The CybOX™ Language Specification

Version 1.0(draft)

Sean Barnum, Robert Martin, Bryan Worrell, Ivan Kirillov

4/13/2012

The Cyber Observable eXpression (CybOX) is a standardized language, being developed in collaboration with any and all interested parties, for the specification, capture, characterization and communication of events or stateful properties that are observable in the operational domain. A wide variety of high-level cyber security use cases rely on such information including: event management/logging, malware characterization, intrusion detection, incident response/management, attack pattern characterization, etc. CybOX provides a common mechanism (structure and content) for addressing cyber observables across and among this full range of use cases improving consistency, efficiency, interoperability and overall situational awareness. To enable such an aggregate solution to be practical for any single use case, numerous flexibility mechanisms are designed into the language. In particular, almost everything is optional such that any single use case could leverage only the portions of CybOX that are relevant for it (from a single field to the entire language or anything in between) without being overwhelmed by the rest. This document defines the requirements and data model for the CybOX Language.

Acknowledgements

The authors Sean Barnum, Robert Martin, Bryan Worrell and Ivan Kirillov wish to thank the CybOX community for its assistance in contributing and reviewing this document.

Trademark Information

CybOX and the CybOX logo are trademarks of The MITRE Corporation. All other trademarks are the property of their respective owners.

Terms of Use

MITRE MAKES CybOX AVAILABLE ON AN "AS IS" BASIS AND THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE MITRE CORPORATION, ITS BOARD OF TRUSTEES, OFFICERS, AGENTS, AND EMPLOYEES, DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION THEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.[1]

Feedback

The MITRE Corporation welcomes any feedback regarding the CybOX Language Specification. Please send any comments, questions, or suggestions to cybox@mitre.org.[2]

---

[1] For more information see http://cybox.mitre.org/about/termsofuse.html

[2] For more information about the CybOX Language, please visit http://cybox.mitre.org

Table of Contents

1 Introduction

Information security is a complex function that consumes significant organizational resources, and is growing increasingly difficult to manage. One of the biggest problems is a lack of standardization among the various activities involved including between the sources of security information, and the tools that consume that information, as well as between the various tools themselves. Often, the exchange of security information is time critical, but is hampered by the variety of incompatible formats in which it is represented.

This lack of standardization gives rise to many challenges across the information security community. One such challenge is the ability to effectively understand and communicate observations in the cyber domain as well as meaningful patterns of potential observations that may indicate some sort of relevant event or state. The concept of observable events or properties in the operational cyber realm is a central, underlying element of a wide array of different activities involved in cyber security. Cyber observables are a critical element of event management, attack pattern & threat characterization, cyber threat indicator sharing, attack detection, incident investigation, malware analysis & management, digital forensics, etc.

Without a uniform, standard mechanism for specifying, capturing, characterizing, and communicating these cyber observables, each activity area, each use case, each organization, each sharing community and often each supporting tool vendor is forced to use its own unique approach for representing data that inhibits consistency, efficiency, interoperability, and overall situational awareness. This requires the IT Security Professional to translate the data produced by the various processes and tools in order to map between users and uses and to correlate all of this data in order to obtain a meaningful holistic situational awareness. It may also be necessary for the data to be manually converted into a format that is usable by another tool which can also be a tedious and error-prone process.

What the industry requires is a standardized method for representing cyber observables. The representation of this information must easily facilitate its generation, sharing, consumption and analysis by software tools. The advantage of such a standard is that it will:

- Bring consistency and transparency to cyber observables produced by sensors
- Bring consistency and transparency to the results produced by analysis tools.
- Enable new levels of correlation analysis heuristics
- Assist in the exchange of information between tools.
- Enable holistic exchange of cyber observables between differing activities and use cases
- Enable new levels of integrated situational awareness and operational understanding
- Reduce the need for IT Security Professionals to learn the proprietary languages of each of the processes and tools that they and their partners use, and instead allow them to learn a single language that is understood by all the processes and tools.

This document presents the CybOX Language as a standard that fulfills these needs and requirements.

1.1 The CybOX Language

The Cyber Observable eXpression (CybOX™) is an international, information security, community standard to promote consistent capture of cyber observable content, and to standardize the transfer of this information across the entire spectrum of security activities, tools and services.

The CybOX Language, developed by a broad spectrum of industry, academia, and government organizations from around the world, standardizes the encoding and communication of high-fidelity information about cyber observables, whether they are dynamic events or stateful measures observable in the operational cyber domain.

The CybOX Language adheres to three overarching principal objectives:

- Develop a common solution for all relevant use cases

  CybOX is not targeted at a single cyber security use case; rather it is intended to be flexible enough to offer a common solution for all cyber security use cases requiring the ability to deal with cyber observables. CybOX is targeted to support a wide range of relevant cyber security domains including: event management, attack pattern & threat characterization, cyber threat indicator sharing, attack detection, incident investigation, malware analysis & management, digital forensics, etc. ***To enable such an aggregate solution to be practical for any single use case, numerous flexibility mechanisms are designed into the language. In particular, almost everything is optional such that any single use case could leverage only the portions of CybOX that are relevant for it (from a single field to the entire language or anything in between) without being overwhelmed by the rest.***

- Develop a solution for both instances of observables as well as potential patterns

  CybOX is also intended to be flexible enough to allow both the ***high-fidelity description of cyber observable instances*** measured in an operational context as well as more ***abstract patterns for potential observables*** that may be targets for observation and analysis apriori. This flexibility has the potential to enable greater synergies between observation and interpretation.

- Develop a solution capable of supporting significant improvements in automation

  By specifying a common structured language mechanism for the cyber observables, the intent is to enable the potential for new levels of detailed ***automatation*** in sharing, mapping, detection and analysis heuristics.

By achieving these objectives the CybOX Language serves as a framework and vocabulary to provide:

- A comprehensive and flexible solution for characterizing cyber observables.
- A standard format that codifies the necessary range of cyber observable characteristics.
- An open alternative to closed, proprietary, and replicated efforts.
- An effort that is supported by a community of security experts, system administrators, and software developers from industry, government, and academia.

All of which leads to a common and structured format that facilitates collaboration and information sharing among the information security community as well as interoperability among security tools.

1.2 Specification Architecture

The CybOX language is defined within a set of specification documents as follows:

- **CybOX Language Core Specification**

  Specifies the purpose, approach, conventions and usage of the CybOX language as well as the detailed language data models for the language core and set of common types.

- **CybOX Language Defined Objects Specfication**

  Restates some language basics from the CybOX Language Core Specification (to give context to readers of just the CybOX Language Defined Objects Specfication) as well as specifying the detailed language data models for the official set of CybOX defined objects.

- **CybOX Language Use Case Specification (coming soon)**

  Identifies and characterizes in summary the target use cases supported by the CybOX language.

1.3 CybOX Language Versioning Conventions

The accepted convention for CybOX Language versioning defines a single major and minor version that applies to the entire CybOX Language. These major and minor components are what allow changes to the language to be classified as either major or minor. Whenever a modification is made to the CybOX Language, the version of the language must change. Major versions are only needed when a change to the CybOX Language is made that is not backwards compatible. It is possible to introduce new capabilities to existing language constructs or make bug fixes within a minor revision regardless of whether backward compatibility is maintained though compatibility is always targeted. There is also the possibility for addressing critical defects that will result in breaking backward compatibility within a minor revision of the CybOX Language. The CybOX Language versioning convention also defines a single subminor version for the CybOX Language Defined Objects Specification that represents an independently incrementing version counter for any changes to the object specifications that are independent of changes to the Core language specification. Any implementation schemas should have their major and minor versions aligned with the major and minor versions of the corresponding Core language spec and should have a subminor version that represents an independently incrementing version counter for minor schema changes, feature additions or bug fixes occuring between specification releases.

Language Specifications

Core

Major version number = Major language changes

Minor version number = Feature additions and minor language changes (including bug fixes that could break backward compatability)

Objects

Major version number = Aligned with major version number of Core specification

Minor version number = Aligned with minor version number of Core specification

Subminor version number = Independently incrementing version counter (Any changes to object specifications independent of changes to the Core specification)

Implementation Schema Core & Common_Types

Major version number = Aligned with major version number of Core specification

Minor version number = Aligned with minor version number of Core specification

Subminor version number = Bug fixes without backward compatibility issues or between specification releases

Implementation Schema Defined Objects

Major version number = Aligned to major version number of related Core schema

Minor version number = Aligned with minor version number of Core specification

Subminor version number = Minor object schema changes, feature additions and bug fixes

Language Releases

Major version number = Aligned with major version number of Core specification

Minor version number = Aligned with minor version number of Core specification

1.4 CybOX Language Naming Conventions

The CybOX Language utilizes the following naming conventions.

Metadata Field Names

Convention: Lowercase with underscores (e.g. object_state)

Data Field Names

Convention: Capitalized with underscores (e.g. Defined_Object)

Type names

Convention:	Camelcase upper start without pretype underscore (e.g. DefinedObjectType)

Enumeration Type names:

Convention:	Camelcase upper start without pretype underscore with “Enum” appended (e.g. DefinedObjectTypeEnum)

Attribute Group names:

Convention:	Camelcase upper start without pretype underscore with “Group” appended (e.g. ObjectAttributeGroup)

Object Names

Convention:	Object specification file names: Capitalized with underscores (e.g. Win_Network_Route_Entry_Object)
Convention:	Object specification root element: Capitalized with underscores without trailing “Object” (e.g. Win_Network_Route_Entry)

Namespace names

Convention:	Camelcase upper start with entire object name with removed underscores (e.g. NetworkRouteEntryObject) The exceptions would be Common_Types which would just be “Common” and the core namespace would just be “cybox”

Namespace abbreviations

Convention:

Camelcase upper start

- with entire object name
  - with removed underscores
  - with Windows abbreviated to Win
  - with Object abbreviated to Obj
  - with Network abbreviated to Net

(e.g. WinNetRouteEntryObj)

1.5 Document Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in *RFC 2119*.[16]

The following font and font style conventions are used throughout the remainder of this document:

- The `Courier New` font is used for writing constructs in the CybOX Language Data Model. Example: `generator`
- The *'italic, with single quotes'* font is used for noting values for CybOX Language properties. Example: *'does not exist'*

This document uses the concept of namespaces[3] to logically group CybOX constructs throughout both the Data Model section of the document, as well as other parts of the specification. The format of these namespaces is `prefix:element`, where the prefix is the namespace component, and the element is the name of the qualified construct. The following table lists the namespaces used in this document:

Data Model	Namespace	Description	Example(s)
CybOX Core	cybox	The CybOX Core data model that captures all of the foundational constructs used in CybOX.	cybox:ObservableType
CybOX Common	Common	The CybOX Common data model that captures all of the common constructs used across the various CybOX object data models	Common:HashType
CybOX Objects	<type>Obj	The CybOX Object data models construct representations of observable and stateful information. Each CybOX Object schema has its own defined namespace and can be used as an extension point for other domain-specific or organizational-specific models.	FileObj:FileObjectType MutexObj:Mutex MemoryObj:Memory_Block

1.6 Document Structure

This document serves as the specification for the CybOX Language defining requirements, data model, and processing model which is organized into the following sections:

- Section 1 – Introduction
- Section 2 –Use Cases for the CybOX Language

---

[3] Namespaces (computer science): http://en.wikipedia.org/wiki/Namespace_(computer_science)

- Section 2 – Requirements for the CybOX Language
- Section 3 – Data Model for the CybOX Langauge
- Section 4 – Representations of the CybOX Language
- Appendix A – Leveraging the CybOX Language Data Model
- Appendix B – Extending the CybOX Language Data Model
- Appendix C – Normative References
- Appendix D – Change Log
- Appendix E –Acronyms

2 Use Cases for the CybOX Language

The following list identifies the key use cases that the CybOX language is targeted to support. These use cases will be further characterized and described within the CybOX Language Use Case Specification. Additional use cases will be documented as they emerge through the continued operational application of CybOX.

- **Use Case Area: Event Management**
    - Producing Event Data
    - Exchanging Event Data
    - Analyzing Event Data
    - Querying Event Data
    - Composing Events
- **Use Case Area: Attack Patterns and Threat Characterization**
    - Characterizing Observable Evidence of Granular Attacker Actions
    - Characterizing Observable Evidence of Attacker Preparatory Probing Techniques
    - Characterizing Observable Evidence of Attacker Obfuscation Techniques
    - Characterizing Observable Evidence of Abstract Attack Patterns
- **Use Case Area: Cyber Threat Indicator Sharing**
    - Generating Cyber Threat Indicators
    - Exchanging Cyber Threat Indicators
- **Use Case Area: Attack Detection**
    - Detecting Dynamic In-Progress Attacks
    - Detecting Past Attacks
- **Use Case Area: Incident Investigation**
    - Correlating Incident Initiation Data
    - Excavating Incident Context
- **Use Case Area: Malware Analysis & Management**
    - Analyzing Malware Instances
    - Analyzing Malware Patterns
    - Hunting Malware Artifacts
    - Metadata Indexing Malware Collections
    - Exchanging Malware Characterizations
- **Use Case Area: Digital Forensics**
    - Conducting Digital Forensic Analysis
    - Managing Evidentiary Process

3 Requirements for the CybOX Language

The following requirements have been developed based upon the goals of CybOX and the needs of the targeted use cases identified on the CybOX website. These requirements apply to the CybOX Language itself and establish the CybOX Language as the standardized framework for expressing cyber observables. At the highest level are the Basic Requirements, which capture the essence of the goals and use cases. Each of these requirements is further expanded and refined in the Detailed Requirements section below.

3.1 Basic Requirements

The basic requirements listed in this section form the foundation of the CybOX Language and are further refined and expanded upon in the Detailed Requirements section of this document.

Supporting Modular and Flexible Use (Type layering & Optionality)

- The language MUST be capable of supporting modular, partial and flexible use.

