

STIX 2.0 Specification

Part 3b: Cyber Observable Objects - Version 2.0-rc3

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

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

Additional Artifacts

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

- STIX Version 2.0 Part 1: STIX Core Concepts
- STIX Version 2.0 Part 2: STIX Objects
- STIX Version 2.0 Part 3a: Cyber Observable Core Concepts
- STIX Version 2.0 Part 3b: Cyber Observable Objects (this document)
- STIX Version 2.0 Part 4: STIX Patterning

Table of Contents

[1. Cyber Observable Objects](#)

[2. Defined Object Data Models](#)

[2.1. Artifact Object](#)

[2.1.1. Properties](#)

[2.1.2. Examples](#)

[2.2. AS Object](#)

- [2.2.1. Properties](#)
- [2.2.2. Examples](#)
- [2.3. Directory Object](#)
 - [2.3.1. Properties](#)
 - [2.3.2. Examples](#)
- [2.4. Domain Name Object](#)
 - [2.4.1. Properties](#)
 - [2.4.2. Examples](#)
- [2.5. Email Address Object](#)
 - [2.5.1. Properties](#)
 - [2.5.2. Examples](#)
- [2.6. Email Message Object](#)
 - [2.6.1. Properties](#)
 - [2.6.2. Email MIME Component Type](#)
 - [2.6.2.1. Properties](#)
 - [2.6.3. Examples](#)
- [2.7. File Object](#)
 - [2.7.1. Properties](#)
 - [2.7.2. Examples](#)
 - [2.7.3. Archive File Extension](#)
 - [2.7.3.1. Properties](#)
 - [2.7.3.2. Example](#)
 - [2.7.4. NTFS File Extension](#)
 - [2.7.4.1. Properties](#)
 - [2.7.4.2. Alternate Data Stream Type](#)
 - [2.7.4.2.1. Properties](#)
 - [2.7.4.3. Example](#)
 - [2.7.5. PDF File Extension](#)
 - [2.7.5.1. Properties](#)
 - [2.7.5.2. Example](#)
 - [2.7.6. Raster Image File Extension](#)
 - [2.7.6.1. Properties](#)
 - [2.7.6.2. Example](#)
 - [2.7.7. Windows™ PE Binary File Extension](#)
 - [2.7.7.1. Properties](#)
 - [2.7.7.2. Windows PE Binary Vocabulary](#)
 - [2.7.7.3. PE Optional Header Type](#)
 - [2.7.7.3.1. Properties](#)
 - [2.7.7.4. Windows PE Section Type](#)

[2.7.7.4.1. Properties](#)

[2.7.7.5. Example](#)

[2.8. IPv4 Address Object](#)

[2.8.1. Properties](#)

[2.8.2. Examples](#)

[2.9. IPv6 Address Object](#)

[2.9.1. Properties](#)

[2.9.2. Examples](#)

[2.10. MAC Address Object](#)

[2.10.1. Properties](#)

[2.10.2. Examples](#)

[2.11. Mutex Object](#)

[2.11.1. Properties](#)

[2.11.2. Examples](#)

[2.12. Network Traffic](#)

[2.12.1. Properties](#)

[2.12.2. Examples](#)

[2.12.3. HTTP Extension](#)

[2.12.3.1. Properties](#)

[2.12.3.2. Example](#)

[2.12.4. ICMP Extension](#)

[2.12.4.1. Properties](#)

[2.12.4.2. Example](#)

[2.12.5. Network Socket Extension](#)

[2.12.5.1. Properties](#)

[2.12.5.2. Network Socket Address Family Enumeration](#)

[2.12.5.3. Network Socket Protocol Family Enumeration](#)

[2.12.5.4. Network Socket Type Enumeration](#)

[2.12.5.5. Example](#)

[2.12.6. TCP Extension](#)

[2.12.6.1. Properties](#)

[2.12.6.2. Example](#)

[2.13. Process Object](#)

[2.13.1. Properties](#)

[2.13.2. Examples](#)

[2.13.3. Windows Process Extension](#)

[2.13.3.1. Properties](#)

[2.13.3.2. Example](#)

[2.13.4. Windows Service Extension](#)

- [2.13.4.1. Properties](#)
 - [2.13.4.2. Windows Service Start Type Enumeration](#)
 - [2.13.4.3. Windows Service Type Enumeration](#)
 - [2.13.4.4. Window Service Status Enumeration](#)
 - [2.13.4.5. Example](#)
 - [2.14. Software Object](#)
 - [2.14.1. Properties](#)
 - [2.14.2. Examples](#)
 - [2.15. URL Object](#)
 - [2.15.1. Properties](#)
 - [2.15.2. Examples](#)
 - [2.16. User Account Object](#)
 - [2.16.1. Properties](#)
 - [2.16.2. Account Type Vocabulary](#)
 - [2.16.3. Examples](#)
 - [2.16.4. UNIX Account Extension](#)
 - [2.16.4.1. Properties](#)
 - [2.16.4.2. Example](#)
 - [2.17. Windows Registry Key Object](#)
 - [2.17.1. Properties](#)
 - [2.17.2. Windows Registry Value Type](#)
 - [2.17.2.1. Properties](#)
 - [2.17.3. Registry Datatype Enumeration](#)
 - [2.17.4. Examples](#)
 - [2.18. X509 Certificate Object](#)
 - [2.18.1. Properties](#)
 - [2.18.2. X509 v3 Extensions Type](#)
 - [2.18.2.1. Properties](#)
 - [2.18.3. Examples](#)
- [3. Conformance](#)
 - [3.1. Defined Object Producers](#)
 - [3.2. Defined Object Consumers](#)
 - [4. Appendix A. Acknowledgments](#)
 - [5. Appendix B. Revision History](#)

1. Cyber Observable Objects

A Cyber Observable Object represents an instance of information observed on a host or network; for example, this could include the properties of a PDF file observed on an endpoint or a network connection between two IP addresses as observed by a firewall.

STIX defines a set of **object data models**, such as the *File Object*, for the normalized capture of observed data as supporting evidence in STIX Domain Objects or outside of STIX in a similar, higher-level language.

Defined **object data models** provide a base set of **properties** which are applicable across a broad spectrum of use cases relevant to the particular data model as well as (in certain cases) a set of object **extensions** targeting more specific use cases. The full set of Defined Object Data Models is defined in §2.

