "dst_ref": "0"
  }
}
```

2.5 Observable Objects

Type Name: `observable-objects`

The Observable Objects type represents 1 or more Observable Objects as a special set of key/value pairs. The keys in the dictionary are references used to refer to the values, which are objects. Each key in the dictionary **SHOULD** be a non-negative monotonically increasing integer, incrementing by 1 from a starting value of 0, and represented as a string within the JSON MTI serialization. However, implementers **MAY** elect to use an alternate key format if necessary.

Examples

```
{
    "0": {
      "type": "email-addr",
      "value": "jdoe@example.com",
      "display_name": "John Doe"
    },
    "1": {
      "type": "email-addr",
      "value": "mary@example.com",
      "display_name": "Mary Smith"
    },
    "2": {
      "type": "email-message",
      "from_ref": "0",
      "to_refs": ["1"],
      "date": "1997-11-21T15:55:06Z",
      "subject": "Saying Hello"
    }
  }
}
```

 Copyright © OASIS Open 2017. All Rights Reserved.

3 Cyber Observable Objects

This section outlines the common properties and behavior across all Cyber Observable Objects.

The JSON MTI serialization uses the JSON object type [RFC7159] when representing Objects.

3.1 Common Properties

Property Name	Type	Description
`type` (required)	`string`	Indicates that this object is an Observable Object. The value of this property **MUST** be a valid Observable Object type name.
`extensions` (optional)	`dictionary`	Specifies any extensions of the object, as a dictionary. Dictionary keys **MUST** identify the extension type by name. The corresponding dictionary values **MUST** contain the contents of the extension instance.

3.2 Object References

Identifiers on Observable Objects are specified as keys in the `observable-objects` type. For more information on how such keys may be defined, see section 2.6.

The `object-ref` type is used to define Observable Object properties that are *references* to other Observable Objects (such as the **`src_ref`** property on the Network Traffic Object). *Resolving* a reference is the process of identifying and obtaining the actual Observable Object referred to by the reference property. References resolve to an object when the value of the property (e.g., **`src_ref`**) is an exact match with the `key` of another Observable Object that resides in the same parent container as the Observable Object that specifies the reference. This specification does not address the implementation of reference resolution.

3.3 Object Property Metadata

3.3.1 String Encoding

Capturing the observed encoding of a particular Observable Object string is useful for attribution, the creation of indicators, and related use cases.

Certain string properties in Observable Objects may contain an additional sibling property with the same base name and a suffix of **`_enc`** that captures the name of the original observed encoding of the property value. All **`_enc`** properties **MUST** specify their encoding using the corresponding name from the the IANA

 Copyright © OASIS Open 2017. All Rights Reserved.

character set registry [Character Sets] . If the preferred MIME name for a character set is defined, this value **MUST** be used; if it is not defined, then the Name value from the registry **MUST** be used instead.

As an example of how this capability may be used in an Object, the **`name`** property in the File Object has the sibling property **`name_enc`**, for capturing the observed encoding of the file name string.

Examples

File with Unicode representation of the filename and a corresponding encoding specification

```
{
  "0": {
    "type": "file",
    "hashes": {
      "SHA-256": "effb46bba03f6c8aea5c653f9cf984f170dcdd3bbbe2ff6843c3e5da0e698766"
    },
    "name": "quêry.dll",
    "name_enc": "windows-1252"
  }
}
```

3.4 Object Relationships

A Cyber Observable Relationship is a connection between two or more Cyber Observable Objects within the scope of a given Observable Objects dictionary. Cyber Observable relationships are references that are represented as properties of a Cyber Observable Object, containing the keys of the target Cyber Observable Object(s).

Cyber Observable Object relationships are implemented in Object properties as either singletons or lists. In the case of singleton relationships, the name of their Object property **MUST** end in **`_ref`**, whereas for lists of relationships the name of their Object property **MUST** end in **`_refs`**.

The target(s) of Cyber Observable relationships may be restricted to a subset of Cyber Observable Object types, as specified in the description of the Observable Object property that defines the relationship. For example, the **`belongs_to_refs`** property on the IPv4 Address Object specifies that the *only* valid target of the relationship is one or more AS Objects.