Expressing Observed Cyber Observable Instances

- The language MUST be capable of expressing the details of specific cyber observable observation instances.

Expressing Apriori Cyber Observable Patterns

- The language MUST be capable of expressing the abstract variation of apriori patterns for cyber observables.

Expressing Cyber Observable Events

- The language MUST be capable of expressing dynamic cyber observable events.

Expressing Cyber Observable Stateful Measures

- The language MUST be capable of expressing static cyber observable properties.

Expressing Relationships Between Actions

- The language MUST be capable of expressing relationships between cyber observable actions.

Expressing Relationships Between Actions and Objects

- The language MUST be capable of expressing relationships between cyber observable actions and related objects.

Expressing Relationships Between Objects

- The language MUST be capable of expressing relationships between cyber observable objects.

Characterizing the Source of the Cyber Observable Data

- The language MUST be capable of characterizing the source of cyber observable data.

Expressing Logical Compositions of Individual Cyber Observables

- The language MUST be capable of expressing logical (AND, OR, NOT, etc.) compositions of individual cyber observables.

Supporting Extensibility

- The language MUST be capable of supporting extensibility by users of the language.

3.2 Detailed Requirements

The detailed requirements expand upon the general requirements listed in the previous section.

Content Creation

- The language MUST require that all content specify the language version which it complies with.
- The language MUST require that all content specify when it was created.
- The language MUST allow content to contain information about the product name and version used to create the content.
- The language MUST allow content to contain additional information that is relevant to the creation of the document.

Flexibility & Modularity

- The language MUST employ a general approach of flexible optionality where almost all portions of the language are not required such that any given user or use case of the language may use only the portions of the language which are relevant to their context.
- The language MUST support a modular layered structure that enables use of independent portions of the language without requiring other portions.
- All major components of the language MUST be reusable.
- Should support centralized pooling and inclusion via reference of events, actions, objects and attributes

Identifiers

- Components of the language MUST have globally unique identifiers.
- Components of the language MUST support definition as a reference to a globally unique identifier.
- Component identifiers MUST be structured to allow individual organizations to dynamically create identifiers without relying on an outside source and be ensured that global uniqueness is maintained.

Composition & Collection

- The language MUST allow for the exchange of collections of CybOX Observables as a single unit of content.
- The language MUST contain the structure and the means to create unbounded logical combinations of individual components.
- The language MUST provide the ability to negate logical statements.
- The language MUST be capable of expressing logical (AND, OR, NOT, etc.) compositions of individual events.

Instances & Patterns

- The language MUST support characterization of the full details of cyber observable instance events, actions and objects.
- The language MUST support characterization of the relevant arguments for cyber observable actions.
- The language MUST support abstraction of specific defined object structure and syntax to support flexible extension of the language.
- The language MUST support characterization of custom (undefined) object attributes.
- The language MUST support characterization of domain-specific (independent of CybOX) object attributes.
- The language MUST support characterization of defined effects of cyber observable actions on objects.
- The language MUST support the definition of patterns on individual object attributes.
- The language MUST support the definition of various conditions on object attribute values (e.g. Equals, DoesNotEqual, Contains, StartsWith, GreaterThan, IsInSet, IsInRange, etc.).
- The language MUST support the specification that an object attribute value is within a set of potential values
- The language MUST support the specification that an object attribute value is within a range of potential values
- The language MUST support the specification of regular expression (Regex) pattern definitions to be applied to individual object attributes.

Characterizing Source of Cyber Observable Data

- The language MUST support characterization of the source of cyber observable data including observable collections, observables, events, actions & objects.
- The language MUST support characterization of the people or organizations that are the source of specific cyber observable data.
- The language MUST support characterization of the time that specific cyber observable data was observed or captured.
- The language MUST support detailed characterization of the tools that are the source of specific cyber observable data.

- The language MUST support characterization of any indicators that are the source of specific cyber observable data.
- The language MUST support characterization of the system on which the specific cyber observable data was collected or characterized.
- The language MUST support characterization of the process instance in which the specific cyber observable data was collected or characterized.

4 Data Model

4.1 Data Model Conventions

The following conventions are used throughout this data model section.

4.1.1 Property Table Notation

Throughout the data model, tables are used to describe each data type. Each property table will consist of a column of property names to identify the property, a type column to reflect the datatype of the property, a multiplicity column to reflect the allowed number of occurrences of the property, and a description column that will describe the property. Values in the type column are either primitive datatypes or other types defied in this document. These values will be cross referenced to the base definition of their types. Below is an example property table.

Table 4-1 Example Property Table

Property	Type	Multiplicity	Description
<PROPERTY NAME>	<DATA TYPE>	0..1	<DESCRIPTION OF THE PROPERTY AND ANY USAGE REQUIREMENTS FOR THE PROPERTY>

4.1.2 Primitive Data Types

The following primitive datatypes are used in the CybOX Language.

- hexBinary – Data of this type conforms to the World Wide Web Consortium (W3C) Recommendation for hex-encoded binary data [1].
- base64Binary – Data of this type conforms to the W3C Recommendation for base-64-encoded binary data [2].
- boolean – Data of this type conforms to the W3C Recommendation for boolean data [3].
- integer – Data of this type conforms to the W3C Recommendation for integer data [4].
- unsigned int – Data of this type represents an unsigned integer value that conforms to the W3C Recommendation for unsigned integer data [5].
- non-negative int – Data of this type represents a non-negative integer value that conforms to the W3C Recommendation for non-negative integer data [6].
- positive int – Data of this type represents a positive integer value that conforms to the W3C Recommendation for positive integer data [7].
- long – Data of this type represents a long integer value that conforms to the W3C Recommendation for long integer data [8].

- unsigned long – Data of this type represents an unsigned long value that conforms to the W3C Recommendation for unsigned long data [9].
- double – Data of this type represents a double value that conforms to the W3C Recommendation for double data [10].
- float – Data of this type represents a float value that confirms to the W3C Recommendation for float data [11].
- time – Data of this type represents a time value that conforms to the W3C Recommendation for time data [12].
- date – Data of this type represents a date value that conforms to the W3C Recommendation for date data [13].
- dateTime – Data of this type represents a date and time value that conforms with the W3C Recommendation for datetime data [14].
- duration – Data of this type represents a duration value that conforms to the W3C Recommendation for duration data [15].
- string – Data of this type conforms to the W3C Recommendation for string data [16].
- QName – Data of this type conforms to the W3C Recommendation for QName data[17]
- URI – Data of this type conforms to the W3C Recommendation for anyURI data [18].

4.1.3 CybOX Primitive Datatype Expansions

The CybOX language within the Common Types data model defines several datatypes to be used for CybOX object attributes in place of language-specific primitive data types. By leveraging a common foundation—`cybox:BaseObjectAttributeType`—each derivation is able to store metadata (e.g., regular expressions, ranges, entropy) to help characterize its stored data.

The following CybOX datatypes have been defined to expand language-specific primitives.

- AnyURIObjectAttributeType
- Base64BinaryObjectAttributeType
- DateObjectAttributeType
- DateTypeObjectAttributeType
- DoubleObjectAttributeType
- DurationObjectAttributeType
- FloatObjectAttributeType
- HexBinaryObjectAttributeType
- IntegerObjectAttributeType
- LongObjectAttributeType
- NameObjectAttributeType
- NonNegativeIntegerObjectAttributeType
- PositiveIntegerObjectAttributeType
- StringObjectAttributeType
- TimeObjectAttributeType
- UnsignedLongObjectAttributeType

- UnsignedIntegerObjectAttributeType

4.1.4 CybOX Identifier Conventions

The CybOX language defines identifier (id) fields as qualified names according to the W3C recommendation for QName data[17] with the added stipulation that the namespace prefix MUST be present.

The CybOX use of the QName type is a colon separated string construct where the nonoptional prefix before the colon is a namespace reference associated with a URI for the defining domain/scope and the postfix after the colon is an identifier string beginning with a letter whose format is specified by the associated namespace domain. Native CybOX content MUST utilize the “cybox” namespace prefix.

Examples:

cybox:guid-fce3cf95-2bc6-45de-b418-c5991e201196

maec:example-obj-1

capec:cybox-59cac3e5-a2bc-481a-9541-adafef920cc9

foo:bar-123

Utilizing this approach, CybOX both ensures global uniqueness of identifiers and enables the flexible use of CybOX content within various different contexts or other information standards that require their own particular identifier syntax.

Currently each specifying domain will define their own format locally. CybOX envisions a future independent registration of valid namespaces and associated domain formats under an organization such as IANA.

4.2 Core Data Types

4.2.1 ObservablesType

The ObservablesType is a complex type representing a collection of cyber observables.

Property	Type	Mult	Description
cybox_major_version	string	1..1	The major_version attribute specifies the major version of the CybOX language utlized for this set of Observables.
cybox_minor_version	string	1..1	The minor_version attribute specifies the minor version of the CybOX language utlized for this set of Observables.
Observable_Package_Source	`Common:MeasureSourceType`	0..1	The Observable_Package_Source element is optional and enables descriptive specification of how this package of Observables was identified and specified.
Observable	cybox:Observable	1..∞	The Observable element represents a description of a single cyber observable.
Pools	`cybox:PoolsType`	0..1	The Pools element enables the description of

			Events, Actions, Objects and Attributes in a space-efficient pooled manner with the actual Observable structures defined in the CybOX schema containing references to the pooled elements. This reduces redundancy caused when identical observable elements occur multiple times within a set of defined Observables.

4.2.2 **ObservableType**

The ObservableType is a complex type representing a description of a single cyber observable.

Property	Type	Mult	Description
id	QName	1..1	The id attribute specifies a unique id for this Observable.
idref	QName	1..1	The idref attribute specifies a unique id reference to an Observable defined elsewhere.
Title	string	0..1	The Title element provides a mechanism to specify a short title or description for this Observable
Description	`Common:StructuredTextType`	0..1	The Description element provides a mechanism to specify a structured text description of this Observable.
Keywords	string	0..∞	The Keywords element enables capture of relevant keywords for this cyber observable.
Observable_Source	`Common:MeasureSourceType`	0..1	The Observable_Source element is optional and enables descriptive specification of how this Observable was identified and specified.
Stateful_Measure	`cybox:StatefulMeasureType`	0..1	The Stateful Measure element enables specification of a cyber observable property that is statically stateful in nature (e.g. a registry key holding a certain value, a specific mutex existing or a file having a specific MD5 hash).
Event	cybox:Event	0..1	The Event element enables specification of a cyber observable event that is dynamic in nature with specific action(s) taken against specific cyber relevant objects (e.g. a file is deleted, a registry key is created or an HTTP Get Request is received).
Observable_Composition	`cybox:ObservableCompositionType`	0..1	The Observable_Composition element enables specification of composite observables made up of logical constructions of atomic observables or other composite observables (e.g. Obs5 = (Obs1 OR Obs2) AND (Obs3 OR Obs4)).
Noisiness	`cybox:NoisinessEnum`	0..1	The Noisiness element is optional and enables simple characterization of how noisy this Observable typically could be. In other words, how likely is it to generate false positives.
Ease_of_Obfuscation	`cybox:EaseOfObfuscationEnum`	0..1	The Ease_of_Obfuscation element is optional and enables simple characterization of how easy it would be for an attacker to obfuscate the observability of this Observable.
Obfuscation_Techniques	`cybox:ObfuscationTechniquesType`	0..1	The Obfuscation_Techniques element is optional and enables the description of potential techniques

			an attacker could leverage to obfuscate the observability of this Observable.

4.2.3 StatefulMeasureType

The StatefulMeasureType is a complex type representing a cyber observable property that is statically stateful in nature (e.g. a registry key holding a certain value, a specific mutex existing or a file having a specific MD5 hash).

Property	Type	Mult	Description
has_changed	boolean	1..1	The has_changed attribute is optional and conveys a targeted observation pattern of whether the associated stateful measure specified has changed. This attribute would be leveraged within a pattern observable triggering on whether the value of a stateful measure comprised of an objet specification has changed.
name	string	1..1	The name attribute is optional and enables the assignment of a relevant name to a specific Stateful Measure.
Description	Common: StructuredTextType	0..1	The Description element provides a mechanism to specify a structured text description of this Stateful Measure.
Object	cybox:Object	1..1	The Object element identifies and specificies the characteristics of a specific cyber-relevant object (e.g. a file, a registry key or a process).

4.2.4 TrendEnum

TrendEnum is a (non-exhaustive) enumeration of trend types.

Restriction base: string

Enumeration Value	Description
Increasing	Specifies an increasing trend.
Decreasing	Specifies a decreasing trend.

4.2.5 EventType

The EventType is a complex type representing a cyber observable event that is dynamic in nature with specific action(s) taken against specific cyber relevant objects (e.g. a file is deleted, a registry key is created or an HTTP Get Request is received).

Property	Type	Mult	Description
id	QName	1..1	The id attribute specifies a unique id for this Event.
idref	QName	1..1	The idref attribute specifies a unique id reference to an Event defined elsewhere.
type	cybox:EventTypeEnum	1..1	The type attribute specifies what kind of Event this is.
Description	Common: StructuredTextType	0..1	The Description element provides a mechanism to specify a structured text description of this Event.
Producer-Observer	Common: MeasureSourceType	0..1	The Producer-Observer element is optional and enables descriptive specification of how this Event was observed (in the case of a Cyber Observable

			Event instance) or could potentially be observed (in the case of a Cyber Observable Event pattern).
Actions	cybox:ActionsType	0..1	The Actions element enables description/specification of one or more cyber observable actions.
Frequency	cybox:FrequencyType	0..1	The Frequency element conveys a targeted observation pattern of the frequency of the associated event or action.
Event	cybox:EventType	1..1	This Event element is included recursively to enable description/specification of composite Events.

4.2.6 EventTypeEnum

EventTypeEnum is a (non-exhaustive) enumeration of cyber observable event types.

