TTA Standard

정보통신단체표준(영문표준)

TTAE.OT-xx.xxxx-Part4 제정일: 2018년 xx월 xx일

# 구조화된 위협 정보 표현 규격(STIX™) 버전 2.0 - 제4부: 사이버 관측 객체

# Structured Threat Information eXpression(STIX™) Version 2.0 - Part4: Cyber Observable Objects

TTA 한국정보통신기술협회
Telecommunications Technology Association

**표준초안 검토 위원회** 사이버보안 프로젝트그룹(PG503)

**표준안 심의 위원회** 정보보호 기술위원회(TC5)

| | 성명 | 소 속 | 직위 | 위원회 및 직위 | 표준번호 |
|---|---|---|---|---|---|
| **표준(과제) 제안** | 김종현 | 한국전자통신연구원 | 책임연구원 | 사이버보안 프로젝트 그룹 위원 | 미정 |
| | 박성민 | 한국인터넷진흥원 | 선임연구원 | 위원 | 미정 |
| | 김낙현 | 한국인터넷진흥원 | 선임연구원 | 위원 | 미정 |
| | 이철호 | 국가보안기술 연구소 | 책임연구원 | 위원 | 미정 |
| **표준 초안 작성자** | 김종현 | 한국전자통신연구원 | 책임연구원 | 사이버보안 프로젝트 그룹 위원 | 미정 |
| | 박성민 | 한국인터넷진흥원 | 선임연구원 | 위원 | 미정 |
| | 김낙현 | 한국인터넷진흥원 | 선임연구원 | 위원 | 미정 |
| | 이철호 | 국가보안기술 연구소 | 책임연구원 | 위원 | 미정 |
| | 염흥열 | 순천향대학교 | 교수 | 위원 | 미정 |
| | 김익균 | 한국전자통신연구원 | 책임연구원 | 위원 | 미정 |
| **사무국 담당** | 박수정 | TTA | 책임연구원 | - | |

(※ '표준번호'는 제정 또는 개정 시의 표준번호를 기입한다.)

(※ 개정된 표준일 경우, 공헌자를 제정 및 개정 표준별로 구분하여 병기할 수 있다.)



본 표준 발간 이전에 접수된 지식재산권 확약서 정보는 본 표준의 '부록(지식재산권 확약서 정보)'에 명시하고 있으며, 이후 접수된 지식재산권 확약서는 TTA 웹사이트에서 확인할 수 있습니다.

본 표준과 관련하여 접수된 확약서 외의 지식재산권이 존재할 수 있습니다.

발행인 : 한국정보통신기술협회 회장

발행처 : 한국정보통신기술협회

13591, 경기도 성남시 분당구 분당로 47

Tel : 031-724-0114, Fax : 031-724-0109

발행일 : 2018.xx

# 서문

## 1 표준의 목적

이 표준은 STIX(Structured Threat Information Expression) 2.0에서 사용하는 STIX 사이버 관측 객체를 정의한다. 또한, STIX 2.0에 포함된 사이버 관측 객체에 대한 기본적인 소비자 및 생산자의 요구사항을 제시한다.

## 2 주요 내용 요약

이 표준은 사이버 관측 객체를 서술한다. STIX 사이버 관측은 특성화된 데이터에 대한 추가 컨텍스트를 제공하기 위하여 다양한 STIX 도메인 객체 (SDO) 에서 사용된다. 이를 통해 STIX 2.0은 공동 위협 분석, 위협 공유 자동화, 탐지 및 대응 자동화와 같은 다양한 기능을 제공하고 개선하도록 설계할 수 있다. 주요 내용으로는 아티팩트 객체 모델을 포함한 18가지 객체 데이터 모델의 특성 (properties) 를 정의하고 관련 적합성을 서술한다.

## 3 인용 표준과의 비교

### 3.1 인용 표준과의 관련성

이 표준은 인용표준(STIX™ Version 2.0. Part4: STIX Cyber Observable Objects)을 영문 그대로 완전 수용하는 표준이다.

### 3.2 인용 표준과 본 표준의 비교표

| TTAE.xx-xx.xxxx | STIX™ Version 2.0. Part4: Cyber Observable Objects | 비고 |
|---|---|---|
| 1. 소개 | 1. Introduction | 동일 |
| 2. 객체 데이터 모델 정의 | 2. Defined Object Data Models | 동일 |
| 3. 적합성 | 3. Conformance | 동일 |
| 부속서 A. 용어 사전 | Appendix A. Glossary | 동일 |
| 부속서 B. 감사의 글 | Appendix B. Acknowledgements | 동일 |
| 부속서 C. 개정이력 | Appendix C. Revision History | 동일 |

# Preface

## 1 Purpose

The standard is to adopt as a TTA standard the STIX™ Version 2.0. Part 4: Cyber Observable Objects published by OASIS.

## 2 Summary

This standard describes STIX Cyber Observable objects. The STIX Cyber observable objects are used in various STIX domain objects (SDOs) to provide additional context. This allows STIX 2.0 to be designed to provide and enhance a variety of features such as common threat analysis, automated threat sharing, and automated detection and response. The main contents define the properties of the 18 object data models including the Artifact object model and describe the relevant conformance.

## 3 Relationship to Reference Standards

### 3.1 The relationship of international standards

The standard is fully equivalent to STIX™ Version 2.0. Part4: STIX Cyber Observable Objects.

### 3.2 Differences between International standards(recommendation) and this standard

| TTAE.xx-xx.xxxx | STIX™ Version 2.0. Part4: Cyber Observable Objects | Remarks |
|---|---|---|
| 1. Introduction | 1. Introduction | Equals |
| 2. Defined Object Data Models | 2. Defined Object Data Models | Equals |
| 3. Conformance | 3. Conformance | Equals |
| Appendix A. Glossary | Appendix A. Glossary | Equals |
| Appendix B. Acknowledgements | Appendix B. Acknowledgements | Equals |
| Appendix C. Revision History | Appendix C. Revision History | Equals |

# 목 차

# STIX™ Version 2.0. Part 4: Cyber Observable Objects

## Committee Specification 01

## 19 July 2017

### Specification URIs

**This version:**

http://docs.oasis-open.org/cti/stix/v2.0/cs01/part4-cyber-observable-objects/stix-v2.0-cs01-part4-cyber-observable-objects.docx (Authoritative)

http://docs.oasis-open.org/cti/stix/v2.0/cs01/part4-cyber-observable-objects/stix-v2.0-cs01-part4-cyber-observable-objects.html

http://docs.oasis-open.org/cti/stix/v2.0/cs01/part4-cyber-observable-objects/stix-v2.0-cs01-part4-cyber-observable-objects.pdf

**Previous version:**

http://docs.oasis-open.org/cti/stix/v2.0/csprd02/part4-cyber-observable-objects/stix-v2.0-csprd02-part4-cyber-observable-objects.docx (Authoritative)

http://docs.oasis-open.org/cti/stix/v2.0/csprd02/part4-cyber-observable-objects/stix-v2.0-csprd02-part4-cyber-observable-objects.html

http://docs.oasis-open.org/cti/stix/v2.0/csprd02/part4-cyber-observable-objects/stix-v2.0-csprd02-part4-cyber-observable-objects.pdf

**Latest version:**

http://docs.oasis-open.org/cti/stix/v2.0/stix-v2.0-part4-cyber-observable-objects.docx (Authoritative)

http://docs.oasis-open.org/cti/stix/v2.0/stix-v2.0-part4-cyber-observable-objects.html

http://docs.oasis-open.org/cti/stix/v2.0/stix-v2.0-part4-cyber-observable-objects.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.
- *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.
- (this document) *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.
- *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. This document defines a set of cyber observable objects that can be used in STIX and elsewhere.

**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-Pt4-Cyb-Objects]**

*STIX™ Version 2.0. Part 4: Cyber Observable Objects*. Edited by Trey Darley and Ivan Kirillov. 19 July 2017. OASIS Committee Specification 01.

http://docs.oasis-open.org/cti/stix/v2.0/cs01/part4-cyber-observable-objects/stix-v2.0-cs01-part4-cyber-observable-objects.html. Latest version:
http://docs.oasis-open.org/cti/stix/v2.0/stix-v2.0-part4-cyber-observable-objects.html.

# Notices

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

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 entity.

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 4: Cyber Observable Objects*) contains the definitions for the various 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 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

**[IPFIX]** IANA, "IP Flow Information Export (IPFIX) Entities", December 2016, [Online]. Available: http://www.iana.org/assignments/ipfix/ipfix.xhtml

**[ISO639-2]** "ISO 639-2:1998 Codes for the representation of names of languages -- Part 2: Alpha-3 code", 1998. [Online]. Available: http://www.iso.org/iso/catalogue_detail?csnumber=4767

[Media Types] N. Freed, M. Kucherawy, M. Baker and B. Hoehrmann, “Media Types”, IANA, December 2016. [Online]. Available: http://www.iana.org/assignments/media-types/media-types.xhtml

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, http://www.rfc-editor.org/info/rfc1034.

[RFC2047] Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, DOI 10.17487/RFC2047, November 1996, http://www.rfc-editor.org/info/rfc2047.

[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.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, http://www.rfc-editor.org/info/rfc3986.

[RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, DOI 10.17487/RFC5322, October 2008, http://www.rfc-editor.org/info/rfc5322.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, DOI 10.17487/RFC5890, August 2010, http://www.rfc-editor.org/info/rfc5890.

[Port Numbers] J.Touch, A. Mankin, E. Kohler, et. al., “Service Name and Transport Protocol Port Number Registry”, IANA, January 2017. [Online]. Available: http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

[NVD] Official Common Platform Enumeration (CPE) Dictionary, National Vulnerability Database [Online]. Available: https://nvd.nist.gov/cpe.cfm

[X.509] X.509 : Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks, ITU, October 2016. [Online]. Available: https://www.itu.int/rec/T-REC-X.509/

## 1.3 Naming Requirements

### 1.3.1 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.

### 1.3.2 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.

## 1.4 Document Conventions

### 1.4.1 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.

### 1.4.2 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.

# 2 Defined Object Data Models

## 2.1 Artifact Object

Type Name: `artifact`

The Artifact Object permits capturing an array of bytes (8-bits), as a base64-encoded string, or linking to a file-like payload. The size of the base64-encoded data captured in the **payload_bin** property MUST be less than or equal to 10MB.

One of **payload_bin** or **url** MUST be provided. It is incumbent on object creators to ensure that the URL is accessible for downstream consumers. If a URL is provided, then the **hashes** property MUST contain the hash of the URL contents.

### 2.1.1 Properties

| Common Properties | | |
|---|---|---|
| **type, extensions** | | |
| Artifact Object Specific Properties | | |
| **mime_type, payload_bin, url, hashes** | | |
| Property Name | Type | Description |
| **type** (required) | `string` | The value of this property MUST be `artifact`. |
| **mime_type** (optional) | `string` | The value of this property MUST be a valid MIME type as specified in the IANA Media Types registry [Media Types]. |
| **payload_bin** (optional) | `binary` | Specifies the binary data contained in the artifact as a base64-encoded string. This property MUST NOT be present if url is provided. |
| **url** (optional) | `string` | The value of this property MUST be a valid URL that resolves to the unencoded content. This property MUST NOT be present if **payload_bin** is provided. |
| **hashes** (optional) | `hashes` | Specifies a dictionary of hashes for the contents of the **url** or the **payload_bin**. This MUST be provided when the **url** property is present. |

Examples

*Basic Image Artifact*

```
{
  "0": {
    "type": "artifact",
    "mime_type": "image/jpeg",
    "payload_bin": "VBORw0KGgoAAAANSUhEUgAAADI== ..."
    }
}
```

## 2.2 AS Object

Type Name: `autonomous-system`

The AS object represents the properties of an Autonomous System (AS).

### 2.2.1 Properties

| Common Properties | | |
|---|---|---|
| **type, extensions** | | |
| AS Object Specific Properties | | |
| **number, name, rir** | | |
| Property Name | Type | Description |
| **type** (required) | `string` | The value of this property MUST be `autonomous-system`. |
| **number** (required) | `integer` | Specifies the number assigned to the AS. Such assignments are typically performed by a Regional Internet Registry (RIR). |
| **name** (optional) | `string` | Specifies the name of the AS. |
| **rir** (optional) | `string` | Specifies the name of the Regional Internet Registry (RIR) that assigned the number to the AS. |

Examples

*Basic AS Object*

```
{
  "0": {
    "type": "autonomous-system",
    "number": 15139,
    "name": "Slime Industries",
    "rir": "ARIN"
  }
}
```

## 2.3 Directory Object

Type Name: `directory`

The Directory Object represents the properties common to a file system directory.

### 2.3.1 Properties

| Common Properties | | |
|---|---|---|
| **type, extensions** | | |
| File Object Specific Properties | | |
| **path, path_enc, created, modified, accessed, contains_refs** | | |
| Property Name | Type | Description |
| **type** (required) | `string` | The value of this property MUST be `directory`. |
| **path** (required) | `string` | Specifies the path, as originally observed, to the directory on the file system. |
| **path_enc** (optional) | `string` | Specifies the observed encoding for the path. The value MUST be specified if the path is stored in a non-Unicode encoding. This value MUST be specified using the corresponding name from the 2013-12-20 |

| | | |
|---|---|---|
| | | revision of the IANA 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. |
| **created** (optional) | `timestamp` | Specifies the date/time the directory was created. |
| **modified** (optional) | `timestamp` | Specifies the date/time the directory was last written to/modified. |
| **accessed** (optional) | `timestamp` | Specifies the date/time the directory was last accessed. |
| **contains_refs** (optional) | `list` of type `object-ref` | Specifies a list of references to other File and/or Directory Objects contained within the directory.<br><br>The objects referenced in this list MUST be of type `file` or `directory`. |

Examples

*Basic directory*

```
{
  "0": {
    "type": "directory",
    "path": "C:\\Windows\\System32"
    }
}
```

## 2.4 Domain Name Object

Type Name: `domain-name`

The Domain Name represents the properties of a network domain name.

### 2.4.1 Properties

| Common Properties | | |
|---|---|---|
| **type, extensions** | | |
| Domain Name Object Specific Properties | | |
| **value, resolves_to_refs** | | |
| Property Name | Type | Description |
| **type** (required) | `string` | The value of this property MUST be `domain-name`. |
| **value** (required) | `string` | Specifies the value of the domain name. The value of this property MUST conform to [RFC1034], and each domain and sub-domain contained within the domain name MUST conform to [RFC5890]. |
| **resolves_to_refs** (optional) | `list` of type `object-ref` | Specifies a list of references to one or more IP addresses or domain names that the domain name resolves to.<br><br>The objects referenced in this list MUST be of type `ipv4-addr` or `ipv6-addr` or `domain-name` (for cases such as CNAME records). |

Examples

*Basic FQDN*

```
{
  "0": {
    "type": "domain-name",
    "value": "example.com",
    "resolves_to_refs": [
      "1"
    ]
  },
  "1": {
    "type": "ipv4-addr",
    "value": "198.51.100.2"
  }
}
```

## 2.5 Email Address Object

Type Name: `email-addr`

The Email Address Object represents a single email address.

### 2.5.1 Properties

| Common Properties | | |
|---|---|---|
| **type, extensions** | | |
| **Email Address Object Specific Properties** | | |
| **value, display_name, belongs_to_ref** | | |
| **Property Name** | **Type** | **Description** |
| **type** (required) | `string` | The value of this property MUST be `email-addr`. |
| **value** (required) | `string` | Specifies a single email address. This MUST NOT include the display name.<br><br>This property corresponds to the *addr-spec* construction in section 3.4 of [RFC5322], for example, `jane.smith@example.com`. |
| **display_name** (optional) | `string` | Specifies a single email display name, i.e., the name that is displayed to the human user of a mail application.<br><br>This property corresponds to the *display-name* construction in section 3.4 of [RFC5322], for example, `Jane Smith`. |
| **belongs_to_ref** (optional) | `object-ref` | Specifies the user account that the email address belongs to, as a reference to a User Account Object.<br><br>The object referenced in this property MUST be of type `user-account`. |

Examples

*Basic Email Address*

```
{
  "0": {
    "type": "email-addr",
    "value": "john@example.com",
```

```
    "display_name": "John Doe"
  }
}
```

## 2.6 Email Message Object

Type Name: `email-message`
The Email Message Object represents an instance of an email message, corresponding to the internet message format described in [RFC5322] and related RFCs.

Header field values that have been encoded as described in section 2 of [RFC2047] MUST be decoded before inclusion in Email Message Object properties. For example, `this is some text` MUST be used instead of `=?iso-8859-1?q?this=20is=20some=20text?=`. Any characters in the encoded value which cannot be decoded into Unicode SHOULD be replaced with the 'REPLACEMENT CHARACTER' (U+FFFD). If it is necessary to capture the header value as observed, this can be achieved by referencing an Artifact Object through the **raw_email_ref** property.

### 2.6.1 Properties

| Common Properties | | |
|---|---|---|
| **type, extensions** | | |
| Email Message Object Specific Properties | | |
| **is_multipart, date, content_type, from_ref, sender_ref, to_refs, cc_refs, bcc_refs, subject, received_lines, additional_header_fields, body, body_multipart, raw_email_ref** | | |
| Property Name | Type | Description |
| **type** (required) | `string` | The value of this property MUST be `email-message`. |
| **is_multipart** (required) | `boolean` | Indicates whether the email body contains multiple MIME parts. |
| **date** (optional) | `timestamp` | Specifies the date/time that the email message was sent. |
| **content_type** (optional) | `string` | Specifies the value of the “Content-Type” header of the email message. |
| **from_ref** (optional) | `object-ref` | Specifies the value of the “From:” header of the email message. The "From:" field specifies the author(s) of the message, that is, the mailbox(es) of the person(s) or system(s) responsible for the writing of the message.<br><br>The object referenced in this property MUST be of type `email-address`. |
| **sender_ref** (optional) | `object-ref` | Specifies the value of the “Sender” field of the email message. The "Sender:" field specifies the mailbox of the agent responsible for the actual transmission of the message.<br><br>The object referenced in this property MUST |

| | | |
|---|---|---|
| | | be of type `email-address`. |
| **to_refs** (optional) | `list` of type `object-ref` | Specifies the mailboxes that are "To:" recipients of the email message.<br><br>The objects referenced in this list MUST be of type `email-address`. |
| **cc_refs** (optional) | `list` of type `object-ref` | Specifies the mailboxes that are "CC:" recipients of the email message.<br><br>The objects referenced in this list MUST be of type `email-address`. |
| **bcc_refs** (optional) | `list` of type `object-ref` | Specifies the mailboxes that are "BCC:" recipients of the email message.<br><br>As per [RFC5322], this list may be empty, which should not be treated the same as the key being absent.<br><br>The objects referenced in this list MUST be of type `email-address`. |
| **subject** (optional) | `string` | Specifies the subject of the email message. |
| **received_lines** (optional) | `list` of type `string` | Specifies one or more "Received" header fields that may be included in the email headers.<br><br>List values MUST appear in the same order as present in the email message. |
| **additional_header_fields** (optional) | `dictionary` | Specifies any other header fields (except for **date**, **received_lines**, **content_type**, **from_ref**, **sender_ref**, **to_refs**, **cc_refs**, **bcc_refs**, and **subject**) found in the email message, as a dictionary.<br><br>Each key/value pair in the dictionary represents the name/value of a single header field or names/values of a header field that occurs more than once. Each dictionary key SHOULD be a case-preserved version of the header field name. For cases where a header field occurs exactly once, the corresponding value for the dictionary key MUST be a `string`. For cases where a header field occurs more than once, the corresponding value for the dictionary key MUST be a `list` of type `string`, where each `string` in the `list` represents a single value of the header field. |
| **body** (optional) | `string` | Specifies a `string` containing the email body. This property MUST NOT be used if **is_multipart** is true. |

| | | |
|---|---|---|
| **body_multipart** (optional) | `list` of type `mime-part-type` | Specifies a list of the MIME parts that make up the email body. This property MUST NOT be used if **is_multipart** is false. |
| **raw_email_ref** (optional) | `object-ref` | Specifies the raw binary contents of the email message, including both the headers and body, as a reference to an Artifact Object.<br><br>The object referenced in this property MUST be of type `artifact`. |

### 2.6.2 Email MIME Component Type

Type Name: `mime-part-type`
Specifies one component of a multi-part email body.