Examples

Network Traffic with Source/Destination IPv4 Addresses and AS

```
{
  "0": {
    "type": "ipv4-addr",
    "value": "1.2.3.4",
    "belongs_to_refs": ["3"]
  },
  "1": {
    "type": "ipv4-addr",
    "value": "2.3.4.5"
  },
```

 Copyright © OASIS Open 2017. All Rights Reserved.

```
  "2": {
    "type": "network-traffic",
    "src_ref": "0",
    "dst_ref": "1",
  }
  "3": {
    "type": "as"
    "number": 42
  }
}
```

3.5 Predefined Object Extensions

Predefined Object Extensions have a specific purpose in Cyber Observable Objects: defining coherent sets of properties beyond the base, e.g., HTTP request information for a Network Traffic object. Accordingly, each Cyber Observable Object may include one or more Predefined Object Extensions.

Each Predefined Object Extension can be defined at most once on a given Observable Object. In an Observable Object instance, each extension is specified under the **`extensions`** property, which is of type `dictionary`. Note that this means that each extension is specified through a corresponding key in the `extensions` property. For example, when specified in a File Object instance, the NTFS extension would be specified using the key value of `ntfs-ext`.

Examples

Basic File with NTFS Extension

```
{
  "0": {
    "type": "file",
    "hashes": {
      "MD5": "3773a88f65a5e780c8dff9cdc3a056f3"
    },
    "size": 25537,
    "extensions": {
      "ntfs-ext": {
        "sid": "1234567"
      }
    }
  }
}
```

 Copyright © OASIS Open 2017. All Rights Reserved.

4 Common Vocabularies

4.1 Encryption Algorithm Vocabulary

Type Name: `encryption-algo-ov`

An open vocabulary of encryption algorithms.

When specifying an encryption algorithm not already defined within the `encryption-algo-ov`, wherever an authoritative name for an encryption algorithm name is defined, it should be used as the value. In cases where no authoritative name exists and/or there is variance in the naming of a particular encryption algorithm, producers should exercise their best judgement.

Vocabulary Value	Description
`AES128-ECB`	Specifies the Advanced Encryption Standard (AES) with Electronic Codebook (ECB) mode, as a defined in [NIST 800-38A].
`AES128-CBC`	Specifies the Advanced Encryption Standard (AES) with Cipher Block Chaining (CBC) mode, as a defined in [NIST 800-38A].
`AES128-CFB`	Specifies the Advanced Encryption Standard (AES) with Cipher Feedback (CFB) mode, as a defined in [NIST 800-38A].
`AES128-OFB`	Specifies the Advanced Encryption Standard (AES) with Output Feedback (OFB) mode, as a defined in [NIST 800-38A].
`AES128-CTR`	Specifies the Advanced Encryption Standard (AES) with counter (CTR) mode, as a defined in [NIST 800-38A].
`AES128-XTS`	Specifies the Advanced Encryption Standard (AES) with XEX Tweakable Block Cipher with Ciphertext Stealing (XTS) mode, as a defined in [NIST 800-38E].
`AES128-GCM`	Specifies the Advanced Encryption Standard (AES) with Galois/Counter (GCM) mode, as a defined in NIST SP 8l00-38D.
`Salsa20`	Specifies the Salsa20 stream cipher, as defined in the [Salsa20] specification.
`Salsa12`	Specifies the Salsa20/12 stream cipher as defined in the [Salsa20/8 20/12] specification.
`Salsa8`	Specifies the Salsa20/8 stream cipher as defined in the [Salsa20/8 20/12] specification.

 Copyright © OASIS Open 2017. All Rights Reserved.

`ChaCha20-Poly1305`	Specifies the ChaCha20-Poly1305 stream cipher, as defined in [RFC 7539].
`ChaCha20`	Specifies the ChaCha20 stream cipher (without poly1305 authentication), as defined in [RFC 7539].
`DES-CBC`	Specifies the Data Encryption Standard algorithm with Cipher Block Chaining (CBC) mode, as defined in [FIPS81].
`3DES-CBC`	Specifies the Triple Data Encryption Standard algorithm with Cipher Block Chaining (CBC) mode, as defined in [NIST 800-67] and [NIST 800-38A].
`DES-ECB`	Specifies the Data Encryption Standard algorithm with Electronic Codebook (ECB) mode, as defined in [FIPS81].
`3DES-ECB`	Specifies the Triple Data Encryption Standard algorithm with Electronic Codebook (ECB) mode, as defined in [NIST 800-67].