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 string, or linking to a file-like payload. The size of the base64-encoded data captured in the **payload** 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
<code>type, description, extended_properties</code>
Artifact Object Specific Properties
<code>mime_type, payload_bin, url, hashes</code>

Property Name	Type	Description
<code>type</code> (required)	string	The value of this property MUST be artifact .
<code>mime_type</code> (optional)	string	The value of this property MUST be a valid MIME type as specified in the IANA Media Types registry .
<code>payload_bin</code> (optional)	binary	Specifies the binary data contained in the artifact as a base64-encoded string. This property MUST NOT be present if <code>url</code> is provided.
<code>url</code> (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 <code>payload_bin</code> is provided.
<code>hashes</code> (optional)	hashes-type	Specifies a dictionary of hashes for the contents of the <code>url</code> or the <code>payload_bin</code> . This MUST be provided when the <code>url</code> property is present.

2.1.2. 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 Systems (AS).

2.2.1. Properties

Common Properties		
type, description, extended_properties		
AS Object Specific Properties		
number, name, rir		
Property Name	Type	Description
type (required)	string	The value of this property MUST be <code>as</code> .
number (required)	integer	Specifies the number assigned to the AS. Such assignments are typically performed by a Regional Internet Registries (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.

2.2.2. Examples

Basic AS Object

```
{
  "0": {
    "type": "as",
    "number": "15139",
    "name": "Wayne Industries",
    "rir": "ARIN"
  }
}
```

2.3. Directory Object

Type Name: `directory`

The Directory Object represents the properties common to a file system directory. A Directory Object **MUST** contain at least one of **path** OR **path_bin**.

2.3.1. Properties

Common Properties		
type, description, extended_properties		
File Object Specific Properties		
path, path_enc, path_bin, created, modified, accessed, contains_refs		
Property Name	Type	Description
type (required)	string	The value of this property MUST be directory .
path (optional)	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 . 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.
path_bin (optional)	binary	Specifies the directory path as a base64-encoded string. This property MUST NOT be specified in conjunction with the path property; only one of path OR path_bin may be used.
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.

2.3.2. 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, description, extended_properties		
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.
resolves_to_refs	list of type object-ref	<p>Specifies a list of references to one or more IP addresses or domain names that the domain name resolves to.</p> <p>The objects referenced in this list SHOULD be of type ipv4-addr or ipv6-addr or MAY be of type domain-name for cases such as CNAME records.</p>

2.4.2. Examples

Basic FQDN

```
{
  "0": {
    "type": "domain-name",
    "value": "www.example.com",
    "resolves_to_refs": [
      "1"
    ]
  },
  "1": {
    "type": "ipv4-addr",
    "value": "1.2.3.4"
  }
}
```

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, description, extended_properties
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 <code>email-addr</code> .
value (required)	string	Specifies a single email address. This MUST not include the display name. This property corresponds to the <i>addr-spec</i> construction in RFC 5322 Section 3.4, for example, <i>jane.smith@example.com</i> .
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. This property corresponds to the <i>display-name</i> construction in RFC 5322 Section 3.4, for example, " <i>Jane Smith</i> ".
belongs_to_ref (optional)	object-ref	Specifies the user account that the email address belongs to, as a reference to a User Account Object. The object referenced in this property MUST be of type <code>user-account</code> .

2.5.2. Examples

```
{
  "0": {
    "type": "email-addr",
    "value": "bruce@wayneindustries.com",
    "display_name": "Bruce Wayne"
  }
}
```

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 RFC 5322 and related RFCs.

Header field values that have been encoded as described in Section 2 of [RFC 2047](#) **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		
<code>type, description, extended_properties</code>		
Email Message Object Specific Properties		
<code>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</code>		
Property Name	Type	Description
<code>type</code> (required)	<code>string</code>	The value of this property MUST be <code>email-message</code> .
<code>is_multipart</code> (required)	<code>boolean</code>	Indicates whether the email body contains multiple MIME parts.
<code>date</code> (optional)	<code>timestamp</code>	Specifies the date/time that the email message was sent.
<code>content_type</code> (optional)	<code>string</code>	Specifies the value of the "Content-Type" header of the email message.
<code>from_ref</code> (optional)	<code>object-ref</code>	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. The object referenced in this property MUST be of type <code>email-address</code> .

<code>sender_ref</code> (optional)	<code>object-ref</code>	Specifies the value of the "From" field of the email message. The "Sender:" field specifies the mailbox of the agent responsible for the actual transmission of the message. The object referenced in this property MUST be of type <code>email-address</code> .
<code>to_refs</code> (optional)	<code>list</code> of type <code>object-ref</code>	Specifies the mailboxes that are "To:" recipients of the email message. The objects referenced in this list MUST be of type <code>email-address</code> .
<code>cc_refs</code> (optional)	<code>list</code> of type <code>object-ref</code>	Specifies the mailboxes that are "CC:" recipients of the email message. The objects referenced in this list MUST be of type <code>email-address</code> .
<code>bcc_refs</code> (optional)	<code>list</code> of type <code>object-ref</code>	Specifies the mailboxes that are "BCC:" recipients of the email message. As per RFC 5322, this list may be empty, which should not be treated the same as the key being absent. The objects referenced in this list MUST be of type <code>email-address</code> .
<code>subject</code> (optional)	<code>string</code>	Specifies the subject of the email message.
<code>received_lines</code> (optional)	<code>list</code> of type <code>string</code>	Specifies one or more <i>Received</i> header fields that may be included in the email headers. List values MUST appear in the same order as present in the email message.
<code>additional_header_fields</code> (optional)	<code>dictionary</code>	Specifies any other header fields (except for <code>date</code> , <code>received_lines</code> , <code>content_type</code> , <code>from_ref</code> , <code>sender_ref</code> , <code>to_refs</code> , <code>cc_refs</code> , <code>bcc_refs</code> , and <code>subject</code>) found in the email message, as a dictionary.