There is no property to capture the value of the "Content-Transfer-Encoding" header field, since the body MUST be decoded before being represented in the **body** property.

One of **body** OR **body_raw_ref** MUST be included.

#### 2.6.2.1 Properties

| Property Name | Type | Description |
|---|---|---|
| **body** (optional) | `string` | Specifies the contents of the MIME part if the **content_type** is not provided or starts with `text/` (e.g., in the case of plain text or HTML email).<br><br>For inclusion in this property, the contents MUST be decoded to Unicode. Note that the charset provided in **content_type** is for informational usage and not for decoding of this property. |
| **body_raw_ref** (optional) | `object-ref` | Specifies the contents of non-textual MIME parts, that is those whose **content_type** does not start with `text/`, as a reference to an Artifact Object or File Object.<br><br>The object referenced in this property MUST be of type `artifact` or `file`. For use cases where conveying the actual data contained in the MIME part is of primary importance, `artifact` SHOULD be used. Otherwise, for use cases where conveying metadata about the file-like properties of the MIME part is of primary importance, `file` SHOULD be used. |
| **content_type** (optional) | `string` | Specifies the value of the "Content-Type" header field of the MIME part.<br><br>Any additional "Content-Type" header field parameters such as `charset` SHOULD be included in this property.<br><br>Example: |

| | | |
|---|---|---|
| | | `text/html; charset=UTF-8` |
| **`content_disposition`** (optional) | `string` | Specifies the value of the "Content-Disposition" header field of the MIME part. |

Examples

*Simple Email Message*

```
{
  "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"],
    "is_multipart": false,
    "date": "1997-11-21T15:55:06.000Z",
    "subject": "Saying Hello"
  }
}
```

*Simple Email Message with Additional Header Properties*

```
{
  "0": {
    "type": "email-addr",
    "value": "joe@example.com",
    "display_name": "Joe Smith"
  },
  "1": {
    "type": "email-addr",
    "value": "bob@example.com",
    "display_name": "Bob Smith"
  },
  "2": {
    "type": "email-message",
    "from_ref": "0",
    "to_refs": [
      "1"
    ],
    "is_multipart": false,
    "date": "2004-04-19T12:22:23.000Z",
    "subject": "Did you see this?",
    "additional_header_fields": {
      "Reply-To": [
        "steve@example.com",
        "jane@example.com"
      ]
    }
  }
}
```

*Complex MIME Email Message*

```
{
  "0": {
    "type": "email-message",
    "is_multipart": true,
```

```
    "received_lines": [
      "from   mail.example.com   ([198.51.100.3])   by   smtp.gmail.com   with   ESMTPSA   id
q23sm23309939wme.17.2016.07.19.07.20.32   (version=TLS1_2   cipher=ECDHE-RSA-AES128-GCM-SHA256
bits=128/128); Tue, 19 Jul 2016 07:20:40 -0700 (PDT)"
    ],
    "content_type": "multipart/mixed",
    "date": "2016-06-19T14:20:40.000Z",
    "from_ref": "1",
    "to_refs": [
      "2"
    ],
    "cc_refs": [
      "3"
    ],
    "subject": "Check out this picture of a cat!",
    "additional_header_fields": {
      "Content-Disposition": "inline",
      "X-Mailer": "Mutt/1.5.23",
      "X-Originating-IP": "198.51.100.3"
    },
    "body_multipart": [
      {
        "content_type": "text/plain; charset=utf-8",
        "content_disposition": "inline",
        "body": "Cats are funny!"
      },
      {
        "content_type": "image/png",
        "content_disposition": "attachment; filename=\"tabby.png\"",
        "body_raw_ref": "4"
      },
      {
        "content_type": "application/zip",
        "content_disposition": "attachment; filename=\"tabby_pics.zip\"",
        "body_raw_ref": "5"
      }
    ]
  },
  "1": {
    "type": "email-addr",
    "value": "jdoe@example.com",
    "display_name": "John Doe"
  },
  "2": {
    "type": "email-addr",
    "value": "bob@example.com",
    "display_name": "Bob Smith"
  },
  "3": {
    "type": "email-addr",
    "value": "mary@example.com",
    "display_name": "Mary Jones"
  },
  "4":{
    "type": "artifact",
    "mime_type": "image/jpeg",
    "payload_bin": "VBORw0KGgoAAAANSUhEUgAAADI== ...",
    "hashes": {
      "SHA-256": "effb46bba03f6c8aea5c653f9cf984f170dcdd3bbbe2ff6843c3e5da0e698766"
    }
  },
  "5": {
    "type": "file",
```

```
    "name": "tabby_pics.zip",
    "magic_number_hex": "504B0304",
    "hashes": {
      "SHA-256": "fe90a7e910cb3a4739bed9180e807e93fa70c90f25a8915476f5e4bfbac681db"
    }
  }
}
```

## 2.7 File Object

Type Name: `file`

The File Object represents the properties of a file. A File Object MUST contain at least one of **hashes** or **name**.

### 2.7.1 Properties

| Common Properties | | |
|---|---|---|
| **type, extensions** | | |
| File Object Specific Properties | | |
| **hashes, size, name, name_enc, magic_number_hex, mime_type, created, modified, accessed, parent_directory_ref, is_encrypted, encryption_algorithm, decryption_key, contains_refs, content_ref** | | |
| Property Name | Type | Description |
| **type** (required) | `string` | The value of this property MUST be `file`. |
| **extensions** (optional) | `dictionary` | The File Object defines the following extensions. In addition to these, producers MAY create their own.<br><br>`ntfs-ext`, `raster-image-ext`, `pdf-ext`, `archive-ext`, `windows-pebinary-ext`<br><br>Dictionary keys MUST identify the extension type by name.<br><br>The corresponding dictionary values MUST contain the contents of the extension instance. |
| **hashes** (optional) | `hashes` | Specifies a dictionary of hashes for the file. |
| **size** (optional) | `integer` | Specifies the size of the file, in bytes. The value of this property MUST NOT be negative. |
| **name** (optional) | `string` | Specifies the name of the file. |
| **name_enc** (optional) | `string` | Specifies the observed encoding for the name of the file. This value MUST be specified using the corresponding name from the 2013-12-20 revision of the IANA character set registry [Character Sets]. If the value from the Preferred MIME Name column for a character set is defined, this value MUST be used; if it is not defined, then the value from the Name column in the registry MUST be used instead. |

| | | |
|---|---|---|
| | | This property allows for the capture of the original text encoding for the file name, which may be forensically relevant; for example, a file on an NTFS volume whose name was created using the windows-1251 encoding, commonly used for languages based on Cyrillic script. |
| **magic_number_hex** (optional) | `hex` | Specifies the hexadecimal constant ("magic number") associated with a specific file format that corresponds to the file, if applicable. |
| **mime_type** (optional) | `string` | Specifies the MIME type name specified for the file, e.g., `application/msword`.<br><br>Whenever feasible, this value SHOULD be one of the values defined in the Template column in the IANA media type registry [Media Types].<br><br>Maintaining a comprehensive universal catalog of all extant file types is obviously not possible. When specifying a MIME Type not included in the IANA registry, implementers should use their best judgement so as to facilitate interoperability. |
| **created** (optional) | `timestamp` | Specifies the date/time the file was created. |
| **modified** (optional) | `timestamp` | Specifies the date/time the file was last written to/modified. |
| **accessed** (optional) | `timestamp` | Specifies the date/time the file was last accessed. |
| **parent_directory_ref** (optional) | `object-ref` | Specifies the parent directory of the file, as a reference to a Directory Object.<br><br>The object referenced in this property MUST be of type `directory`. |
| **is_encrypted** (optional) | `boolean` | Specifies whether the file is encrypted. |
| **encryption_algorithm** (optional) | `open-vocab` | Specifies the name of the encryption algorithm used to encrypt the file. This is an open vocabulary and values SHOULD come from the `encryption-algo-ov` vocabulary.<br><br>This property MUST NOT be used if **is_encrypted** is `false` or not included. |
| **decryption_key** (optional) | `string` | Specifies the decryption key used to decrypt the file.<br><br>This property MUST NOT be used if **is_encrypted** is `false` or not included. |
| **contains_refs** (optional) | `list` of type `object-ref` | Specifies a list of references to other Observable Objects contained within the file, such as another file that is appended to the end of the file, or an IP address that is contained somewhere in the the file. |

| | | |
|---|---|---|
| | | This is intended for use cases other than those targeted by the Archive extension. |
| **content_ref** (optional) | `object-ref` | Specifies the content of the file, represented as an Artifact Object.<br><br>The object referenced in this property MUST be of type `artifact`. |

Examples

*Basic file with file system properties without observed encoding*

```
{
  "0": {
    "type": "file",
    "hashes": {
      "SHA-256": "fe90a7e910cb3a4739bed9180e807e93fa70c90f25a8915476f5e4bfbac681db"
    },
    "size": 25536,
    "name": "foo.dll"
  }
}
```

*Basic file with file system properties with observed encoding*

```
{
  "0": {
    "type": "file",
    "hashes": {
      "SHA-256": "841a8921140aba50671ebb0770fecc4ee308c4952cfeff8de154ab14eeef4649"
    },
    "name": "quêry.dll",
    "name_enc": "windows-1252"
  }
}
```

In this example, the file name would have originally appeared using the bytes 71 75 ea 72 79 2e 64 6c 6c. Representing it in UTF-8, as required for JSON, would use the bytes 71 75 c3 aa 72 79 2e 64 6c 6c.

*Basic file with parent directory*

```
{
  "0": {
    "type": "directory",
    "path": "C:\\Windows\\System32"
  },
  "1": {
    "type": "file",
    "hashes": {
      "SHA-256": "ceafbfd424be2ca4a5f0402cae090dda2fb0526cf521b60b60077c0f622b285a"
    },
    "parent_directory_ref": "0",
    "name": "qwerty.dll"
  }
}
```

### 2.7.2 Archive File Extension

Type Name: `archive-ext`

The Archive File extension specifies a default extension for capturing properties specific to archive files. The key for this extension when used in the **extensions** dictionary MUST be

archive-ext.

#### 2.7.2.1 Properties

| Property Name | Type | Description |
|---|---|---|
| **contains_refs** (required) | list of type object-ref | Specifies the files contained in the archive, as a reference to one or more other File Objects.<br><br>The objects referenced in this list MUST be of type file. |
| **version** (optional) | string | Specifies the version of the archive type used in the archive file. |
| **comment** (optional) | string | Specifies a comment included as part of the archive file. |

Examples

*Basic unencrypted ZIP Archive*

```
{
  "0": {
    "type": "file",
    "hashes": {
      "SHA-256": "ceafbfd424be2ca4a5f0402cae090dda2fb0526cf521b60b60077c0f622b285a"
    }
  },
  "1": {
    "type": "file",
    "hashes": {
      "SHA-256": "19c549ec2628b989382f6b280cbd7bb836a0b461332c0fe53511ce7d584b89d3"
    }
  },
  "2": {
    "type": "file",
    "hashes": {
      "SHA-256": "0969de02ecf8a5f003e3f6d063d848c8a193aada092623f8ce408c15bcb5f038"
    }
  },
  "3": {
    "type": "file",
    "name": "foo.zip",
    "hashes": {
      "SHA-256": "35a01331e9ad96f751278b891b6ea09699806faedfa237d40513d92ad1b7100f"
    },
    "mime_type": "application/zip",
    "extensions": {
      "archive-ext": {
        "contains_refs": [
          "0",
          "1",
          "2"
        ],
        "version": "5.0"
      }
    }
  }
}
```

### 2.7.3 NTFS File Extension

Type Name: ntfs-ext

The NTFS file extension specifies a default extension for capturing properties specific to the storage of the file on the NTFS file system. The key for this extension when used in the **extensions** dictionary MUST be `ntfs-ext`. An object using the NTFS File Extension MUST contain at least one property from this extension.

#### 2.7.3.1 Properties

| Property Name | Type | Description |
|---|---|---|
| **sid** (optional) | `string` | Specifies the security ID (SID) value assigned to the file. |
| **alternate_data_streams** (optional) | `list` of type `alternate-data-stream-type` | Specifies a list of NTFS alternate data streams that exist for the file. |

### 2.7.3.2 Alternate Data Stream Type

Type Name: `alternate-data-stream-type`

The Alternate Data Stream type represents an NTFS alternate data stream.

#### 2.7.3.2.1 Properties

| Property Name | Type | Description |
|---|---|---|
| **name** (required) | `string` | Specifies the name of the alternate data stream. |
| **hashes** (optional) | `hashes` | Specifies a dictionary of hashes for the data contained in the alternate data stream. |
| **size** (optional) | `integer` | Specifies the size of the alternate data stream, in bytes. The value of this property MUST NOT be negative. |

Examples

*NTFS File with a single alternate data stream*

```
{
  "0": {
    "type": "file",
    "hashes": {
      "SHA-256": "35a01331e9ad96f751278b891b6ea09699806faedfa237d40513d92ad1b7100f"
    },
    "extensions": {
      "ntfs-ext": {
        "alternate_data_streams": [
          {
            "name": "second.stream",
            "size": 25536
          }
        ]
      }
    }
  }
}
```

## 2.7.4 PDF File Extension

Type Name: `pdf-ext`

The PDF file extension specifies a default extension for capturing properties specific to PDF files. The key for this extension when used in the **extensions** dictionary MUST be `pdf-ext`. An object using the PDF File Extension MUST contain at least one property from this extension.

#### 2.7.4.1 Properties

| Property Name | Type | Description |
|---|---|---|
| **version** (optional) | string | Specifies the decimal version number of the string from the PDF header that specifies the version of the PDF specification to which the PDF file conforms. E.g., `1.4`. |
| **is_optimized** (optional) | boolean | Specifies whether the PDF file has been optimized. |
| **document_info_dict** (optional) | dictionary | Specifies details of the PDF document information dictionary (DID), which includes properties like the document creation data and producer, as a dictionary. Each key in the dictionary SHOULD be a case-preserved version of the corresponding entry in the document information dictionary without the prepended forward slash, e.g., `Title`. The corresponding value for the key MUST be the value specified for the document information dictionary entry, as a string. |
| **pdfid0** (optional) | string | Specifies the first file identifier found for the PDF file. |
| **pdfid1** (optional) | string | Specifies the second file identifier found for the PDF file. |

Examples

*Basic PDF file*

```
{
  "0": {
    "type": "file",
    "hashes": {
      "SHA-256": "35a01331e9ad96f751278b891b6ea09699806faedfa237d40513d92ad1b7100f"
    },
    "extensions": {
      "pdf-ext": {
        "version": "1.7",
        "document_info_dict": {
          "Title": "Sample document",
          "Author": "Adobe Systems Incorporated",
          "Creator": "Adobe FrameMaker 5.5.3 for Power Macintosh",
          "Producer": "Acrobat Distiller 3.01 for Power Macintosh",
          "CreationDate": "20070412090123-02"
        },
        "pdfid0": "DFCE52BD827ECF765649852119D",
        "pdfid1": "57A1E0F9ED2AE523E313C"
      }
    }
  }
}
```

### 2.7.5 Raster Image File Extension

Type Name: `raster-image-ext`

The Raster Image file extension specifies a default extension for capturing properties specific to raster image files. The key for this extension when used in the **extensions** dictionary MUST be `raster-image-ext`. An object using the Raster Image File Extension

MUST contain at least one property from this extension.

#### 2.7.5.1 Properties

| Property Name | Type | Description |
|---|---|---|
| **image_height** (optional) | integer | Specifies the height of the image in the image file, in pixels. |
| **image_width** (optional) | integer | Specifies the width of the image in the image file, in pixels. |
| **bits_per_pixel** (optional) | integer | Specifies the sum of bits used for each color channel in the image file, and thus the total number of pixels used for expressing the color depth of the image. |
| **image_compression_algorithm** (optional) | string | Specifies the name of the compression algorithm used to compress the image in the image file, if applicable. |
| **exif_tags** (optional) | dictionary | Specifies the set of EXIF tags found in the image file, as a dictionary. Each key/value pair in the dictionary represents the name/value of a single EXIF tag. Accordingly, each dictionary key MUST be a case-preserved version of the EXIF tag name, e.g., `XResolution`. Each dictionary value MUST be either an `integer` (for int* EXIF datatypes) or a `string` (for all other EXIF datatypes). |

Examples

*Simple Image File with EXIF Data*

```
{
  "0": {
    "type": "file",
    "name": "picture.jpg",
    "hashes": {
      "SHA-256": "35a01331e9ad96f751278b891b6ea09699806faedfa237d40513d92ad1b7100f"
    },
    "extensions": {
      "raster-image-ext": {
        "exif_tags": {
          "Make": "Nikon",
          "Model": "D7000",
          "XResolution": 4928,
          "YResolution": 3264
        }
      }
    }
  }
}
```

### 2.7.6 Windows™ PE Binary File Extension

Type Name: `windows-pebinary-ext`

The Windows™ PE Binary File extension specifies a default extension for capturing properties specific to Windows portable executable (PE) files. The key for this extension when used in the **extensions** dictionary MUST be `windows-pebinary-ext`.