`CAST128-CBC`	Specifies the CAST-128 algorithm with Cipher Block Chaining (CBC) mode, as defined in [RFC 2144].
`CAST256-CBC`	Specifies the CAST-256 algorithm with Cipher Block Chaining (CBC) mode, as defined in [RFC 2612].
`RSA`	Specifies the RSA symmetric encryption algorithm, as defined by [RFC 8017]
`DSA`	Specifies the Digital Signature Algorithm, as defined by [FIPS186-4].

Copyright © OASIS Open 2017. All Rights Reserved.

5 Customizing Cyber Observables

There are three means to customize Cyber Observable Objects: custom object extensions, custom observable objects, and custom properties. Custom object extensions provide a mechanism and requirements for the specification of extensions not defined by this specification (including relationships) on Observable Objects. Custom Observable Objects provide a mechanism and requirements to create Observable Objects not defined by this specification. Custom properties, as in the rest of STIX, provide a mechanism to add individual properties anywhere in the data model.

Custom Observable Object properties **SHOULD** be used for cases where it is necessary to add one or more simple additional properties (i.e. key/value pairs) on an Observable Object. On the other hand, Custom Observable Object extensions **SHOULD** be used for cases where it is necessary to describe more complex additional properties (i.e., those with potentially multiple levels of hierarchy). As an example, a vendor-specific property that expresses some custom threat score for a File Object should be added directly to the Observable Object as a custom property, whereas a set of properties that represent metadata around a new file system to the File Object should be done as a custom extension.

A consumer that receives a STIX document containing Custom Cyber Observable Properties, Extensions, or Objects it does not understand **MAY** refuse to process the document or **MAY** ignore those properties or objects and continue processing the document.

5.1 Custom Observable Objects

There will be cases where certain information exchanges can be improved by adding objects that are not specified nor reserved in this document; these objects are called Custom Observable Objects. This section provides guidance and requirements for how producers can use Custom Observable Objects and how consumers should interpret them in order to extend STIX in an interoperable manner.

5.1.1 Requirements

- Producers **MAY** include any number of Custom Observable Objects in an Observable Objects entity.
- The type property in a Custom Observable Object **MUST** be in ASCII and **MUST** only contain the characters a-z (lowercase ASCII), 0-9, and hyphen (-).
- The type property **MUST NOT** contain a hyphen (-) character immediately following another hyphen (-) character.
- Custom Observable Object names **MUST** have a minimum length of 3 ASCII characters.
- Custom Observable Object names **MUST** be no longer than 250 ASCII characters in length.
- The value of the **`type`** property in a Custom Observable Object **SHOULD** start with “x-” followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. For example: `x-example-com-customobject`.
- A Custom Observable Object whose name is not prefixed with “x-” **MAY** be used in a future version of the specification with a different meaning. Therefore, if compatibility with future versions of this specification is required, the “x-” prefix **MUST** be used.
- A Custom Observable Object **MUST** have one or more Custom Properties:
    - Custom Property names **MUST** be in ASCII and **MUST** only contain the characters a–z (lowercase ASCII), 0–9, and underscore (_).
    - Custom Property names **MUST** have a minimum length of 3 ASCII characters.
    - Custom Property names **MUST** be no longer than 250 ASCII characters in length.
- Custom Observable Objects **SHOULD** only be used when there is no existing Observable Object defined by the STIX specification that fulfills that need.
- Custom Observable Object property values **MUST** be a valid primitive, type, or a homogenous list of types.

 Copyright © OASIS Open 2017. All Rights Reserved.

Examples

Simple Custom Observable Object

```
{
  "0": {
    "type": "x-example",
    "foo": "bar",
    "vals": ["this",
             "is",
             "an",
             "example"]
  }
}
```

5.2 Custom Object Extensions

In addition to the Predefined Cyber Observable Object extensions specified in [STIX™ Version 2.0. Part 4: Cyber Observable Objects](), STIX supports user-defined custom extensions for Cyber Observable Objects. As with Predefined Object Extensions, custom extension data **MUST** be conveyed under the **`extensions`** property.

5.2.1 Requirements

- An Observable Object **MAY** have any number of Custom Extensions.
- Custom Extension names **MUST** be in ASCII and are limited to characters a-z (lowercase ASCII), 0-9, and hyphen (-).
- Custom Extension names **SHOULD** start with “x-” followed by a source unique identifier (like a domain name), a hyphen and then the name. For example: `x-example-com-customextension`.
