CybOX 3.0 Specification
Pre-Draft

Version 0.1

Current Status/Intent

This document serves to gain consensus on pre-draft concepts of CybOX 3.0. Please feel free to poke holes and comment

Feature Roadmap

Capability	MVP for 3.0	Status	Consensus
CybOX Core			
Relationships	**Yes**	TBD	No consensus
Patterning	**Yes**	Development	No consensus
Object Type	**Yes**	Proposal	No consensus
• Object Extensions	**Yes**		
Actions	TBD	Proposal	Not discussed
Vocabularies	**Yes**	TBD	No consensus
• Hashes	**Yes**	Proposal	*Nearing consensus*
Object Field Properties	TBD	TBD	Not discussed
• Obfuscation Properties	TBD	TBD	No consensus
• Observed Encoding	TBD	TBD	Not discussed
CybOX Objects			
API Object	TBD	TBD	Not discussed
Artifact Object	TBD	Proposal	Not discussed
AS Object	**Yes**	Proposal	Not discussed

Code Object	TBD	TBD	Not discussed
Custom Object	TBD	TBD	Not discussed
Device Object	**Yes**	Proposal	Not discussed
Disk Partition	TBD	TBD	Not discussed
DNS Record	**Yes**	TBD	Not discussed
Domain Name	**Yes**	Proposal	Not discussed
Email Address	**Yes**	Draft Complete	**Consensus reached**
File	**Yes**	Draft Complete	**Consensus reached**
• Metadata Extension	**Yes**	Draft Complete	**Consensus reached**
• EXT3 File Extension	TBD	Proposal	Not discussed
• NTFS File Extension	**Yes**	Proposal	No consensus
• Image File Extension	TBD	Proposal	Not discussed
• PDF File Extension	**Yes**	Proposal	Not discussed
• Archive File Extension	TBD	Proposal	Not discussed
• PE Binary File Extension	**Yes**	TBD	Not discussed
GUI	TBD	TBD	Not discussed
• GUI Window Extension	TBD	TBD	Not discussed
Hostname	**Yes**	Proposal	No consensus
IPv4 Address	**Yes**	Draft Complete	**Consensus reached**
IPv6 Address	**Yes**	Draft Complete	**Consensus reached**
Link	**Yes**	TBD	Not discussed
MAC Address	**Yes**	Draft Complete	**Consensus reached**
Message	**Yes**	TBD	Not discussed
• Email Message Extension	**Yes**	TBD	Not discussed
• SMS Message Extension	TBD	TBD	Not discussed
Memory	**Yes**	TBD	Not discussed

Mutex	**Yes**	Proposal	Not discussed
Win Mutex Extension	TBD	TBD	Not discussed
Network Connection	**Yes**	Proposal	Not discussed
• Extensions	TBD	TBD	Not discussed
Network Socket	TBD	TBD	Not discussed
Pipe	TBD	TBD	Not discussed
• Win Pipe Extension	TBD	TBD	Not discussed
Port	**Yes**	TBD	Not discussed
Process	**Yes**	TBD	Not discussed
• Win Process Extension	**Yes**	TBD	Not discussed
Product	TBD	TBD	Not discussed
Semaphore	TBD	TBD	Not discussed
System	**Yes**	TBD	Not discussed
• Win System Extension	TBD	TBD	Not discussed
URL	**Yes**	Proposal	No consensus
User Account	TBD	TBD	Not discussed
User Session	TBD	TBD	Not discussed
WHOIS	**Yes**	TBD	Not discussed
Win Driver	**Yes**	TBD	Not discussed
Win Handle	**Yes**	TBD	Not discussed
Win Kernel Hook	**Yes**	TBD	Not discussed
Win Network Share	**Yes**	TBD	Not discussed
Win Registry Key	**Yes**	Proposal	*Nearing consensus*
Win Service	**Yes**	TBD	Not discussed
Win Semaphore	TBD	TBD	Not discussed
Win Thread	TBD	TBD	Not discussed

X509 Certificate	**Yes**	TBD	Not discussed

Overview

CybOX 3.0 is a refactored version of the Cyber Observable eXpression (™) language originally developed by MITRE, a U.S. Federally Funded Research & Development Center (FFRDC) supported by the U.S. Department of Homeland Security (DHS). CybOX 3.0 has been refactored in conjunction with its sister standards, the Structured Threat Information eXpression (STIX) and the Trusted Automated Exchange for Indicator Information (TAXII) under the auspices of the Cyber Threat Intelligence Technical Committee (CTI-TC) of the Organization for the Advancement of Structured Information Systems (OASIS).