#### 2.7.6.1 Properties

| Property Name | Type | Description |
|---|---|---|
| **pe_type** (required) | open-vocab | Specifies the type of the PE binary. This is an open vocabulary and values SHOULD come from the windows-pebinary-type-ov vocabulary. |
| **imphash** (optional) | string | Specifies the special import hash, or 'imphash', calculated for the PE Binary based on its imported libraries and functions. For more information on the imphash algorithm, see the original article by Mandiant/FireEye: https://www.fireeye.com/blog/threat-research/2014/01/tracking-malware-import-hashing.html. |
| **machine_hex** (optional) | hex | Specifies the type of target machine. |
| **number_of_sections** (optional) | integer | Specifies the number of sections in the PE binary, as a non-negative integer. |
| **time_date_stamp** (optional) | timestamp | Specifies the time when the PE binary was created. The timestamp value MUST be precise to the second. |
| **pointer_to_symbol_table_hex** (optional) | hex | Specifies the file offset of the COFF symbol table. |
| **number_of_symbols** (optional) | integer | Specifies the number of entries in the symbol table of the PE binary, as a non-negative integer. |
| **size_of_optional_header** (optional) | integer | Specifies the size of the optional header of the PE binary. The value of this property MUST NOT be negative. |
| **characteristics_hex** (optional) | hex | Specifies the flags that indicate the file's characteristics. |
| **file_header_hashes** (optional) | hashes | Specifies any hashes that were computed for the file header. |
| **optional_header** (optional) | windows-pe-optional-header-type | Specifies the PE optional header of the PE binary. |
| **sections** (optional) | list of type windows-pe-section-type | Specifies metadata about the sections in the PE file. |

### 2.7.6.2 Windows™ PE Binary Vocabulary

Vocabulary Name: `windows-pebinary-type-ov`

An open vocabulary of Windows PE binary types.

| Value | Description |
|---|---|
| `exe` | Specifies that the PE binary is an executable image (i.e., not an OBJ or DLL). |
| `dll` | Specifies that the PE binary is a dynamically linked library (DLL). |
| `sys` | Specifies that the PE binary is a device driver (SYS). |

### 2.7.6.3 Windows™ PE Optional Header Type

Type Name: `windows-pe-optional-header-type`

The Windows PE Optional Header type represents the properties of the PE optional header.

#### 2.7.6.3.1 Properties

| Property Name | Type | Description |
|---|---|---|
| **`magic_hex`** (optional) | `hex` | Specifies the hex value that indicates the type of the PE binary. |
| **`major_linker_version`** (optional) | `integer` | Specifies the linker major version number. |
| **`minor_linker_version`** (optional) | `integer` | Specifies the linker minor version number. |
| **`size_of_code`** (optional) | `integer` | Specifies the size of the code (text) section. If there are multiple such sections, this refers to the sum of the sizes of each section. The value of this property MUST NOT be negative. |
| **`size_of_initialized_data`** (optional) | `integer` | Specifies the size of the initialized data section. If there are multiple such sections, this refers to the sum of the sizes of each section. The value of this property MUST NOT be negative. |
| **`size_of_uninitialized_data`** (optional) | `integer` | Specifies the size of the uninitialized data section. If there are multiple such sections, this refers to the sum of the sizes of each section. The value of this property MUST NOT be negative. |
| **`address_of_entry_point`** (optional) | `integer` | Specifies the address of the entry point relative to the image base when the executable is loaded into memory. |
| **`base_of_code`** (optional) | `integer` | Specifies the address that is relative to the image base of the beginning-of-code section when it is loaded into memory. |
| **`base_of_data`** (optional) | `integer` | Specifies the address that is relative to the |

| | | |
|---|---|---|
| | | image base of the beginning-of-data section when it is loaded into memory. |
| **image_base** (optional) | integer | Specifies the preferred address of the first byte of the image when loaded into memory. |
| **section_alignment** (optional) | integer | Specifies the alignment (in bytes) of PE sections when they are loaded into memory. |
| **file_alignment** (optional) | integer | Specifies the factor (in bytes) that is used to align the raw data of sections in the image file. |
| **major_os_version** (optional) | integer | Specifies the major version number of the required operating system. |
| **minor_os_version** (optional) | integer | Specifies the minor version number of the required operating system. |
| **major_image_version** (optional) | integer | Specifies the major version number of the image. |
| **minor_image_version** (optional) | integer | Specifies the minor version number of the image. |
| **major_subsystem_version** (optional) | integer | Specifies the major version number of the subsystem. |
| **minor_subsystem_version** (optional) | integer | Specifies the minor version number of the subsystem. |
| **win32_version_value_hex** (optional) | hex | Specifies the reserved win32 version value. |
| **size_of_image** (optional) | integer | Specifies the size of the image in bytes, including all headers, as the image is loaded in memory. The value of this property MUST NOT be negative. |
| **size_of_headers** (optional) | integer | Specifies the combined size of the MS-DOS, PE header, and section headers, rounded up to a multiple of the value specified in the file_alignment header. The value of this property MUST NOT be negative. |
| **checksum_hex** (optional) | hex | Specifies the checksum of the PE binary. |
| **subsystem_hex** (optional) | hex | Specifies the subsystem (e.g., GUI, device driver, etc.) that is required to run this image. |
| **dll_characteristics_hex** (optional) | hex | Specifies the flags that characterize the PE binary. |
| **size_of_stack_reserve** (optional) | integer | Specifies the size of the stack to reserve, in bytes. The value of this property MUST NOT be negative. |
| **size_of_stack_commit** (optional) | integer | Specifies the size of the stack to commit, in bytes. The value of this property MUST NOT be negative. |

| | | |
|---|---|---|
| **size_of_heap_reserve** (optional) | integer | Specifies the size of the local heap space to reserve, in bytes.. The value of this property MUST NOT be negative. |
| **size_of_heap_commit** (optional) | integer | Specifies the size of the local heap space to commit, in bytes. The value of this property MUST NOT be negative. |
| **loader_flags_hex** (optional) | hex | Specifies the reserved loader flags. |
| **number_of_rva_and_sizes** (optional) | integer | Specifies the number of data-directory entries in the remainder of the optional header. |
| **hashes** (optional) | hashes | Specifies any hashes that were computed for the optional header. |

#### 2.7.6.4 Windows™ PE Section Type

Type Name: `windows-pe-section-type`

The Windows PE Section type specifies metadata about a PE file section.

##### 2.7.6.4.1 Properties

| Property Name | Type | Description |
|---|---|---|
| **name** (required) | string | Specifies the name of the section. |
| **size** (optional) | integer | Specifies the size of the section, in bytes. The value of this property MUST NOT be negative. |
| **entropy** (optional) | float | Specifies the calculated entropy for the section, as calculated using the Shannon algorithm (https://en.wiktionary.org/wiki/Shannon_entropy). The size of each input character is defined as a byte, resulting in a possible range of 0 through 8. |
| **hashes** (optional) | hashes | Specifies any hashes computed over the section. |

Examples

*Typical EXE File*

```
{
  "0": {
    "type": "file",
    "hashes": {
      "SHA-256": "35a01331e9ad96f751278b891b6ea09699806faedfa237d40513d92ad1b7100f"
    },
    "extensions": {
      "windows-pebinary-ext": {
        "pe_type": "exe",
        "machine_hex": "014c",
        "number_of_sections": 4,
        "time_date_stamp": "2016-01-22T12:31:12Z",
        "pointer_to_symbol_table_hex": "74726144",
        "number_of_symbols": 4542568,
        "size_of_optional_header": 224,
        "characteristics_hex": "818f",
        "optional_header": {
          "magic_hex": "010b",
          "major_linker_version": 2,
```

```
          "minor_linker_version": 25,
          "size_of_code": 512,
          "size_of_initialized_data": 283648,
          "size_of_uninitialized_data": 0,
          "address_of_entry_point": 4096,
          "base_of_code": 4096,
          "base_of_data": 8192,
          "image_base": 14548992,
          "section_alignment": 4096,
          "file_alignment": 4096,
          "major_os_system_version": 1,
          "minor_os_system_version": 0,
          "major_image_version": 0,
          "minor_image_version": 0,
          "major_subsystem_version": 4,
          "minor_subsystem_version": 0,
          "win32_version_value_hex": "00",
          "size_of_image": 299008,
          "size_of_headers": 4096,
          "checksum_hex": "00",
          "subsystem_hex": "03",
          "dll_characteristics_hex": "00",
          "size_of_stack_reserve": 100000,
          "size_of_stack_commit": 8192,
          "size_of_heap_reserve": 100000,
          "size_of_heap_commit": 4096,
          "loader_flags_hex": "abdbffde",
          "number_of_rva_and_sizes": 3758087646
        },
        "sections": [
          {
            "name": "CODE",
            "entropy": 0.061089
          },
          {
            "name": "DATA",
            "entropy": 7.980693
          },
          {
            "name": "NicolasB",
            "entropy": 0.607433
          },
          {
            "name": ".idata",
            "entropy": 0.607433
          }
        ]
      }
    }
  }
}
```


## 2.8 IPv4 Address Object

Type Name: `ipv4-addr`

The IPv4 Address Object represents one or more IPv4 addresses expressed using CIDR notation.

### 2.8.1 Properties

| Common Properties |
|---|

| type, extensions | | |
|---|---|---|
| IPv4 Address Object Specific Properties | | |
| value, resolves_to_refs, belongs_to_refs | | |
| Property Name | Type | Description |
| **type** (required) | string | The value of this property MUST be ipv4-addr. |
| **value** (required) | string | Specifies one or more IPv4 addresses expressed using CIDR notation.<br><br>If a given IPv4 Address Object represents a single IPv4 address, the CIDR /32 suffix MAY be omitted.<br><br>Example: 10.2.4.5/24 |
| **resolves_to_refs** (optional) | list of type object-ref | Specifies a list of references to one or more Layer 2 Media Access Control (MAC) addresses that the IPv4 address resolves to.<br><br>The objects referenced in this list MUST be of type mac-addr. |
| **belongs_to_refs** (optional) | list of type object-ref | Specifies a list of reference to one or more autonomous systems (AS) that the IPv4 address belongs to.<br><br>The objects referenced in this list MUST be of type autonomous-system. |

Examples

*IPv4 Single Address*

```
{
  "0": {
    "type": "ipv4-addr",
    "value": "198.51.100.3"
  }
}
```

*IPv4 CIDR Block*

```
{
  "0": {
    "type": "ipv4-addr",
    "value": "198.51.100.0/24"
  }
}
```

## 2.9 IPv6 Address Object

Type Name: ipv6-addr

The IPv6 Address Object represents one or more IPv6 addresses expressed using CIDR notation.

### 2.9.1 Properties

| Common Properties |
|---|
| type, extensions |

| IPv6 Address Object Specific Properties | | |
|---|---|---|
| **value, resolves_to_refs, belongs_to_refs** | | |
| Property Name | Type | Description |
| **type** (required) | `string` | The value of this property MUST be `ipv6-addr`. |
| **value** (required) | `string` | Specifies one or more IPv6 addresses expressed using CIDR notation.<br><br>If a given IPv6 Address Object represents a single IPv6 address, the CIDR /128 suffix MAY be omitted. |
| **resolves_to_refs** (optional) | `list` of type `object-ref` | Specifies a list of references to one or more Layer 2 Media Access Control (MAC) addresses that the IPv6 address resolves to.<br><br>The objects referenced in this list MUST be of type `mac-addr`. |
| **belongs_to_refs** (optional) | `list` of type `object-ref` | Specifies a list of reference to one or more autonomous systems (AS) that the IPv6 address belongs to.<br><br>The objects referenced in this list MUST be of type `autonomous-system`. |

Examples

*IPv6 Single Address*

```
{
  "0": {
    "type": "ipv6-addr",
    "value": "2001:0db8:85a3:0000:0000:8a2e:0370:7334"
  }
}
```

*IPv6 CIDR block*

```
{
  "0": {
    "type": "ipv6-addr",
    "value": "2001:0db8::/96"
  }
}
```

## 2.10 MAC Address Object

Type Name: `mac-addr`

The MAC Address Object represents a single Media Access Control (MAC) address.

### 2.10.1 Properties

| Common Properties | | |
|---|---|---|
| **type, extensions** | | |
| MAC Address Object Specific Properties | | |
| **value** | | |
| Property Name | Type | Description |

| | | |
|---|---|---|
| **type** (required) | `string` | The value of this property MUST be `mac-addr`. |
| **value** (required) | `string` | Specifies a single MAC address.<br><br>The MAC address value MUST be represented as a single colon-delimited, lowercase MAC-48 address, which MUST include leading zeros for each octet.<br><br>Example: `00:00:ab:cd:ef:01` |

Examples

*Typical MAC address*

```
{
  "0": {
    "type": "mac-addr",
    "value": "d2:fb:49:24:37:18"
  }
}
```

## 2.11 Mutex Object

Type Name: `mutex`

The Mutex Object represents the properties of a mutual exclusion (mutex) object.

### 2.11.1 Properties

| Common Properties | | |
|---|---|---|
| **type, extensions** | | |
| File Object Specific Properties | | |
| **name** | | |
| Property Name | Type | Description |
| **type** (required) | `string` | The value of this property MUST be `mutex`. |
| **name** (required) | `string` | Specifies the name of the mutex object. |

Examples

*Malware mutex*

```
{
  "0": {
    "type": "mutex",
    "name": "__CLEANSWEEP__"
  }
}
```

## 2.12 Network Traffic Object

Type Name: `network-traffic`

The Network Traffic Object represents arbitrary network traffic that originates from a source and is addressed to a destination. The network traffic MAY or MAY NOT constitute a valid unicast, multicast, or broadcast network connection. This MAY also include traffic that is not established, such as a SYN flood.

To allow for use cases where a source or destination address may be sensitive and not suitable for sharing, such as addresses that are internal to an organization's network, the

source and destination properties (**src_ref** and **dst_ref**, respectively) are defined as optional in the properties table below. However, a Network Traffic Object MUST contain the **protocols** property and at least one of the **src_ref** or **dst_ref** properties and SHOULD contain the **src_port** and **dst_port** properties.

### 2.12.1 Properties

| Common Properties | | |
|---|---|---|
| **type, extensions** | | |
| Network Traffic Specific Properties | | |
| **start, end, is_active, src_ref, dst_ref, src_port, dst_port, protocols, src_byte_count, dst_byte_count, src_packets, dst_packets, ipfix, src_payload_ref, dst_payload_ref, encapsulates_refs, encapsulated_by_ref** | | |
| Property Name | Type | Description |
| **type** (required) | `string` | The value of this property MUST be `network-traffic`. |
| **extensions** (optional) | `dictionary` | The Network Traffic Object defines the following extensions. In addition to these, producers MAY create their own.<br><br>`http-request-ext`, `tcp-ext`, `icmp-ext`, `socket-ext`<br><br>Dictionary keys MUST identify the extension type by name.<br><br>The corresponding dictionary values MUST contain the contents of the extension instance. |
| **start** (optional) | `timestamp` | Specifies the date/time the network traffic was initiated, if known. |
| **end** (optional) | `timestamp` | Specifies the date/time the network traffic ended, if known.<br><br>If the **is_active** property is true, then the **end** property MUST NOT be included. |
| **is_active** (optional) | `boolean` | Indicates whether the network traffic is still ongoing. |
| **src_ref** (optional) | `object-ref` | Specifies the source of the network traffic, as a reference to one or more Observable Objects.<br><br>The objects referenced in this list MUST be of type `ipv4-addr` or `ipv6-addr` or `mac-addr` or `domain-name` (for cases where the IP address for a domain name is unknown). |
| **dst_ref** (optional) | `object-ref` | Specifies the destination of the network traffic, as a reference to one or more Observable Objects.<br><br>The objects referenced in this list MUST be of type `ipv4-addr` or `ipv6-addr` or `mac-addr` or `domain-name` (for cases where the IP address for a domain name |

| | | |
|---|---|---|
| | | is unknown). |
| **src_port** (optional) | integer | Specifies the source port used in the network traffic, as an integer. The port value MUST be in the range of 0 - 65535. |
| **dst_port** (optional) | integer | Specifies the destination port used in the network traffic, as an integer. The port value MUST be in the range of 0 - 65535. |
| **protocols** (required) | list of type string | Specifies the protocols observed in the network traffic, along with their corresponding state.<br><br>Protocols MUST be listed in low to high order, from outer to inner in terms of packet encapsulation. That is, the protocols in the outer level of the packet, such as IP, MUST be listed first.<br><br>The protocol names SHOULD come from the service names defined in the Service Name column of the IANA Service Name and Port Number Registry [Port Numbers]. In cases where there is variance in the name of a network protocol not included in the IANA Registry, content producers should exercise their best judgement, and it is recommended that lowercase names be used for consistency with the IANA registry.<br><br>Examples:<br>`ipv4, tcp, http`<br>`ipv4, udp`<br>`ipv6, tcp, http`<br>`ipv6, tcp, ssl, https` |
| **src_byte_count** (optional) | integer | Specifies the number of bytes sent from the source to the destination. |
| **dst_byte_count** (optional) | integer | Specifies the number of bytes sent from the destination to the source. |
| **src_packets** (optional) | integer | Specifies the number of packets sent from the source to the destination. |
| **dst_packets** (optional) | integer | Specifies the number of packets sent destination to the source. |
| **ipfix** (optional) | dictionary | Specifies any IP Flow Information Export [IPFIX] data for the traffic, as a dictionary. Each key/value pair in the dictionary represents the name/value of a single IPFIX element. Accordingly, each dictionary key SHOULD be a case-preserved version of the IPFIX element name, e.g., `octetDeltaCount`. Each dictionary value MUST be either an integer or a string, as well as a valid IPFIX property. |
| **src_payload_ref** (optional) | object-ref | Specifies the bytes sent from the source to the destination.<br><br>The object referenced in this property MUST be of |

| | | |
|---|---|---|
| | | type `artifact`. |
| **`dst_payload_ref`** (optional) | `object-ref` | Specifies the bytes sent from the destination to the source.<br><br>The object referenced in this property MUST be of type `artifact`. |
| **`encapsulates_refs`** (optional) | `list` of type `object-ref` | Links to other `network-traffic` objects encapsulated by this `network-traffic` object.<br><br>The objects referenced in this property MUST be of type `network-traffic`. |
| **`encapsulated_by_ref`** (optional) | `object-ref` | Links to another `network-traffic` object which encapsulates this object.<br><br>The object referenced in this property MUST be of type `network-traffic`. |

Examples

*Basic TCP Network Traffic*

```
{
  "0": {
    "type": "ipv4-addr",
    "value": "198.51.100.2"
  },
  "1":{
    "type": "ipv4-addr",
    "value": "198.51.100.3"
  },
  "2": {
    "type": "network-traffic",
    "src_ref": "0",
    "dst_ref": "1",
    "protocols": [
      "tcp"
    ]
  }
}
```

*Basic HTTP Network Traffic*

```
{
  "0": {
    "type": "domain-name",
    "value": "example.com"
  },
  "1": {
    "type": "network-traffic",
    "dst_ref": "0",
    "protocols": [
      "ipv4",
      "tcp",
      "http"
    ]
  }
}
```

*Network Traffic with Netflow Data*

```
{
  "0": {
    "type": "ipv4-addr",
```

```
    "value": "203.0.113.1"
  },
  "1": {
    "type": "ipv4-addr",
    "value": "203.0.113.5"
  },
  "2": {
    "type": "network-traffic",
    "src_ref": "0",
    "dst_ref": "1",
    "protocols": [
      "ipv4",
      "tcp"
    ],
    "src_byte_count": 147600,
    "src_packets": 100,
    "ipfix": {
      "minimumIpTotalLength": 32,
      "maximumIpTotalLength": 2556
    }
  }
}
```