Restriction base: string

Enumeration Value	Description
File Ops (CRUD)	Specifies the class of events dealing with file operations.
Registry Ops	Specifies the class of events dealing with registry operations.
Memory Ops	Specifies the class of events dealing with memory operations.
Process Mgt	Specifies the class of events dealing with process management.
Thread Mgt	Specifies the class of events dealing with thread management.
Service Mgt	Specifies the class of events dealing with service management.
Session Mgt	Specifies the class of events dealing with session management.
API Calls	Specifies the class of events dealing with API calls.
Port Scan	Specifies the class of events dealing with port scanning.
IP Ops	Specifies the class of events dealing with IP Operations.
DNS Lookup Ops	Specifies the class of events dealing with DNS Lookup operations.
Socket Ops	Specifies the class of events dealing with thread management.
IPC	Specifies the class of events dealing with thread management.
Configuration Management	Specifies the class of events dealing with configuration management.
User/Password Mgt	Specifies the class of events dealing with user/password management.
Account Ops (App Layer)	Specifies the class of events dealing with account operations at the application layer.
HTTP Traffic	Specifies the class of events dealing with HTTP traffic.
App Layer Traffic	Specifies the class of events dealing with Application Layer traffic.
Packet Traffic	Specifies the class of events dealing with packet traffic.
Data Flow	Specifies the class of events dealing with data flow.
Anomoly Events	Specifies the class of events dealing with anomoly events.
Technical Compliance	Specifies the class of events dealing with Technical compliance.
Procedural Compliance	Specifies the class of events dealing with procedural compliance.
GUI/KVM	Specifies the class of events dealing with the GUI/Kernel-based Virtual Machine (KVM).
Autorun	Specifies the class of events dealing with Autorun.
USB/Media Detection	Specifies the class of events dealing with USB and/or Media detection.
SQL	Specifies the class of events dealing with the SQL language.
DHCP	Specifies the class of events dealing with the Dynamic Host Configuration Protocol (DHCP).
Redirection	Specifies the class of events dealing with redirection.
Authentication Ops	Specifies the class of events dealing with authentication operations.
Authorization (ACL)	Specifies the class of events dealing with authorization via Access Control Lists (ACL).
Privilege Ops	Specifies the class of events dealing with privilege operations.

Basic System Ops	Specifies the class of events dealing with basic system operations.
Signature Detection	Specifies the class of events dealing with signature detection.
Auto-update Ops	Specifies the class of events dealing with auto-update operations.
Application Logic	Specifies the class of events dealing with application logic.
Email Ops	Specifies the class of events dealing with e-mail operations.

4.2.7 FrequencyType

The FrequencyType is a complex type representing the specification of a frequency for a given action or event..

Property	Type	Mult	Description
rate	float	1..1	This attribute specifies the rate for this defined frequency.
scale	string	1..1	This attribute specifies the time scale for this defined frequency.
trend	cybox:TrendEnum	1..1	This attribute is optional and conveys a targeted observation pattern of the nature of any trend in the frequency of the associated event or action. This attribute would be leveraged within an event or action pattern observable triggering on the matching of a specified trend in the frequency of an event or action.
units	string	1..1	This attribute specifies the units for this defined frequency.

4.2.8 ActionsType

The ActionsType is a complex type representing a set of cyber observable actions.

Property	Type	Mult	Description
Action	cybox:Action	1..∞	The Action element enables description/specification of a single cyber observable action.

4.2.9 ActionType

The ActionType is a complex type representing a single cyber observable action.

Property	Type	Mult	Description
action_status	cybox: ActionStatusTypeEnum	1..1	The action_status attribute enables description of the status of the action being described.
context	cybox: ActionContextTypeEnum	1..1	The context attribute is optional and enables simple characterization of the broad operational context in which the Action is relevant
id	QName	1..1	The id attribute specifies a unique id for this Action.
idref	QName	1..1	The idref attribute specifies a unique id reference to an Action defined elsewhere.
network_protocol	cybox: NetworkProtocolEnum	1..1	The network_protocol attribute is optional and (where the Context is Network) enables the description of the relevant network protocol involved in the Action.
ordinal_position	positiveInteger	1..1	The ordinal_position attribute is intended to

			reference the ordinal position of the action with within a series of actions.
timestamp	time	1..1	The timestamp attribute represents the local or relative time at which the action occurred or was observed.
type	`cybox:ActionTypeEnum`	1..1	The type attribute specifies the basic type of action performed.
*******		1..1	The "any" attribute enables the capture of custom attributes describing this Action.
Action_Name	`cybox: ActionNameType`	0..1	The Action_Name element is optional and identifies/characterizes the specific action performed.
Description	`Common: StructuredTextType`	0..1	The Description element contains a textual description of the action.
Action_Aliases	`cybox: ActionAliasesType`	0..1	The Action_Aliases element is optional and enables identification of other potentially used names for this Action.
Action_Arguments	`cybox: ActionArgumentsType`	0..1	The Action_Arguments element is optional and enables the specification of relevant arguments/parameters for this Action.
Discovery_Method	`Common: MeasureSourceType`	0..1	The Discovery_Method element is optional and enables descriptive specification of how this Action was observed (in the case of a Cyber Observable Action instance) or could potentially be observed (in the case of a Cyber Observable Action pattern).
Associated_Objects	`cybox: AssociatedObjectsType`	0..1	The Associated_Objects element is optional and enables the description/specification of cyber Objects relevant (either initiating or affected by) this Action.
Relationships	`cybox: RelationshipsType`	0..1	The Relationships element is optional and enables description of other cyber observable actions that are related to this Action.
Frequency	`cybox: FrequencyType`	0..1	The Frequency element conveys a targeted observation pattern of the frequency of the associated event or action.

4.2.10 ActionTypeEnum

ActionTypeEnum is a (non-exhaustive) enumeration of cyber observable action types.

Restriction base: string

Enumeration Value	Description
Accept	Specifies the atomic action of accepting an object or value.
Access	Specifies the atomic action of accessing an object.
Alert	Specifies the atomic action of issuing an alert.
Allocate	Specifies the atomic action of allocating an object.
Archive	Specifies the atomic action of archiving an object or data.
Assign	Specifies the atomic action of assigning a value to an object.
Audit	Specifies the atomic action of auditing an object.
Backup	Specifies the atomic action of backing up an object or data.
Bind	Specifies the atomic action of binding two objects.
Block	Specifies the atomic action of blocking access to an object or resource.

Call	Specifies the atomic action of calling an object or resource.
Clean	Specifies the atomic action of cleaning an object, such as a file system.
Click	Specifies the atomic action of clicking an object, as with a mouse.
Close	Specifies the atomic action of closing an object, such as a window handle.
Compare	Specifies the atomic action of comparing two objects.
Compress	Specifies the atomic action of compressing an object.
Configure	Specifies the atomic action of configuring a resource.
Connect	Specifies the atomic action of connecting to an object, such as a service or resource.
Control	Specifies the atomic action of controlling an object or data.
Copy/Duplicate	Specifies the atomic action of copying or duplicating an object or data EXCEPT in cases where the object is considered a thread or process as a whole.
Create	Specifies the atomic action of creating an object or data.
Decode	Specifies the atomic action of decoding an object or data.
Decompress	Specifies the atomic action of decompressing an object, such as an archive.
Decrypt	Specifies the atomic action of decrypting an object.
Deny	Specifies the atomic action of denying access to a object or resource.
Depress	Specifies the atomic action of depressing an object that has been pressed, such a button.
Detect	Specifies the atomic action of detecting an object.
Disconnect	Specifies the atomic action of disconnecting from a service or resource.
Download	Specifies the atomic action of an object or data.
Draw	Specifies the atomic action of drawing an object.
Drop	Specifies the atomic action of dropping an object, such as a connection.
Encode	Specifies the atomic action of encoding an object or data.
Encrypt	Specifies the atomic action of encrypting an object or data.
Enumerate	Specifies the atomic action of enumerating a list of objects.
Execute	Specifies the atomic action of executing an object, such as an executable file.
Filter	Specifies the atomic action of filtering an object or data.
Find	Specifies the atomic action of finding an object or data.
Flush	Specifies the atomic action of flushing an object or data, such as a cache.
Fork	Specifies the atomic action of forking, as with a process. Because this is usually associated with processes and threads and does not generalize to objects, it is DIFFERENT from Copy/Duplicate.
Free	Specifies the atomic action of freeing an object.
Get	Specifies the atomic action of getting a value from an object.
Hook	Specifies the atomic action of hooking an object to another object.
Hide	Specifies the atomic action of hiding an object.
Impersonate	Specifies the atomic action of impersonation, in which an object performs actions that assume the character or appearance of another object.
Initialize	Specifies the atomic action of initializing an object.
Install	Specifies the atomic action of installing an object, such as an application, program, patch, or other resource.
Interleave	Specifies the atomic action of interleaving an object, i.e. the action of arranging data in a non-contiguous way to increase performance.
Join	Specifies the atomic action of joining one object to another object.
Kill	Specifies the atomic action of killing an object, as with a thread or program.
Listen	Specifies the atomic action of listening to an object, such as to a port on a network connection.
Load	Specifies the atomic action of loading an object.
Lock	Specifies the atomic action of locking an object.
Login/Logon	Specifies the atomic action of logging into an object, such as into a system or application.

Logout/Logoff	Specifies the atomic action of logging out of an object, such as a system or application.
Map	Specifies the atomic action of mapping an object to another object or data.
Merge	Specifies the atomic action of merging one object to another object.
Modify	Specifies the atomic action of modifying an object.
Monitor	Specifies the atomic action of monitoring the state of an object.
Move	Specifies the atomic action of moving an object.
Open	Specifies the atomic action of opening an object.
Pack	Specifies the atomic action of packing an object.
Pause	Specifies the atomic action of pausing an object, such as a thread or process.
Press	Specifies the atomic action of pressing an object, such as a button.
Quarantine	Specifies the atomic action of placing an object in quarantine, that is, to store the object in an isolated area away from other objects it can operate on.
Query	Specifies the atomic action of querying an object.
Queue	Specifies the atomic action of querying an object.
Raise	Specifies the atomic action of raising an object.
Read	Specifies the atomic action of reading an object.
Receive	Specifies the atomic action of receiving an object.
Release	Specifies the atomic action of releasing an object.
Rename	Specifies the atomic action of renaming an object.
Remove/Delete	Specifies the atomic action of removing or deleting an object.
Replicate	Specifies the atomic action of replicating an object.
Restore	Specifies the atomic action of restoring an object.
Resume	Specifies the atomic action of resuming an object, as with a process or thread.
Run	Specifies the atomic action of running an object, such as an application.
Save	Specifies the atomic action of saving an object.
Scan	Specifies the atomic action of scanning for an object or data.
Schedule	Specifies the atomic action of scheduling an object, such as an event.
Search	Specifies the atomic action of searching for an object.
Send	Specifies the atomic action of sending an object.
Set	Specifies the atomic action of setting an object to a value.
Snapshot	Specifies the atomic action taking a snapshot of an object.
Start	Specifies the atomic action of starting an object, such as a thread or process.
Stop	Specifies the atomic action of stopping an object, such as a thread or process.
Suspend	Specifies the atomic action of suspending an object, such an account or privileges for an account.
Synchronize	Specifies the atomic action of synchronizing an object.
Throw	Specifies the atomic action of throwing an object, such as an exception in a programming language.
Transmit	Specifies the atomic action of transmitting an object.
Unblock	Specifies the atomic action of unblocking an object.
Unhide	Specifies the atomic action of unhiding an object.
Unhook	Specifies the atomic action of unhooking an object from another object, that is, to detach.
Unload	Specifies the atomic action of unloading an object.
Unlock	Specifies the atomic action of unlocking an object.
Unmap	Specifies the atomic action of unmapping an object from another object or data.
Unpack	Specifies the atomic action of unpacking an object, such as an archive.
Update	Specifies the atomic action of updating an object.
Upgrade	Specifies the atomic action of upgrading an object.
Upload	Specifies the atomic action of uploading an object.
Wipe/Destroy/Purge	Specifies the atomic action of wiping, destroying, or purging an object.

Write	Specifies the atomic action of writing an object.

4.2.11 ActionStatusTypeEnum

ActionStatusTypeEnum is a (non-exhaustive) enumeration of cyber observable action status types.

Restriction base: string

Enumeration Value	Description
Success	Specifies a cyber observable action that was successful.
Fail	Specifies a cyber observable action that failed.
Error	Specifies a cyber observable action that resulted in an error.
Complete/Finish	Specifies a cyber observable action that completed or finished.
Pending	Specifies a cyber observable action is pending.
Ongoing	Specifies a cyber observable action that is ongoing.
Unknown	Specifies a cyber observable action with an unknown status.

4.2.12 ActionContextTypeEnum

ActionContextTypeEnum is a (non-exhaustive) enumeration of cyber observable action contexts.

Restriction base: string

Enumeration Value	Description
Host	Specifies that the cyber observable action occurred on a host.
Network	Specifies that the cyber observable action occurred on a network.

4.2.13 NetworkProtocolEnum

NetworkProtocolEnum is a (non-exhaustive) enumeration of network protocols.

Restriction base: string

Enumeration Value	Description
TCP/IP	Specifies the TCP/IP protocol.
UDP	Specifies the UDP protocol.
DNS	Specifies the DNS protocol.
TELNET	Specifies the TELNET protocol.
HTTP	Specifies the HTTP protocol.
FTP	Specifies the FTP protocol.
IRC	Specifies the IRC protocol.
IDENT	Specifies the IDENT protocol.
POP	Specifies the POP protocol.
IMAP	Specifies the IMAP protocol.
SMB	Specifies the SMB protocol.
ARP	Specifies the ARP protocol.
Other	Specifies a network protocol other than those listed.

4.2.14 ActionNameType

The ActionNameType identifies/characterizes the specific action performed.