OASIS is a nonprofit consortium that drives the development, convergence and adoption of open standards for the global information society. OASIS promotes industry consensus and produces worldwide standards for security, Internet of Things (IoT), cloud computing, energy, content technologies, emergency management, and other areas. The consortium has more than 5,000 participants representing over 600 organizations and individual members in more than 65 countries.

CybOX objects are designed to fulfill an important need in the CTI ecosystem. They provide a structured language for the specification, capture, characterization and communication of events or stateful properties that are observed in the cyber domain. CybOX objects are used to describe a fact in many different functional domains, including but not limited to:

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

A CybOX object describes the facts about a single event or Object. This may be the information about a file, a file was created by a process, a network connection existed between two IPs, or that there was a failed login attempt. An object describes **WHAT** happened, but not the who, when, or why.

This document provides the details on the refactored objects for Version 3.0 and a roadmap for subsequent 3.x objects.

Contributors

Ivan Kirillov
Trey Darley
Sean Barnum
John-Mark Gurney
Jane Ginn
Jason Keirstead

Document Conventions

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 - `package`
    - property names are in bold style - **`created_at`**
    - literals are in green with a green background - `IP Watchlist`
- In property tables, if the property is inherited, its row has a light grey background. If the property is being redefined in some way, then the background is dark grey.

All type names, property names and literals are in lower-case. Words in property names are separated with an underscore (_), while words in type names and string enumerations are separated with a dash (-).

Examples are included, using the JSON serialization. They are in `Consolas` 9 pt font, with black text and a light blue background. JSON examples have a 2 character space indentation.

Features for Future Releases

Table of Contents

1. CybOX Core

CybOX Base (`cybox-base`)

The “base” type that all other CybOX types inherit from.

Property Name	Type	Description
id	`string`	
type	`string`	
spec_version	`string`	Indicates the version of the CybOX specification that the entity conforms to.
created_time	`timestamp`	Indicates the date/time that the CybOX entity was created.

Object (`cybox-object`)

Inherits From	Inherited Properties	
`cybox-base`	**type, id, spec_version, created_time**	
Property Name	**Type**	**Description**
type (required)	`string`	Indicates that this object is a CybOX Object. The value of this field **MUST** be a valid CybOX object type.
extended_properties (optional)	dictionary (or a dict of dicts, in the case of multiple extended_property sets)	The set of extended properties specified for the Object.

Action (`cybox-action`)

Inherits From	Inherited Properties
`cybox-base`	**type, id, spec_version, created_time**

Property Name	Type	Description
type (required)	`string`	Indicates that this object is a CybOX Action. The value of this field **MUST** be `cybox-action`

2. CybOX Common

Hashes (`hashes`)

The Hashes class represents 1 or more hashes, as a dictionary. Accordingly, the name of each hashing algorithm **MUST** be specified as a key in the dictionary. This name **MUST** either be one of the values defined in the `hash-algo-enum` OR a custom value prepended with "x_" (e.g., "x_custom_hash"). The value of the key **MUST** be the hash as a lowercase hexadecimal string.

As an example, the following represents how an MD5 hash and a custom hash, foo_hash, would be captured in a single `hashes` dictionary:

```
{
    "md5": "3773a88f65a5e780c8dff9cdc3a056f3",
    "x_foo_hash": "aaaabbbbccccddddeeeeffff0123457890"
},
```

Hashing Algorithm Enumeration (`hash-algo-enum`)

An enumeration of hashing algorithms.