*Basic Tunneled Network Traffic*

```
{
  "0": {
    "type": "ipv4-addr",
    "value": "198.51.100.2"
  },
  "1": {
    "type": "ipv4-addr",
    "value": "203.0.113.1"
  },
  "2": {
    "type": "ipv4-addr",
    "value": "203.0.113.2"
  },
  "3": {
    "type": "network-traffic",
    "src_ref": "0",
    "dst_ref": "1",
    "src_port": 2487,
    "dst_port": 1723,
    "protocols": [
      "ipv4",
      "pptp"
    ],
    "src_byte_count": 35779,
    "dst_byte_count": 935750,
    "encapsulates_refs": [
      "4"
    ]
  },
  "4": {
    "type": "network-traffic",
    "src_ref": "0",
    "dst_ref": "2",
    "src_port": 24678,
    "dst_port": 80,
    "protocols": [
      "ipv4",
      "tcp",
      "http"
```

```
    ],
    "src_packets": 14356,
    "dst_packets": 14356,
    "encapsulated_by_ref": "3"
  }
}
```

*Web traffic tunneled over DNS*

```
{
  "0": {
    "type": "ipv4-addr",
    "value": "203.0.113.1"
  },
  "1": {
    "type": "ipv4-addr",
    "value": "198.51.100.34"
  },
  "2": {
    "type": "ipv4-addr",
    "value": "198.51.100.54"
  },
  "3": {
    "type": "network-traffic",
    "src_ref": "0",
    "dst_ref": "1",
    "src_port": 2487,
    "dst_port": 53,
    "protocols": [
      "ipv4",
      "udp",
      "dns"
    ],
    "src_byte_count": 35779,
    "dst_byte_count": 935750,
    "encapsulates_refs": [
      "4"
    ]
  },
  "4": {
    "type": "network-traffic",
    "src_ref": "0",
    "dst_ref": "2",
    "src_port": 24678,
    "dst_port": 443,
    "protocols": [
      "ipv4",
      "tcp",
      "ssl",
      "http"
    ],
    "src_packets": 14356,
    "dst_packets": 14356,
    "encapsulated_by_ref": "3"
  }
}
```

### 2.12.2 HTTP Request Extension

Type Name: `http-request-ext`

The HTTP request extension specifies a default extension for capturing network traffic properties specific to HTTP requests. The key for this extension when used in the

**extensions** dictionary MUST be `http-request-ext`.

#### 2.12.2.1 Properties

| Property Name | Type | Description |
|---|---|---|
| **request_method** (required) | `string` | Specifies the HTTP method portion of the HTTP request line, as a lowercase string. |
| **request_value** (required) | `string` | Specifies the value (typically a resource path) portion of the HTTP request line. |
| **request_version** (optional) | `string` | Specifies the HTTP version portion of the HTTP request line, as a lowercase string. |
| **request_header** (optional) | `dictionary` | Specifies all of the HTTP header fields that may be found in the HTTP client request, as a dictionary.<br><br>Each key in the dictionary MUST be the name of the header field and SHOULD preserve case, e.g., `User-Agent`. The corresponding value for each dictionary key MUST be a `string`. |
| **message_body_length** (optional) | `integer` | Specifies the length of the HTTP message body, if included, in bytes. |
| **message_body_data_ref** (optional) | `object-ref` | Specifies the data contained in the HTTP message body, if included.<br><br>The object referenced in this property MUST be of type `artifact`. |

Examples

*Basic HTTP Request*

```
{
  "0": {
    "type": "ipv4-addr",
    "value": "198.51.100.53"
  },
  "1": {
    "type": "network-traffic",
    "dst_ref": "0",
    "protocols": [
      "tcp",
      "http"
    ],
    "extensions": {
      "http-request-ext": {
        "request_method": "get",
        "request_value": "/download.html",
        "request_version": "http/1.1",
        "request_header": {
          "Accept-Encoding": "gzip,deflate",
          "User-Agent":   "Mozilla/5.0   (Windows;   U;   Windows   NT   5.1;   en-US;   rv:1.6)
Gecko/20040113",
          "Host": "www.example.com"
        }
      }
    }
  }
}
```

### 2.12.3 ICMP Extension

Type Name: `icmp-ext`

The ICMP extension specifies a default extension for capturing network traffic properties specific to ICMP. The key for this extension when used in the **`extensions`** dictionary MUST be `icmp-ext`.

#### 2.12.3.1 Properties

| Property Name | Type | Description |
|---|---|---|
| **`icmp_type_hex`** (required) | `hex` | Specifies the ICMP type byte. |
| **`icmp_code_hex`** (required) | `hex` | Specifies the ICMP code byte. |

Examples

*Basic ICMP Traffic*

```
{
  "0": {
    "type": "ipv4-addr",
    "value": "198.51.100.9"
  },
  "1": {
    "type": "ipv4-addr",
    "value": "203.0.113.5"
  },
  "2": {
    "type": "network-traffic",
    "src_ref": "0",
    "dst_ref": "1",
    "protocols": [
      "icmp"
    ],
    "extensions": {
      "icmp-ext": {
        "icmp_type_hex": "08",
        "icmp_code_hex": "00"
      }
    }
  }
}
```

### 2.12.4 Network Socket Extension

Type Name: `socket-ext`

The Network Socket extension specifies a default extension for capturing network traffic properties associated with network sockets. The key for this extension when used in the extensions dictionary MUST be `socket-ext`.

#### 1.1.1.1 2.12.4.1 Properties

| Property Name | Type | Description |
|---|---|---|
| **`address_family`** (required) | `network-socket-address-family-enum` | Specifies the address family (AF_*) that the socket is configured for. |

| | | |
|---|---|---|
| **is_blocking** (optional) | `boolean` | Specifies whether the socket is in blocking mode. |
| **is_listening** (optional) | `boolean` | Specifies whether the socket is in listening mode. |
| **protocol_family** (optional) | `network-socket-protocol-family-enum` | Specifies the protocol family (PF_*) that the socket is configured for. |
| **options** (optional) | `dictionary` | Specifies any options (SO_*) that may be used by the socket, as a dictionary. Each key in the dictionary SHOULD be a case-preserved version of the option name, e.g., `SO_ACCEPTCONN`. Each key value in the dictionary MUST be the value for the corresponding options key. |
| **socket_type** (optional) | `network-socket-type-enum` | Specifies the type of the socket. |
| **socket_descriptor** (optional) | `integer` | Specifies the socket file descriptor value associated with the socket, as a non-negative integer. |
| **socket_handle** (optional) | `integer` | Specifies the handle or inode value associated with the socket. |

#### 2.12.4.2 Network Socket Address Family Enumeration

Enumeration Name: `network-socket-address-family-enum`

An enumeration of network socket address family types.

| Vocabulary Value | Description |
|---|---|
| `AF_UNSPEC` | Specifies an unspecified address family. |
| `AF_INET` | Specifies the IPv4 address family. |
| `AF_IPX` | Specifies the IPX (Novell Internet Protocol) address family. |
| `AF_APPLETALK` | Specifies the APPLETALK DDP address family. |
| `AF_NETBIOS` | Specifies the NETBIOS address family. |
| `AF_INET6` | Specifies the IPv6 address family. |
| `AF_IRDA` | Specifies IRDA sockets. |
| `AF_BTH` | Specifies BTH sockets. |

#### 2.12.4.3 Network Socket Protocol Family Enumeration

Enumeration Name: `network-socket-protocol-family-enum`

An enumeration of network socket protocol family types.

| Vocabulary Value | Description |
|---|---|

| | |
|---|---|
| PF_INET | Specifies the IP protocol family. |
| PF_AX25 | Specifies the amateur radio AX.25 family. |
| PF_IPX | Specifies the Novell Internet Protocol family. |
| PF_INET6 | Specifies the IP version 6 family. |
| PF_APPLETALK | Specifies the Appletalk DDP protocol family. |
| PF_NETROM | Specifies the Amateur radio NetROM protocol family. |
| PF_BRIDGE | Specifies the Multiprotocol bridge protocol family. |
| PF_ATMPVC | Specifies the ATM PVCs protocol family. |
| PF_X25 | Specifies the protocol family reserved for the X.25 project. |
| PF_ROSE | Specifies the PF_KEY key management API family. |
| PF_DECNET | Specifies the protocol family reserved for the DECnet project. |
| PF_NETBEUI | Specifies the protocol family reserved for the 802.2LLC project. |
| PF_SECURITY | Specifies the Security callback pseudo AF protocol family. |
| PF_KEY | Specifies the PF_KEY key management API protocol family. |
| PF_NETLINK | Specifies the netlink routing API family. |
| PF_ROUTE | Specifies the PF_ROUTE routing API family. |
| PF_PACKET | Specifies the packet family. |
| PF_ASH | Specifies the Ash family. |
| PF_ECONET | Specifies the Acorn Econet family. |
| PF_ATMSVC | Specifies the ATM SVCs protocol family. |
| PF_SNA | Specifies the Linux SNA Project protocol family. |
| PF_IRDA | Specifies IRDA sockets. |
| PF_PPPOX | Specifies PPPoX sockets. |
| PF_WANPIPE | Specifies Wanpipe API sockets. |
| PF_BLUETOOTH | Specifies Bluetooth sockets. |

#### 2.12.4.4 Network Socket Type Enumeration

Enumerations Name: `network-socket-type-enum`

An enumeration of network socket types.

| Vocabulary Value | Description |
|---|---|
| SOCK_STREAM | Specifies a pipe-like socket which operates over a connection with a particular remote socket, and transmits data reliably as a stream of bytes. |
| SOCK_DGRAM | Specifies a socket in which individually-addressed packets are sent |

| | |
|---|---|
| | (datagram). |
| SOCK_RAW | Specifies raw sockets which allow new IP protocols to be implemented in user space. A raw socket receives or sends the raw datagram not including link level headers. |
| SOCK_RDM | Specifies a socket indicating a reliably-delivered message. |
| SOCK_SEQPACKET | Specifies a datagram congestion control protocol socket. |

Examples

*Basic Stream Socket*

```
{
  "0": {
    "type": "ipv4-addr",
    "value": "198.51.100.2"
  },
  "1": {
    "type": "network-traffic",
    "src_ref": "0",
    "src_port": 223,
    "protocols": [
      "ip",
      "tcp"
    ],
    "extensions": {
      "socket-ext": {
        "is_listening": true,
        "address_family": "AF_INET",
        "protocol_family": "PF_INET",
        "socket_type": "SOCK_STREAM"
      }
    }
  }
}
```

### 2.12.5 TCP Extension

Type Name: `tcp-ext`

The TCP extension specifies a default extension for capturing network traffic properties specific to TCP. The key for this extension when used in the **extensions** dictionary MUST be `tcp-ext`. An object using the TCP Extension MUST contain at least one property from this extension.

#### 2.12.5.1 Properties

| Property Name | Type | Description |
|---|---|---|
| **src_flags_hex** (optional) | hex | Specifies the source TCP flags, as the union of all TCP flags observed between the start of the traffic (as defined by the **start** property) and the end of the traffic (as defined by the **end** property).<br><br>If the start and end times of the traffic are not specified, this property SHOULD be interpreted as the union of all TCP flags observed over the entirety of the network traffic being reported upon. |
| **dst_flags_hex** (optional) | hex | Specifies the destination TCP flags, as the union of all TCP flags observed between the start of the traffic (as defined by |

| | | |
|---|---|---|
| | | the **start** property) and the end of the traffic (as defined by the **end** property).<br><br>If the start and end times of the traffic are not specified, this property SHOULD be interpreted as the union of all TCP flags observed over the entirety of the network traffic being reported upon. |

Examples

*Basic TCP Traffic*

```
{
  "0": {
    "type": "ipv4-addr",
    "value": "198.51.100.5"
  },
  "1": {
    "type": "ipv4-addr",
    "value": "198.51.100.6"
  },
  "2": {
    "type": "network-traffic",
    "src_ref": "0",
    "dst_ref": "1",
    "src_port": 3372,
    "dst_port": 80,
    "protocols": [
      "tcp"
    ],
    "extensions": {
      "tcp-ext": {
        "src_flags_hex": "00000002"
      }
    }
  }
}
```

## 2.13 Process Object

Type Name: `process`

The Process Object represents common properties of an instance of a computer program as executed on an operating system. A Process Object MUST contain at least one property (other than **type**) from this object (or one of its extensions).

### 2.13.1 Properties

| Common Properties | | |
|---|---|---|
| **type, extensions** | | |
| Process Object Specific Properties | | |
| **is_hidden, pid, name, created, cwd, arguments, command_line, environment_variables, opened_connection_refs, creator_user_ref, binary_ref, parent_ref, child_refs** | | |
| Property Name | Type | Description |
| **type** (required) | `string` | The value of this property MUST be `process`. |
| **extensions** (optional) | `dictionary` | The Process Object defines the following |

| | | |
|---|---|---|
| | | extensions. In addition to these, producers MAY create their own.<br><br>`windows-process-ext`, `windows-service-ext`<br><br>Dictionary keys MUST identify the extension type by name.<br><br>The corresponding dictionary values MUST contain the contents of the extension instance. |
| **is_hidden** (optional) | `boolean` | Specifies whether the process is hidden. |
| **pid** (optional) | `integer` | Specifies the Process ID, or PID, of the process. |
| **name** (optional) | `string` | Specifies the name of the process. |
| **created** (optional) | `timestamp` | Specifies the date/time at which the process was created. |
| **cwd** (optional) | `string` | Specifies the current working directory of the process. |
| **arguments** (optional) | `list` of type `string` | Specifies the list of arguments used in executing the process. Each argument MUST be captured separately as a string. |
| **command_line** (optional) | `string` | Specifies the full command line used in executing the process, including the process name (depending on the operating system). |
| **environment_variables** (optional) | `dictionary` | Specifies the list of environment variables associated with the process as a dictionary. Each key in the dictionary MUST be a case preserved version of the name of the environment variable, and each corresponding value MUST be the environment variable value as a string. |
| **opened_connection_refs** (optional) | `list` of type `object-ref` | Specifies the list of network connections opened by the process, as a reference to one or more Network Traffic Objects.<br><br>The objects referenced in this list MUST be of type `network-traffic`. |
| **creator_user_ref** (optional) | `object-ref` | Specifies the user that created the process, as a reference to a User Account Object.<br><br>The object referenced in this property MUST be of type `user-account`. |
| **binary_ref** (optional) | `object-ref` | Specifies the executable binary that was executed as the process, as a reference to a File Object.<br><br>The object referenced in this property MUST be of type `file`. |

| | | |
|---|---|---|
| **parent_ref** (optional) | `object-ref` | Specifies the other process that spawned (i.e. is the parent of) this one, as reference to a Process Object.<br><br>The object referenced in this property MUST be of type `process`. |
| **child_refs** (optional) | `list` of type `object-ref` | Specifies the other processes that were spawned by (i.e. children of) this process, as a reference to one or more other Process Objects.<br><br>The objects referenced in this list MUST be of type `process`. |

Examples

*Basic Process*

```
{
  "0": {
    "type": "file",
    "hashes": {
      "SHA-256": "35a01331e9ad96f751278b891b6ea09699806faedfa237d40513d92ad1b7100f"
    },
  },
  "1": {
    "type": "process",
    "pid": 1221,
    "name": "gedit-bin",
    "created": "2016-01-20T14:11:25.55Z",
    "arguments" :[
      "--new-window"
    ],
    "binary_ref": "0"
  }
}
```

### 2.13.2 Windows™ Process Extension

Type Name: `windows-process-ext`

The Windows Process extension specifies a default extension for capturing properties specific to Windows processes. The key for this extension when used in the **extensions** dictionary MUST be `windows-process-ext`. An object using the Windows Process Extension MUST contain at least one property from this extension.

#### 2.13.2.1 Properties

| Property Name | Type | Description |
|---|---|---|
| **aslr_enabled** (optional) | `boolean` | Specifies whether Address Space Layout Randomization (ASLR) is enabled for the process. |
| **dep_enabled** (optional) | `boolean` | Specifies whether Data Execution Prevention (DEP) is enabled for the process. |
| **priority** (optional) | `string` | Specifies the current priority class of the process in Windows. This value SHOULD be a string that ends in `_CLASS`. |
| **owner_sid** (optional) | `string` | Specifies the Security ID (SID) value of the owner |

| | | |
|---|---|---|
| | | of the process. |
| **window_title** (optional) | string | Specifies the title of the main window of the process. |
| **startup_info** (optional) | dictionary | Specifies the STARTUP_INFO struct used by the process, as a dictionary. Each name/value pair in the struct MUST be represented as a key/value pair in the dictionary, where each key MUST be a case-preserved version of the original name. For example, given a name of "lpDesktop" the corresponding key would be `lpDesktop`. |

Examples

*Basic Windows Process*

```
{
  "0": {
    "type": "process",
    "pid": 314,
    "name": "foobar.exe",
    "extensions": {
      "windows-process-ext": {
        "aslr_enabled": true,
        "dep_enabled": true,
        "priority": "HIGH_PRIORITY_CLASS",
        "owner_sid": "S-1-5-21-186985262-1144665072-74031268-1309"
      }
    }
  }
}
```

### 2.13.3 Windows™ Service Extension

Type Name: `windows-service-ext`

The Windows Service extension specifies a default extension for capturing properties specific to Windows services. The key for this extension when used in the **extensions** dictionary MUST be `windows-service-ext`.