Property	Type	Mult	Description
Defined_Name	`cybox: DefinedActionNameEnum`	0..1	The Defined_Name element is optional and utilizes a standardized defined name to identify/characterize the specific action performed. Wherever possible, standardized defined action

			names should be utilized.
Undefined_Name	string	0..1	The Undefined_Name element is optional and utilizes a non-standardized undefined name to identify/characterize the specific action performed.

4.2.15 DefinedActionNameEnum

The DefinedActionNameEnum type is an enumeration of defined action names.

Restriction base: string

Enumeration Value	Description
Accept Socket Connection	Specifies the defined action of accepting a socket connection.
Add Scheduled Task	Specifies the defined action of adding a scheduled task.
Allocate Virtual Memory in Process	Specifies the defined action of allocating virtual memory in a process.
Bind Address to Socket	Specifies the defined action of binding an address to a socket.
Change Service Configuration	Specifies the defined action of changing the service configuration.
Check for Remote Debugger	Specifies the defined action of checking for a remote debugger.
Close Port	Specifies the defined action of closing a port.
Close Registry Key	Specifies the defined action of closing a registry key.
Close Socket	Specifies the defined action of closing a socket.
Configure Service	Specifies the defined action of configuring a service.
Connect to Named Pipe	Specifies the defined action of connecting to a named pipe.
Connect to Socket	Specifies the defined action of connecting to a socket.
Control Driver	Specifies the defined action of controlling a driver.
Control Service	Specifies the defined action of controlling a service.
Copy File	Specifies the defined action of copying a file.
Create Dialog Box	Specifies the defined action of creating a dialog box.
Create Directory	Specifies the defined action of creating a directory.
Create Event	Specifies the defined action of creating an event.
Create File	Specifies the defined action of creating a file.
Create File Alternate Data Stream	Specifies the defined action of creating a file alternate data stream.
Create File Symbolic Link	Specifies the defined action of creating a file symbolic link.
Create Mailslot	Specifies the defined action of creating a mailslot.
Create Mutex	Specifies the defined action of creating a mutex.
Create Named Pipe	Specifies the defined action of creating a named pipe.
Create Process	Specifies the defined action of creating a process.
Create Process as User	Specifies the defined action of creating a process as user.
Create Registry Key	Specifies the defined action of creating a registry key.
Create Remote Thread in Process	Specifies the defined action of creating a remote thread in a process.
Create Service	Specifies the defined action of creating a service.
Create Socket	Specifies the defined action of creating a socket.
Create Symbolic Link	Specifies the defined action of creating a symbolic link.
Create Thread	Specifies the defined action of creating a thread.
Create Window	Specifies the defined action of creating a window.
Delete Directory	Specifies the defined action of deleting a directory.
Delete File	Specifies the defined action of deleting a file.
Delete Named Pipe	Specifies the defined action of deleting a named pipe.
Delete Registry Key	Specifies the defined action of deleting a registry key.

Delete Registry Key Value	Specifies the defined action of deleting a registry key value.
Delete Service	Specifies the defined action of deleting a service.
Disconnect from Named Pipe	Specifies the defined action of disconnecting from a named pipe.
Disconnect from Socket	Specifies the defined action of disconnecting from a socket.
Enumerate Protocols	Specifies the defined action of enumerating protocols.
Enumerate Registry Key Subkeys	Specifies the defined action of enumerating registry key subkeys.
Enumerate Registry Key Values	Specifies the defined action of enumerating registry key values.
Enumerate Threads in Process	Specifies the defined action of enumerating threads in a process.
Enumerate Processes	Specifies the defined action of enumerating processes.
Enumerate Services	Specifies the defined action of enumerating services.
Enumerate Threads	Specifies the defined action of enumerating threads.
Enumerate Windows	Specifies the defined action of enumerating windows.
Find File	Specifies the defined action of finding a file.
Find Window	Specifies the defined action of finding a window.
Flush Process Instruction Cache	Specifies the defined action of flushing the Process Instruction Cache.
Free Library	Specifies the defined action of freeing a library.
Free Process Virtual Memory	Specifies the defined action of freeing virtual memory from a process.
Get Disk Free Space	Specifies the defined action of getting the amount of free space available on a disk.
Get Disk Type	Specifies the defined action of getting the disk type.
Get Elapsed System Up Time	Specifies the defined action of getting the elapsed system up-time.
Get File Attributes	Specifies the defined action of getting file attributes.
Get Function Address	Specifies the defined action of getting the function address.
Get System Global Flags	Specifies the defined action of getting system global flags.
Get Host By Address	Specifies the defined action of getting host by address.
Get Host By Name	Specifies the defined action of getting host by name.
Get Host Name	Specifies the defined action of getting the host name.
Get Library File Name	Specifies the defined action of getting the library file name.
Get Library Handle	Specifies the defined action of getting the library handle.
Get NetBIOS Name	Specifies the defined action of getting the NetBIOS name.
Get Process Current Directory	Specifies the defined action of getting the process's current directory.
Get Process Environment Variable	Specifies the defined action of getting the process environment variable.
Get Process Startup Information	Specifies the defined action of getting the process startup information.
Get Processes Snapshot	Specifies the defined action of getting the processes snapshot.
Get Service Status	Specifies the defined action of getting the service status.
Get System Host Name	Specifies the defined action of getting the system host name.
Get System Network Parameters	Specifies the defined action of getting the system network parameters.
Get System Time	Specifies the defined action of getting the system time.
Get Thread Context	Specifies the defined action of getting the thread context.
Get Thread Username	Specifies the defined action of getting the thread username.
Get Windows Directory	Specifies the defined action of getting a windows directory.
Get Windows System Directory	Specifies the defined action of getting a windows System directory.
Get Windows Temporary Files Directory	Specifies the defined action of getting the Windows Temporary Files Directory.
Hide Window	Specifies the defined action of hiding a window.
Impersonate Process	Specifies the defined action of impersonating a process.

Kill Process	Specifies the defined action of killing a process.
Kill Thread	Specifies the defined action of killing a thread.
Kill Window	Specifies the defined action of killing a window.
Listen on Socket	Specifies the defined action of listening on a socket.
Load Driver	Specifies the defined action of loading a driver.
Load Library	Specifies the defined action of loading a library.
Lock File	Specifies the defined action of locking a file.
Map File	Specifies the defined action of mapping a file.
Modify File	Specifies the defined action of modifying a file.
Modify Named Pipe	Specifies the defined action of modifying a named pipe.
Modify Process	Specifies the defined action of modifying a process.
Modify Service	Specifies the defined action of modifying a service.
Monitor Registry Key	Specifies the defined action of monitoring a registry key.
Move File	Specifies the defined action of moving a file.
Open File	Specifies the defined action of opening a file.
Open Mutex	Specifies the defined action of opening a mutex.
Open Port	Specifies the defined action of opening a port.
Open Process	Specifies the defined action of opening a process.
Open Registry Key	Specifies the defined action of opening a registry key.
Open Service	Specifies the defined action of opening a service.
Open Service Control Manager	Specifies the defined action of opening a service control manager.
Protect Virtual Memory	Specifies the defined action of protecting virtual memory.
Query Disk Attributes	Specifies the defined action of querying disk attributes.
Query DNS	Specifies the defined action of querying DNS.
Query Process Virtual Memory	Specifies the defined action of querying process virtual memory.
Queue APC in Thread	Specifies the defined action of querying the Asynchronized Procedure Call (APC) in the context of a thread.
Read File	Specifies the defined action of reading a file.
Read From Named Pipe	Specifies the defined action of reading from a named pipe.
Read From Process Memory	Specifies the defined action of reading from process memory.
Read Registry Key Value	Specifies the defined action of reading a registry key value.
Receive Data on Socket	Specifies the defined action of receiving data on a socket.
Release Mutex	Specifies the defined action of releasing a mutex.
Rename File	Specifies the defined action of renaming a file.
Send Control Code to File	Specifies the defined action of sending control code to a file.
Send Control Code to Service	Specifies the defined action of sending control code to a service.
Send Data on Socket	Specifies the defined action of sending data on a socket.
Send Data to Address on Socket	Specifies the defined action of sending data to the address on a socket.
Set File Attributes	Specifies the defined action of setting file attributes.
Set NetBIOS Name	Specifies the defined action of setting the NetBIOS name.
Set Process Current Directory	Specifies the defined action of setting the process current directory.
Set Process Environment Variable	Specifies the defined action of setting the process environment variable.
Set System Global Flags	Specifies the defined action of setting system global flags.
Set System Host Name	Specifies the defined action of setting the system host name.
Set System Time	Specifies the defined action of setting the system time.
Set Thread Context	Specifies the defined action of setting the thread context.
Show Window	Specifies the defined action of showing a window.
Start Service	Specifies the defined action of starting a service.

Unload Driver	Specifies the defined action of unloading a driver.
Unlock File	Specifies the defined action of unlocking a file.
Unmap File	Specifies the defined action of unmapping a file.
Write Registry Key Value	Specifies the defined action of writing a registry key value.
Write to File	Specifies the defined action of writing to a file.
Write to Process Virtual Memory	Specifies the defined action of writing to process virtual memory.

4.2.16 ActionAliasesType

The ActionAliasesType enables identification of other potentially used names for this Action.

Property	Type	Mult	Description
Action_Alias	string	1..∞	The Action_Alias element is optional and enables identification of a single other potentially used name for this Action.

4.2.17 ActionArgumentsType

The ActionArgumentsType enables the specification of relevant arguments/parameters for this Action.

Property	Type	Mult	Description
Action_Argument	`cybox: ActionArgumentType`	1..∞	The Action_Argument element is optional and enables the specification of a single relevant argument/parameter for this Action.

4.2.18 ActionArgumentType

The ActionArgumentType enables the specification of a single relevant argument/parameter for this Action.

Property	Type	Mult	Description
Argument_Name-Defined	`cybox: DefinedArgumentNameEnum`	0..1	The Argument_Name-Defined element is optional and utilizes a standardized defined name to identify/characterize the specific action argument utilized. Wherever possible, standardized defined argument names should be utilized.
Argument_Name-Undefined	string	0..1	The Argument_Name-Undefined element is optional and utilizes a non-standardized undefined name to identify/characterize the specific action argument utilized.
Argument_Value	string	0..1	The Argument_Value attribute specifies the value for this action argument/parameter.

4.2.19 DefinedArgumentNameEnum

The DefinedArgumentNameEnum type is an enumeration of defined argument names.

Restriction base: string

Enumeration Value	Description
API	Specifies an argument called API.
Creation Flags	Specifies an argument called Creation Flags.
Access Mode	Specifies an argument called Access Mode.
Share Mode	Specifies an argument called Share Mode.

Callback Address	Specifies an argument called Callback Address.
Source Address	Specifies an argument called Source Address.
Destination Address	Specifies an argument called Destination Address.
Starting Address	Specifies an argument called Starting Address.
Size (bytes)	Specifies an argument called Size (bytes).
Control Parameter	Specifies an argument called Control Parameter.
Host Name	Specifies an argument called Host Name.
Function Name	Specifies an argument called Function Name.
Function Address	Specifies an argument called Function Address.
Options	Specifies an argument called Options.
Transfer Flags	Specifies an argument called Transfer Flags.
Control Code	Specifies an argument called Control Code.
APC Mode	Specifies an argument called APC Mode.
APC Address	Specifies an argument called APC Address.
Base Address	Specifies an argument called Base Address.

4.2.20 AssociatedObjectsType

The AssociatedObjectsType enables the description/specification of cyber Objects relevant to an Action.

Property	Type	Mult	Description
Associated_Object	cybox:AssociatedObjectType	1..∞	The Associated_Object element enables the description of cyber Objects associated with this Action. This could include Objects that initiated the action, are the target Objects affected by the Action, are utilized by the Action or are the returned result of the Action.

4.2.21 AssociatedObjectType (extends cybox:ObjectType)

The AssociatedObjectType is a complex type representing the characterization of a cyber observable Object associated with a given cyber observable Action.

Property	Type	Mult	Description
association_type	cybox:AssociationTypeEnum	1..1	The association_type attribute specifies the kind of association this Object holds for this Action.
ActionPertinentObjectAttributes	cybox:ActionPertinentObjectAttributesType	0..1	The Action-Pertinent_Object_Attributes element is optional and identifies which of the Attributes of this Object are specifically pertinent to this Action.

4.2.22 AssociationTypeEnum

AssociationTypeEnum is a (non-exhaustive) enumeration of types of object-action associations.

Restriction base: string

Enumeration Value	Description
Initiating	Specifies that the associated object initiated the action.
Affected	Specifies that the associated object was affected by the action.
Utilized	Specifies that the associated object was utilized by the action.
Returned	Specifies that the associated object was the result of the action.

4.2.23 ActionPertinentObjectAttributesType

The ActionPertinentObjectAttributesType identifies which of the Attributes of this Object are specifically pertinent to this Action.

Property	Type	Mult	Description
Attribute	cybox: ActionPertinent ObjectAttributeType	1..∞	The Attribute element identifies a single Object Attribute that is specifically pertinent to this Action.

4.2.24 ActionPertinentObjectAttributeType

The ActionPertinentObjectAttributeType identifies one of the Attributes of an Object that specifically pertinent to an Action.

Property	Type	Mult	Description
name	string	1..1	The name attribute specifies the field name for the pertinent Object Attribute.
xpath	string	1..1	The xpath attribute specifies the XPath expression identifying the pertinent attribute within the Defined_Object schema for this object type.

4.2.25 RelationshipsType

The RelationshipsType enables description of other cyber observable actions that are related to this Action.

Property	Type	Mult	Description
Relationship	cybox: ActionRelationshipType	1..∞	The Relationship element is optional and enables description of a single other cyber observable action that is related to this Action.

4.2.26 ActionRelationshipType

The ActionRelationshipType is a complex type characterizing a relationship between a specified cyber observable action and another cyber observable action.