Value	Description
md5	Specifies the MD5 message digest algorithm. (As the security of MD5 has been compromised, it **SHOULD NOT** be used.)
md6	Specifies the MD6 message digest algorithm.
ripemd-160	Specifies the RIPEMD-160 (RACE Integrity Primitives Evaluation Message Digest) cryptographic hash function.
sha-1	Specifies the SHA-1 (secure-hash algorithm 1) cryptographic hash function.

sha-224	Specifies the SHA-224 cryptographic hash function (part of the SHA-2 family).
sha-256	Specifies the SHA-256 cryptographic hash function (part of the SHA-2 family).
sha-384	Specifies the SHA-384 cryptographic hash function (part of the SHA-2 family).
sha-512	Specifies the SHA-512 cryptographic hash function (part of the SHA-2 family).
sha3-224	Specifies the SHA3-224 cryptographic hash function.
sha3-256	Specifies the SHA3-256 cryptographic hash function.
sha3-384	Specifies the SHA3-384 cryptographic hash function.
sha3-512	Specifies the SHA3-512 cryptographic hash function.
ssdeep	Specifies the ssdeep fuzzy hashing algorithm.
whirlpool	Specifies the whirlpool cryptographic hash function.

3. CybOX Vocabularies

4. Objects

4.1 Included Objects

The following list represents the Objects and their corresponding extensions that will be included in CybOX 3.0.

- API
- Artifact
- AS
- Code
- Custom
- Device
- Disk Partition
- DNS Record
- Domain Name
- Email Address
- File
    - File Metadata Extension
    - UFS File Extension

    - NTFS File Extension
    - Image File Extension
    - PDF File Extension
    - Archive File Extension
    - PE Binary File Extension
- GUI
    - GUI Window Extension
- Hostname
- IPv4 Address
- IPv6 Address
- Link
- MAC Address
- Message
    - Email Message Extension
    - SMS Message Extension
- Memory
- Mutex
    - Windows Mutex Extension
- Network Connection
    - HTTP Session Extension
    - DNS Query Extension
    - Network Flow Extension
- Network Socket
- Pipe
    - Windows Pipe Extension
- Port
- Process
    - Windows Process Extension
- Product
- Semaphore
- System
    - Windows System Extension
- URI
- User Account
- User Session
- Whois
- Windows Driver
- Windows Executable File
- Windows Handle
- Windows Kernel Hook
- Windows Network Share
- Windows Registry Key
- Windows Service

- Windows Semaphore
- Windows Thread
- X509 Certificate

4.1.1 Other Objects

The following list represents the Objects that were included in previous CybOX releases, but will NOT be included in CybOX 3.0. **Note**: this is not say that these Objects are deprecated, rather, it is likely that many of these Objects will be refactored and included in future CybOX minor releases.

- Account
- ARP Cache
- DNS Cache
- GUI Dialog Box
- Linux Package
- Network Packet
- Network Route Entry
- Network Route
- Network Subnet
- Unix File
- Unix Network Route
- Unix Pipe
- Unix Process
- Unix User Account
- Unix Volume
- URL History
- Volume
- Windows Computer Account
- Windows Critical Section
- Windows Event Log
- Windows Filemapping
- Windows File
- Windows Hook
- Windows Kernel
- Windows Mailslot
- Windows Memory Page Region
- Windows Network Route Entry
- Windows Prefetch
- Windows System Restore
- Windows Task
- Windows User Account
- Windows Volume
- Windows Waitable Timer

4.2 Object Extensions

Each CybOX Object may have one to many *extensions* defined for it. Such extensions are intended to capture properties that do not belong on the base Object type, including those pertaining to a particular domain or sub-type of the Object. Accordingly, each extension may be used only in conjunction with the base Object that is defined for, and is intended to represent an exclusive set of properties; that is, extensions should, ideally, not overlap with each other with regards to the context and properties that they characterize. The sole exception to this principle is with regards to extensions for related entities, for example, different versions of the same file system. In such cases, overlapping extension properties are permitted.

4.2.1 Mutual Exclusivity

It would be semantically inaccurate to use certain extensions together in an Object instance, and therefore extensions MAY be mutual exclusive with each other. Such mutual exclusivity is defined between the extensions for a particular Object; for example, the Image and PDF File Object extensions are mutually exclusive with each other, as a file cannot be an image and a PDF document at the same time.

4.2.2 Extension Hierarchies

Extensions MAY be defined as subclasses of other extensions, as needed, for cases where it makes sense for an extension to inherit the properties of its parent. For example, if a particular file system version builds upon the previous version, it would be logical for it to be directly derived from this version. Parent extensions are mutually exclusive with their children in an Object instance; that is, when specifying subclasses of extensions, the parent extension MUST NOT also be specified.