		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 dictionary key name. For cases where a header field occurs exactly once, the corresponding value for the dictionary key MUST be a <code>string</code> . For cases where a header field occurs more than once, the corresponding value for the dictionary key MUST be a <code>list</code> of type <code>string</code> , where each <code>string</code> in the <code>list</code> represents a single value of the header field.
<code>body</code> (optional)	<code>string</code>	Specifies a <code>string</code> containing the email body. This field MAY only be used if <code>is_multipart</code> is false.
<code>body_multipart</code> (optional)	<code>list</code> of type <code>mime-part-type</code>	Specifies a list of the MIME parts that make up the email body. This property MAY only be used if <code>is_multipart</code> is true.
<code>raw_email_ref</code> (optional)	<code>object-ref</code>	Specifies the raw binary contents of the email message, including both the headers and body, as a reference to an Artifact Object. The object referenced in this field MUST be of type <code>artifact</code> .

2.6.2. Email MIME Component Type

Type Name: `mime-part-type`

Specifies a 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	<p>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).</p> <p>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.</p>
body_raw_ref (optional)	object-ref	<p>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.</p> <p>The object referenced in this field MUST be of type artifact or file. For use cases where conveying the actual data contained in the MIME part is of primary importance, the artifact SHOULD be used. Otherwise, for use cases where conveying metadata about the file-like properties of the MIME part is of primary importance, the file SHOULD be used.</p>
content_type (optional)	string	<p>Specifies the value of the “Content-Type” header field of the MIME part.</p> <p>Any additional “Content-Type” header field parameters such as charset SHOULD be included in this property.</p> <p>Example: <code>text/html; charset=UTF-8</code></p>
content_disposition (optional)	string	<p>Specifies the value of the “Content-Disposition” header field of the MIME part.</p>

2.6.3. Examples

Simple message

```

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

```

Simple message with Additional Header Properties

```

{
  "0": {
    "type": "email-address",
    "value": "joe@example.com",
    "display_name": "Joe Smith"
  },
  "1": {
    "type": "email-address",
    "value": "bob@example.net",
    "display_name": "Bob Smith"
  },
  "2": {
    "type": "email-message",
    "from_ref": "0",
    "to_refs": [
      "1"
    ],
    "date": "2004-04-19T12:22:23Z",
    "subject": "Did you see this?",
    "additional_header_fields": {
      "Reply-To": [
        "steve@example.com",
        "jane@example.com"
      ]
    }
  }
}

```

Complex MIME Message

```
{
  "0": {
    "type": "email-message",
    "is_multipart": true,
    "received_lines": [
      "from mail.wayneindustries.com ([1.2.3.4]) 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": "2016-06-19T14:20:40Z",
    "from_ref": "1",
    "to_refs": [
      "2"
    ],
    "cc_refs": [
      "3"
    ],
    "subject": "Check out this picture of the Riddler!",
    "additional_header_fields": {
      "Content-Disposition": "inline",
      "X-Mailer": "Mutt/1.5.23",
      "X-Originating-IP": "1.2.3.4"
    },
    "body_multipart": [
      {
        "content_type": "text/plain; charset=utf-8",
        "content_disposition": "inline",
        "body": "The Riddler is such a funny guy!"
      },
      {
        "content_type": "image/png",
        "content_disposition": "attachment; filename=\"riddler.png\"",
        "body_raw_ref": "4"
      },
      {
        "content_type": "application/zip",
        "content_disposition": "attachment; filename=\"riddler_pics.zip\"",
        "body_raw_ref": "5"
      }
    ],
    "1": {
      "type": "email-address",
      "value": "bwayne@wayneindustries.com",
      "display_name": "Bruce Wayne"
    },
    "2": {
      "type": "email-address",
      "value": "robin@batcave.com",

```

```

    "display_name": "Robin"
  },
  "3": {
    "type": "email-address",
    "value": "apennyworth@wayneindustries.com",
    "display_name": "Alfred Pennyworth"
  },
  "4": {
    "type": "artifact",
    "mime_type": "image/jpeg",
    "payload_bin": "VBORw0KGgoAAAANSUgAAADI== ...",
    "hashes": {
      "MD5": "69D0D97D02A03C43782DD571394E6869"
    }
  },
  "5": {
    "type": "file",
    "name": "riddler_pics.zip",
    "magic_number_hex": "504B0304",
    "hashes": {
      "MD5": "22A0FB8F3879FB569F8A3FF65850A82E"
    }
  }
}

```

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` OR `name_bin`.

2.7.1. Properties

Common Properties		
type, description, extended_properties		
File Object Specific Properties		
hashes, size, name, name_enc, name_bin, magic_number, 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 .
extended_properties (optional)	dictionary	<p>The File Object defines the following extensions. In addition to these, producers MAY create their own.</p> <p>ntfs-ext, raster-image-ext, pdf-ext, archive-ext, windows-pebinary-ext</p> <p>Dictionary keys MUST identify the extension type by name.</p> <p>The corresponding dictionary values MUST contain the contents of the extension instance.</p>
hashes (optional)	hashes-type	Specifies a dictionary of hashes for the file.
size (optional)	integer	Specifies the size of the file, in bytes, as a non-negative integer.
name (optional)	string	Specifies the name of the file.
name_enc (optional)	string	<p>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. 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.</p> <p>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</p>
name_bin (optional)	binary	Specifies the name of the file as a base64-encoded string. This property MUST NOT be specified in conjunction with the name property; only one of name OR name_bin may be used.

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	<p>Specifies the MIME type name specified for the file, e.g., “application/msword”.</p> <p>Whenever feasible, this value SHOULD be one of the values defined in the Template column in the IANA media type registry.</p> <p>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.</p>
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	<p>Specifies the parent directory of the file, as a reference to a Directory Object.</p> <p>The object referenced in this property MUST be of type directory.</p>
is_encrypted (optional)	boolean	Specifies whether the file is encrypted.
encryption_algorithm (optional)	open-vocab	<p>Specifies the name of the encryption algorithm used to encrypt the file. This is an open vocabulary and values SHOULD come from the encryption-algorithm-ov vocabulary.</p> <p>This property MUST NOT be used if is_encrypted is false or not included.</p>
decryption_key (optional)	string	<p>Specifies the decryption key used to decrypt the archive file.</p> <p>This property MUST NOT be used if</p>

		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. The object referenced in this property MUST be of type artifact .