Property	Type	Mult	Description
type	cybox: ActionRelationshipTypeEnum	1..1	The type attribute describes the nature of the relationship between this Action and the related Action.
Action_Reference	cybox:ActionReferenceType	1..∞	The Action_Reference element captures references to other Actions.

4.2.27 RelationshipTypeEnum

RelationshipTypeEnum is a (non-exhaustive) enumeration of types of relationships between cyber observable elements.

Restriction base: string

Enumeration Value	Description
Parent_Of	Specifies that this entity (e.g. Action) is the parent of the associated entity.
Child_Of	Specifies that this entity (e.g. Action) is a child of the associated entity.
Preceded_By	Specifies that this entity (e.g. Action) is preceded by the associated entity.

Followed_By	Specifies that this entity (e.g. Action) is followed by the associated entity.
Downloaded_From	Specifies that this entity (e.g. Action) is downloaded from the associated entity.
Uploaded_To	Specifies that this entity (e.g. Action) is uploaded to the associated entity.
Related_To	Specifies that this entity (e.g. Action) is related to the associated entity.
Dropped_By	Specifies that this entity (e.g. Action) is dropped by the associated entity.
Contained_Within	Specifies that this entity (e.g. Action) is contained within the associated entity.
Installed_By	Specifies that this entity (e.g. Action) is installed by the associated entity.
Dependent_On	Specifies that this entity (e.g. Action) is dependent on the associated entity.
Resolves_To	Specifies that this entity (e.g. Action) resolves to the associated entity.

4.2.28 ActionRelationshipTypeEnum

The ActionRelationshipTypeEnum is an enumeration of types of relationships between actions.

Restriction base: cybox:RelationshipTypeEnum

Enumeration Value	Description
Preceded_By	Specifies that this action is preceded by the related action.
Followed_By	Specifies that this action is followed by the related action.
Related_To	Specifies that this action is simply related to the related action in some way.
Dependent_On	Specifies that this action is dependent on the related action.

4.2.29 ActionReferenceType

ActionReferenceType is intended to serve as a method for linking to actions.

Property	Type	Mult	Description
action_id	QName	1..1	The action_id attribute refers to the id of the action being referenced.

4.2.30 ObjectType

The ObjectType is a complex type representing the characteristics of a specific cyber-relevant object (e.g. a file, a registry key or a process).

Property	Type	Mult	Description
id	QName	1..1	The id attribute specifies a unique id for this Object.
idref	QName	1..1	The idref attribute specifies a unique id reference to an Object defined elsewhere.
object_state	cybox:ObjectStateTypeEnum	1..1	The object_state attribute enables description of the current state of the object.
type	cybox:ObjectTypeEnum	1..1	The type attribute specifies what kind of object this is.
*******		1..1	The "any" attribute enables the capture of custom attributes describing this Object.
Description	Common:StructuredTextType	0..1	The Description element provides a mechanism to specify a structured text description of this Object.
Defined_Object	Common:DefinedObjectType	0..1	The Defined_Object element is an abstract placeholder for various predefined Object type schemas (e.g. File, Process or System) that can be instantiated in its place through extension of the DefinedObjectType. This mechanism enables the specification of a broad range of Object types with consistent Object Attribute naming and structure.

			The set of Defined_Object schemas are maintained independent of the core CybOX schema.
Domain-specific_Object_Attributes	cybox: DomainSpecificObjectAttributesType	0..1	The Domain_Specific_Object_Attributes element is of an Abstract type placeholder within the CybOX schema enabling the inclusion of domain-specific metadata for an object through the use of a custom type defined as an extension of this base Abstract type. This enables domains utilizing CybOX such as malware analysis or forensics to incorporate non-generalized object metadata from their domains into CybOX objects.
Custom_Attributes	cybox: CustomAttributesType	0..1	The Custom_Attributes element is optional and enables the specification of a set of custom Object Attributes that may not be defined in existing Defined_Object schemas.
Related_Objects	cybox: RelatedObjectsType	0..1	The Related_Objects element is optional and enables the identification and/or specification of Objects with relevant relationships with this Object.
Defined_Effect	cybox: DefinedEffectType	0..1	The Defined_Effect element is an abstract placeholder for various predefined Object Effect types (e.g. DataReadEffect, ValuesEnumeratedEffect or StateChangeEffect) that can be instantiated in its place through extension of the DefinedEffectType. This mechanism enables the specification of a broad range of types of potential complex action effects on Objects. The set of Defined_Effect types (extending the DefeinedEffectType) are maintained as part of the core CybOX schema.
Discovery_Method	Common: MeasureSourceType	0..1	The Discovery_Method element is optional and enables descriptive specification of how this Object was observed (in the case of a Cyber Observable Object instance) or could potentially be observed (in the case of a Cyber Observable Object pattern).

4.2.31 ObjectTypeEnum

ObjectTypeEnum is a (non-exhaustive) enumeration of cyber observable object types.

Restriction base: string

Enumeration Value	Description
File	Specifies a file object.
Directory	Specifies a directory object.
Module	Specifies a module object.
Network Packet	Specifies a network packet object.
Key/Key Group	Specifies a key or key group object.
Hive	Specifies a hive object.
Process	Specifies a process object.
Thread	Specifies a thread object.
Mutex	Specifies a mutex object.
Event log	Specifies an event log object.
Service/Daemon	Specifies a Service/Daemon object.
Library	Specifies a library object.

Package	Specifies a package object.
Pipe	Specifies a pipe object.
Socket	Specifies a socket object.
IP Address	Specifies a IP Address object.
Port	Specifies a port object.
Protocol	Specifies a protocol object.
ASN	Specifies an Autonomous System Number object.
URI	Specifies a URI object.
Host	Specifies a host object.
Session	Specifies a session object.
Session Token	Specifies a session token object.
Account	Specifies an account object.
Device (physical)	Specifies a physical device object.
Handle	Specifies a handle object.
Heap	Specifies a heap object.
Memory Address	Specifies a memory address object.
Memory Page	Specifies a memory page object.
Window	Specifies a window object.
Dialog	Specifies a dialog object.
Parameter	Specifies a parameter object.
Authentication Token	Specifies an authentication token object.
Encryption Token	Specifies an encryption token object.
Web Query	Specifies a web query object.
Protocol Header	Specifies a protocol header object.
Protocol Field	Specifies a protocol field object.
Link	Specifies a link object.
SQL Query	Specifies an SQL query object.
Database	Specifies a database object.
ACL	Specifies an Access Control List object.
Role	Specifies a role object.
System	Specifies a system object.
VM	Specifies a Virtual Machine (VM) object.
Signature	Specifies a signature object.
Channel	Specifies a channel object.
API	Specifies an API object.
Environment Variable	Specifies an environment variable object.
Application	Specifies an application object.
Network	Specifies a network object.
Configuration	Specifies a configuration object.
Policy	Specifies a policy object.
Task	Specifies a task object.
Malware	Specifies a malware object.
Message	Specifies a message object.
Email Message	Specifies an e-mail message object.
Media	Specifies a media object.
Operating System	Specifies an Operating System (OS) object.
Query	Specifies a query object.
Domain	Specifies a domain object.
Critical Section	Specifies a critical section object.

Mailslot	Specifies a mailslot object.
NamedPipe	Specifies a named pipe object.
Semaphore	Specifies a semaphore object.
WaitableTimer	Specifies a waitable timer object.
Volume	Specifies a volume object.
Disk	Specifies a disk object.
DiskPartition	Specifies a disk partition object.
Other	Specifies an object of Other type.

4.2.32 ObjectStateTypeEnum

ObjectStateTypeEnum is a (non-exhaustive) enumeration of cyber observable object states.

Restriction base: string

Enumeration Value	Description
Exists	Specifies that the object exists.
Does Not Exist	Specifies that the object does not exist.
Open	Specifies that the object is open.
Closed	Specifies that the object is closed.
Active	Specifies that the object is active.
Inactive	Specifies that the object is inactive.
Locked	Specifies that the object is locked.
Unlocked	Specifies that the object is unlocked.
Started	Specifies that the object has started.
Stopped	Specifies that the object has stopped.

4.2.33 DomainSpecificObjectAttributesType (abstract)

The DomainSpecificObjectAttributesType is an Abstract type placeholder within the CybOX schema enabling the inclusion of domain-specific metadata for an object through the use of a custom type defined as an extension of this base Abstract type. This enables domains utilizing CybOX such as malware analysis or forensics to incorporate non-generalized object metadata from their domains into CybOX objects.

4.2.34 CustomAttributesType

The CustomAttributesType enables the specification of a set of custom Object Attributes that may not be defined in existing Defined_Object schemas.

Property	Type	Mult	Description
Attribute	cybox:Attribute	1..∞	The Attribute element enables the specification of a single Object Attribute.

4.2.35 RelatedObjectsType

The RelatedObjectsType enables the identification and/or specification of Objects with relevant relationships with this Object.

Property	Type	Mult	Description
Related_Object	`cybox:RelatedObjectType`	1..∞	The Related_Objects element enables the identification and/or specification of an Object with a relevant relationship with this Object.

4.2.36 RelatedObjectType (extends cybox:ObjectType)

The RelatedObjectType enables the identification and/or specification of an Object with a relevant relationship with this Object.

Property	Type	Mult	Description
relationship	`cybox:ObjectRelationshipEnum`	1..1	The Relationship attribute specifies the nature of the relationship between this Object and the Related_Object.

4.2.37 ObjectRelationshipEnum

ObjectRelationshipEnum is a (non-exhaustive) enumeration of interobject relationships.

Restriction base: string

Enumeration Value	Description
Created_By	Specifies that this object was created by the related object.
Deleted_By	Specifies that this object was deleted by the related object.
Properties_Modified_By	Specifies that the properties of this object were modified by the related object.
Read_From	Specifies that this object was read from the related object.
Read_From_By	Specifies that this object was read from by the related object.
Written_To_By	Specifies that this object was written to by the related object.
Downloaded_From	Specifies that this object was downloaded from the related object.
Downloaded_To	Specifies that this object downloaded the related object.
Downloaded_By	Specifies that this object was downloaded by the related object.
Uploaded_To	Specifies that this object was uploaded to the related object.
Uploaded_By	Specifies that this object was uploaded by the related object.
Suspended_By	Specifies that this object was suspended by the related object.
Paused_By	Specifies that this object was paused by the related object.
Resumed_By	Specifies that this object was resumed by the related object.
Opened_By	Specifies that this object was opened by the related object.
Closed_By	Specifies that this object was closed by the related object.
Copied_From	Specifies that this object was copied from the related object.
Copied_To	Specifies that this object was copied to the related object.
Copied_By	Specifies that this object was copied by the related object.
Moved_From	Specifies that this object was moved from the related object.
Moved_To	Specifies that this object was moved to the related object.
Moved_By	Specifies that this object was moved by the related object.
Searched_For	Specifies that this object searched for the related object.
Searched_For_By	Specifies that this object was searched for by the related object.
Allocated_By	Specifies that this object was allocated by the related object.
Initialized_To	Specifies that this object was initialized to the related object.
Initialized_By	Specifies that this object was initialized by the related object.
Sent_To	Specifies that this object was sent to the related object.
Sent_From	Specifies that this object was sent from the related object.
Sent_By	Specifies that this object was sent by the related object.
Received_From	Specifies that this object was received from the related object.
Received_By	Specifies that this object was received by the related object.
Mapped_Into	Specifies that this object was mapped into the related object.
Mapped_By	Specifies that this object was mapped by the related object.
Properties_Queried	Specifies that the object queried properties of the related object.

Properties_Queried_By	Specifies that the properties of this object were queried by the related object.
Values_Enumerated	Specifies that the object enumerated values of the related object.
Values_Enumerated_By	Specifies that the values of the object were enumerated by the related object.
Bound_By	Specifies that this object was bound by the related object.
Freed_By	Specifies that this object was freed by the related object.
Killed_By	Specifies that this object was killed by the related object.
Encrypted_By	Specifies that this object was encrypted by the related object.
Encrypted_To	Specifies that this object was encrypted to the related object.
Decrypted_By	Specifies that this object was decrypted by the related object.
Decrypted_To	Specifies that this object decrypted the related object.
Unpacked_By	Specifies that this object was unpacked by the related object.
Packed_By	Specifies that this object was packed by the related object.
Encoded_By	Specifies that this object was encoded by the related object.
Decoded_By	Specifies that this object was decoded by the related object.
Compressed_Into	Specifies that this object was compressed into the related object.
Compressed_By	Specifies that this object was compressed by the related object.
Decompressed_Into	Specifies that this object was decompressed into the related object.
Decompressed_By	Specifies that this object was decompressed by the related object.
Joined_By	Specifies that this object was joined by the related object.
Joined_Into	Specifies that this object was joined into the related object.
Merged_Into	Specifies that this object was merged into the related object.
Merged_By	Specifies that this object was merged by the related object.
Locked_By	Specifies that this object was locked by the related object.
Unlocked_By	Specifies that this object was unlocked by the related object.
Hooked_By	Specifies that this object was hooked by the related object.
Unhooked_By	Specifies that this object was unhooked by the related object.
Monitored_By	Specifies that this object was monitored by the related object.
Listened_On	Specifies that this object listened on the related object.
Listened_On_By	Specifies that this object was listened on by the related object.
Renamed_From	Specifies that this object was renamed from the related object.
Renamed_To	Specifies that this object was renamed to the related object.
Renamed_By	Specifies that this object was renamed by the related object.
Injected_Into	Specifies that this object injected into the related object.
Injected_As	Specifies that this object injected as the related object.
Injected_By	Specifies that this object was injected by the related object.
Deleted_From	Specifies that this object was deleted from the related object.
Loaded_Into	Specifies that this object loaded into the related object.
Loaded_From	Specifies that this object was loaded from the related object.
Set_To	Specifies that this object was set to the related object.
Resolved_To	Specifies that this object was resolved to the related object.
Parent_Of	Specifies that this object is a parent of the related object.
Child_Of	Specifies that this object is a child of the related object.
Related_To	Specifies that this object is related to the related object.
Dropped_By	Specifies that this object was dropped by the related object.
Contained_Within	Specifies that this object is contained within the related object.
Installed_By	Specifies that this object was installed by the related object.
Connected_To	Specifies that this object connected to the related object.