4.2.3 Implementation

Given the mutually exclusive nature of each Object extension, this entails that each such extension can be defined at most once on each Object. Accordingly, in an Object instance, this is specified through the **`extended_properties`** field, which is of type *dictionary* and inherited from the `object` class defined in CybOX Core. Note that this means that each extension is specified through a corresponding key in the **`extended_properties`** field. For example, when specified in a File Object instance, the file metadata extension would be specified using the key value of **`metadata`**.

4.2.4 Custom Extensions

Custom extensions, i.e. those not included with the default set as specified for the Object, can likewise be included in the **`extended_properties`** field. The key value for such extensions MUST be prepended with “custom_” to indicate that they are not part of the default set; e.g., the key value for a custom “foo” extension would be **`custom_foo`**.

4.2.5 Example

The following is an example of how a File with a set of extensions (captured in the **`extended_properties`** field) would be represented using JSON.

```
{
    "type": "file-object",
    "hashes": {
        "md5": "3773a88f65a5e780c8dff9cdc3a056f3"
    },
    "size": 25537,
    "extended_properties": {
        "metadata": {"mime_type": "vnd.microsoft.portable-executable"},
        "ext3": {"inode": "34483923"},
        "pebinary": {"exports": [{"name": "foo_app"}]}
    }
}
```

4.3 IPv4 Address Object (`ipv4addr-object`)

The IPv4 Address Object represents a single IPv4 address OR a range of IPv4 addresses, specified using CIDR notation.

Properties

Inherits From	Inherited Properties		
`cybox-object`	**all**		
Property Name	**Type**	**Description**	**Pattern**
type (inherited from `cybox-object`)	`string`	Indicates that this object is a CybOX IPv4 Address Object. The value of this field **MUST** be `ipv4addr-object`	n/a
value	`string`	Specifies a single IPv4 address value OR a range of addresses specified using CIDR notation. If a single IPv4 address is specified, this is implicitly equivalent to a /32 CIDR block.	^(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$

4.4 IPv6 Address Object (`ipv6addr-object`)

The IPv6 Address Object represents a single IPv6 address OR a range of IPv6 addresses, specified using CIDR notation.

Properties

Inherits From	Inherited Properties		
`cybox-object`	**all**		
Property Name	**Type**	**Description**	**Pattern**
type (inherited from `cybox-object`)	`string`	Indicates that this object is a CybOX IPv6 Address Object. The value of this field **MUST** be `ipv6addr-object`	n/a
value	`string`	Specifies a single IPv6 address value OR a range of addresses specified using CIDR	^(?:[A-F0-9]{1,4}:){6}(?:[A-F0-9]{1,4}:[

		notation. If a single IPv6 address is specified, this is implicitly equivalent to a /128 CIDR block.	A-F0-9]{1,4}|(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?))$

4.5 MAC Address Object (`macaddr-object`)

The MAC Address Object represents a single media access control (MAC) address.

Properties

Inherits From	Inherited Properties		
`cybox-object`	**all**		
Property Name	**Type**	**Description**	**Pattern**
type (inherited from `cybox-object`)	`string`	Indicates that this object is a CybOX MAC Address Object. The value of this field **MUST** be `macaddr-object`	n/a
value	`string`	Specifies a single hyphen delimited, lowercase MAC-48 address value, including leading zeros.	^([0-9a-f]{2}[-]){5}([0-9a-f]{2})$

4.6 Email Address Object (`emailaddr-object`)

The Email Address Object represents a single email address.

Properties

Inherits From	Inherited Properties		
`cybox-object`	**all**		
Property Name	**Type**	**Description**	**Pattern**

type (inherited from `cybox-object`)	`string`	Indicates that this object is a CybOX Email Address Object. The value of this field **MUST** be `emailaddr-object`	n/a
value	`string`	Specifies a single full email address value (including local and domain components).	<insert regex here>

4.7 File Object (`file-object`)

The File Object represents the properties of a file OR directory.

Properties

The base File Object type that defines the set of properties common to any file.

Inherits From	Inherited Properties		
`cybox-object`	**all**		
Property Name	**Type**	**Description**	
type (inherited from `cybox-object`)	`string`	Indicates that this object is a CybOX File Object. The value of this field **MUST** be `file-object`	
hashes	`hashes`	Specifies a list of hashes for the file.	
size	`integer`	Specifies the size of the file, in bytes.	
mime_type	`string`	Specifies the MIME type name specified for the file, e.g., “msword”. This value MUST be one of the values found in the IANA media type registry, located at [IANA]..	
file_system_properties	`file-system-properties`	Specifies the basic properties associated with the storage of the file on a file system.	