2.7.2. Examples

Basic file with file system properties without observed encoding

```
{
  "0": {
    "type": "file",
    "hashes": {
      "MD5": "4472ea40dc71e5bb701574ea215a81a1"
    },
    "size": 25536,
    "name": "foo.dll"
  }
}
```

Basic file with file system properties with observed encoding

```
{
  "0": {
    "type": "file",
    "hashes": {
      "MD5": "66e2ea40dc71d5ba701574ea215a81f1"
    },
    "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": {
      "MD5": "A2FD2B3F4D5A1BD5E7D283299E01DCE9"
    },
    "parent_directory_ref": "0",
    "name": "qwerty.dll"
  }
}
```

2.7.3. 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 `extended_properties` dictionary **MUST** be `archive-ext`.

2.7.3.1. Properties

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

2.7.3.2. Example

Basic unencrypted ZIP Archive

```
{
  "0": {
    "type": "file",
    "hashes": {
      "MD5": "66e2ea40dc71d5ba701574ea215a81f1"
    }
  },
  "1": {
    "type": "file",
    "hashes": {
      "MD5": "22A0FB8F3879FB569F8A3FF65850A82E"
    }
  },
  "2": {
    "type": "file",
    "hashes": {
      "MD5": "8D98A25E9D0662B1F4CA3BF22D6F53E9"
    }
  },
  "3": {
    "type": "file",
    "hashes": {
      "MD5": "B365B9A80A06906FC9B400C06C33FF43"
    }
  },
  "mime_type": "application/zip",
  "extended_properties": {
    "archive-ext": {
      "contains_refs": [
        "0",
        "1",
        "2"
      ]
    }
  },
  "version": "5.0"
}
```

2.7.4. 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 **extended_properties** dictionary MUST be *ntfs-ext*.

2.7.4.1. Properties

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

2.7.4.2. Alternate Data Stream Type

Type Name: `alternate-data-stream`

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

2.7.4.2.1. Properties

Property Name	Type	Description
name (required)	<code>string</code>	Specifies the name of the alternate data stream.
hashes (optional)	<code>hashes-type</code>	Specifies a dictionary of hashes for the data contained in the alternate data stream.
size (optional)	<code>integer</code>	Specifies the size of the alternate data stream, in bytes, as a non-negative integer.

2.7.4.3. Example

NTFS File with a single alternate data stream

```
{
  "0": {
    "type": "file",
    "hashes": {
      "MD5": "B4D33B0C7306351B9ED96578465C5579"
    },
    "extended_properties": {
```


pdfid1 (optional)	string	Specifies the second file identifier found for the PDF file.
--------------------------	---------------	--

2.7.5.2. Example

Basic PDF file

```
{
  "0": {
    "type": "file",
    "hashes": {
      "MD5": "66e2ea40dc71d5ba701574ea215a81f1"
    },
    "extended_properties": {
      "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"
      },
    },
  },
}
```

2.7.6. Raster Image File Extension

Type Name: **raster-image-ext**

The Raster Image file extension specifies a default extension for capturing properties specific to image files. The key for this extension when used in the **extended_properties** dictionary **MUST** be *raster-image-ext*.

2.7.6.1. 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.
<code>bits_per_pixel</code> (optional)	<code>integer</code>	Specifies the sum of bits used for each color channel in the image in the image file, and thus the total number of pixels used for expressing the color depth of the image.
<code>image_compression_algorithm</code> (optional)	<code>string</code>	Specifies the name of the compression algorithm used to compress the image in the image file, if applicable.
<code>exif_tags</code> (optional)	<code>dictionary</code>	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 <code>integer</code> (for int* EXIF datatypes) or a <code>string</code> (for all other EXIF datatypes).

2.7.6.2. Example

Simple Image File with EXIF Data

```
{
  "0": {
    "type": "file",
    "name": "picture.jpg",
    "hashes": {
      "MD5": "B4D33B0C7306351B9ED96578465C5579"
    },
    "extended_properties": {
      "raster-image-ext": {
        "exif_tags": {
          "Make": "Nikon",
          "Model": "D7000",
          "XResolution": 4928,
          "YResolution": 3264
        }
      }
    }
  }
}
```

2.7.7. 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 `extended_properties` dictionary **MUST** be `windows-pebinary-ext`.

2.7.7.1. Properties

Property Name	Type	Description
<code>pe_type</code> (required)	<code>open-vocab</code>	Specifies the type of the PE binary. This is an open vocabulary and values SHOULD come from the <code>windows-pebinary-type-ov</code> vocabulary.
<code>imphash</code> (optional)	<code>string</code>	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
<code>machine_hex</code> (optional)	<code>hex</code>	Specifies the type of target machine.
<code>number_of_sections</code> (optional)	<code>integer</code>	Specifies the number of sections in the PE binary, as a non-negative integer.
<code>time_date_stamp</code> (optional)	<code>timestamp</code>	Specifies the time when the PE binary was created. The timestamp value MUST BE precise to the second.
<code>pointer_to_symbol_table_hex</code> (optional)	<code>hex</code>	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.
characteristics_hex (optional)	hex	Specifies the flags that indicate the file's characteristics.
file_header_hashes (optional)	hashes-type	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	Specifies metadata about the sections in the PE file.

2.7.7.2. Windows PE Binary Vocabulary

Type 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).

2.7.7.3. 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.

2.7.7.3.1. Properties

Property Name	Type	Description
---------------	------	-------------

magic_hex (optional)	hex	Specifies the unsigned integer 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.
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.
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.
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	integer	Specifies the major version number of the

(optional)		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, in bytes, of the image, including all headers, as the image is loaded in memory.
size_of_headers (optional)	integer	Specifies the combined size of the MS-DOS, PE header, and section headers, rounded up a multiple of the value specified in the file_alignment header.
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.
size_of_stack_commit (optional)	integer	Specifies the size of the stack to commit.
size_of_heap_reserve (optional)	integer	Specifies the size of the local heap space to reserve.
size_of_heap_commit	integer	Specifies the size of the local heap space

(optional)		to commit.
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-type	Specifies any hashes that were computed for the optional header.

2.7.7.4. Windows PE Section Type

Type Name: windows-pe-section

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

2.7.7.4.1. 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.
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-type	Specifies any hashes computed over the section.