4.2.38 DefinedEffectType (abstract)

The DefinedEffectType is an abstract placeholder for various predefined Object Effect types (e.g. DataReadEffect, ValuesEnumeratedEffect or StateChangeEffect) that can be instantiated in its place through extension of the DefinedEffectType. This mechanism enables the specification of a broad range of types of potential complex action effects on Objects. The set of Defined_Effect types (extending the DefinedEffectType) are maintained as part of the core CybOX schema.

Property	Type	Mult	Description
effect_type	cybox:EffectTypeEnum	1..1	The effect_type attribute specifies the nature of the Defined Effect instantiated in the place of the Defined_Effect element.

4.2.39 EffectTypeEnum

EffectTypeEnum is a (non-exhaustive) enumeration of effect types.

Restriction base: string

Enumeration Value	Description
State_Changed	Specifies that the associated Action had an effect on the Object of changing its state.
Data_Read	Specifies that the associated Action had an effect on the Object of reading data from it.
Data_Written	Specifies that the associated Action had an effect on the Object of writing data to it.
Data_Sent	Specifies that the associated Action had an effect on the Object of sending data to it.
Data_Received	Specifies that the associated Action had an effect on the Object of receiving data from it.
Properties_Read	Specifies that the associated Action had an effect on the Object of reading properties from it.
Properties_Enumerated	Specifies that the associated Action had an effect on the Object of enumeraring properies from it.
Values_Enumerated	Specifies that the associated Action had an effect on the Object of enumerating values from it.
ControlCode_Sent	Specifies that the associated Action had an effect on the Object of having a control code sent to it.

4.2.40 StateChangeEffectType (extends cybox:DefinedEffectType)

The StateChangeEffectType is intended as a generic way of characterizing the effects of actions upon objects where the some state of the object is changed.

Property	Type	Mult	Description
Old_State	cybox:StateType	0..1	The Old_State element specifies the object and its attributes as they were before the state change effect occurred.
New_State	cybox:StateType	1..1	The New_State element specifies the object and its attributes as they are after the state change effect occurred.

4.2.41 StateType

The StateType characterizes the state of an Object.

Property	Type	Mult	Description
Object	cybox:ObjectType	1..1	The Object element identifies and specificies the

			characteristics of a specific cyber-relevant object (e.g. a file, a registry key or a process).
Defined_Object	Common: DefinedObjectType	1..1	The Defined_Object element is an abstract placeholder for various predefined Object type schemas (e.g. File, Process or System) that can be instantiated in its place through extension of the DefinedObjectType. This mechanism enables the specification of a broad range of Object types with consistent Object Attribute naming and structure. The set of Defined_Object schemas are maintained independent of the core CybOX schema.
Object_IDRef	QName	1..1	The Object_IDRef element specifies a unique id reference to an Object defined elsewhere.

4.2.42 DataReadEffectType (extends cybox:DefinedEffectType)

The DataReadEffectType type is intended to characterize the effects of actions upon objects where some data is read, such as from a file or a pipe.

Property	Type	Mult	Description
Data	Common:DataSegmentType	1..1	The Data element specifies the data that was read from the object by the action.

4.2.43 DataWrittenEffectType (extends cybox:DefinedEffectType)

The DataWrittenEffectType type is intended to characterize the effects of actions upon objects where some data is written, such as to a file or a pipe.

Property	Type	Mult	Description
Data	Common:DataSegmentType	1..1	The Data element specifies the data that was written to the object by the action.

4.2.44 DataSentEffectType (extends cybox:DefinedEffectType)

The DataSentEffectType type is intended to characterize the effects of actions upon objects where some data is sent, such as a byte sequence on a socket.

Property	Type	Mult	Description
Data	Common:DataSegmentType	1..1	The Data element specifies the data that was sent on the object, or from the object, by the action.

4.2.45 DataReceivedEffectType (extends cybox:DefinedEffectType)

The DataReceivedEffectType type is intended to characterize the effects of actions upon objects where some data is received, such as a byte sequence on a socket.

Property	Type	Mult	Description
Data	Common:DataSegmentType	1..1	The Data element specifies the data that was received on the object, or from the object, by the action.

4.2.46 PropertyReadEffectType (extends cybox:DefinedEffectType)

The PropertyReadEffectType type is intended to characterize the effects of actions upon objects where some specific property is read from an object, such as the current running state of a process.

Property	Type	Mult	Description
Name	string	0..1	The Name element specifies the Name of the property being read.
Value	string	0..1	The Value element specifies the value of the property being read.

4.2.47 PropertiesEnumeratedEffectType (extends cybox:DefinedEffectType)

The PropertiesEnumeratedEffectType type is intended to characterize the effects of actions upon objects where some properties of the object are enumerated, such as the startup parameters for a process.

Property	Type	Mult	Description
Properties	`cybox:PropertiesType`	1..1	The Properties element specifies the properties that were enumerated as a result of the action on the object.

4.2.48 PropertiesType

The PropertiesType specifies the properties that were enumerated as a result of the action on the object.

Property	Type	Mult	Description
Property	string	1..∞	The Property elemet specifies a single property that was enumerated as a result of the action on the object.

4.2.49 ValuesEnumeratedEffectType (extends cybox:DefinedEffectType)

The ValuesEnumeratedEffectType type is intended to characterize the effects of actions upon objects where some values of the object are enumerated, such as the values of a registry key.

Property	Type	Mult	Description
Values	`cybox:ValuesType`	1..1	The Values element specifies the values that were enumerated as a result of the action on the object.

4.2.50 ValuesType

The ValuesType specifies the values that were enumerated as a result of the action on the object.

Property	Type	Mult	Description
Value	string	1..∞	The Value element specifies a single value that was enumerated as a result of the action on the object.

4.2.51 SendControlCodeEffectType (extends cybox:DefinedEffectType)

The SendControlCodeEffectType is intended to characterize the effects of actions upon objects where some control code, or other control-oriented communication signal, is sent to the object. For example, an action may send a control code to change the running state of a process.

Property	Type	Mult	Description
Control_Code	string	1..1	The Control_Code element specifies the actual control code that was sent to the object.

4.2.52 AttributeType (extends Common:BaseObjectAttributeType)

The AttibuteType is a complex type representing the specification of a single Object Attribute.

Property	Type	Mult	Description
name	string	1..1	The name attribute specifies a name for this attribute.

4.2.53 ObservableCompositionType

The ObservablesCompositionType enables the specification of higher-order composite observables composed of logical combinations of other observables.

Property	Type	Mult	Description
operator	`cybox:OperatorTypeEnum`	1..1	The operator attribute enables the specification of complex compositional cyber observables by providing logical operators for defining interrelationships between constituent cyber observables defined utilizing the recursive Observable element.
Observable	`cybox:ObservableType`	0..∞	The Observable element represents a description of a single cyber observable.

4.2.54 OperatorTypeEnum

OperatorTypeEnum is a (non-exhaustive) enumeration of operators.

Restriction base: string

Enumeration Value	Description
AND	Specifies the AND logical composition operation.
OR	Specifies the OR logical composition operation.
NOT	Specifies the NOT logical composition operation.

4.2.55 PoolsType

The PoolsType enables the description of Events, Actions, Objects and Attributes in a space-efficient pooled manner with the actual Observable structures defined in the CybOX schema containing references to the pooled elements. This reduces redundancy caused when identical observable elements occur multiple times within a set of defined Observables.

Property	Type	Mult	Description
Event_Pool	`cybox:EventPoolType`	0..1	The Event_Pool element enables the description of CybOX Events in a space-efficient pooled manner with the actual Observable structures defined in the CybOX schema containing references to the pooled Event elements. This reduces redundancy caused when identical Events occur multiple times within a set of defined Observables.
Action_Pool	`cybox:ActionPoolType`	0..1	The Action_Pool element enables the description

			of CybOX Actions in a space-efficient pooled manner with the actual Observable structures defined in the CybOX schema containing references to the pooled Action elements. This reduces redundancy caused when identical Actions occur multiple times within a set of defined Observables.
Object_Pool	cybox:ObjectPoolType	0..1	The Object_Pool element enables the description of CybOX Objects in a space-efficient pooled manner with the actual Observable structures defined in the CybOX schema containing references to the pooled Object elements. This reduces redundancy caused when identical Objects occur multiple times within a set of defined Observables.
Attribute_Pool	cybox:AttributePoolType	0..1	The Attribute_Pool element enables the description of CybOX Attributes in a space-efficient pooled manner with the actual Observable structures defined in the CybOX schema containing references to the pooled Attributes elements. This reduces redundancy caused when identical Attributes occur multiple times within a set of defined Observables.

4.2.56 EventPoolType

The EventPoolType enables the description of CybOX Events in a space-efficient pooled manner with the actual Observable structures defined in the CybOX schema containing references to the pooled Event elements. This reduces redundancy caused when identical Events occur multiple times within a set of defined Observables.

Property	Type	Mult	Description
Event	cybox:EventType	1..∞	The Event element enables specification of a cyber observable event that is dynamic in nature with specific action(s) taken against specific cyber relevant objects (e.g. a file is deleted, a registry key is created or an HTTP Get Request is received).

4.2.57 ActionPoolType

The ActionPoolType enables the description of CybOX Actions in a space-efficient pooled manner with the actual Observable structures defined in the CybOX schema containing references to the pooled Action elements. This reduces redundancy caused when identical Actions occur multiple times within a set of defined Observables.

Property	Type	Mult	Description
Action	cybox:ActionType	1..∞	The Action element enables description/specification of a single cyber observable action.

4.2.58 ObjectPoolType

The ObjectPoolType enables the description of CybOX Objects in a space-efficient pooled manner with the actual Observable structures defined in the CybOX schema containing references to the pooled Object elements. This reduces redundancy caused when identical Objects occur multiple times within a set of defined Observables.

Property	Type	Mult	Description
Object	cybox:ObjectType	1..∞	The Object element identifies and specificies the characteristics of a specific cyber-relevant object (e.g. a file, a registry key or a process).

4.2.59 AttributePoolType

The AttributePoolType enables the description of CybOX Attributes in a space-efficient pooled manner with the actual Observable structures defined in the CybOX schema containing references to the pooled Attributes elements. This reduces redundancy caused when identical Attributes occur multiple times within a set of defined Observables.

Property	Type	Mult	Description
Attribute	cybox:AttributeType	1..∞	The Attribute element enables the specification of a single Object Attribute.

4.2.60 NoisinessEnum

NoisinessEnum is a (non-exhaustive) enumeration of potential levels of noisiness for a given observable pattern.

Restriction base: string

Enumeration Value	Description
High	Specifies that this observable has a high level of noisiness meaning a potentially high level of false positives.
Medium	Specifies that this observable has a medium level of noisiness meaning a potentially medium level of false positives.
Low	Specifies that this observable has a low level of noisiness meaning a potentially low level of false positives.

4.2.61 ObfuscationTechniquesType

The ObfuscationTechniquesType enables the description of a set of potential techniques an attacker could leverage to obfuscate the observability of this Observable.

Property	Type	Mult	Description
Obfuscation_Technique	cybox: ObfuscationTechniqueType	1..∞	The Obfuscation_Technique element is optional and enables the description of a single potential technique an attacker could leverage to obfuscate the observability of this Observable.

4.2.62 ObfuscationTechniqueType

The ObfuscationTechniqueType enables the description of a single potential technique an attacker could leverage to obfuscate the observability of this Observable.

Property	Type	Mult	Description
Description	Common: StructuredTextType	1..1	The Description element captures a structured text description of the obfuscation technique.
Observables	cybox:ObservablesType	0..1	The Observables element is optional and enables description of potential cyber observables that could indicate the use of this particular obfuscation technique.

4.2.63 EaseOfObfuscationEnum

The EaseOfObfuscationEnum is a (non-exhaustive) enumeration of simple characterizations of how easy it would be for an attacker to obfuscate the observability of this Observable.

Restriction base: string

Enumeration Value	Description
High	Specifies that this observable is very easy to obfuscate and hide.
Medium	Specifies that this observable is somewhat easy to obfuscate and hide.
Low	Specifies that this observable is not very easy to obfuscate and hide.

1.1 Common Data Types

4.2.64 MeasureSourceType

The MeasureSourceType is a complex type representing a description of a single cyber observation source.

Property	Type	Mult	Description
analysis_method	Common: AnalysisMethodTypeEnum	1..1	The analysis_method attribute is optional and (when analysis is used) enables identification of the method of analysis utilized as part of this cyber observation source.
analysis_type	Common:AnalysisTypeEnum	1..1	The analysis_type attribute is optional and (when analysis is used) enables identification of the type of analysis utilized as part of this cyber observation source.
class	Common: SourceClassTypeEnum	1..1	The class attribute is optional and enables identification of the high-level class of this cyber observation source.
information_source_type	Common: Information SourceTypeEnum	1..1	The information_sourceType attribute is optional and enables identification of the type of information source leveraged for this cyber observation source.
name	string	1..1	The name attribute is optional and enables the assignment of a relevant name to a this Discovery Method.
source_type	Common: SourceTypeEnum	1..1	The source_type attribute is optional and enables identification of the broad type of this cyber observation source.
tool_type	Common:ToolTypeEnum	1..1	The tool_type attribute is optional and (when tools are used) enables identification of the type of tool leveraged as part of this cyber observation source.