- Custom Extension names **MUST** have a minimum length of 3 ASCII characters.
- Custom Extension names **MUST** be no longer than 250 ASCII characters in length.
- Custom Extension names that are not prefixed with “x-” may be used in a future version of the specification for a different meaning. If compatibility with future versions of this specification is required, the “x-” prefix **MUST** be used.
- Custom Extensions **SHOULD** only be used when there is no existing extension defined by the STIX 2.0 specification that fulfills that need.
- A Custom Extension **MUST** have one or more Custom Properties:
    - Custom Property names **MUST** be in ASCII and **MUST** only contain the characters a–z (lowercase ASCII), 0–9, and underscore (_).
    - Custom Property names **MUST** have a minimum length of 3 ASCII characters.
    - Custom Property names **MUST** be no longer than 250 ASCII characters in length.

Examples

Custom File Object Extension

```
{
  "0": {
    "type": "file",
    "hashes": {
      "SHA-256": "effb46bba03f6c8aea5c653f9cf984f170dcdd3bbbe2ff6843c3e5da0e698766"
    },
    "extensions": {
      "x-example-com-foo": {
```

 Copyright © OASIS Open 2017. All Rights Reserved.

```
        "foo_val": "foo",
        "bar_val": "bar"
      }
    }
  }
}
```

5.3 Custom Object Properties

There will be cases where certain information exchanges can be improved by adding properties to Observable Objects that are neither specified nor reserved in this document; these properties are called Custom Object Properties. This section provides guidance and requirements for how producers can use Custom Object Properties and how consumers should interpret them in order to extend Cyber Observable Objects in an interoperable manner.

5.3.1 Requirements

- A Cyber Observable Object **MAY** have any number of Custom Properties.
- Custom Property names **MUST** be in ASCII and MUST only contain the characters a–z (lowercase ASCII), 0–9, and underscore (_).
- Custom Property names **SHOULD** start with “x_” followed by a source unique identifier (such as a domain name with dots replaced by underscores), an underscore and then the name. For example, **`x_example_com_customfield`**.
- Custom Property names **MUST** have a minimum length of 3 ASCII characters.
- Custom Property names **MUST** be no longer than 250 ASCII characters in length.
- Custom Property names that do not start with “x_” may be used in a future version of the specification for a different meaning. If compatibility with future versions of this specification is required, the “x_” prefix **MUST** be used.
- Custom Properties **SHOULD** only be used when there are no existing properties defined by the STIX 2.0 specification that fulfils that need.
- Custom Properties **SHOULD** only be used to define simple properties (e.g., those of string or integer type)
- For Custom Properties that use the `hex` type, the property name **MUST** end with '_hex'.
- For Custom Properties that use the `binary` type, the property name **MUST** end with '_bin'.

Examples

File Object with Custom Properties

```
{
  "0": {
    "type": "file",
    "hashes": {
      "SHA-256": "effb46bba03f6c8aea5c653f9cf984f170dcdd3bbbe2ff6843c3e5da0e698766"
    },
    "x_example_com_foo": "bar",
    "x_example_com_bar": 27
  }
}
```

 Copyright © OASIS Open 2017. All Rights Reserved.

6 Reserved Names

This section defines names that are reserved for future use in revisions of this document. The names defined in this section **MUST NOT** be used for the name of any Custom Cyber Observable Object or Property.

The following object names are reserved:

- `action`

 Copyright © OASIS Open 2017. All Rights Reserved.

7 Conformance

7.1 Producers and Consumers

A "Cyber Observable Producer" is any software that creates Cyber Observable content and conforms to the following normative requirements:

1. It **MUST** be able to create content encoded as JSON.
2. All properties marked required in the property table for the Cyber Observable Object or type **MUST** be present in the created content.
3. All properties **MUST** conform to the specified data type and normative requirements.
4. It **MUST** support at least one defined Cyber Observable Object per the Conformance section in *STIX™ Version 2.0. Part 4: Cyber Observable Objects*.

A "Cyber Observable Consumer" is any software that consumes Cyber Observable content and conforms to the following normative requirements:

1. It **MUST** support parsing all required properties for the content that it consumes.

 Copyright © OASIS Open 2017. All Rights Reserved.