2.7.7.5. Example

Typical EXE File

```
{
```

```
"0": {
  "type": "file",
  "hashes": {
    "MD5": "1C19FC56AEF2048C1CD3A5E67B099350"
  },
  "extended_properties": {
    "windows-pebinary-ext": {
      "pe_type": "exe",
      "file_header": {
        "machine_hex": "014c",
        "number_of_sections": 4,
        "time_date_stamp": "2016-01-22T12:31:12",
        "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_entrypoint": 4096,
        "base_of_code": 4096,
        "base_of_data": 8192,
        "image_base": 14548992,
        "section_alignment": 4096,
        "file_alignment": 4096,
        "major_operating_system_version": 1,
        "minor_operating_system_version": 0,
        "major_image_version": 0,
        "minor_image_version": 0,
        "major_subsystem_version": 4,
        "minor_subsystem_version": 0,
        "win32_version_value": "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_hex": "dffffdde"
      },
      "sections": [
```


		IPv4 address, the CIDR /32 suffix MAY be omitted. 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. The objects referenced in this list MUST be of type mac-addr .
belongs_to_refs (optional)	list of type object-ref	Specifies a reference to one or more autonomous systems (AS) that the IPv4 address belongs to. The objects referenced in this list MUST be of type as .

2.8.2. Examples

IPv4 Single Address

```
{
  "0": {
    "type": "ipv4-addr",
    "value": "1.2.3.4"
  }
}
```

IPv4 CIDR Block

```
{
  "0": {
    "type": "ipv4-addr",
    "value": "192.168.0.0/16"
  }
}
```

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, description, extended_properties		
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 <code>ipv6-addr</code> .
value (required)	string	Specifies one or more IPv6 addresses expressed using CIDR notation. 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. The objects referenced in this list MUST be of type <code>mac-addr</code> .
belongs_to_refs (optional)	list of type object-ref	Specifies a reference to one or more autonomous systems (AS) that the IPv6 address belongs to. The objects referenced in this list MUST be of type <code>as</code> .

2.9.2. 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:db8::/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, description, extended_properties		
MAC Address Object Specific Properties		
value		
Property Name	Type	Description
type (required)	string	The value of this property MUST be <code>mac-addr</code> .
value (required)	string	Specifies a single MAC address. The MAC address value MUST be represented as a single colon-delimited, lowercase MAC-48 address, which MUST include leading zeros for each octet. Example: 00:00:ab:cd:ef:01

2.10.2. 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		
<code>type, description, extended_properties</code>		
File Object Specific Properties		
<code>name</code>		
Property Name	Type	Description
<code>type</code> (required)	<code>string</code>	The value of this property MUST be <code>mutex</code> .
<code>name</code> (required)	<code>string</code>	Specifies the name of the mutex object.

2.11.2. Examples

Malware mutex

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

2.12. Network Traffic

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 amenable 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		
<code>type</code> , <code>description</code> , <code>extended_properties</code>		
Network Traffic Specific Properties		
<code>start</code> , <code>end</code> , <code>is_active</code> , <code>src_ref</code> , <code>dst_ref</code> , <code>src_port</code> , <code>dst_port</code> , <code>protocols</code> , <code>src_byte_count</code> , <code>dst_byte_count</code> , <code>src_packets</code> , <code>dst_packets</code> , <code>ipfix</code> , <code>src_payload_ref</code> , <code>dst_payload_ref</code> , <code>encapsulates_refs</code> , <code>encapsulated_by_ref</code>		
Property Name	Type	Description
<code>type</code> (required)	<code>string</code>	The value of this property MUST be <code>network-traffic</code> .
<code>extended_properties</code> (optional)	<code>dictionary</code>	<p>The Network Traffic Object defines the following extensions. In addition to these, producers MAY create their own.</p> <p><code>http-ext</code>, <code>tcp-ext</code>, <code>icmp-ext</code>, <code>socket-ext</code></p> <p>Dictionary keys MUST identify the extension type by name.</p> <p>The corresponding dictionary values MUST contain the contents of the extension instance.</p>
<code>start</code> (optional)	<code>timestamp</code>	Specifies the date/time the network traffic was initiated, if known.
<code>end</code> (optional)	<code>timestamp</code>	<p>Specifies the date/time the network traffic ended, if known.</p> <p>If the <code>is_active</code> property is true,</p>

		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	<p>Specifies the source of the network traffic, as a reference to one or more Observable Objects.</p> <p>The objects referenced in this list SHOULD be of type ipv4-addr, ipv6-addr, mac-addr or MAY be of type domain-name for cases where the IP address for a domain name is unknown.</p>
dst_ref (optional)	object-ref	<p>Specifies the destination of the network traffic, as a reference to one or more Observable Objects.</p> <p>The objects referenced in this list SHOULD be of type ipv4-addr, ipv6-addr, mac-addr or MAY be of type domain-name for cases where the IP address for a domain name is unknown.</p>
src_port (optional)	integer	Specifies the source port used in the network traffic, as an integer in the range of 0 - 65535.
dst_port (optional)	integer	Specifies the destination port used in the network traffic, as an integer in the range of 0 - 65535.
protocols (optional)	list of type string	<p>Specifies the protocols observed in the network traffic, along with their corresponding state.</p> <p>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.</p>

		<p>The protocol names SHOULD come from the service names defined in the Service Name column of the IANA Service Name and Port Number Registry. 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.</p> <p>Examples: ipv4, tcp, http ipv4, udp ipv6, tcp, http ipv6, tcp, ssl, https</p>
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.
src_payload_ref (optional)	object-ref	<p>Specifies the bytes sent from the source to the destination.</p> <p>The object referenced in this property</p>

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

2.12.2. Examples

Basic TCP Network Traffic

```

{
  "0": {
    "type": "ipv4-addr",
    "value": "1.2.3.4"
  },
  "1": {
    "type": "ipv4-addr",
    "value": "2.3.4.5"
  },
  "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": "1",
  "protocols": [
    "ipv4",
    "tcp",
    "http"
  ]
}
}
```

Network Traffic with Netflow Data

```
{
  "0": {
    "type": "ipv4-addr",
    "value": "192.168.43.9"
  },
  "1": {
    "type": "ipv4-addr",
    "value": "192.168.22.101"
  },
  "2": {
    "type": "network-traffic",
    "src_ref": "0",
    "dst_ref": "1",
    "protocols": [
      "ipv4",
      "tcp"
    ],
    "src_bytes": 147600,
    "src_packets": 100,
    "ipfix": {
      "minimumIpTotalLength": 32,
      "maximumIpTotalLength": 2556
    }
  }
}
```