4.7.2 File System Properties Object (`file-system-properties`)

The set of basic properties common to the storage of files on most file systems.

Properties

Property Name	Type	Description
type (required)	`string`	Indicates that this object is a CybOX File System Properties object. The value of this field **MUST** be `file-system-properties`
is_directory (required)	`string`	Specifies whether the file object instance represents a directory (if `true`) or a file (if `false`).
file_name	`string`	Specifies the name of the file, including its extension (if known), but excluding its path. This field MAY be included only if the **is_directory** property is set to `false`.
file_path	`file-path`	Specifies the path to the file on the file system, excluding its name and extension. If this property is included without the **file_name** field, this property specifies a path to a directory.
modified_time	`timestamp`	Specifies the date/time the file was last written to/modified.
accessed_time	`timestamp`	Specifies the date/time the file was last accessed.
created_time	`timestamp`	Specifies the date/time the file was created.

4.7.3 File Path Object (`file-path`)

A representation of a delimited file or directory path.

Properties

Property Name	Type	Description
type (required)	`string`	Indicates that this object is a CybOX File System Properties object. The value of this field **MUST** be `file-path`
delimiter	`string`	Specifies the delimiter used in the file path string captured in the **components** property.
components	`array` of type `string`	Specifies a list of strings that represent the components of the file path string, when split

		using the delimiter specified in the **delimiter** property.

4.7.4 Default Extensions

The following represents a list of default extensions and their associated types specified for the File Object.

4.7.4.1 File Metadata (`file-metadata-extension`)

The File Metadata extension specifies a default extension for capturing general classes of file metadata. The key for this extension when used in the **`extended_properties`** dictionary MUST be *metadata*.

Properties

Property Name	Type	Description
magic_number	`string`	Specifies the hexadecimal constant ("magic number") associated with a specific file format that corresponds to the file, if applicable.
has_mismatch	`boolean`	Indicates that there is a mismatch between one or more stated and reported properties of the file. For example, a mismatch between the MIME type of the file and its file extension.
mismatch_type	`file-mismatch-enum`	Specifies the specific type of file mismatch that was found. This field MUST be used if the **has_mismatch** property is set to `true`.

4.7.4.1.1 File Mismatch Enumeration (`file-mismatch-enum`)

The File Mismatch Enumeration represents an enumeration of types of file mismatches.

Value	Description
extension/type	A mismatch between the MIME type reported for the file and its file extension. For example, if the reported MIME type (as captured in the **mime_type** property) for the file is 'vnd.microsoft.portable-executable' and the file extension (as captured in the **file_name** property) is 'txt'.
magic/extension	A mismatch between the magic number reported for the file and its file extension. For example, if the reported magic number (as captured in the

	magic_number property) for the file is '25504446', indicating a PDF file, and the file extension (as captured in the **file_name** property) is 'txt'.
magic/type	A mismatch between the reported MIME type and magic number for the file. For example, if the reported MIME type (as captured in the **mime_type** property) for the file is 'JPEG' and the reported magic number is '424D' (as captured in the **magic_number** property, indicating a bitmap file).

4.7.4.2 EXT3 File (`ext3-file-extension`)

The EXT3 File extension specifies a default extension for capturing properties specific to the storage of the file on the EXT3 file system. The key for this extension when used in the **`extended_properties`** dictionary MUST be *ext3*.

Properties

Property Name	Type	Description
inode	`string`	Specifies the index node (inode) value assigned to the file; this value is commonly, but not always, an integer.

4.7.4.3 NTFS File (`ntfs-file-extension`)

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*.

Properties

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

4.7.4.3.1 AlternateDataStream (`alternate-data-stream`)

The AlternateDataStream object represents an NTFS alternate data stream.

Properties

Property Name	Type	Description
type (required)	`string`	Indicates that this object is a CybOX NTFS

		Alternate Data Stream object. The value of this field **MUST** be `alternate-data-stream`
hashes	`hashes`	Specifies a list of hashes for the data contained in the alternate data stream.
name	`string`	Specifies the name of the alternate data stream.
size	`integer`	Specifies the size of the alternate data stream, in bytes.