Appendix A. Glossary

CAPEC - Common Attack Pattern Enumeration and Classification

Consumer - Any entity that receives STIX content

CTI - Cyber Threat Intelligence

Embedded Relationship - A link (an "edge" in a graph) between one STIX Object and another represented as a property on one object containing the ID of another object

Entity - Anything that has a separately identifiable existence (e.g., organization, person, group, etc.)

IEP - FIRST (Forum of Incident Response and Security Teams) Information Exchange Policy

Instance - A single occurrence of a STIX object version

MTI - Mandatory To Implement

MVP - Minimally Viable Product

Object Creator - The entity that created or updated a STIX object (see section 3.3 of *STIX™ Version 2.0. Part 1: STIX Core Concepts*).

Object Representation - An instance of an object version that is serialized as STIX

Producer - Any entity that distributes STIX content, including object creators as well as those passing along existing content

SDO - STIX Domain Object (a "node" in a graph)

SRO - STIX Relationship Object (one mechanism to represent an "edge" in a graph)

STIX - Structured Threat Information Expression

STIX Content - STIX documents, including STIX Objects, STIX Objects grouped as bundles, etc.

STIX Object - A STIX Domain Object (SDO) or STIX Relationship Object (SRO)

STIX Relationship - A link (an "edge" in a graph) between two STIX Objects represented by either an SRO or an embedded relationship

TAXII - An application layer protocol for the communication of cyber threat information

TLP - Traffic Light Protocol

TTP - Tactic, technique, or procedure; behaviors and resources that attackers use to carry out their attacks

 Copyright © OASIS Open 2017. All Rights Reserved.

Appendix B. Acknowledgments

Cyber Observable Subcommittee Chairs:

Trey Darley, Kingfisher Operations, sprl

Ivan Kirillov, MITRE Corporation

STIX Subcommittee Chairs:

Sarah Kelley, Center for Internet Security (CIS)

John Wunder, MITRE Corporation

Special Thanks:

Substantial contributions to this specification from the following individuals are gratefully acknowledged:

Sarah Kelley, Center for Internet Security (CIS)

Terry MacDonald, Cosive

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

Richard Struse, DHS Office of Cybersecurity and Communications

Iain Brown, GDS

Jason Keirstead, IBM

Tim Casey, Intel

Trey Darley, Kingfisher Operations, sprl

Allan Thomson, LookingGlass Cyber

Greg Back, MITRE Corporation

Ivan Kirillov, MITRE Corporation

Jon Baker, MITRE Corporation

John Wunder, MITRE Corporation

Sean Barnum, MITRE Corporation

Richard Piazza, MITRE Corporation

Christian Hunt, New Context Services, Inc.

John-Mark Gurney, New Context Services, Inc.

Aharon Chernin, Perch

Dave Cridland, Surevine

Bret Jordan, Symantec Corp.

Participants:

The following individuals were members of the OASIS CTI Technical Committee during the creation of this specification and their contributions are gratefully acknowledged:

David Crawford, Aetna

Marcos Orallo, Airbus Group SAS

Roman Fiedler, AIT Austrian Institute of Technology

 Copyright © OASIS Open 2017. All Rights Reserved.

Florian Skopik, AIT Austrian Institute of Technology
Russell Spitler, AlienVault
Ryan Clough, Anomali
Nicholas Hayden, Anomali
Wei Huang, Anomali
Angela Nichols, Anomali
Hugh Njemanze, Anomali
Katie Pelusi, Anomali
Dean Thompson, Australia and New Zealand Banking Group (ANZ Bank)
Alexander Foley, Bank of America
Sounil Yu, Bank of America
Vicky Laurens, Bank of Montreal
Humphrey Christian, Bay Dynamics
Ryan Stolte, Bay Dynamics
Alexandre Dulaunoy, CIRCL
Andras Iklody, CIRCL
Rapha‘l Vinot, CIRCL
Sarah Kelley, CIS
Syam Appala, Cisco Systems
Ted Bedwell, Cisco Systems
David McGrew, Cisco Systems
Mark-David McLaughlin, Cisco Systems
Pavan Reddy, Cisco Systems
Omar Santos, Cisco Systems
Jyoti Verma, Cisco Systems
Doug DePeppe, Cyber Threat Intelligence Network, Inc. (CTIN)
Jane Ginn, Cyber Threat Intelligence Network, Inc. (CTIN)
Ben Othman, Cyber Threat Intelligence Network, Inc. (CTIN)
Jeff Odom, Dell
Sreejith Padmajadevi, Dell
Ravi Sharda, Dell
Will Urbanski, Dell
Sean Sobieraj, DHS Office of Cybersecurity and Communications (CS&C)
Richard Struse, DHS Office of Cybersecurity and Communications (CS&C)
Marlon Taylor, DHS Office of Cybersecurity and Communications (CS&C)
Jens Aabol, Difi-Agency for Public Management and eGovernment
Wouter Bolsterlee, EclecticIQ
Marko Dragoljevic, EclecticIQ
Oliver Gheorghe, EclecticIQ
Joep Gommers, EclecticIQ
Sergey Polzunov, EclecticIQ
Rutger Prins, EclecticIQ

 Copyright © OASIS Open 2017. All Rights Reserved.