Basic Tunneled Network Traffic

```
{
  "0": {
    "type": "ipv4-addr",
    "value": "172.16.12.34"
  },
  "1": {
    "type": "ipv4-addr",

```

```
    "value": "192.168.1.34"
  },
  "2": {
    "type": "ipv4-addr",
    "value": "192.168.1.54"
  },
  "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": "172.16.12.34"
  },
  "1": {
    "type": "ipv4-addr",
    "value": "192.168.1.34"
  },
  "2": {
```

```

    "type": "ipv4-addr",
    "value": "192.168.1.54"
  },
  "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"
  }
}

```

2.12.3. HTTP Extension

Type Name: `http-ext`

The HTTP extension specifies a default extension for capturing network traffic properties specific to HTTP. The key for this extension when used in the **extended_properties** dictionary MUST be *http-ext*.

2.12.3.1. Properties

Property Name	Type	Description
<code>request_method</code> (required)	<code>string</code>	Specifies the HTTP method portion of the HTTP request line, as a lowercase string.
<code>request_value</code> (required)	<code>string</code>	Specifies the value (typically a resource path) portion of the HTTP request line.
<code>request_version</code> (optional)	<code>string</code>	Specifies the HTTP version portion of the HTTP request line, as a lowercase string.
<code>request_header</code> (optional)	<code>dictionary</code>	<p>Specifies all of the HTTP header fields that may be found in the HTTP client request, as a dictionary.</p> <p>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 <code>string</code>.</p>
<code>message_body_length</code> (optional)	<code>integer</code>	Specifies the length of the HTTP message body, if included, in bytes.
<code>message_body_data_ref</code> (optional)	<code>object-ref</code>	<p>Specifies the data contained in the HTTP message body, if included.</p> <p>The object referenced in this property MUST be of type <code>artifact</code>.</p>

2.12.3.2. Example

```
{
  "0": {
    "type": "ipv4-addr",
    "value": "192.0.2.53"
  },
  "1": {
    "type": "network-traffic",
    "dst_ref": "0",
    "protocols": [
      "tcp",
      "http"
    ],
    "extended_properties": {
      "http-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"
        }
      }
    }
  }
}
```

2.12.4. 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 `extended_properties` dictionary MUST be `icmp-ext`.

2.12.4.1. Properties

Property Name	Type	Description
<code>icmp_type_hex</code> (required)	<code>hex</code>	Specifies the ICMP type byte.
<code>icmp_code_hex</code> (required)	<code>hex</code>	Specifies the ICMP code byte.

2.12.4.2. Example

```
{
  "0": {
    "type": "ipv4-addr",
    "value": "192.168.43.9"
  },
  "1": {
    "type": "ipv4-addr",
    "value": "8.8.8.8"
  },
  "2": {
    "type": "network-traffic",
    "src_ref": "0",
    "dst_ref": "1",
    "protocols": [
      "icmp"
    ],
    "extended_properties": {
      "icmp-ext": {
        "icmp_type_hex": "08",
        "icmp_code_hex": "00"
      }
    }
  }
}
```

2.12.5. 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 `extended_properties` dictionary **MUST** be `socket-ext`.

2.12.5.1. Properties

Property Name	Type	Description
<code>address_family</code> (required)	<code>socket-address-family-enum</code>	Specifies the address family (AF_*) that the socket is configured for.
<code>is_blocking</code> (optional)	<code>boolean</code>	Specifies whether the socket is in blocking mode.

<code>is_listening</code> (optional)	<code>boolean</code>	Specifies whether the socket is in listening mode.
<code>protocol_family</code> (optional)	<code>socket-protocol-family-enum</code>	Specifies the protocol family (PF_*) that the socket is configured for.
<code>options</code> (optional)	<code>dictionary</code>	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.
<code>socket_type</code> (optional)	<code>network-socket-type-enum</code>	Specifies the type of the socket.
<code>socket_descriptor</code> (optional)	<code>integer</code>	Specifies the socket file descriptor value associated with the socket, as a non-negative integer.
<code>socket_handle</code> (optional)	<code>integer</code>	Specifies the handle or inode value associated with the socket.

2.12.5.2. Network Socket Address Family Enumeration

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

An enumeration of network socket address family types.

Vocabulary Value	Description
<code>AF_UNSPEC</code>	Specifies an unspecified address family.
<code>AF_INET</code>	Specifies the IPv4 address family.
<code>AF_IPX</code>	Specifies the IPX (Novell Internet Protocol) address family.
<code>AF_APPLETALK</code>	Specifies the APPLETALK DDP address family.
<code>AF_NETBIOS</code>	Specifies the NETBIOS address family.

AF_INET6	Specifies the IPv6 address family.
AF_IRDA	Specifies IRDA sockets.
AF_BTH	Specifies BTH sockets.

2.12.5.3. Network Socket Protocol Family Enumeration

Type 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.

2.12.5.4. Network Socket Type Enumeration

Type 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.

2.12.5.5. Example

Basic Stream Socket

```
{
  "0": {
    "type": "ipv4-addr",
    "value": "192.168.1.2"
  },
  "1": {
    "type": "network-traffic",
```

```

"src_ref": "0",
"src_port": 223,
"protocols": [
  "ip",
  "tcp"
],
"extended_properties": {
  "socket": {
    "is_listening": "true",
    "address_family": "AF_INET",
    "protocol_family": "PF_INET",
    "socket_type": "SOCK_STREAM"
  }
}
}
}
}

```

2.12.6. 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 `extended_properties` dictionary **MUST** be `tcp-ext`.