Description	Common: StructuredTextType	0..1	The Description element is optional and enables a generalized but structured description of this syber observation source.
Contributors	Common:PersonnelType	0..1	The Contributors element is optional and enables description of the individual contributors involved in this cyber observation source.
Time	Common:TimeType	0..1	The Time element is optional and enables description of various time-related attributes for this cyber observation source instance.
Tools	Common: ToolsInformationType	0..1	The Tools element is optional and enables description of the tools utilized for this cyber observation source.
Indicators	Common:IndicatorsType	0..1	The Indicators element is optional and enables the inclusion of varying specifications for indicators contributing to this cyber observation.
Platform	Common: CPESpecificationType	0..1	The Platform element is optional and enables a formal, standardized specification of the platform for this cyber observation srouce.
System	SystemObj: SystemObjectType	0..1	The System element is optional and enables characterization of the system on which the mechanism of cyber observation executed.
Instance	ProcessObj: ProcessObjectType	0..1	The Instance element is optional and enables characterization of the process instance in which the mechanism of cyber observation executed.

4.2.65 SourceClassTypeEnum

The SourceClassTypeEnum is a (non-exhaustive) enumeration of cyber observation source classes.

Restriction base: string

Enumeration Value	Description
Network	Describes a Network-based cyber observation.
System	Describes a System-based cyber observation.
Software	Describes a Software-based cyber observation.

4.2.66 SourceTypeEnum

The SourceTypeEnum is a (non-exhaustive) enumeration of cyber observation source types.

Restriction base: string

Enumeration Value	Description
Tool	Describes a cyber observation made using various tools, such as scanners, firewalls, gateways, protection systems, and detection systems. See ToolTypeEnum for a more complete list of tools that CybOX supports.
Analysis	Describes a cyber observation made from analysis methods, such as Static and Dynamic methods. See AnalysisMethodTypeEnum for a more complete list of methods that CybOX supports.
InformationSource	Describes a cyber observation made using other information sources, such as logs, Device Driver APIs, and TPM output data. See InformationSourceTypeEnum for a more complete list of information sources that CybOX supports.

4.2.67 ToolTypeEnum

The ToolTypeEnum is a (non-exhaustive) enumeration of cyber observation source tool types.

Restriction base: string

Enumeration Value	Description
NIDS	The NIDS value specifies the Network Intrusion Detection System tool.
NIPS	The NIPS value specifies the Network Intrusion Protection System tool.
HIDS	The HIDS value specifies the Host-based Intrusion Detection System tool.
HIPS	The HIPS value specifies the Host-based Intrusion Protection System tool.
Firewall	The Firewall value specifies a cyber observation made using a firewall.
Router	The Router value specifies a cyber observation made using a router.
Proxy	The Proxy value specifies a cyber observation made using a network proxy.
Gateway	The Gateway value specifies a cyber observation made using a network gateway.
SNMP/MIBs	The SNMP/MIBs value specifies a cyber observation made using the Simple Network Management Protocol or via the Management Information Bases.
A/V	The A/V value specifies a cyber observation made using Anti-Virus tools and/or software.
DBMS Monitor	The DBMS value specifies a cyber observation made using a Database Management System monitor.
Vulnerability Scanner	The Vulnerability Scanner value specifies a cyber observation made using a vulnerability scanner.
Configuration Scanner	The Configuration Scanner value specifies a cyber observation made using a configuration scanner.
Asset Scanner	The Asset Scanner value specifies a cyber observation made using an asset scanner.
SIM	The SIM value specifies a cyber observation made using Security Information Management tools.
SEM	The SEM value specifies a cyber observation made using Security Event Management tools.

4.2.68 AnalysisTypeEnum

The AnalysisTypeEnum is a (non-exhaustive) enumeration of types of cyber observation source analysis.

Restriction base: string

Enumeration Value	Description
Anomaly Detection	The Anomaly Detection value specifies anomaly detection analysis.

4.2.69 AnalysisMethodTypeEnum

The AnalysisMethodTypeEnum is a (non-exhaustive) enumeration of cyber observation source analysis methods.

Restriction base: string

Enumeration Value	Description
Static Analysis	The Static Analysis value specifies a cyber observation made using static analysis methods.
Dynamic Analysis	The Dynamic Analysis value specifies a cyber observation made using dynamic analysis methods.
Other	The Other value specifies a cyber observation made using other analysis methods.

4.2.70 InformationSourceTypeEnum

The InformationSourceTypeEnum is a (non-exhaustive) enumeration of cyber observation information source types.

Restriction base: string

Enumeration Value	Description
Comm Logs	The Comm Logs value specifies a cyber observation coming from communications logs.
Application Logs	The Application Logs value specifies a cyber observation coming from application logs.
Web Logs	The Web Logs value specifies a cyber observation coming from web logs.
DBMS Log	The DBMS Log value specifies a cyber observation coming from the Database Management System log.
OS/Device Driver APIs	The OS/Device Driver APIs value specifies a cyber observation coming from OS/Device Driver APIs.
Frameworks	The Frameworks value specifies a cyber observation coming from Frameworks.
VM Hypervisor	The VM Hypervisor value specifies a cyber observation coming from the VM hypervisor data.
TPM	The TPM value specifies a cyber observation made using TPM output data.
Application Framework	The Application Framework value specifies a cyber observation coming from an application framework.
Help Desk	The Help Desk value specifies a cyber observation coming from an human or automated help desk.
Incident Management	The Incident Management value specifies a cyber observation made using information provided by Incident Management services.
IAVM	The IAVM value specifies a cyber observation made using information provided by Information Assurance Vulnerability Management mechanisms.

4.2.71 ContributorType

The ContributorType represents a description of an individual who contributed as a source of cyber observation data.

Property	Type	Mult	Description
Role	string	0..1	This field describes the role played by this contributor.
Name	string	0..1	This field contains the name of this contributor.
Email	string	0..1	This field contains the email of this contributor.
Phone	string	0..1	This field contains a telephone number of this contributor.
Organization	string	0..1	This field contains the organization name of this contributor.
Date	`Common:DateRangeType`	0..1	This field contains a description (bounding) of the timing of this contributor's involvement.
Contribution_Location	string	0..1	This field contains information describing the location at which the contributory activity occured.

4.2.72 DateRangeType

The DateRangeType specifies a range of dates.

Property	Type	Mult	Description
Start_Date	date	0..1	This field contains the start date for this contributor's involvement.
End_Date	date	0..1	This field contains the end date for this contributor's involvement.

4.2.73 PersonnelType

The PersonnelType is an abstracted data type to standardize the description of sets of personnel.

Property	Type	Mult	Description
Contributor	Common:ContributorType	1..∞	This field contains information describing the identify, resources and timing of involvement for a single contributor.

4.2.74 TimeType

The TimeType specifies various time properties for a cyber observation source.

Property	Type	Mult	Description
Start_Time	dateTime	0..1	The Start_Time element is optional and describes the starting time for this cyber observation source instance.
End_Time	dateTime	0..1	The End_Time element is optional and describes the ending time for this cyber observation source instance.
Produced_Time	dateTime	0..1	The Produced_Time element is optional and describes the time that this cyber observation source instance was produced.
Received_Time	dateTime	0..1	The Received_Time element is optional and describes the time that this cyber observation source instance was received.

4.2.75 ToolSpecificDataType (abstract)

The ToolSpecificDataType is an Abstract type placeholder within the CybOX schema enabling the inclusion of metadata for a specific type of tool through the use of a custom type defined as an extension of this base Abstract type.

4.2.76 ToolsInformationType

The ToolsInformationType represents a description of a set of automated tools.

Property	Type	Mult	Description
Tool	Common: ToolInformationType	1..∞	The Tool element is optional and enables description of a single tool utilized for this cyber observation source.

4.2.77 ToolInformationType

The ToolInformationType represens a description of a single automated tool.

Property	Type	Mult	Description
id	QName	1..1	The id attribute specifies a unique ID for this Tool.
idref	QName	1..1	The idref attribute specifies reference to a unique ID for this Tool.
Vendor	string	0..1	This field contains information identifying the vendor organization for this tool.
Name	string	0..1	This field contains the name of the tool leveraged.
Version	string	0..1	This field contains an appropriate version descriptor of this tool.

Service_Pack	string	0..1	This field contains an appropriate service pack descriptor for this tool.
Tool-specific_Data	Common: ToolSpecificDataType	0..1	This is an abstract type provided to a flexible mechanism for enabling tool-specific data to be included.
Tool_Hashes	Common: HashListType	0..1	This field contains a hash value computed on the tool file content in order to verify its integrity.
Tool_Configuration	Common: ToolConfigurationType	0..1	This field contains information describing the configuration and usage of the tool.
Execution_Environment	Common: ExecutionEnvironmentType	0..1	This field contains information describing the execution environment of the tool.
Errors	Common: ErrorsType	0..1	This field captures any errors generated during the run of the tool.
Metadata	Common: MetadataType	0..∞	This field captures other relevant metadata including tool-specific fields.

4.2.78 ToolConfigurationType

The ToolConfigurationType characterizes the configuration for a tool used as a cyber observation source.

Property	Type	Mult	Description
Configuration_Settings	Common: ConfigurationSettingsType	1..1	This field describes the configuration settings of this tool instance.
Dependencies	Common: DependenciesType	0..1	This field contains information describing the relevant dependencies for this tool.
Usage_Context_Assumptions	Common: UsageContextAssumptionsType	0..1	This field contains descriptions of the various relevant usage context assumptions for this tool .
Internationalization_Settings	Common: InternationalizationSettingsType	0..1	This field contains information describing relevant internationalization setting for this tool .
Build_Information	Common: BuildInformationType	0..1	This field contains information describing how this tool was built.

4.2.79 ConfigurationSettingsType

The ConfigurationSettingsType is a modularized data type used to provide a consistent approach to describing configuration settings for a tool, application or other cyber object

Property	Type	Mult	Description
Configuration_Setting	Common: ConfigurationSettingType	1..∞	This field contains a single configuration setting instance.

4.2.80 ConfigurationSettingType

The ConfigurationSettingType is a modularized data type used to provide a consistent approach to describing a particular configuration setting for a tool, application or other cyber object

Property	Type	Mult	Description
Item_Name	string	1..1	This field contains the name of the configuration item referenced by this configuration setting instance.
Item_Value	string	1..1	This field contains the value of this configuration

			setting instance.
Item_Type	string	0..1	This field contains the type of the configuration item referenced in this configuration setting instance.
Item_Description	string	0..1	This field contains a description of the configuration item referenced in this configuration setting instance.

4.2.81 DependenciesType

The DependenciesType contains information describing a set of dependencies for this tool.

Property	Type	Mult	Description
Dependency	Common: DependencyType	1..∞	This field contains information describing a single dependency for this tool.

4.2.82 DependencyType

The DependencyType contains information describing a single dependency for this tool.

Property	Type	Mult	Description
Dependency_Type	string	0..1	This field describes the type of this dependency instance.
Dependency_Description	Common: StructuredTextType	1..1	This field contains a description of this dependency instance.

4.2.83 UsageContextAssumptionsType

The UsageContextAssumptionsType contains descriptions of the various relevant usage context assumptions for this tool

Property	Type	Mult	Description
Usage_Context_Assumption	Common: StructuredTextType	1..∞	This field contains a single usage context assumption for this tool.

4.2.84 InternationalizationSettingsType

The InternationalizationSettingsType contains information describing relevant internationalization setting for this tool

Property	Type	Mult	Description
Internal_Strings	Common: InternalStringsType	1..∞	This field contains a single internal string instance for this internationalization setting instance.

4.2.85 InternalStringsType

The InternalStringsType contains a single internal string instance for this internationalization setting instance.

Property	Type	Mult	Description
Key	string	1..1	This field contains the actual key of this internal string instance.
Content	string	1..1	This field contains the actual content of this internal string instance.

4.2.86 BuildInformationType

The BuildInformationType contains information describing how this tool was built.

Property	Type	Mult	Description
Build_ID	string	0..1	This field contains an externally defined unique identifier of this build of this application instance.
Build_Project	string	0..1	This field contains the project name of this build of this application instance.
Build_Utility	Common: BuildUtilityType	0..1	This field contains information identifying the utility used to build this application.
Build_Version	string	0..1	This field contains the appropriate version descriptor of this build of this application instance.
Build_Label	string	0..1	This field contains any relevant label for this build of this application instance.
Compilers	Common: CompilersType	0..1	This field describes the compilers utilized during this build of this application.
Compilation_Date	dateTime	0..1	This field identifies the compilation date for the build of the tool.
Build_Configuration	Common: BuildConfigurationType	0..1	This field describes how the build utility was configured for this build of this application.
Build_Script	string	0..1	This field contains the actual build script for this build of this application instance.
Libraries	Common:LibrariesType	0..1	This field identifies the libraries incorporated into the build of the tool.
Build_Output_Log	string	0..1	This field contains a capture of the output log of the build process.

4.2.87 BuildUtilityType

The BuildUtilityType contains information identifying the utility used to build this application.

Property	Type	Mult	Description
Build_Utility_Name	string	1..1	This field contains the informally defined name of the utility used to build this application instance.
Build_Utility_CPE_Specification	Common: CPESpecificationType	1..1	This field contains the CPE specification data to formally define the build utility used to build this application.

4.2.88 CompilersType

The CompilersType describes the compilers utilized during this build of this application.

Property	Type	Mult	Description
Compiler	Common:CompilerType	1..∞	This field describes a single compiler utilized during this build of this application.

4.2.89 CompilerType

The CompilerType describes a single compiler utilized during this build of this application.

Property	Type	Mult	Description
Compiler_Informal_	Common: CompilerInformal	0..1	This field contains the informal description of this

Description	DescriptionType		compiler instance.
Compiler_CPE_ Specification	Common: CPESpecificationType	0..1	This field contains the CPE specification data to formally define this compiler instance.

4.2.90 CompilerInformalDescriptionType

The CompilerInformalDescriptionType contains the informal description of this compiler instance.

Property	Type	Mult	Description
Compiler_Name	string	1..1	This field contains the name of the compiler.
Compiler_Version	string	0..1	This field contains the version of the compiler.