Andrei S”rghi, EclecticIQ
Raymon van der Velde, EclecticIQ
Ben Sooter, Electric Power Research Institute (EPRI)
Chris Ricard, Financial Services Information Sharing and Analysis Center (FS-ISAC)
Phillip Boles, FireEye, Inc.
Prasad Gaikwad, FireEye, Inc.
Rajeev Jha, FireEye, Inc.
Anuj Kumar, FireEye, Inc.
Shyamal Pandya, FireEye, Inc.
Paul Patrick, FireEye, Inc.
Scott Shreve, FireEye, Inc.
Jon Warren, FireEye, Inc.
Remko Weterings, FireEye, Inc.
Gavin Chow, Fortinet Inc.
Steve Fossen, Fortinet Inc.
Kenichi Terashita, Fortinet Inc.
Ryusuke Masuoka, Fujitsu Limited
Daisuke Murabayashi, Fujitsu Limited
Derek Northrope, Fujitsu Limited
Jonathan Algar, GDS
Iain Brown, GDS
Adam Cooper, GDS
Mike McLellan, GDS
Tyrone Nembhard, GDS
Chris O'Brien, GDS
James Penman, GDS
Howard Staple, GDS
Chris Taylor, GDS
Laurie Thomson, GDS
Alastair Treharne, GDS
Julian White, GDS
Bethany Yates, GDS
Robert van Engelen, Genivia
Eric Burger, Georgetown University
Allison Miller, Google Inc.
Mark Risher, Google Inc.
Yoshihide Kawada, Hitachi, Ltd.
Jun Nakanishi, Hitachi, Ltd.
Kazuo Noguchi, Hitachi, Ltd.
Akihito Sawada, Hitachi, Ltd.
Yutaka Takami, Hitachi, Ltd.
Masato Terada, Hitachi, Ltd.

 Copyright © OASIS Open 2017. All Rights Reserved.

Peter Allor, IBM
Eldan Ben-Haim, IBM
Allen Hadden, IBM
Sandra Hernandez, IBM
Jason Keirstead, IBM
John Morris, IBM
Laura Rusu, IBM
Ron Williams, IBM
Paul Martini, iboss, Inc.
Jerome Athias, Individual
Peter Brown, Individual
Joerg Eschweiler, Individual
Stefan Hagen, Individual
Elysa Jones, Individual
Sanjiv Kalkar, Individual
Terry MacDonald, Individual
Alex Pinto, Individual
Tim Casey, Intel Corporation
Kent Landfield, Intel Corporation
Karin Marr, Johns Hopkins University Applied Physics Laboratory
Julie Modlin, Johns Hopkins University Applied Physics Laboratory
Mark Moss, Johns Hopkins University Applied Physics Laboratory
Mark Munoz, Johns Hopkins University Applied Physics Laboratory
Nathan Reller, Johns Hopkins University Applied Physics Laboratory
Pamela Smith, Johns Hopkins University Applied Physics Laboratory
David Laurance, JPMorgan Chase Bank, N.A.
Russell Culpepper, Kaiser Permanente
Beth Pumo, Kaiser Permanente
Michael Slavick, Kaiser Permanente
Trey Darley, Kingfisher Operations, sprl
Gus Creedon, Logistics Management Institute
Wesley Brown, LookingGlass
Jamison Day, LookingGlass
Kinshuk Pahare, LookingGlass
Allan Thomson, LookingGlass
Ian Truslove, LookingGlass
Chris Wood, LookingGlass
Greg Back, Mitre Corporation
Jonathan Baker, Mitre Corporation
Sean Barnum, Mitre Corporation
Desiree Beck, Mitre Corporation
Michael Chisholm, Mitre Corporation

 Copyright © OASIS Open 2017. All Rights Reserved.