2.12.6.1. Properties

Property Name	Type	Description
<code>src_flags_hex</code> (optional)	hex	<p>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).</p> <p>If the start and end times of the traffic are not specified, it is assumed that this represents the union of all TCP flags observed over the entirety of the network traffic being reported upon.</p>
<code>dst_flags_hex</code> (optional)	hex	Specifies the destination TCP flags, as the union of all TCP

		<p>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).</p> <p>If the start and end times of the traffic are not specified, it is assumed that this represents the union of all TCP flags observed over the entirety of the network traffic being reported upon.</p>
--	--	---

2.12.6.2. Example

```

{
  "0": {
    "type": "ipv4-addr",
    "value": "1.2.3.4"
  },
  "1": {
    "type": "ipv4-addr",
    "value": "2.3.4.5"
  },
  "2": {
    "type": "network-traffic",
    "src_ref": "0",
    "dst_ref": "1",
    "src_port": 3372,
    "dst_port": 80,
    "protocols": [
      "tcp"
    ],
    "extended_properties": {
      "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.

2.13.1. Properties

Common Properties		
<code>type, description, extended_properties</code>		
Process Object Specific Properties		
<code>is_hidden, pid, name, created, cwd, arguments, environment_variables, opened_connection_refs, creator_user_ref, binary_ref, parent_ref, child_refs</code>		
Property Name	Type	Description
<code>type</code> (required)	<code>string</code>	The value of this property MUST be <code>process</code> .
<code>extended_properties</code> (optional)	<code>dictionary</code>	<p>The Process Object defines the following extensions. In addition to these, producers MAY create their own.</p> <p><code>windows-process-ext</code>, <code>windows-service-ext</code></p> <p>Dictionary keys MUST identify the extension type by name.</p> <p>The corresponding dictionary values MUST contain the contents of the extension instance.</p>
<code>is_hidden</code> (optional)	<code>boolean</code>	Specifies whether the process is hidden.
<code>pid</code> (optional)	<code>integer</code>	Specifies the Process ID, or PID, of the process.
<code>name</code> (optional)	<code>string</code>	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 should be captured separately as a string.
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 Connection Objects. The objects referenced in this list MUST be of type network-connection .
creator_user_ref (optional)	object-ref	Specifies the user that created the process, as a reference to a User Account Object.

		The object referenced in this property MUST be of type <code>user-account</code> .
<code>binary_ref</code> (optional)	<code>object-ref</code>	Specifies the executable binary that was executed as the process, as a reference to a File Object. The object referenced in this property MUST be of type <code>file</code> .
<code>parent_ref</code> (optional)	<code>object-ref</code>	Specifies the other process that spawned (i.e. is the parent of) this one, as represented by a Process Object. The object referenced in this property MUST be of type <code>process</code> .
<code>child_refs</code> (optional)	<code>list of type object-ref</code>	Specifies the other processes that were spawned by (i.e. children of) this process, as a reference to one or more other Process Objects. The objects referenced in this list MUST be of type <code>process</code> .

2.13.2. Examples

```
{
  "0": {
    "type": "file",
    "hashes": {
      "MD5": "B4D33B0C7306351B9ED96578465C5579"
    },
  },
  "1": {
    "type": "process",
```

```

    "pid": 1221,
    "name": "gedit-bin",
    "created": "2016-01-20T14:11:25.55Z",
    "arguments" :[
        "--new-window"
    ],
    "binary_ref": "0"
}
}
}

```

2.13.3. 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 `extended_properties` dictionary MUST be `windows-process-ext`.

2.13.3.1. Properties

Property Name	Type	Description
<code>aslr_enabled</code> (optional)	<code>boolean</code>	Specifies whether Address Space Layout Randomization (ASLR) is enabled for the process.
<code>dep_enabled</code> (optional)	<code>boolean</code>	Specifies whether Data Execution Prevention (DEP) is enabled for the process.
<code>priority</code> (optional)	<code>string</code>	Specifies the current priority class of the process in Windows. This value SHOULD be a string that ends in “_CLASS”.
<code>owner_sid</code> (optional)	<code>string</code>	Specifies the Security ID (SID) value of the owner of the process.
<code>window_title</code> (optional)	<code>string</code>	Specifies the title of the main window of the process.
<code>startup_info</code> (optional)	<code>dictionary</code>	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. For example., given a name of 'lpDesktop' the corresponding key would be 'lpDesktop'.
--	--	--

2.13.3.2. Example

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

2.13.4. 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 `extended_properties` dictionary **MUST** be `windows-service-ext`.

2.13.4.1. Properties

Property Name	Type	Description
<code>service_name</code> (required)	<code>string</code>	Specifies the name of the service.
<code>descriptions</code> (optional)	<code>list</code> of type <code>string</code>	Specifies the descriptions defined for the service.

display_name (optional)	<code>string</code>	Specifies the displayed name of the service in Windows GUI controls.
group_name (optional)	<code>string</code>	Specifies the name of the load ordering group of which the service is a member.
start_command_line (optional)	<code>string</code>	Specifies the full command line used to start the service.
start_type (optional)	<code>windows-service-start-enum</code>	Specifies the start options defined for the service.
service_dll_refs (optional)	<code>list</code> of type <code>object-ref</code>	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 <code>file</code> .
service_type (optional)	<code>windows-service-enum</code>	Specifies the type of the service.
service_status (optional)	<code>windows-service-status-enum</code>	Specifies the current status of the service.

2.13.4.2. Windows Service Start Type Enumeration

Type Name: `windows-service-start-enumeration`

An enumeration of Windows service start types.

Vocabulary Value	Description
<code>SERVICE_AUTO_START</code>	A service started automatically by the service control manager during system startup.
<code>SERVICE_BOOT_START</code>	A device driver started by the system loader. This value is valid only for driver services.
<code>SERVICE_DEMAND_START</code>	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.

2.13.4.3. Windows Service Type Enumeration

Type Name: `windows-service-enum`

An enumeration of Windows service start 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.

2.13.4.4. Window Service Status Enumeration

Type 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.

2.13.4.5. Example

```
{
  "0": {
    "type": "file",
    "hashes": {
      "MD5": "B4D33B0C7306351B9ED96578465C5579"
    },
    "name": "sirvizio.exe"
  },
  "1": {
    "type": "process",
    "pid": 2217,
    "name": "sirvizio",
    "binary_ref": "0",
    "extended_properties": {
      "windows-service-ext": {
        "display_name": "Sirvizio",
        "start_command_line": "C:\\Windows\\System32\\sirvizio.exe /s",
        "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, description, extended_properties		
Software Object Specific Properties		
name, language, vendor, version, swid		
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 . While the CPE dictionary does not contain entries for <i>all</i> software, whenever it <i>does</i> contain an identifier for a given instance of software, this property SHOULD be present.
language (optional)	string	Specifies the language of the software. The value of this property MUST be an ISO 639-2 language code.
vendor (optional)	string	Specifies the name of the vendor of the software.
version (optional)	string	Specifies the version of the software.