4.7.4.4 Image File (`image-file-extension`)

The 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 *image*.

Properties

Property Name	Type	Description
image_is_compressed	`boolean`	Specifies whether the image in the image file is compressed.
image_height	`integer`	Specifies the height of the image in the image file, in pixels.
image_width	`integer`	Specifies the width of the image in the image file, in pixels.
bits_per_pixel	`integer`	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.
image_compression_algorithm	`string`	Specifies the name of the compression algorithm used to compress the image in the image file, if applicable.

4.7.4.5 PDF File (`pdf-file-extension`)

The PDF File extension specifies a default extension for capturing properties specific to PDF files. The key for this extension when used in the **`extended_properties`** dictionary MUST be *pdf*.

Properties

Property Name	Type	Description
version	`double`	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”.
metadata	`pdf-file-metadata`	Specifies metadata associated with the PDF file.

4.7.4.5.1 PDF File Metadata (`pdf-file-metadata`)

The PDF File Metadata object captures metadata regarding a PDF file.

Properties

Property Name	Type	Description
is_encrypted	`boolean`	Specifies whether the PDF file is encrypted.
is_optimized	`boolean`	Specifies whether the PDF file has been optimized.
document_information_dictionary	`document-information-dictionary`	Specifies details of the PDF document information dictionary (DID), which includes properties like the document creation data and producer.
pdfid0	`string`	Specifies the first file identifier found for the PDF file.
pdfid1	`string`	Specifies the second file identifier found for the PDF file.

4.7.4.5.2 Document Information Dictionary (`document-information-dictionary`)

The Document Information Dictionary object captures details of the PDF Document Information Dictionary, used for storing metadata associated with the PDF document.

Properties

Property Name	Type	Description
title	`string`	Specifies the title of the PDF document.
author	`string`	Specifies the name of the person who

		created the PDF document.
subject	`string`	Specifies the subject of the PDF document.
keywords	`string`	Specifies the keywords associated with the PDF document.
creator	`string`	Specifies the name of the application that created the original document, for cases where the original document was then converted to PDF.
producer	`string`	Specifies the name of the application that converted the document to PDF, for cases where the original document was then converted to PDF.
creationdate	`string`	Specifies the date and time that the document was created.
moddate	`string`	Specifies the date and time that the document was most recently modified.
trapped	`string`	Specifies a name object indicating whether the document has been modified to include trapping information.

4.7.4.6 Archive File (`archive-file-extension`)

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*.

Properties

Property Name	Type	Description
version	`string`	Specifies the version of the archive type used in the archive file.
file_count	`integer`	Specifies the number of file contained within the archive.
encryption_algorithm	`encryption-algorithm-enum`	Specifies the name of the encryption algorithm used to encrypt the archive file.

decryption_key	`string`	Specifies the decryption key used to decrypt the archive file.
comment	`string`	Specifies a comment included as part of the archive file.

4.7.4.7 PE Binary File (`pebinary-file-extension`)

The Archive File extension specifies a default extension for capturing properties specific to portable executable (PE) files. The key for this extension when used in the **`extended_properties`** dictionary MUST be *pebinary*.

Properties

Property Name	Type	Description
version	`string`	Specifies the version of the archive type used in the archive file.

4.8 Registry Key Object (`win-registry-key-object`)

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

Properties

Inherits From	Inherited Properties		
`cybox-object`	**all**		
Property Name	**Type**	**Description**	
type (inherited from `cybox-object`)	`string`	Indicates that this object is a CybOX File Object. The value of this field **MUST** be `win-registry-key-object`	
key	`string`	Specifies the full registry key value as a lowercase string, including the hive. The hive MUST be fully expanded and not truncated; e.g., hkey_local_machine must be	

		used instead of hklm.	
number_of_values	`integer`	Specifies the number of values found under the registry key.	
values	`array` of type `registry-value`	Specifies the values found under the registry key.	
modified_time	`timestamp`	Specifies the last date/time that the registry key was modified.	
creator_username	`string`	Specifies the name of the user who created the registry key.	
number_of_subkeys	`integer`	Specifies the number of subkeys contained under the registry key.	