Nicole Gong, Mitre Corporation
Ivan Kirillov, Mitre Corporation
Michael Kouremetis, Mitre Corporation
Chris Lenk, Mitre Corporation
Richard Piazza, Mitre Corporation
Larry Rodrigues, Mitre Corporation
Jon Salwen, Mitre Corporation
Charles Schmidt, Mitre Corporation
Alex Tweed, Mitre Corporation
Emmanuelle Vargas-Gonzalez, Mitre Corporation
John Wunder, Mitre Corporation
James Cabral, MTG Management Consultants, LLC.
Scott Algeier, National Council of ISACs (NCI)
Denise Anderson, National Council of ISACs (NCI)
Josh Poster, National Council of ISACs (NCI)
Mike Boyle, National Security Agency
Joe Brule, National Security Agency
Jessica Fitzgerald-McKay, National Security Agency
David Kemp, National Security Agency
Shaun McCullough, National Security Agency
John Anderson, NC4
Michael Butt, NC4
Mark Davidson, NC4
Daniel Dye, NC4
Angelo Mendonca, NC4
Michael Pepin, NC4
Natalie Suarez, NC4
Benjamin Yates, NC4
Daichi Hasumi, NEC Corporation
Takahiro Kakumaru, NEC Corporation
Lauri Korts-P_rn, NEC Corporation
John-Mark Gurney, New Context Services, Inc.
Christian Hunt, New Context Services, Inc.
Daniel Riedel, New Context Services, Inc.
Andrew Storms, New Context Services, Inc.
Stephen Banghart, NIST
David Darnell, North American Energy Standards Board
Cory Casanave, Object Management Group
Aharon Chernin, Perch
Dave Eilken, Perch
Sourabh Satish, Phantom
Josh Larkins, PhishMe Inc.

Copyright © OASIS Open 2017. All Rights Reserved.

John Tolbert, Queralt Inc.
Ted Julian, Resilient Systems, Inc..
Igor Baikalov, Securonix
Joseph Brand, Semper Fortis Solutions
Duncan Sparrell, sFractal Consulting LLC
Thomas Schreck, Siemens AG
Rob Roel, Southern California Edison
Dave Cridland, Surevine Ltd.
Bret Jordan, Symantec Corp.
Curtis Kostrosky, Symantec Corp.
Juha Haaga, Synopsys
Masood Nasir, TELUS
Greg Reaume, TELUS
Alan Steer, TELUS
Crystal Hayes, The Boeing Company
Wade Baker, ThreatConnect, Inc.
Cole Iliff, ThreatConnect, Inc.
Andrew Pendergast, ThreatConnect, Inc.
Ben Schmoker, ThreatConnect, Inc.
Jason Spies, ThreatConnect, Inc.
Ryan Trost, ThreatQuotient, Inc.
Patrick Coughlin, TruSTAR Technology
Chris Roblee, TruSTAR Technology
Mark Angel, U.S. Bank
Brian Fay, U.S. Bank
Joseph Frazier, U.S. Bank
Mark Heidrick, U.S. Bank
Mona Magathan, U.S. Bank
Yevgen Sautin, U.S. Bank
Richard Shok, U.S. Bank
James Bohling, US Department of Defense (DoD)
Eoghan Casey, US Department of Defense (DoD)
Gary Katz, US Department of Defense (DoD)
Jeffrey Mates, US Department of Defense (DoD)
Evette Maynard-Noel, US Department of Homeland Security
Robert Coderre, VeriSign
Kyle Maxwell, VeriSign
Eric Osterweil, VeriSign
Patrick Maroney, Wapack Labs LLC
Anthony Rutkowski, Yanna Technologies LLC

 Copyright © OASIS Open 2017. All Rights Reserved.

Appendix C. Revision History

Revision	Date	Editor	Changes Made
01	2017-01-20	Bret Jordan, John Wunder, Rich Piazza, Ivan Kirillov, Trey Darley	Initial Version
02	2017-04-24	Bret Jordan, John Wunder, Rich Piazza, Ivan Kirillov, Trey Darley	Changes made from first public review

 Copyright © OASIS Open 2017. All Rights Reserved.