#### 2.13.3.1 Properties

| Property Name | Type | Description |
|---|---|---|
| **service_name** (required) | string | Specifies the name of the service. |
| **descriptions** (optional) | list of type string | Specifies the descriptions defined for the service. |
| **display_name** (optional) | string | Specifies the displayed name of the service in Windows GUI controls. |
| **group_name** (optional) | string | Specifies the name of the load ordering group of which the service is a member. |
| **start_type** (optional) | windows-service-start-type-enum | Specifies the start options defined for the service. |
| **service_dll_refs** (optional) | list of type object-ref | Specifies the DLLs loaded by the service, as a reference to one or more File Objects. |

| | | |
|---|---|---|
| | | The objects referenced in this property MUST be of type `file`. |
| **`service_type`** (optional) | `windows-service-type-enum` | Specifies the type of the service. |
| **`service_status`** (optional) | `windows-service-status-enum` | Specifies the current status of the service. |

#### 2.13.3.2 Windows™ Service Start Type Enumeration

Enumeration Name: `windows-service-start-type-enum`

An enumeration of Windows service start types.

| Vocabulary Value | Description |
|---|---|
| `SERVICE_AUTO_START` | A service started automatically by the service control manager during system startup. |
| `SERVICE_BOOT_START` | A device driver started by the system loader. This value is valid only for driver services. |
| `SERVICE_DEMAND_START` | A service started by the service control manager when a process calls the StartService function. |
| `SERVICE_DISABLED` | A service that cannot be started. Attempts to start the service result in the error code ERROR_SERVICE_DISABLED. |
| `SERVICE_SYSTEM_ALERT` | A device driver started by the IoInitSystem function. This value is valid only for driver services. |

#### 2.13.3.3 Windows™ Service Type Enumeration

Enumeration Name: `windows-service-type-enum`

An enumeration of Windows service types.

| Vocabulary Value | Description |
|---|---|
| `SERVICE_KERNEL_DRIVER` | The service is a device driver. |
| `SERVICE_FILE_SYSTEM_DRIVER` | The service is a file system driver. |
| `SERVICE_WIN32_OWN_PROCESS` | The service runs in its own process. |
| `SERVICE_WIN32_SHARE_PROCESS` | The service shares a process with other services. |

#### 2.13.3.4 Windows™ Service Status Enumeration

Enumeration Name: `windows-service-status-enum`

An enumeration of Windows service statuses.

| Value | Description |
|---|---|
| SERVICE_CONTINUE_PENDING | The service continue is pending. |
| SERVICE_PAUSE_PENDING | The service pause is pending. |
| SERVICE_PAUSED | The service is paused. |
| SERVICE_RUNNING | The service is running. |
| SERVICE_START_PENDING | The service is starting. |
| SERVICE_STOP_PENDING | The service is stopping. |
| SERVICE_STOPPED | The service is not running. |

Examples

*Basic Windows Service*

```
{
  "0": {
    "type": "file",
    "hashes": {
      "SHA-256": "35a01331e9ad96f751278b891b6ea09699806faedfa237d40513d92ad1b7100f"
    },
    "name": "sirvizio.exe"
  },
  "1":{
    "type": "process",
    "pid": 2217,
    "name": "sirvizio",
    "command_line": "C:\\Windows\\System32\\sirvizio.exe /s",
    "binary_ref": "0",
    "extensions": {
      "windows-service-ext": {
        "service_name": "sirvizio",
        "display_name": "Sirvizio",
        "start_type": "SERVICE_AUTO_START",
        "service_type": "SERVICE_WIN32_OWN_PROCESS",
        "service_status": "SERVICE_RUNNING"
      }
    }
  }
}
```

## 2.14 Software Object

Type Name: `software`

The Software Object represents high-level properties associated with software, including software products.

### 2.14.1 Properties

| Common Properties | | |
|---|---|---|
| **type, extensions** | | |
| Software Object Specific Properties | | |
| **name, cpe, languages, vendor, version** | | |
| Property Name | Type | Description |

| | | |
|---|---|---|
| **type** (required) | `string` | The value of this property MUST be `software`. |
| **name** (required) | `string` | Specifies the name of the software. |
| **cpe** (optional) | `string` | Specifies the Common Platform Enumeration (CPE) entry for the software, if available. The value for this property MUST be a CPE v2.3 entry from the official NVD CPE Dictionary [NVD] .<br><br>While the CPE dictionary does not contain entries for *all* software, whenever it *does* contain an identifier for a given instance of software, this property SHOULD be present. |
| **languages** (optional) | `list` of type `string` | Specifies the languages supported by the software. The value of each list member MUST be an ISO 639-2 language code [ISO639-2] . |
| **vendor** (optional) | `string` | Specifies the name of the vendor of the software. |
| **version** (optional) | `string` | Specifies the version of the software. |

Examples

*Typical Software Instance*

```
{
  "0": {
    "type": "software",
    "name": "Word",
    "cpe": "cpe:2.3:a:microsoft:word:2000:*:*:*:*:*:*:*",
    "version": "2002",
    "vendor": "Microsoft"
  }
}
```

## 2.15 URL Object

Type Name: `url`

The URL Object represents the properties of a uniform resource locator (URL).

### 2.15.1 Properties

| Common Properties | | |
|---|---|---|
| **type, extensions** | | |
| URL Object Specific Properties | | |
| **value** | | |
| Property Name | Type | Description |
| **type** (required) | `string` | The value of this property MUST be `url`. |
| **value** (required) | `string` | Specifies the value of the URL. The value of this property MUST conform to [RFC3986], more specifically section 1.1.3 with reference to the definition for "Uniform Resource Locator". |

Examples

*Typical URL*

```
{
  "0": {
    "type": "url",
    "value": "https://example.com/research/index.html"
  }
}
```

## 2.16 User Account Object

Type Name: `user-account`

The User Account Object represents an instance of any type of user account, including but not limited to operating system, device, messaging service, and social media platform accounts.

### 2.16.1 Properties

| Common Properties | | |
|---|---|---|
| **type, extensions** | | |
| User Account Object Specific Properties | | |
| **user_id, account_login, account_type, display_name, is_service_account, is_privileged, can_escalate_privs, is_disabled, account_created, account_expires, password_last_changed, account_first_login, account_last_login** | | |
| Property Name | Type | Description |
| **type** (required) | `string` | The value of this property MUST be `user-account`. |
| **extensions** (optional) | `dictionary` | The User Account Object defines the following extensions. In addition to these, producers MAY create their own.<br><br>`unix-account-ext`<br><br>Dictionary keys MUST identify the extension type by name.<br><br>The corresponding dictionary values MUST contain the contents of the extension instance. |
| **user_id** (required) | `string` | Specifies the identifier of the account. The format of the identifier depends on the system the user account is maintained in, and may be a numeric ID, a GUID, an account name, an email address, etc. The **user_id** property should be populated with whatever field is the unique identifier for the system the account is a member of. For example, for UNIX systems it would be populated with the UID. |
| **account_login** (optional) | `string` | Specifies the account login string, used in cases where the **user_id** property specifies something other than what a user would type when they login. |

| | | |
|---|---|---|
| | | For example, in the case of a Unix account with user_id 0, the account_login might be “root”. |
| **account_type** (optional) | `open-vocab` | Specifies the type of the account.<br><br>This is an open vocabulary and values SHOULD come from the `account-type-ov` vocabulary. |
| **display_name** (optional) | `string` | Specifies the display name of the account, to be shown in user interfaces, if applicable.<br><br>On Unix, this is equivalent to the GECOS field. |
| **is_service_account** (optional) | `boolean` | Indicates that the account is associated with a network service or system process (daemon), not a specific individual. |
| **is_privileged** (optional) | `boolean` | Specifies that the account has elevated privileges (i.e., in the case of root on Unix or the Windows Administrator account). |
| **can_escalate_privs** (optional) | `boolean` | Specifies that the account has the ability to escalate privileges (i.e., in the case of sudo on Unix or a Windows Domain Admin account) |
| **is_disabled** (optional) | `boolean` | Specifies if the account is disabled. |
| **account_created** (optional) | `timestamp` | Specifies when the account was created. |
| **account_expires** (optional) | `timestamp` | Specifies the expiration date of the account. |
| **password_last_changed** (optional) | `timestamp` | Specifies when the account password was last changed. |
| **account_first_login** (optional) | `timestamp` | Specifies when the account was first accessed. |
| **account_last_login** (optional) | `timestamp` | Specifies when the account was last accessed. |

### 2.16.2 Account Type Vocabulary

Vocabulary Name: `account-type-ov`

An open vocabulary of User Account types.

| Vocabulary Value | Description |
|---|---|
| `unix` | Specifies a POSIX account. |
| `windows-local` | Specifies a Windows local account. |
| `windows-domain` | Specifies a Windows domain account. |
| `ldap` | Specifies an LDAP account. |
| `tacacs` | Specifies a TACACS account. |
| `radius` | Specifies a RADIUS account. |

| | |
|---|---|
| `nis` | Specifies a NIS account |
| `openid` | Specifies an OpenID account. |
| `facebook` | Specifies a Facebook account. |
| `skype` | Specifies a Skype account. |
| `twitter` | Specifies a Twitter account. |

Examples

*Basic Unix Account*

```
{
  "0": {
    "type": "user-account",
    "user_id": "1001",
    "account_login": "jdoe",
    "account_type": "unix",
    "display_name": "John Doe",
    "is_service_account": false,
    "is_privileged": false,
    "can_escalate_privs": true,
    "account_created": "2016-01-20T12:31:12Z",
    "password_last_changed": "2016-01-20T14:27:43Z",
    "account_first_login": "2016-01-20T14:26:07Z",
    "account_last_login": "2016-07-22T16:08:28Z"
  }
}
```

*Basic Twitter Account*

```
{
  "0": {
    "type": "user-account",
    "user_id": "thegrugq_ebooks",
    "account_login": "thegrugq_ebooks",
    "account_type": "twitter",
    "display_name": "the grugq"
  }
}
```

### 2.16.3 UNIX™ Account Extension

Type Name: `unix-account-ext`

The UNIX account extension specifies a default extension for capturing the additional information for an account on a UNIX system. The key for this extension when used in the **extensions** dictionary MUST be `unix-account-ext`. An object using the UNIX Account Extension MUST contain at least one property from this extension.

#### 2.16.3.1 Properties

| Property Name | Type | Description |
|---|---|---|
| **gid** (optional) | `integer` | Specifies the primary group ID of the account. |
| **groups** (optional) | `list` of type `string` | Specifies a list of names of groups that the account is a member of. |
| **h o m e _ d i r** | `string` | Specifies the home directory of the account. |

| | | |
|---|---|---|
| (optional) | | |
| **shell** (optional) | `string` | Specifies the account's command shell. |

Examples

*Basic UNIX Account*

```
{
  "0": {
    "type": "user-account",
    "user_id": "1001",
    "account_login": "jdoe",
    "account_type": "unix",
    "display_name": "John Doe",
    "is_service_account": false,
    "is_privileged": false,
    "can_escalate_privs": true,
    "extensions": {
      "unix-account-ext": {
        "gid": 1001,
        "groups": ["wheel"],
        "home_dir": "/home/jdoe",
        "shell": "/bin/bash"
      }
    }
  }
}
```

## 2.17 Windows™ Registry Key Object

Type Name: `windows-registry-key`

The Registry Key Object represents the properties of a Windows registry key.

### 2.17.1 Properties

| Common Properties | | |
|---|---|---|
| **type, extensions** | | |
| File Object Specific Properties | | |
| **key, values, modified, creator_user_ref, number_of_subkeys** | | |
| Property Name | Type | Description |
| **type** (required) | `string` | The value of this property MUST be `windows-registry-key`. |
| **key** (required) | `string` | Specifies the full registry key including the hive.<br><br>The value of the key, including the hive portion, SHOULD be case-preserved. The hive portion of the key MUST be fully expanded and not truncated; e.g., HKEY_LOCAL_MACHINE must be used instead of HKLM. |
| **values** (optional) | `list` of type `windows-registry-value-type` | Specifies the values found under the registry key. |
| **modified** (optional) | `timestamp` | Specifies the last date/time that the registry key |

| | | |
|---|---|---|
| | | was modified. |
| creator_user_ref (optional) | object-ref | Specifies a reference to the user account (represented as a User Account Object) that created the registry key.<br><br>The object referenced in this property MUST be of type user-account. |
| number_of_subkeys (optional) | integer | Specifies the number of subkeys contained under the registry key. |

### 2.17.2 Windows™ Registry Value Type

Type Name: windows-registry-value-type

The Windows Registry Value type captures the properties of a Windows Registry Key Value.

#### 2.17.2.1 Properties

| Property Name | Type | Description |
|---|---|---|
| **name** (required) | string | Specifies the name of the registry value. For specifying the default value in a registry key, an empty string MUST be used. |
| **data** (optional) | string | Specifies the data contained in the registry value. |
| **data_type** (optional) | windows-registry-datatype-enum | Specifies the registry (REG_*) data type used in the registry value. |

### 2.17.3 Windows™ Registry Datatype Enumeration

Enumeration Name: windows-registry-datatype-enum

An enumeration of Windows registry data types.

| Vocabulary Value | Description |
|---|---|
| REG_NONE | No defined value type. |
| REG_SZ | A null-terminated string. This will be either a Unicode or an ANSI string, depending on whether you use the Unicode or ANSI functions. |
| REG_EXPAND_SZ | A null-terminated string that contains unexpanded references to environment variables (for example, "%PATH%"). It will be a Unicode or ANSI string depending on whether you use the Unicode or ANSI functions. |
| REG_BINARY | Binary data in any form. |
| REG_DWORD | A 32-bit number. |

| | |
|---|---|
| REG_DWORD_BIG_ENDIAN | A 32-bit number in big-endian format. |
| REG_LINK | A null-terminated Unicode string that contains the target path of a symbolic link. |
| REG_MULTI_SZ | A sequence of null-terminated strings, terminated by an empty string (₩0). |
| REG_RESOURCE_LIST | A series of nested lists designed to store a resource list used by a hardware device driver or one of the physical devices it controls. This data is detected and written into the ResourceMap tree by the system and is displayed in Registry Editor in hexadecimal format as a Binary Value. |
| REG_FULL_RESOURCE_DESCRIPTION | A series of nested lists designed to store a resource list used by a physical hardware device. This data is detected and written into the HardwareDescription tree by the system and is displayed in Registry Editor in hexadecimal format as a Binary Value. |
| REG_RESOURCE_REQUIREMENTS_LIST | Device driver list of hardware resource requirements in Resource Map tree. |
| REG_QWORD | A 64-bit number. |
| REG_INVALID_TYPE | Specifies an invalid key. |

Examples

*Simple registry key*

```
{
  "0": {
    "type": "windows-registry-key",
    "key": "HKEY_LOCAL_MACHINE\\System\\Foo\\Bar"
    }
}
```

*Registry key with values*

```
{
  "0": {
    "type": "windows-registry-key",
    "key": "hkey_local_machine\\system\\bar\\foo",
    "values": [
      {
        "name": "Foo",
        "data": "qwerty",
        "data_type": "REG_SZ"
      },
      {
        "name": "Bar",
        "data": "42",
        "data_type": "REG_DWORD"
      }
    ]
  }
}
```

## 2.18 X.509 Certificate Object

Type Name: `x509-certificate`

The X.509 Certificate Object represents the properties of an X.509 certificate, as defined by ITU recommendation X.509 [X.509]. An X.509 Certificate Object MUST contain at least one property (other than **type**) from this object.

### 2.18.1 Properties

| Common Properties | | |
|---|---|---|
| **type, extensions** | | |
| File Object Specific Properties | | |
| **is_self_signed, hashes, version, serial_number, signature_algorithm, issuer, validity_not_before, validity_not_after, subject, subject_public_key_algorithm, subject_public_key_modulus, subject_public_key_exponent, x509_v3_extensions** | | |
| Property Name | Type | Description |
| **type** (required) | `string` | The value of this property MUST be `x509-certificate`. |
| **is_self_signed** (optional) | `boolean` | Specifies whether the certificate is self-signed, i.e., whether it is signed by the same entity whose identity it certifies. |
| **hashes** (optional) | `hashes` | Specifies any hashes that were calculated for the entire contents of the certificate. |
| **version** (optional) | `string` | Specifies the version of the encoded certificate. |
| **serial_number** (optional) | `string` | Specifies the unique identifier for the certificate, as issued by a specific Certificate Authority. |
| **signature_algorithm** (optional) | `string` | Specifies the name of the algorithm used to sign the certificate. |
| **issuer** (optional) | `string` | Specifies the name of the Certificate Authority that issued the certificate. |
| **validity_not_before** (optional) | `timestamp` | Specifies the date on which the certificate validity period begins. |
| **validity_not_after** (optional) | `timestamp` | Specifies the date on which the certificate validity period ends. |
| **subject** (optional) | `string` | Specifies the name of the entity associated with the public key stored in the subject public key field of the certificate. |
| **subject_public_key_algorithm** (optional) | `string` | Specifies the name of the algorithm with which to encrypt data being sent to the subject. |
| **subject_public_key_modulus** (optional) | `string` | Specifies the modulus portion of the subject's public RSA key. |
| **subject_public_key_exponent** | `integer` | Specifies the exponent portion of the |

| | | |
|---|---|---|
| (optional) | | subject's public RSA key, as an integer. |
| **x509_v3_extensions** (optional) | `x509-v3-extensions-type` | Specifies any standard X.509 v3 extensions that may be used in the certificate. |

### 2.18.2 X.509 v3 Extensions Type

Type Name: `x509-v3-extensions-type`

The X.509 v3 Extensions type captures properties associated with X.509 v3 extensions, which serve as a mechanism for specifying additional information such as alternative subject names. An object using the X.509 v3 Extensions type MUST contain at least one property from this type.

Note that the X.509 v3 Extensions type is not a STIX Cyber Observables extension, it is a type that describes X.509 extensions.

#### 2.18.2.1 Properties

| Property Name | Type | Description |
|---|---|---|
| **basic_constraints** (optional) | `string` | Specifies a multi-valued extension which indicates whether a certificate is a CA certificate. The first (mandatory) name is CA followed by *TRUE* or *FALSE*. If CA is *TRUE* then an optional pathlen name followed by an non-negative value can be included. Also equivalent to the object ID (OID) value of 2.5.29.19. |
| **name_constraints** (optional) | `string` | Specifies a namespace within which all subject names in subsequent certificates in a certification path MUST be located. Also equivalent to the object ID (OID) value of 2.5.29.30. |
| **policy_constraints** (optional) | `string` | Specifies any constraints on path validation for certificates issued to CAs. Also equivalent to the object ID (OID) value of 2.5.29.36. |
| **key_usage** (optional) | `string` | Specifies a multi-valued extension consisting of a list of names of the permitted key usages. Also equivalent to the object ID (OID) value of 2.5.29.15. |
| **extended_key_usage** (optional) | `string` | Specifies a list of usages indicating purposes for which the certificate public key can be used for. Also equivalent to the object ID (OID) value of 2.5.29.37. |
| **subject_key_identifier** (optional) | `string` | Specifies the identifier that provides a means of identifying certificates that contain a particular public key. Also equivalent to the object ID (OID) value of 2.5.29.14. |

| | | |
|---|---|---|
| **authority_key_identifier** (optional) | string | Specifies the identifier that provides a means of identifying the public key corresponding to the private key used to sign a certificate. Also equivalent to the object ID (OID) value of 2.5.29.35. |
| **subject_alternative_name** (optional) | string | Specifies the additional identities to be bound to the subject of the certificate. Also equivalent to the object ID (OID) value of 2.5.29.17. |
| **issuer_alternative_name** (optional) | string | Specifies the additional identities to be bound to the issuer of the certificate. Also equivalent to the object ID (OID) value of 2.5.29.18. |
| **subject_directory_attributes** (optional) | string | Specifies the identification attributes (e.g., nationality) of the subject. Also equivalent to the object ID (OID) value of 2.5.29.9. |
| **crl_distribution_points** (optional) | string | Specifies how CRL information is obtained. Also equivalent to the object ID (OID) value of 2.5.29.31. |
| **inhibit_any_policy** (optional) | string | Specifies the number of additional certificates that may appear in the path before anyPolicy is no longer permitted. Also equivalent to the object ID (OID) value of 2.5.29.54. |
| **private_key_usage_period_not_before** (optional) | timestamp | Specifies the date on which the validity period begins for the private key, if it is different from the validity period of the certificate. |
| **private_key_usage_period_not_after** (optional) | timestamp | Specifies the date on which the validity period ends for the private key, if it is different from the validity period of the certificate. |
| **certificate_policies** (optional) | string | Specifies a sequence of one or more policy information terms, each of which consists of an object identifier (OID) and optional qualifiers. Also equivalent to the object ID (OID) value of 2.5.29.32. |
| **policy_mappings** (optional) | string | Specifies one or more pairs of OIDs; each pair includes an issuerDomainPolicy and a subjectDomainPolicy. The pairing indicates whether the issuing CA considers its issuerDomainPolicy equivalent to the subject CA's subjectDomainPolicy. Also equivalent to the object ID (OID) value of 2.5.29.33. |

Examples

*Basic X.509 certificate*

```
{
  "0": {
    "type": "x509-certificate",
    "issuer": "C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc, OU=Certification
```

```
Services Division, CN=Thawte Server CA/emailAddress=server-certs@thawte.com",
    "validity_not_before": "2016-03-12T12:00:00Z",
    "validity_not_after": "2016-08-21T12:00:00Z",
    "subject":    "C=US,    ST=Maryland,    L=Pasadena,    O=Brent    Baccala,    OU=FreeSoft,
CN=www.freesoft.org/emailAddress=baccala@freesoft.org"
    }
}
```

# 3 Conformance

## 3.1 Defined Object Producers

A "Defined Object Producer" that creates an Object from section 2 (Defined Object Data Models) is a “Producer” of that Object. Defined Object Producers MUST conform to all normative requirements in the section for that Object along with all of the general requirements pertaining to Objects as defined in section 3 of *STIX™ Version 2.0. Part 3: Cyber Observable Core Concepts*.