4.8.2 Registry Value (`registry-value`)

Properties

Property Name	Type	Description
name	`string`	Specifies the name of the registry value, as a lowercase string. For specifying the default value in a registry key, an empty string should be used.
data	`string`	Specifies the data contained in the registry value.
datatype	`registry-datatype-enum`	Specifies the registry (REG_*) datatype used in the registry value.

4.8.3 Registry Datatype Enum (`registry-value`)

Values

Enum 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. Some UNIX systems support big-endian architectures.
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 arrays 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_DESCRIPTOR	A series of nested arrays 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. See http://www.mdgx.com/reg.htm.
REG_QWORD	A 64-bit number.
REG_INVALID_TYPE	Specifies an invalid key.

4.9 URL Object (`url-object`)

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

Properties

Inherits From	Inherited Properties		
`cybox-object`	**all**		
Property Name	**Type**	**Description**	
type (inherited from `cybox-object`)	`string`	Indicates that this object is a CybOX URI Object. The value of this field **MUST** be `uri-object`	
value	`string`	Specifies the value of the URL.	

4.10 Domain Name Object (`domain-name-object`)

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

Properties

Inherits From	Inherited Properties		
`cybox-object`	**all**		
Property Name	**Type**	**Description**	
type (inherited from `cybox-object`)	`string`	Indicates that this object is a CybOX Domain Name Object. The value of this field **MUST** be `domain-name-object`	
domain_name_type	`domain-name-type-enum`	Specifies the type of domain name.	
value	`string`	Specifies the value of the domain name.	

4.10.1 Domain Name Type Enum (`domain-name-type-enum`)

Values

Enum Value	Description
fqdn	Specifies a fully-qualified domain name, e.g., "www.abcd.com".
tld	Specifies a top-level domain, e.g., ".com".

4.11 System Object (`system-object`)

The System object represents the properties of a system.

Properties

Inherits From	Inherited Properties		
`cybox-object`	**all**		
Property Name	**Type**	**Description**	
type (inherited from `cybox-object`)	`string`	Indicates that this object is a CybOX System Object. The value of this field **MUST** be `system-object`	
hostnames	`array` of type `hostname`	Specifies the hostname(s) associated with the system.	

4.11.1 Hostname Type

Properties

Property Name	Type	Description	
is_domain_name	`boolean`	Specifies if this hostname is also a valid domain name.	
value	`string`	Specifies the value of the hostname.	
naming_system	`string`	Specifies a relevant naming system for the hostname (e.g.,	

		DNS, NIS, NetBIOS, etc.).	

4.12 AS Object (`as-object`)

The AS object represents the properties of an autonomous system (AS).

Properties

Inherits From	Inherited Properties		
`cybox-object`	**all**		
Property Name	**Type**	**Description**	
type (inherited from `cybox-object`)	`string`	Indicates that this object is a CybOX AS Object. The value of this field **MUST** be `as-object`	
number	`integer`	Specifies the number assigned to the autonomous system (AS). Such assignments are typically performed by a regional internet registry (RIR).	
name	`string`	Specifies the name of the autonomous system (AS).	
handle	`string`	Specifies the handle for the autonomous system (AS), which is typically the AS number prepended with the string 'AS'.	
regional_internet_registry	`string`	Specifies the name of the regional internet registry (RIR) that assigned the number to the autonomous system (AS).	

4.13 Device Object (`device-object`)

The Device Object represents the properties of a hardware device.

Properties

Inherits From	Inherited Properties		
`cybox-object`	**all**		
Property Name	**Type**	**Description**	
type (inherited from `cybox-object`)	`string`	Indicates that this object is a CybOX Device Object. The value of this field **MUST** be `device-object`	
device_type	`string`	Specifies the type of the device.	
manufacturer	`string`	Specifies the manufacturer of the device.	
model	`string`	Specifies the model identifier of the device.	
serial_number	`string`	Specifies the serial number of the Device.	
firmware_version	`string`	Specifies the version of the firmware running on the device.	

4.14 Mutex Object (`mutex-object`)

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

Properties

Inherits From	Inherited Properties		
`cybox-object`	**all**		
Property Name	**Type**	**Description**	
type (inherited from	`string`	Indicates that this object is a CybOX Mutex Object. The value of this field **MUST** be	

cybox-object)		mutex-object	
name	string	Specifies the name of the mutex object.	