2.14.2. 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, description, extended_properties		
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.

2.15.2. Examples

Basic URL

```
{
  "0": {
    "type": "url",
    "value": "https://wayneindustries.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, description, extended_properties
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 <code>user-account</code> .
extended_properties (optional)	dictionary	<p>The User Account Object defines the following extensions. In addition to these, producers MAY create their own.</p> <p><code>unix-account-ext</code></p> <p>Dictionary keys MUST identify the extension type by name.</p> <p>The corresponding dictionary values MUST contain the contents of the extension instance.</p>
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 <code>user_id</code> property should be populated with whatever field is the unique identifier for the system the account is a member of. For example, on UNIX systems it would be populated with the UID.
account_login (optional)	string	<p>Specifies the account login string, used in cases where the <code>user_id</code> property specifies something other than what a user would type when they login.</p> <p>For example, in the case of a Unix account with <code>user_id</code> 0, the <code>account_login</code> might be "root".</p>
account_type (optional)	open-vocab	<p>Specifies the type of the account.</p> <p>This is an open vocabulary and values SHOULD come from the <code>account-type-ov</code> vocabulary.</p>

display_name (optional)	string	Specifies the display name of the account, to be shown in user interfaces, if applicable. 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

Type 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.
kavi	Specifies an OASIS Kavi account.

2.16.3. Examples

Basic Unix Account

```
{
  "0": {
    "type": "user-account",
    "user_id": "1001",
    "account_login": "bwayne",
    "account_type": "unix",
    "display_name": "Bruce Wayne",
    "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"
}
}
}

```

2.16.4. 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 `extended_properties` dictionary MUST be `unix`.

2.16.4.1. Properties

Property Name	Type	Description
<code>gid</code> (optional)	<code>number</code>	Specifies the primary group ID of the account.
<code>groups</code> (optional)	<code>list</code> of type <code>string</code>	Specifies a list of names of groups that the account is a member of.
<code>home_dir</code> (optional)	<code>string</code>	Specifies the home directory of the account.
<code>shell</code> (optional)	<code>string</code>	Specifies the account's command shell.

2.16.4.2. Example

```

{
  "0": {
    "type": "user-account",
    "user_id": "1001",
    "user_login": "bwayne",
    "account_type": "unix",
    "display_name": "Bruce Wayne",
    "is_service_account": false,
    "is_privileged": false,
    "can_escalate_privs": true,

```

```

    "extended_properties": {
      "unix-account-ext": {
        "gid": 1001,
        "groups": ["wheel"],
        "home_dir": "/home/bwayne",
        "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		
<code>type, description, extended_properties</code>		
File Object Specific Properties		
<code>key, values, modified, creator_ref, number_of_subkeys</code>		
Property Name	Type	Description
type (required)	<code>string</code>	The value of this property MUST be <code>windows-registry-key</code> .
key (required)	<code>string</code>	Specifies the full registry key including the hive. 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)	<code>list</code> of type <code>windows-registry-value-type</code>	Specifies the values found under the registry key.

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

2.17.2. Windows Registry Value Type

Type Name: `windows-registry-value-type`

2.17.2.1. Properties

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

2.17.3. Registry Datatype Enumeration

Type Name: `windows-registry-datatype-enum`

An enumeration of Windows registry data types.

Vocabulary Value	Description
<code>REG_NONE</code>	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.

2.17.4. 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",  
        "value": "qwerty",  
        "data_type": "REG_SZ"  
      },  
      {  
        "name": "Bar",  
        "value": "42",  
        "data_type": "REG_DWORD"  
      }  
    ]  
  }  
}
```

2.18. X509 Certificate Object

Type Name: `x509-certificate`

The X509 Certificate Object represents the properties of an X.509 certificate, as defined by [ITU recommendation X.509](#).

2.18.1. Properties

Common Properties		
type, description, extended_properties		
File Object Specific Properties		
is_self_signed, hashes, version, serial_number, signature_algorithm, issuer, validity_not_before, validity_not_after, subject, 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-type	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 (optional)	integer	Specifies the exponent portion of the 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. X509 v3 Extensions Type

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

2.18.2.1. Properties

Property Name	Type	Description
basic_constraints (optional)	<code>string</code>	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 a non-negative value can be included. Also equivalent to the object ID (OID) value of 2.5.29.19.
name_constraints (optional)	<code>string</code>	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)	<code>string</code>	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)	<code>string</code>	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)	<code>string</code>	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)	<code>string</code>	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.
----------------------------	--------	---

2.18.3. 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",
    "subject": "C=US, ST=Maryland, L=Pasadena, O=Brent Baccala, OU=FreeSoft, CN=www.freesoft.org/emailAddress=baccala@freesoft.org",
    "validity_not_before": "2016-03-12T12:00:00Z",
    "validity_not_after": "2016-08-21T12:00:00Z"
  }
}
```

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 Part 3a Section 3 (Cyber Observable Objects).

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 2.1, 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 Part 3a Section 3 (Cyber Observable Objects).

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 2.20, Network Traffic Object.

4. Appendix A. Acknowledgments

Cybox Subcommittee Chairs:

Trey Darley (trey@kingfisherops.com), Kingfisher Operations, sprl

Ivan Kirillov, (ikirillov@mitre.org), MITRE Corporation

Contributors

The following individuals made substantial contributions in the form of normative text and proofing of this specification, their contributions are gratefully acknowledged:

- Bret Jordan, Blue Coat Systems, Inc.
- Terry MacDonald, Cosive
- Jane Ginn, Cyber Threat Intelligence Network, Inc. (CTIN)
- Trey Darley, Kingfisher Operations, sprl
- Jason Keirstead, IBM
- Allan Thomson, LookingGlass Cyber
- John-Mark Gurney, New Context Services, Inc.
- Christian Hunt, New Context Services, Inc.
- Greg Back, MITRE Corporation
- Sean Barnum, MITRE Corporation
- Ivan Kirillov, MITRE Corporation
- John Wunder, MITRE Corporation
- Dave Cridland, Surevine

5. Appendix B. Revision History