*For example, a "Defined Object Producer" that can produce File Object is a "File Object Producer". That producer has to conform to all normative requirements in Cyber Observable Objects section 2.7, File Object.*

## 3.2 Defined Object Consumers

A "Defined Object Consumer" that receives an Object from section 2 (Defined Object Data Models) is a "Consumer" of that Object. Defined Object Consumers MUST conform to all normative requirements in the section for that Object along with all of the general requirements pertaining to Objects as defined in section 3 of *STIX™ Version 2.0. Part 3: Cyber Observable Core Concepts*.

*For example, an "Object Consumer" that can receive Network Traffic Objects is a "Network Traffic Object Consumer". That consumer has to conform to all normative requirements in Cyber Observable Objects Section 2.12, Network Traffic Object.*

# 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

# 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
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
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.
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
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.
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

# Appendix C. Revision History

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

# 부 록 Ⅰ-1

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

# 지식재산권 확약서 정보

## Ⅰ-1.1 지식재산권 확약서(1)

- 해당 사항 없음

## Ⅰ-1.2 지식재산권 확약서(2)

- 해당 사항 없음

※ 상기 기재된 지식재산권 확약서 이외에도 본 표준이 발간된 후 접수된 확약서가 있을 수 있으니, TTA 웹사이트에서 확인하시기 바랍니다.

# 부 록 Ⅰ-2

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

# 시험인증 관련 사항

## Ⅰ-2.1 시험인증 대상 여부

- 해당 사항 없음

## Ⅰ-2.2 시험표준 제정 현황

- 해당 사항 없음

# 부 록 Ⅰ-3

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

# 본 표준의 연계(family) 표준

**Ⅰ-3.1 TTA, 구조화된 위협 정보 표현 규격(STIX™) 버전 2.0 - 제1부: STIX 핵심 개념**

STIX의 핵심 개념을 정의하는 문서로 공통 데이터 형식, STIX 객체, 데이터 표시 등에 대한 설명을 제공

**Ⅰ-3.2 TTA, 구조화된 위협 정보 표현 규격(STIX™) 버전 2.0 - 제2부: STIX 객체**

STIX의 도메인 Objects 집합을 정의하는 문서로 Objects 의 구성요소와 구성요소에 대한 설명을 제공

**Ⅰ-3.3 TTA, 구조화된 위협 정보 표현 규격(STIX™) 버전 2.0 - 제3부: STIX 사이버 관측 코어 개념**

STIX의 사이버 관측 코어 개념을 정의하는 문서로 핵심 개념을 구성하는 필드와 필드에 대한 설명을 제공

**Ⅰ-3.4 TTA, 구조화된 위협 정보 표현 규격(STIX™) 버전 2.0 - 제5부: STIX 패터닝**

STIX의 Indicator 지원 패턴을 정의하는 문서로 Indicator 지원 패턴을 구성하는 필드와 필드에 대한 설명을 제공

# 부 록 Ⅰ-4

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

## 참고 문헌


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

[2] IANA, “IP Flow Information Export (IPFIX) Entities”, December 2016, [Online]. Available: http://www.iana.org/assignments/ipfix/ipfix.xhtml

[3] “ISO 639-2:1998 Codes for the representation of names of languages -- Part 2: Alpha-3 code”, 1998. [Online]. Available: http://www.iso.org/iso/catalogue_detail? csnumber=4767

[4] N. Freed, M. Kucherawy, M. Baker and B. Hoehrmann, “Media Types”, IANA, December 2016. [Online]. Available: http://www.iana.org/assignments/media-types/media-types.xhtml

[5] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, http://www.rfc-editor.org/info/rfc1034.

[6] Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, DOI 10.17487/RFC2047, November 1996, http://www.rfc-editor.org/info/rfc2047.

[7] 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.

[8] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, http://www.rfc-editor.org/info/rfc3986.

[9] Resnick, P., Ed., "Internet Message Format", RFC 5322, DOI 10.17487/RFC5322, October 2008, http://www.rfc-editor.org/info/rfc5322.

[10] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, DOI 10.17487/RFC5890, August 2010, http://www.rfc-editor.org/info/rfc5890.

[11] J.Touch, A. Mankin, E. Kohler, et. al., “Service Name and Transport Protocol Port Number Registry”, IANA, January 2017. [Online]. Available: http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

[12] Official Common Platform Enumeration (CPE) Dictionary, National Vulnerability Database [Online]. Available: https://nvd.nist.gov/cpe.cfm

[13] X.509 : Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks, ITU, October 2016. [Online]. Available: https://www.itu.int/rec/T-REC-X.509/

# 부 록 Ⅰ-5

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

# 영문표준 해설서

## Ⅰ-5.1 개요

STIX는 공동 위협 분석, 위협 공유 자동화, 탐지 및 대응 자동화와 같은 다양한 기능을 제공하고 개선하도록 설계되었다.

STIX 사이버 관측은 특성화된 데이터에 대한 추가 컨텍스트를 제공하기 위하여 다양한 STIX 도메인 객체 (SDO) 에서 사용된다. 예를 들어, 관측된 데이터 SDO(Observed Data SDO)는 원시 데이터가 특정 시간 및 특정 객체에서 관측되었음을 나타낸다.

STIX 2.0에 포함시키기 위해 선택된 사이버 관측 객체는 기본적인 소비자 및 생산자 요구사항을 충족시키는 최소 실행 가능 제품 (MVP)을 나타낸다. STIX 2.0에는 포함되어 있지 않지만 커뮤니티에서 필요로하는 객체 및 속성이 향후 릴리스에 포함될 예정이다.

이 표준은 사이버 관측 객체의 정의를 서술한다.

## Ⅰ-5.2. 사용자 정의 객체 데이터 모델

### Ⅰ-5.2.1 아티팩트 객체 (Artifact Object)

아티팩트 객체를 통해 base64 인코딩 문자열인 바이트(8비트) 배열을 포착하거나 파일 같은 페이로드에 연결할 수 있다. payload_bin 속성에 포착된 base64 인코딩 데이터의 크기는 10MB 이하이어야 한다.

payload_bin 또는 url 중 하나를 제공해야 한다. 이는 URL을 다운스트림 소비자가 액세스할 수 있도록 하기 위해 객체 작성자에 대한 의무이다. URL을 제공하는 경우 hashes 속성은 URL 콘텐츠의 해시를 포함해야 한다.

<표 Ⅰ-5.2-1> 아티팩트 객체 공통 속성

| 공통 속성 | | |
|---|---|---|
| type, extensions | | |
| 아티팩트 객체 고유 속성 | | |
| mime_type, payload_bin, url, hashes | | |
| 속성 이름 | 형식 | 설명 |
| type(필수) | string | 이 속성의 값은 artifact이어야 한다. |
| mime_type(선택 사항) | string | 이 속성의 값은 IANA 미디어 형식[Media Types] 레지스트리에 지정된 유효한 MIME 형식이어야 한다. |

| | | |
|---|---|---|
| payload_bin(선택 사항) | binary | base64 인코딩 문자열인 아티팩트에 포함된 이진 데이터를 지정한다. URL이 제공되는 경우 이 속성이 없어야 한다. |
| url(선택 사항) | string | 이 속성의 값은 인코딩되지 않은 콘텐츠로 확인되는 유효한 URL이어야 한다. payload_bin이 제공되는 경우 이 속성이 없어야 한다. |
| hashes(선택 사항) | hashes | url 또는 payload_bin의 콘텐츠에 대한 해시의 속성을 지정한다. 이 속성은 url 속성이 존재할 경우 반드시 제공해야 한다. |

예제

기본 이미지 아티팩트

```
{
  "0": {
    "type": "artifact",
    "mime_type": "image/jpeg",
    "payload_bin": "VBORw0KGgoAAAANSUhEUgAAADI== ..."
    }
}
```

### I-5.2.2 AS 객체(Autonomous System Object)

AS 객체는 AS(자율 시스템, Autonomous System)의 속성을 표시한다.

<표 I-5.2-2> AS 객체 공통 속성

| 공통 속성 | | |
|---|---|---|
| type, extensions | | |
| AS 객체 고유 속성 | | |
| number, name, rir | | |
| 속성 이름 | 형식 | 설명 |
| type(필수) | string | 이 속성의 값은 autonomous-system이어야 한다. |
| number(필수) | integer | AS에 할당된 번호를 지정한다. 그러한 할당은 일반적으로 RIR(Regional Internet Registry)에 의해 수행된다. |
| name(선택 사항) | string | AS의 이름을 지정한다. |
| rir(선택 사항) | string | AS에 번호를 할당한 RIR의 이름을 지정한다. |

예제

기본 AS 객체

```
{
  "0": {
    "type": "autonomous-system",
    "number": 15139,
    "name": "Slime Industries",
    "rir": "ARIN"
  }
}
```

### I-5.2.3 디렉토리 객체(Directory Object)

디렉터리 객체는 파일 시스템 디렉터리에 공통인 속성을 표시한다.

<표 I-5.2-3> 디렉토리 객체 공통 속성

| 공통 속성 | | |
|---|---|---|
| type, extensions | | |
| 파일 객체 고유 속성 | | |
| path, path_enc, created, modified, accessed, contains_refs | | |
| 속성 이름 | 형식 | 설명 |
| type(필수) | string | 이 속성의 값은 directory이어야 한다. |
| path(필수) | string | 원래 관측한 경로를 파일 시스템의 디렉터리로 지정한다. |
| path_enc(선택 사항) | string | 경로에 대해 관측된 인코딩을 지정한다. 경로가 비 유니코드 인코딩에 저장된 경우 값을 지정해야 한다. 이 값은 IANA 문자 집합 레지스트리의 2013년 12월 20일 개정판에서 해당 이름을 사용하여 지정해야 한다. 문자 집합에 대해 선호하는 MIME 이름이 정의된 경우 이 값을 사용해야 하며, 정의되지 않은 경우 레지스트리의 이름 값을 대신 사용해야 한다. |
| created(선택 사항) | timestamp | 디렉터리가 생성된 날짜/시간을 지정한다. |
| modified(선택 사항) | timestamp | 디렉터리에 마지막 수정한 날짜/시간을 지정한다. |
| accessed(선택 사항) | timestamp | 디렉터리를 마지막 액세스한 날짜/시간을 지정한다. |
| contains_refs(선택 사항) | object-ref 형식의 list | 디렉터리에 포함된 다른 파일 및/또는 디렉터리 객체에 대한 참조의 목록을 지정한다.<br><br>이 목록에 참조된 객체는 file 또는 directory 형식이어야 한다. |

예제

기본 디렉터리

```
{
  "0": {
    "type": "directory",
    "path": "C:₩₩Windows₩₩System32"
    }
}
```

### I-5.2.4 도메인 이름 객체(Domain Name Object)

도메인 이름은 네트워크 도메인 이름의 속성을 표시한다.

<표 I-5.2-4> 도메인 이름 객체 공통 속성

| 공통 속성 | | |
|---|---|---|
| type, extensions | | |
| 도메인 이름 객체 고유 속성 | | |
| value, resolves_to_refs | | |
| 속성 이름 | 형식 | 설명 |
| type(필수) | string | 이 속성의 값은 domain-name이어야 한다. |
| value(필수) | string | 도메인 이름의 값을 지정한다. 이 속성의 값은 [RFC1034]를 준수해야 하며 도메인 이름에 포함된 각 도메인과 하위 도메인은 [RFC5890]을 준수해야 한다. |
| resolves_to_refs (선택 사항) | object-ref 형식의 list | 도메인 이름이 확인되는 하나 이상의 IP 주소 또는 도메인 이름에 대한 참조의 목록을 지정한다.<br><br>이 목록에 참조된 객체는 ipv4-addr 또는 ipv6-addr 또는 domain-name(CNAME 레코드 같은 경우) 형식이어야 한다. |

예제

기본 FQDN

```
{
  "0": {
    "type": "domain-name",
    "value": "example.com",
    "resolves_to_refs": [
      "1"
    ]
  },
  "1": {
    "type": "ipv4-addr",
    "value": "198.51.100.2"
  }
}
```

### I-5.2.5 이메일 주소 객체(Email Address Object)

이메일 주소 객체는 단일 이메일 주소를 표시한다.

<표 I-5.2-5> 이메일 주소 객체 공통 속성

| 공통 속성 | | |
|---|---|---|
| type, extensions | | |
| 이메일 주소 객체 고유 속성 | | |
| value, display_name, belongs_to_ref | | |
| 속성 이름 | 형식 | 설명 |
| type(필수) | string | 이 속성의 값은 email-addr이어야 한다. |
| value(필수) | string | 단일 이메일 주소를 지정한다. 표시 이름을 포함해야 한다.<br><br>이 속성은 [RFC5322] 섹션 3.4의 addr-spec 구성에 해당한다(예: jane.smith@example.com). |
| display_name (선택 사항) | string | 단일 이메일 표시 이름, 즉 메일 응용 프로그램의 사람 사용자에게 표시되는 이름을 지정한다.<br><br>이 속성은 [RFC5322] 섹션 3.4의 display-name 구성에 해당한다(예: Jane Smith). |
| belongs_to_ref (선택 사항) | object-ref | 이메일 주소가 속한 사용자 계정을 사용자 계정 객체에 대한 참조로 지정한다.<br><br>이 속성에 참조된 객체는 user-account 형식이어야 한다. |

예제

기본 이메일 주소

```
{
  "0": {
    "type": "email-addr",
    "value": "john@example.com",
    "display_name": "John Doe"
  }
}
```

### I-5.2.6 이메일 메시지 객체(Email Message Object)

이메일 메시지 객체는 [RFC5322] 및 관련 RFC에 기술된 인터넷 메시지 형식에 해당하는 이메일 메시지의 인스턴스를 표시한다.
[RFC2047]의 섹션 2에 기술된 대로 인코딩한 헤더 필드 값은 이메일 메시지 객체 속성에 포함하기 전에 디코딩해야 한다. 예를 들어 =?iso-8859-1?q?this=20is=20some=20text?=

대신 this is some text를 사용해야 한다. 유니코드로 디코딩할 수 없는 인코딩된 값의 문자는 대체 문자(U+FFFD)로 바꾸어야 한다. 헤더 값을 관측한 대로 포착해야 하는 경우, raw_email_ref 속성을 통해 아티팩트 객체를 참조하여 이 목적을 달성할 수 있다.

<표 I-5.2-6> 이메일 메시지 객체 공통 속성

| 공통 속성 | | |
|---|---|---|
| type, extensions | | |
| 이메일 메시지 객체 고유 속성 | | |
| is_multipart, date, content_type, from_ref, sender_ref, to_refs, cc_refs, bcc_refs, subject, received_lines, additional_header_fields, body, body_multipart, raw_email_ref | | |
| 속성 이름 | 형식 | 설명 |
| type(필수) | string | 이 속성의 값은 email-message이어야 한다. |
| is_multipart(필수) | boolean | 이메일 본문에 여러 MIME 부분을 포함하는지 여부를 나타낸다. |
| date(선택 사항) | timestamp | 이메일 메시지를 전송한 날짜/시간을 지정한다. |
| content_type(선택 사항) | string | 이메일 메시지의 “콘텐츠 유형” 헤더의 값을 지정한다. |
| from_ref(선택 사항) | object-ref | 이메일 메시지의 “보낸 사람:” 헤더의 값을 지정한다. "보낸 사람:" 필드는 메시지의 작성자, 즉 메시지 쓰기를 책임지는 사람 또는 시스템의 사서함을 지정한다.<br><br>이 속성에 참조된 객체는 email-address 형식이어야 한다. |
| sender_ref(선택 사항) | object-ref | 이메일 메시지의 “보낸 사람” 필드의 값을 지정한다. "보낸 사람:" 필드는 메시지의 실제 전송을 책임지는 대행자의 사서함을 지정한다.<br><br>이 속성에 참조된 객체는 email-address 형식이어야 한다. |
| to_refs(선택 사항) | object-ref 형식의 list | 이메일 메시지의 “받는 사람:”인 사서함을 지정한다.<br><br>이 목록에 참조된 객체는 email-address 형식이어야 한다. |
| cc_refs(선택 사항) | object-ref 형식의 list | 이메일 메시지의 “참조:”인 사서함을 지정한다.<br><br>이 목록에 참조된 객체는 email-address 형식이어야 한다. |
| bcc_refs(선택 사항) | object-ref 형식의 list | 이메일 메시지의 “숨은 참조:”인 사서함을 지정한다.<br><br>[RFC5322]에 따라 이 목록이 비어 있을 수 |

| | | |
|---|---|---|
| | | 있으며, 이것을 없는 키와 같다고 취급하면 안 된다.<br><br>이 목록에 참조된 객체는 email-address 형식이어야 한다. |
| subject(선택 사항) | string | 이메일 메시지의 제목을 지정한다. |
| received_lines(선택 사항) | string 형식의 list | 이메일 헤더에 포함될 수 있는 하나 이상의 "받은 날짜" 헤더 필드를 지정한다.<br><br>목록 값은 이메일 메시지에 있는 것과 같은 순서로 나타나야 한다. |
| additional_header_fields(선택 사항) | dictionary | 이메일 메시지에서 발견된 다른 헤더 필드(date, received_lines, content_type, from_ref, sender_ref, to_refs, cc_refs, bcc_refs 및 subject 제외)를 사전으로 지정한다.<br><br>사전의 각 키/값 쌍은 단일 헤더 필드의 이름/값 또는 두 번 이상 발생한 헤더 필드의 이름/값을 나타낸다. 각 사전 키는 헤더 필드 이름의 대/소문자 유지 버전이어야 한다. 헤더 필드가 정확히 한 번 발생하는 경우 사전 키에 대한 해당 값은 string이어야 한다. 헤더 필드가 두 번 이상 발생하는 경우 사전 키에 대한 해당 값은 string 형식의 list이어야 하며, 각 list의 string은 헤더 필드의 단일 값을 표시한다. |
| body(선택 사항) | string | 이메일 본문이 포함된 string을 지정한다. 이 속성은 is_multipart가 false인 경우에만 사용할 수 있다. |
| body_multipart(선택 사항) | mime-part-type 형식의 list | 이메일 본문을 구성하는 MIME 부분의 목록을 지정한다. 이 속성은 is_multipart가 true인 경우에만 사용할 수 있다. |
| raw_email_ref(선택 사항) | object-ref | 이메일 메시지의 원시 이진 콘텐츠(헤더와 본문 모두 포함)를 아티팩트 객체에 대한 참조로 지정한다.<br><br>이 속성에 참조된 객체는 artifact 형식이어야 한다. |

### I-5.2.7 파일 객체(File Object)

파일 객체는 파일의 속성을 표시한다. 파일 객체는 hashes 또는 name 중 적어도 하나를 포함해야 한다.

<표 I-5.2-7> 파일 객체 공통 속성

| 공통 속성 |
|---|
| type, extensions |

| 파일 객체 고유 속성 | | |
|---|---|---|
| hashes, size, name, name_enc, magic_number_hex, mime_type, created, modified, accessed, parent_directory_ref, is_encrypted, encryption_algorithm, decryption_key, contains_refs, content_ref | | |
| 속성 이름 | 형식 | 설명 |
| type(필수) | string | 이 속성의 값은 file이어야 한다. |
| extensions(선택 사항) | dictionary | 파일 객체는 다음과 같은 확장명을 정의한다. 이러한 확장명에 더하여 생산자는 자기만의 확장명을 만들 수 있다<br><br>ntfs-ext, raster-image-ext, pdf-ext, archive-ext, windows-pebinary-ext<br><br>사전 키는 확장 형식을 이름에 의해 식별해야 한다.<br><br>해당 사전 값은 확장 인스턴스의 콘텐츠를 포함해야 한다. |
| hashes(선택 사항) | hashes | 파일에 대한 해시의 사전을 지정한다. |
| size(선택 사항) | integer | 파일의 크기(단위: 바이트)를 지정한다. 이 속성의 값은 음수이면 안 된다. |
| name(선택 사항) | string | 파일의 이름을 지정한다. |
| name_enc(선택 사항) | string | 파일의 이름에 대해 관측한 인코딩을 지정한다. 이 값은 IANA 문자 집합 레지스트리의 2013년 12월 20일 개정판에서 해당 이름을 사용하여 지정해야 한다. 문자 집합에 대해 선호하는 MIME 이름 열의 값이 정의된 경우 이 값을 반드시 사용해야 하며, 정의되지 않은 경우 레지스트리의 이름 열의 값을 대신 사용해야 한다.<br><br>이 속성을 사용하여 파일 이름에 대한 원본 텍스트 인코딩(과학 수사에 관련될 수 있음)을 포착할 수 있다. 예: Windows-1251 인코딩을 사용하여 그 이름을 생성하고 키릴 (Cyrillic) 자모 스크립트을 사용하는 NTFS 볼륨의 파일. |
| magic_number_hex(선택 사항) | hex | 적용 가능한 경우 파일에 해당하는 특정 파일 형식에 연결된 16진 상수("매직 넘버")를 지정한다. |
| mime_type(선택 사항) | string | 파일에 대해 지정된 MIME 형식 이름(예: application/msword)을 지정한다.<br><br>타당하다면 언제나 이 값은 IANA 미디어 형식 레지스트리 [Media Types]의 템플릿 열에 정의된 값 중 하나인 것이 바람직하다.<br><br>모든 현존하는 파일 형식의 포괄적인 범용 카탈로그를 유지하는 것은 분명 불가능하다. IANA 레지스트리에 포함되지 않은 MIME 형식을 지정할 때 구현자는 상호운용성을 촉진하기 위해 최선의 결정을 내려야 한다. |
| created(선택 사항) | timestamp | 파일이 생성된 날짜/시간을 지정한다. |

| | | |
|---|---|---|
| modified(선택 사항) | timestamp | 파일에 마지막 수정한 날짜/시간을 지정한다. |
| accessed(선택 사항) | timestamp | 파일을 마지막 액세스한 날짜/시간을 지정한다. |
| parent_directory_ref (선택 사항) | object-ref | 파일을 상위 디렉터리를 디렉터리 객체에 대한 참조로 지정한다.<br><br>이 속성에 참조된 객체는 directory 형식이어야 한다. |
| is_encrypted(선택 사항) | boolean | 파일이 암호화되는지 여부를 지정한다. |
| encryption_algorithm (선택 사항) | open-vocab | 파일을 암호화하는 데 사용된 암호화 알고리즘의 이름을 지정한다. 이는 개방형 어휘이며 값은 encryption-algo-ov 어휘에서 가져오는 것이 바람직하다.<br><br>이 속성은 is_encrypted가 false이거나 포함되지 않은 경우 사용하지 않아야 한다. |
| decryption_key(선택 사항) | string | 파일을 복호하는 데 사용되는 설명 키를 지정한다.<br><br>이 속성은 is_encrypted가 false이거나 포함되지 않은 경우 사용하지 않아야 한다. |
| contains_refs(선택 사항) | object-ref 형식의 list | 파일의 끝에 추가된 다른 파일 또는 파일의 다른 곳에 포함된 IP 주소 등 파일에 포함된 다른 관측 가능 객체에 대한 참조의 목록을 지정한다.<br><br>이는 보관 확장의 목표인 것 이외의 사용 사례를 위한 것이다. |
| content_ref(선택 사항) | object-ref | 아티팩트 객체로 표현된 파일의 콘텐츠를 지정한다.<br><br>이 속성에 참조된 객체는 artifact 형식이어야 한다. |

예제

관측된 인코딩이 없는 파일 시스템 속성을 가진 기본 파일

```
{
  "0": {
    "type": "file",
    "hashes": {
      "SHA-256": "fe90a7e910cb3a4739bed9180e807e93fa70c90f25a8915476f5e4bfbac681db"
    },
    "size": 25536,
    "name": "foo.dll"
  }
}
```

관측된 인코딩을 포함하고 파일 시스템 속성을 가진 기본 파일

```
{
  "0": {
    "type": "file",
    "hashes": {
      "SHA-256": "841a8921140aba50671ebb0770fecc4ee308c4952cfeff8de154ab14eeef4649"
    },
    "name": "quêry.dll",
    "name_enc": "windows-1252"
  }
}
```

### I-5.2.8 IP 주소 객체(IP Address Object)

IPv4 주소 객체는 CIDR 표기를 사용하여 표시된 하나 이상의 IPv4 주소를 표시한다.

<표 I-5.2-8> IPv4 주소 객체 공통 속성

| 공통 속성 | | |
|---|---|---|
| type, extensions | | |
| IPv4 주소 객체 고유 속성 | | |
| value, resolves_to_refs, belongs_to_refs | | |
| 속성 이름 | 형식 | 설명 |
| type(필수) | string | 이 속성의 값은 ipv4-addr이어야 한다. |
| value(필수) | string | CIDR 표기법을 사용하여 표시된 하나 이상의 IPv4 주소를 지정한다.<br><br>지정된 IPv4 주소 객체가 단일 IPv4 주소를 표시하는 경우 CIDR /32 접미사를 생략할 수 있다.<br><br>예: 10.2.4.5/24 |
| resolves_to_refs (선택 사항) | list of type | IPv4 주소가 확인되는 하나 이상의 레이어 2 MAC(Media Access Control) 주소에 대한 참조의 목록 |

| | | |
|---|---|---|
| | object-ref | 을 지정한다.<br><br>이 목록에 참조된 객체는 mac-addr 형식이어야 한다. |
| belongs_to_refs<br>(선택 사항) | object-ref<br>형식의 list | IPv4 주소가 속한 하나 이상의 AS(Autonomous System)에 대한 참조의 목록을 지정한다.<br><br>이 목록에 참조된 객체는 autonomous-system 형식이어야 한다. |

예제

IPv4 단일 주소

```
{
  "0": {
    "type": "ipv4-addr",
    "value": "198.51.100.3"
  }
}
```

IPv4 CIDR 블록

```
{
  "0": {
    "type": "ipv4-addr",
    "value": "198.51.100.0/24"
  }
}
```

### I-5.2.9 IPv6 주소 객체(IPv6 Address Object)

IPv6 주소 객체는 CIDR 표기를 사용하여 표시된 하나 이상의 IPv6 주소를 표시한다.

<표 I-5.2-9> IPv6 주소 객체 공통 속성

| 공통 속성 | | |
|---|---|---|
| type, extensions | | |
| IPv6 주소 객체 고유 속성 | | |
| value, resolves_to_refs, belongs_to_refs | | |
| 속성 이름 | 형식 | 설명 |
| type(필수) | string | 이 속성의 값은 ipv6-addr이어야 한다. |
| value(필수) | string | CIDR 표기법을 사용하여 표시된 하나 이상의 IPv6 주소를 지정한다.<br><br>지정된 IPv6 주소 객체가 단일 IPv6 주소를 표시하는 경우 CIDR /128 접미사를 생략할 수 있다. |
| resolves_to_refs<br>(선택 사항) | object-ref<br>형식의 list | IPv6 주소가 확인되는 하나 이상의 레이어 2 MAC(Media Access Control) 주소에 대한 참조의 목록을 지정한다. |

| | | |
|---|---|---|
| | | 이 목록에 참조된 객체는 mac-addr 형식이어야 한다. |
| belongs_to_refs (선택 사항) | object-ref 형식의 list | IPv6 주소가 속한 하나 이상의 AS(Autonomous System)에 대한 참조의 목록을 지정한다.<br><br>이 목록에 참조된 객체는 autonomous-system 형식이어야 한다. |

예제

IPv6 단일 주소

```
{
  "0": {
    "type": "ipv6-addr",
    "value": "2001:0db8:85a3:0000:0000:8a2e:0370:7334"
  }
}
```

IPv6 CIDR 블록

```
{
  "0": {
    "type": "ipv6-addr",
    "value": "2001:0db8::/96"
  }
}
```

### I-5.2.10 MAC 주소 객체(MAC Address Object)

MAC 주소 객체는 단일 MAC(미디어 액세스 제어, Media Access Control) 주소를 표시한다.

<표 I-5.2-10> MAC 주소 객체 공통 속성

| 공통 속성 | | |
|---|---|---|
| type, extensions | | |
| MAC 주소 객체 고유 속성 | | |
| value | | |
| 속성 이름 | 형식 | 설명 |
| type(필수) | string | 이 속성의 값은 mac-addr이어야 한다. |
| value(필수) | string | 단일 MAC 주소를 지정한다.<br><br>MAC 주소 값은 단일 콜론으로 구분된 소문자 MAC-48 주소로 표시해야 하며, 각 8진수에 대해 선행 0을 포함해야 한다.<br><br>예: 00:00:ab:cd:ef:01 |

예제

대표적 MAC 주소

```
{
  "0": {
    "type": "mac-addr",
    "value": "d2:fb:49:24:37:18"
  }
}
```

### I-5.2.11 뮤텍스 객체(Mutex Object)

뮤텍스 객체는 상호 배제(뮤텍스) 객체의 속성을 표시한다.

<표 I-5.2-11> 뮤텍스 객체 공통 속성

| 공통 속성 | | |
|---|---|---|
| type, extensions | | |
| 뮤텍스 객체 고유 속성 | | |
| name | | |
| 속성 이름 | 형식 | 설명 |
| type(필수) | string | 이 속성의 값은 mutex이어야 한다. |
| name(필수) | string | 뮤텍스 객체의 이름을 지정한다. |

예제

멀웨어 뮤텍스

```
{
  "0": {
    "type": "mutex",
    "name": "__CLEANSWEEP__"
  }
}
```

### I-5.2.12 네트워크 트래픽 객체(Network Traffic Object)

네트워크 트래픽 객체는 소스에서 발원하여 목적지로 지정되는 임시 네트워크 트래픽을 표시한다. 네트워크 트래픽은 유효한 유니캐스트, 멀티캐스트 또는 브로드캐스트 네트워크 연결을 구성할 수도 있고 그렇지 않을 수도 있다. 이 트래픽에 SYN 서비스 장애 등 설정되지 않은 트래픽을 포함할 수도 있다.

<표 I-5.2-12> 네트워크 트래픽 객체 공통 속성

| 공통 속성 | | |
|---|---|---|
| type, extensions | | |
| 네트워크 트래픽 고유 속성 | | |
| start, end, is_active, src_ref, dst_ref, src_port, dst_port, protocols, src_byte_count, dst_byte_count, src_packets, dst_packets, ipfix, src_payload_ref, dst_payload_ref, encapsulates_refs, encapsulated_by_ref | | |
| 속성 이름 | 형식 | 설명 |
| type(필수) | string | 이 속성의 값은 network-traffic이어야 한다. |
| extensions(선택 사항) | dictionary | 네트워크 트래픽 객체는 다음과 같은 확장명을 정의한다. 이러한 확장명에 더하여 생산자는 자기만의 확장명을 만들 수 있다<br><br>http-ext, tcp-ext, icmp-ext, socket-ext<br><br>사전 키는 확장 형식을 이름에 의해 식별해야 한다.<br><br>해당 사전 값은 확장 인스턴스의 콘텐츠를 포함해야 한다. |
| start(선택 사항) | timestamp | 네트워크 트래픽이 시작된 날짜/시간(알려진 경우)을 지정한다. |
| end(선택 사항) | timestamp | 네트워크 트래픽이 종료된 날짜/시간(알려진 경우)을 지정한다.<br><br>is_active 속성이 true인 경우 end 속성을 포함하면 안 된다. |
| is_active(선택 사항) | boolean | 네트워크 트래픽이 여전히 진행 중인지 여부를 나타낸다. |
| src_ref(선택 사항) | object-ref | 네트워크 트래픽의 소스를 하나 이상의 관측 가능 객체에 대한 참조로 지정한다.<br><br>이 목록에 참조된 객체는 ipv4-addr, ipv6-addr, mac-addr 또는 domain-name(도메인에 대한 IP 주소를 모르는 경우) 형식이어야 한다. |
| dst_ref(선택 사항) | object-ref | 네트워크 트래픽의 대상을 하나 이상의 관측 가능 객체에 대한 참조로 지정한다.<br><br>이 목록에 참조된 객체는 ipv4-addr, ipv6-addr, mac-addr 또는 domain-name(도메인에 대한 IP 주소를 모르는 경우) 형식이어야 한다. |
| src_port(선택 사항) | integer | 네트워크 트래픽에 사용된 소스 포트를 정수로 지정한다. 포트는 0 ~ 65535 범위이어야 한다. |
| dst_port(선택 사항) | integer | 네트워크 트래픽에 사용된 대상 포트를 정수로 지정한다. 포트는 0 ~ 65535 범위이어야 한다. |

| | | |
|---|---|---|
| protocols(필수) | string 형식의 list | 네트워크 트래픽에서 관측된 프로토콜을 해당 상태와 함께 지정한다.<br><br>프로토콜은 외측에서 내측까지 낮은 것에서 높은 것의 순서로 패킷 캡슐화에 의해 열거해야 한다. 즉, 패킷의 외측 레벨의 프로토콜(IP 등)을 먼저 열거해야 한다.<br><br>프로토콜 이름은 IANA 서비스 이름 및 포트 번호 레지스트리[Port Numbers]의 Service Name(서비스 이름) 열에 정의된 서비스 이름에서 가져오는 것이 바람직하다. IANA 레지스트리에 포함되지 않은 네트워크 프로토콜의 이름에 변화가 있는 경우 콘텐츠 생산자는 자신이 최선의 판단을 해야 하며, IANA 레지스트리와의 일관성을 확보하기 위해 소문자 이름을 사용하는 것이 좋다.<br><br>예:<br>ipv4, tcp, http<br>ipv4, udp<br>ipv6, tcp, http<br>ipv6, tcp, ssl, https |
| src_byte_count(선택 사항) | integer | 소스에서 대상으로 보낸 바이트 수를 지정한다. |
| dst_byte_count(선택 사항) | integer | 대상에서 소스로 보낸 바이트 수를 지정한다. |
| src_packets(선택 사항) | integer | 소스에서 대상으로 보낸 패킷 수를 지정한다. |
| dst_packets(선택 사항) | integer | 대상에서 소스로 보낸 패킷 수를 지정한다. |
| ipfix(선택 사항) | dictionary | 트래픽에 대한 IP 흐름 정보 내보내기(IP Flow Information Export)[IPFIX] 데이터를 사전으로 지정한다. 사전의 각 키/값 쌍은 단일 IPFIX 요소의 이름/값을 나타낸다. 따라서 각 사전 키는 IPFIX 요소 이름(예: octetDeltaCount)의 대/소문자 유지 버전인 것이 바람직하다. 각 사전 값은 integer 또는 string 및 유효한 IPFIX 속이어야 한다. |
| src_payload_ref(선택 사항) | object-ref | 소스에서 대상으로 보낸 바이트 수를 지정한다.<br><br>이 속성에 참조된 객체는 artifact 형식이어야 한다. |
| dst_payload_ref(선택 사항) | object-ref | 대상에서 소스로 보낸 바이트 수를 지정한다.<br><br>이 속성에 참조된 객체는 artifact 형식이어야 한다. |
| encapsulates_refs(선택 사항) | list of type object-ref | 이 network-traffic 객체에 의해 캡슐화된 다른 network-traffic 객체에 연결한다.<br><br>이 속성에 참조된 객체는 network-traffic 형식이어야 한다. |
| encapsulated_by_ref(선택 사항) | object-ref | 이 객체를 캡슐화하는 다른 network-traffic 객체에 연결한다.<br><br>이 속성에 참조된 객체는 network-traffic 형식이어야 한다. |

예제

기본 TCP 네트워크 트래픽

```
{
  "0": {
    "type": "ipv4-addr",
    "value": "198.51.100.2"
  },
  "1":{
    "type": "ipv4-addr",
    "value": "198.51.100.3"
  },
  "2": {
    "type": "network-traffic",
    "src_ref": "0",
    "dst_ref": "1",
    "protocols": [
      "tcp"
    ]
  }
}
```

예제

기본 HTTP 네트워크 트래픽

```
{
  "0": {
    "type": "domain-name",
    "value": "example.com"
  },
  "1": {
    "type": "network-traffic",
    "dst_ref": "0",
    "protocols": [
      "ipv4",
      "tcp",
      "http"
    ]
  }
}
```

Netflow 데이터 포함 네트워크 트래픽

```
{
  "0": {
    "type": "ipv4-addr",
    "value": "203.0.113.1"
  },
  "1": {
    "type": "ipv4-addr",
    "value": "203.0.113.5"
  },
  "2": {
    "type": "network-traffic",
    "src_ref": "0",
    "dst_ref": "1",
    "protocols": [
      "ipv4",
      "tcp"
    ],
    "src_byte_count": 147600,
    "src_packets": 100,
    "ipfix": {
      "minimumIpTotalLength": 32,
      "maximumIpTotalLength": 2556
    }
  }
}
```

예제

기본 터널링된(Tunneled) 네트워크 트래픽


```
{
  "0": {
    "type": "ipv4-addr",
    "value": "198.51.100.2"
  },
  "1": {
    "type": "ipv4-addr",
    "value": "203.0.113.1"
  },
  "2": {
    "type": "ipv4-addr",
    "value": "203.0.113.2"
  },
  "3": {
    "type": "network-traffic",
    "src_ref": "0",
    "dst_ref": "1",
    "src_port": 2487,
    "dst_port": 1723,
    "protocols": [
      "ipv4",
      "pptp"
    ],
    "src_byte_count": 35779,
    "dst_byte_count": 935750,
    "encapsulates_refs": [
      "4"
    ]
  },
  "4": {
    "type": "network-traffic",
    "src_ref": "0",
    "dst_ref": "2",
    "src_port": 24678,
    "dst_port": 80,
    "protocols": [
      "ipv4",
      "tcp",
      "http"
    ],
    "src_packets": 14356,
    "dst_packets": 14356,
    "encapsulated_by_ref": "3"
  }
}
```

예제

DNS 상에 터널링된 웹 트래픽

```
{
  "0": {
    "type": "ipv4-addr",
    "value": "203.0.113.1"
  },
  "1": {
    "type": "ipv4-addr",
    "value": "198.51.100.34"
  },
  "2": {
    "type": "ipv4-addr",
    "value": "198.51.100.54"
  },
  "3": {
    "type": "network-traffic",
    "src_ref": "0",
    "dst_ref": "1",
    "src_port": 2487,
    "dst_port": 53,
    "protocols": [
      "ipv4",
      "udp",
      "dns"
    ],
    "src_byte_count": 35779,
    "dst_byte_count": 935750,
    "encapsulates_refs": [
      "4"
    ]
  },
  "4": {
    "type": "network-traffic",
    "src_ref": "0",
    "dst_ref": "2",
    "src_port": 24678,
    "dst_port": 443,
    "protocols": [
      "ipv4",
      "tcp",
      "ssl",
      "http"
    ],
    "src_packets": 14356,
    "dst_packets": 14356,
    "encapsulated_by_ref": "3"
  }
}
```

### I-5.2.13 프로세스 객체(Process Object)

프로세스 객체는 운영 체제에서 실행될 때 컴퓨터 프로그램 인스턴스의 공통 속성을 표시한다. 프로세스 객체는 이 객체(또는 그 확장 중 하나)에서 적어도 한 개의 속성(type 제외)을 포함해야 한다.

<표 I-5.2-13> 프로세스 객체 공통 속성

| 공통 속성 | | |
|---|---|---|
| type, extensions | | |
| 프로세스 객체 고유 속성 | | |
| is_hidden, pid, name, created, cwd, arguments, command_line, environment_variables, opened_connection_refs, creator_user_ref, binary_ref, parent_ref, child_refs | | |
| 속성 이름 | 형식 | 설명 |
| type(필수) | string | 이 속성의 값은 process이어야 한다. |
| extensions(선택 사항) | dictionary | 프로세스 객체는 다음과 같은 확장명을 정의한다. 이러한 확장명에 더하여 생산자는 자기만의 확장명을 만들 수 있다<br><br>windows-process-ext, windows-service-ext<br><br>사전 키는 확장 형식을 이름에 의해 식별해야 한다.<br><br>해당 사전 값은 확장 인스턴스의 콘텐츠를 포함해야 한다. |
| is_hidden(선택 사항) | boolean | 프로세스가 숨겨졌는지 여부를 지정한다. |
| pid(선택 사항) | integer | 프로세스의 프로세스 ID(일명 PID)를 지정한다. |
| name(선택 사항) | string | 프로세스의 이름을 지정한다. |
| created(선택 사항) | timestamp | 프로세스가 생성된 날짜/시간을 지정한다. |
| cwd(선택 사항) | string | 프로세스의 현재 작업 디렉터리를 지정한다. |
| arguments(선택 사항) | string 형식의 list | 프로세스를 실행하는 데 사용되는 인수의 목록을 지정한다. 각 인수는 문자열로 따로 포착되어야 한다. |
| command_line(선택 사항) | string | 프로세스 이름(운영 체제에 따라 달라짐)을 포함하여 프로세스를 실행하는 데 사용되는 전체 명령줄을 지정한다. |
| environment_variables(선택 사항) | dictionary | 프로세스와 연결된 환경 변수의 목록을 사전으로 지정한다. 사전의 각 키는 환경 변수 이름의 대/소문자 유지 버전이어야 하며 각 해당 값은 문자열인 환경 변수 값이어야 한다. |
| opened_connection_refs(선택 사항) | object-ref 형식의 list | 프로세스가 연 네트워크 연결의 목록을 하나 이상의 네트워크 트래픽 객체에 대한 참조로 지정한다.<br><br>이 목록에 참조된 객체는 network-traffic 형식이어야 한다. |
| creator_user_ref(선택 사항) | object-ref | 프로세스를 생성한 사용자를 사용자 계정 객체에 대한 참조로 지정한다. |

| | | |
|---|---|---|
| | | 이 속성에 참조된 객체는 user-account 형식이어야 한다. |
| binary_ref(선택 사항) | object-ref | 프로세스로 실행되는 실행 이진 파일을 파일 객체에 대한 참조로 지정한다.<br><br>이 속성에 참조된 객체는 file 형식이어야 한다. |
| parent_ref(선택 사항) | object-ref | 이 프로세스를 생성한 다른 프로세스를 프로세스 객체에 대한 참조로 지정한다.<br><br>이 속성에 참조된 객체는 process 형식이어야 한다. |
| child_refs(선택 사항) | object-ref 형식의 list | 이 프로세스에 의해 생성된 다른 프로세스(즉, 이 프로세스의 자식)를 하나 이상의 다른 프로세스 객체에 대한 참조로 지정한다.<br><br>이 목록에 참조된 객체는 process 형식이어야 한다. |

예제

```
기본 프로세스
{
  "0": {
    "type": "file",
    "hashes": {
      "SHA-256": "35a01331e9ad96f751278b891b6ea09699806faedfa237d40513d92ad1b7100fSHA"
    },
  },
  "1": {
    "type": "process",
    "pid": 1221,
    "name": "gedit-bin",
    "created": "2016-01-20T14:11:25.55Z",
    "arguments" :[
      "--new-window"
    ],
    "binary_ref": "0"
  }
}
```

### I-5.2.14 소프트웨어 객체(Software Object)

소프트웨어 객체는 소프트웨어(소프트웨어 제품 포함)와 연결된 고급 속성을 표시한다.

<표 I-5.2-14> 소프트웨어 객체 공통 속성

| 공통 속성 |
|---|
| type, extensions |

| 소프트웨어 객체 고유 속성 | | |
|---|---|---|
| name, cpe, languages, vendor, version | | |
| 속성 이름 | 형식 | 설명 |
| type(필수) | string | 이 속성의 값은 software이어야 한다. |
| name(필수) | string | 소프트웨어의 이름을 지정한다. |
| cpe(선택 사항) | string | 사용 가능한 경우 소프트웨어에 대한 CPE(Common Platform Enumeration) 항목을 지정한다. 이 속성에 대한 값은 공식 NVD CPE 디렉터리[NVD]의 CPE v2.3 항목이어야 한다.<br><br>CPE 사전이 모든 소프트웨어에 대한 항목을 포함하는 것은 아니지만, 소프트웨어의 지정된 인스턴스에 대한 ID를 포함하고 있는 경우에는 언제나 이 속성이 존재하는 것이 바람직하다. |
| languages(선택 사항) | string 형식의 list | 소프트웨어가 지원하는 언어를 지정한다. 각 목록 구성원의 값은 ISO 639-2 언어 코드[ISO639-2]이어야 한다. |
| vendor(선택 사항) | string | 소프트웨어의 공급업체 이름을 지정한다. |
| version(선택 사항) | string | 소프트웨어의 버전을 지정한다. |

예제

대표적 소프트웨어 인스턴스

```
{
  "0": {
    "type": "software",
    "name": "Word",
    "cpe": "cpe:2.3:a:microsoft:word:2000:*:*:*:*:*:*:*",
    "version": "2002",
    "vendor": "Microsoft"
  }
}
```

### I-5.2.15 URL 객체(URL Object)

URL 객체는 URL(Uniform Resource Locator)의 속성을 표시한다.

<표 I-5.2-15> URL 객체 공통 속성

| 공통 속성 |
|---|
| type, extensions |
| URL 객체 고유 속성 |
| value |

| 속성 이름 | 형식 | 설명 |
|---|---|---|
| type(필수) | string | 이 속성의 값은 url이어야 한다. |
| value(필수) | string | URL의 값을 지정한다. 이 속성의 값은 [RFC3986]을 준수해야 한다. 더 구체적인 사항은 섹션 1.1.3에서 "Uniform Resource Locator"에 대한 정의를 참조한다. |

예제

대표적 URL

```
{
  "0": {
    "type": "url",
    "value": "https://example.com/research/index.html"
  }
}
```

### I-5.2.16 사용자 계정 객체(User Account Object)

사용자 계정 객체는 운영 체제, 장치, 메시지 서비스 및 소셜 미디어 플랫폼 계정을 포함하되 이에 국한되지 않고, 임의 유형의 사용자 계정의 인스턴스를 표시한다.

<표 I-5.2-16> 사용자 계정 객체 공통 속성

| 공통 속성 | | |
|---|---|---|
| type, extensions | | |
| 사용자 계정 객체 고유 속성 | | |
| user_id, account_login, account_type, display_name, is_service_account, is_privileged, can_escalate_privs, is_disabled, account_created, account_expires, password_last_changed, account_first_login, account_last_login | | |
| 속성 이름 | 형식 | 설명 |
| type(필수) | string | 이 속성의 값은 user-account이어야 한다. |
| extensions(선택 사항) | dictionary | 사용자 계정 객체는 다음과 같은 확장명을 정의한다. 이러한 확장명에 더하여 생산자는 자기만의 확장명을 만들 수 있다<br><br>unix-account-ext<br><br>사전 키는 확장 형식을 이름에 의해 식별해야 한다.<br><br>해당 사전 값은 확장 인스턴스의 콘텐츠를 포함해야 한다. |
| user_id(필수) | string | 계정의 ID를 지정한다. ID의 형식은 사용자 계정이 유지되는 시스템에 따라 달라지면 숫자 ID, GUID, 계정 이름, 이메일 주소 등일 수 있다. |

| | | |
|---|---|---|
| | | user_id 속성은 계정이 그 구성원인 시스템의 고유 ID인 필드로 채워진다. 예를 들어 UNIX 시스템의 경우 UID로 채워진다. |
| account_login(선택 사항) | string | user_id 속성이 사용자가 로그인할 때 입력하는 내용 이 외의 것을 지정하는 경우 사용되는 계정 로그인 문자열을 지정한다.<br><br>예를 들어 user_id 0인 Unix 계정의 경우 account_login은 “root”이다. |
| account_type(선택 사항) | open-vocab | 계정의 형식을 지정한다.<br><br>이는 개방형 어휘이며 값은 account-type-ov 어휘에서 가져오는 것이 바람직하다. |
| display_name(선택 사항) | string | 해당하는 경우 사용자 인터페이스에 표시할 계정의 표시 이름을 지정한다.<br>Unix에서는 이 속성이 GECOS 필드와 동등하다. |
| is_service_account(선택 사항) | boolean | 계정이 특정 개인이 아닌 네트워크 서비스 또는 시스템 프로세스(디먼)와 연결되었음을 나타낸다. |
| is_privileged(선택 사항) | boolean | 계정이 권한을 상승시켰음을 지정한다(예를 들어 Unix 또는 Windows 관리자 계정의 루트인 경우). |
| can_escalate_privs(선택 사항) | boolean | 계정이 권한을 상승시킬 수 있는 기능을 가지고 있음을 지정한다(예를 들어 Unix 또는 Windows 도메인 관리자 계정의 sudo의 경우). |
| is_disabled(선택 사항) | boolean | 계정이 비활성화되었는지 여부를 지정한다. |
| account_created(선택 사항) | timestamp | 계정이 생성된 시기를 지정한다. |
| account_expires(선택 사항) | timestamp | 계정의 만료 날짜를 지정한다. |
| password_last_changed(선택 사항) | timestamp | 계정 암호가 마지막 변경된 시기를 지정한다. |
| account_first_login(선택 사항) | timestamp | 계정이 처음 액세스된 시기를 지정한다. |
| account_last_login(선택 사항) | timestamp | 계정이 마지막 액세스된 시기를 지정한다. |

### I-5.2.17 윈도우 레지스트리 키 객체(Windows Registry Key Object)

레지스트리 키 객체는 Windows 레지스트리 키의 속성을 표시한다.

<표 I-5.2-17> 윈도우 레지스트리 키 객체 공통 속성

| 공통 속성 |
|---|
| type, extensions |
| 파일 객체 고유 속성 |

| key, values, modified, creator_user_ref, number_of_subkeys | | |
|---|---|---|
| 속성 이름 | 형식 | 설명 |
| type(필수) | string | 이 속성의 값은 windows-registry-key이어야 한다. |
| key(필수) | string | hive를 포함한 전체 레지스트리 키를 지정한다.<br><br>키의 값은 하이브 부분을 포함하여 대/소문자를 유지하는 것이 바람직하다. 키의 하이브 부분은 전체 확장되어야 하며 잘리면 안 된다. 예를 들어 HKLM 대신에 HKEY_LOCAL_MACHINE을 사용해야 한다. |
| values(선택 사항) | windows-registry-value-type 형식의 list | 레지스트리 키 아래에서 발견된 값을 지정한다. |
| modified(선택 사항) | timestamp | 레지스트리 키가 수정된 마지막 날짜/시간을 지정한다. |
| creator_user_ref(선택 사항) | object-ref | 레지스트리 키를 만든 사용자 계정(사용자 계정 객체로 표시됨)에 대한 참조를 지정한다.<br><br>이 속성에 참조된 객체는 user-account 형식이어야 한다. |
| number_of_subkeys(선택 사항) | integer | 레지스트리 키 아래에 포함된 하위 키 수를 지정한다. |

### I-5.2.18 X.509 인증서 객체(X.509 Certificate Object)

X.509 인증서 객체는ITU 권고 X.509[X.509]에 정의된 X.509 인증서의 속성을 표시한다. X.509 인증서 객체는 이 객체(또는 그 확장 중 하나)에서 적어도 한 개의 속성(type 제외)을 포함해야 한다.

<표 I-5.2-18> X.509 인증서 객체 공통 속성

| 공통 속성 | | |
|---|---|---|
| type, extensions | | |
| 파일 객체 고유 속성 | | |
| is_self_signed, hashes, version, serial_number, signature_algorithm, issuer, validity_not_before, validity_not_after, subject, subject_public_key_algorithm, subject_public_key_modulus, subject_public_key_exponent, x509_v3_extensions | | |
| 속성 이름 | 형식 | 설명 |
| type(필수) | string | 이 속성의 값은 x509-certificate이어야 한다. |
| is_self_signed(선택 사항) | boolean | 인증서가 자체 서명되었는지 여부, 즉 그 ID를 인증하는 엔터티와 동일 |

| | | |
|---|---|---|
| | | 한 엔터티가 서명했는지 여부를 지정한다. |
| hashes(선택 사항) | hashes | 인증서의 전체 콘텐츠에 대해 계산된 해시를 지정한다. |
| version(선택 사항) | string | 인코딩된 인증서의 버전을 지정한다. |
| serial_number(선택 사항) | string | 특정 인증기관이 발행한 인증서에 대한 고유 ID를 지정한다. |
| signature_algorithm(선택 사항) | string | 인증서에 서명하는 데 사용된 알고리즘의 이름을 지정한다. |
| issuer(선택 사항) | string | 인증서를 발급한 인증기관의 이름을 지정한다. |
| validity_not_before(선택 사항) | timestamp | 인증서 유효 기간이 시작되는 날짜를 지정한다. |
| validity_not_after(선택 사항) | timestamp | 인증서 유효 기간이 종료되는 날짜를 지정한다. |
| subject(선택 사항) | string | 인증서의 주체 공개 키 필드에 저장된 공개 키와 연결된 엔터티의 이름을 지정한다. |
| subject_public_key_algorithm(선택 사항) | string | 주체에게 전송하는 데이터를 암호화할 때 사용하는 알고리즘의 이름을 지정한다. |
| subject_public_key_modulus(선택 사항) | string | 주체의 공개 RSA 키의 modulus 부분을 지정한다. |
| subject_public_key_exponent(선택 사항) | integer | 주체의 공개 RSA 키의 지수 부분을 정수로 지정한다. |
| x509_v3_extensions(선택 사항) | x509-v3-extensions-type | 인증서에 사용될 수 있는 표준 X.509 v3 확장을 지정한다. |

### I-5.3. 적합성

#### I-5.3.1 정의된 객체 생산자(Defined Object Producers)

제 2 절(정의된 객체 데이터 모델)에서 객체를 생성하는 "정의된 객체 생산자"는 해당 객체의 “생산자”이다. 정의된 객체 생산자는 STIX™ Version 2.0. Part 3: Cyber Observable Core Concepts 제3절에 정의된 객체에 속하는 모든 일반 요구사항과 함께 해당 객체에 대한 절의 모든 표준 요구사항을 준수해야 한다.

예를 들어 파일 객체를 생성할 수 있는 "정의된 객체 생산자"는 "파일 객체 생산자"이다. 이 생산자는 사이버 관측 가능 객체 2.7절, 파일 객체의 모든 표준 요구사항을 준수해야 한다.

#### I-5.3.2 정의된 객체 소비자(Defined Object Consumers)

제2절(정의된 객체 데이터 모델)에서 객체를 수신하는 "정의된 객체 소비자"는 해당 객체의 “소비자”이다. 정의된 객체 소비자는 STIX™ Version 2.0. Part 3: Cyber Observable Core

Concepts 3절에 정의된 객체에 속하는 모든 일반 요구사항과 함께 해당 객체에 대한 섹션의 모든 표준 요구사항을 준수해야 한다.
예를 들어 네트워크 트래픽 객체를 수신할 수 있는 "객체 소비자"는 "네트워크 트래픽 객체 소비자"이다. 이 소비자는 사이버 관측 가능 객체 2.12절, 네트워크 트래픽 객체의 모든 표준 요구사항을 준수해야 한다.

# 부 록 I-6

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

## 표준의 이력

| 판수 | 채택일 | 표준번호 | 내용 | 담당 위원회 |
|---|---|---|---|---|
| 제1판 | 2018.XX.XX | 제정<br>TTAE.OT-xx.xxxx | - | 사이버보안<br>프로젝트 그룹<br>(PG503),<br>정보보호<br>기술위원회(TC5) |