4.2.91 BuildConfigurationType

The BuildConfigurationType describes how the build utility was configured for this build of this application.

Property	Type	Mult	Description
Configuration_Setting_ Description	string	0..1	This field contains the description of the configuration settings for this build of this application instance.
Configuration_Settings	Common: ConfigurationSettingsType	1..1	This field contains the configuration settings for this build of this application instance.

4.2.92 LibrariesType

The LibrariesType identifies the libraries incorporated into the build of the tool.

Property	Type	Mult	Description
Library	Common:LibraryType	0..1	This field identifies a library incorporated into the build of the tool.

4.2.93 LibraryType

The LibraryType identifies a single library incorporated into the build of the tool.

Property	Type	Mult	Description
name	string	1..1	This field identifies the name of the library.
version	string	1..1	This field identifies the version of the library.

4.2.94 ExecutionEnvironmentType

The ExecutionEnvironmentType contains information describing the execution environment of the tool.

Property	Type	Mult	Description
System	SystemObj: SystemObjectType	0..1	This field contains information describing the system on which the tool was executed.
User_Account_Info	UserAccountObj: UserAccountObjectType	0..1	This field contains information describing the user account that executed the tool.
Command_Line	string	0..1	This field specifies the command line string used to run the tool.
Start_Time	dateTime	0..1	Thie field specifies when the tool was run.

4.2.95 ErrorsType

The ErrorsType captures any errors generated during the run of the tool.

Property	Type	Mult	Description
Error	Common:ErrorType	1..∞	This field captures a single type of error generated during the run of the tool.

4.2.96 ErrorType

The ErrorType captures a single error generated during the run of the tool.

Property	Type	Mult	Description
Error_Type	string	1..1	This field specifies the the type for this tool run error.
Error_Count	integer	0..1	This field specifies the count of instances for this error in the tool run.
Error_Instances	Common: ErrorInstancesType	0..1	This field captures the actual error output for each instance of this type of error.

4.2.97 ErrorInstancesType

The ErrorInstancesType captures the actual error output for each instance of this type of error.

Property	Type	Mult	Description
Error_Instance	string	1..∞	This field captures the actual error output for a single instance of this type of error.

4.2.98 IndicatorsType

The IndicatorsType identifies any indicators that contributed to this cyber observation source.

Property	Type	Mult	Description
Indicator	Common:IndicatorType	1..∞	The Indicator element is optional and enables the inclusion of varying specifications for a single indicator contributing to this cyber observation.

4.2.99 IndicatorType (abstract)

The IndicatorType is an Abstract type placeholder within the CybOX schema enabling the inclusion of varying specifications for indicators contributing to this cyber observation. Externally defined indicator structures can be defined through the use of a custom type defined as an extension of this base Abstract type.

4.2.100 DefinedObjectType (abstract)

The DefinedObjectType is an Abstract type placeholder within the CybOX schema enabling the inclusion of contextually varying object descriptions. This Abstract type is leveraged as the extension base for all predefined CybOX object schemas. Through this extension mechanism any object instance data based on an object schema extended from DefinedObjectType (e.g. File_Object, Address_Object, etc.) can be directly integrated into any instance document where a field is defined as DefinedObjectType. For flexibility and extensibility purposes any user of CybOX can specify their own externally defined object schemas (outside of or derived from the set of predefined objects) extended from DefinedObjectType and utilize them as part of their CybOX content.

Property	Type	Mult	Description
object_reference	QName	1..1	The ObjectReference attribute specifies a unique ID

			reference to an Object defined elsewhere. This construct enables flexibility in defining Object association within Actions as well as specification of Objects within other Objects.
***		1..1	The "any" attribute enables the capture of custom attributes describing this Defined Object specification.

4.2.101 BaseObjectAttributeType (abstract) (extends xs:anySimpleType)

The BaseObjectAttibuteType is a complex type representing a common typing foundation for the specification of a single Object Attribute.

4.2.102 IntegerObjectAttributeType (restriction Common:BaseObjectAttributeType)

The IntegerObjectAttributeType is a complex type (extended from BaseObjectAttributeType) representing the specification of a single Object attribute whose core value is of type Int. This type will be assigned to any attribute of a CybOX object that should contain content of type Integer and enables the use of relevant metadata for the atrribute.

Data restrictions: int, Common:EmptyStringType

Property	Type	Mult	Description
datatype	`Common:DatatypeEnum`	1..1	This attribute is optional and specifies the expected type for the value of the specified element.

4.2.103 StringObjectAttributeType (restriction Common:BaseObjectAttributeType)

The StringObjectAttributeType is a complex type (extended from BaseObjectAttributeType) representing the specification of a single Object attribute whose core value is of type String. This type will be assigned to any attribute of a CybOX object that should contain content of type String and enables the use of relevant metadata for the atrribute.

Property	Type	Mult	Description
datatype	`Common:DatatypeEnum`	1..1	This attribute is optional and specifies the expected type for the value of the specified element.

4.2.104 NameObjectAttributeType (restriction Common:BaseObjectAttributeType)

The NameObjectAttributeType is a complex type (extended from BaseObjectAttributeType) representing the specification of a single Object attribute whose core value is of type Name. This type will be assigned to any attribute of a CybOX object that should contain content of type Name and enables the use of relevant metadata for the atrribute.

Property	Type	Mult	Description
datatype	`Common:DatatypeEnum`	1..1	This attribute is optional and specifies the expected type for the value of the specified element.

4.2.105 DateObjectAttributeType (restriction Common:BaseObjectAttributeType)

The DateObjectAttributeType is a complex type (extended from BaseObjectAttributeType) representing the specification of a single Object attribute whose core value is of type Date. This type will be assigned to any attribute of a CybOX object that should contain content of type Date and enables the use of relevant metadata for the atrribute.

Property	Type	Mult	Description
datatype	`Common:DatatypeEnum`	1..1	This attribute is optional and specifies the expected type for the value of the specified element.

4.2.106 DateTimeObjectAttributeType (restriction Common:BaseObjectAttributeType)

The DateTimeObjectAttributeType is a complex type (extended from BaseObjectAttributeType) representing the specification of a single Object attribute whose core value is of type DateTime. This type will be assigned to any attribute of a CybOX object that should contain content of type DateTime and enables the use of relevant metadata for the atrribute.

Property	Type	Mult	Description
datatype	`Common:DatatypeEnum`	1..1	This attribute is optional and specifies the expected type for the value of the specified element.

4.2.107 FloatObjectAttributeType (restriction Common:BaseObjectAttributeType)

The FloatObjectAttributeType is a complex type (extended from BaseObjectAttributeType) representing the specification of a single Object attribute whose core value is of type Float. This type will be assigned to any attribute of a CybOX object that should contain content of type Float and enables the use of relevant metadata for the atrribute.

Data restrictions: float, Common:EmptyStringType

Property	Type	Mult	Description
datatype	`Common:DatatypeEnum`	1..1	This attribute is optional and specifies the expected type for the value of the specified element.

4.2.108 DoubleObjectAttributeType (restriction Common:BaseObjectAttributeType)

The DoubleObjectAttributeType is a complex type (extended from BaseObjectAttributeType) representing the specification of a single Object attribute whose core value is of type Double. This type will be assigned to any attribute of a CybOX object that should contain content of type Double and enables the use of relevant metadata for the atrribute.

Data restrictions: double, Common:EmptyStringType

Property	Type	Mult	Description
datatype	`Common:DatatypeEnum`	1..1	This attribute is optional and specifies the expected type for the value of the specified element.

4.2.109 UnsignedLongObjectAttributeType (restriction Common:BaseObjectAttributeType)

The UnsignedLongObjectAttributeType is a complex type (extended from BaseObjectAttributeType) representing the specification of a single Object attribute whose core value is of type UnsignedLong. This type will be assigned to any attribute of a CybOX object that should contain content of type UnsignedLong and enables the use of relevant metadata for the atrribute.

Data restrictions: unsignedLong, Common:EmptyStringType

Property	Type	Mult	Description

datatype	Common:DatatypeEnum	1..1	This attribute is optional and specifies the expected type for the value of the specified element.

4.2.110 UnsignedIntegerObjectAttributeType (restriction Common:BaseObjectAttributeType)

The UnsignedIntegerObjectAttributeType is a complex type (extended from BaseObjectAttributeType) representing the specification of a single Object attribute whose core value is of type UnsignedInt. This type will be assigned to any attribute of a CybOX object that should contain content of type UnsignedInteger and enables the use of relevant metadata for the atrribute.

Data restrictions: unsignedInt, Common:EmptyStringType

Property	Type	Mult	Description
datatype	Common:DatatypeEnum	1..1	This attribute is optional and specifies the expected type for the value of the specified element.

4.2.111 PositiveIntegerObjectAttributeType (restriction Common:BaseObjectAttributeType)

The PositiveIntegerObjectAttributeType is a complex type (extended from BaseObjectAttributeType) representing the specification of a single Object attribute whose core value is of type PositveInteger. This type will be assigned to any attribute of a CybOX object that should contain content of type PositiveInteger and enables the use of relevant metadata for the atrribute.

Data restrictions: positiveInteger, Common:EmptyStringType

Property	Type	Mult	Description
datatype	Common:DatatypeEnum	1..1	This attribute is optional and specifies the expected type for the value of the specified element.

4.2.112 HexBinaryObjectAttributeType (restriction Common:BaseObjectAttributeType)

The HexBinaryObjectAttributeType is a complex type (extended from BaseObjectAttributeType) representing the specification of a single Object attribute whose core value is of type HexBinary. This type will be assigned to any attribute of a CybOX object that should contain content of type HexBinary and enables the use of relevant metadata for the atrribute.

Property	Type	Mult	Description
datatype	Common:DatatypeEnum	1..1	This attribute is optional and specifies the expected type for the value of the specified element.

4.2.113 LongObjectAttributeType (restriction Common:BaseObjectAttributeType)

The LongObjectAttributeType is a complex type (extended from BaseObjectAttributeType) representing the specification of a single Object attribute whose core value is of type Long. This type will be assigned to any attribute of a CybOX object that should contain content of type Long and enables the use of relevant metadata for the atrribute.

Data restrictions: long, Common:EmptyStringType

Property	Type	Mult	Description
datatype	Common:DatatypeEnum	1..1	This attribute is optional and specifies the expected type for the value of the specified element.

4.2.114 NonNegativeIntegerObjectAttributeType (restriction Common:BaseObjectAttributeType)

The NonNegativeIntegerObjectAttributeType is a complex type (extended from BaseObjectAttributeType) representing the specification of a single Object attribute whose core value is of type nonNegativeInteger. This type will be assigned to any attribute of a CybOX object that should contain content of type NonNegativeInteger and enables the use of relevant metadata for the atrribute.

Data restrictions: nonNegativeInteger, Common:EmptyStringType

Property	Type	Mult	Description
datatype	Common:DatatypeEnum	1..1	This attribute is optional and specifies the expected type for the value of the specified element.

4.2.115 AnyURIObjectAttributeType (restriction Common:BaseObjectAttributeType)

The AnyURIObjectAttributeType is a complex type (extended from BaseObjectAttributeType) representing the specification of a single Object attribute whose core value is of type anyURI. This type will be assigned to any attribute of a CybOX object that should contain content of type AnyURI and enables the use of relevant metadata for the atrribute.

Property	Type	Mult	Description
datatype	Common:DatatypeEnum	1..1	This attribute is optional and specifies the expected type for the value of the specified element.

4.2.116 DurationObjectAttributeType (restriction Common:BaseObjectAttributeType)

The DurationObjectAttributeType is a complex type (extended from BaseObjectAttributeType) representing the specification of a single Object attribute whose core value is of type duration. This type will be assigned to any attribute of a CybOX object that should contain content of type Duration and enables the use of relevant metadata for the atrribute.

Property	Type	Mult	Description
datatype	Common:DatatypeEnum	1..1	This attribute is optional and specifies the expected type for the value of the specified element.

4.2.117 TimeObjectAttributeType (restriction Common:BaseObjectAttributeType)

The TimeObjectAttributeType is a complex type (extended from BaseObjectAttributeType) representing the specification of a single Object attribute whose core value is of type time. This type will be assigned to any attribute of a CybOX object that should contain content of type Time and enables the use of relevant metadata for the atrribute.

Property	Type	Mult	Description
datatype	Common:DatatypeEnum	1..1	This attribute is optional and specifies the expected type for the value of the specified element.

4.2.118 Base64BinaryObjectAttributeType (restriction Common:BaseObjectAttributeType)

The Base64BinaryObjectAttributeType is a complex type (extended from BaseObjectAttributeType) representing the specification of a single Object attribute whose core value is of type base64Binary. This type will be assigned to any attribute of a CybOX object that should contain content of type Base64Binary and enables the use of relevant metadata for the atrribute.

Property	Type	Mult	Description
datatype	Common:DatatypeEnum	1..1	This attribute is optional and specifies the expected type for the value of the specified element.

4.2.119 ObjectAttributeGroup

The ObjectAttributeGroup is a simple attribute group aggregating a set of attributes for Object Attributes.

Property	Type	Mult	Description
appears_random	boolean	1..1	This attribute is optional and conveys whether the associated object attribute value appears to somewhat random in nature. An object attribute with this attribute set to TRUE need not provide any further information including a value. If more is known about the particular variation of randomness, a regex value could be provided to outline what is known of the structure.
condition	Common: ConditionTypeEnum	1..1	This attribute is optional and defines the relevant condition to apply to the value of this Object Attribute.
datatype	Common: DatatypeEnum	1..1	This attribute is optional and specifies the expected type for the value of the specified element.
end_range	Common: RangeValueType	1..1	This attribute is optional and defines the ending range for the element. This is applicable only if the Condition attribute is set to 'IsInRange' or 'IsNotInRange'.
has_changed	boolean	1..1	This attribute is optional and conveys a targeted observation pattern of whether the associated object attribute value has changed. This attribute would be leveraged within a pattern observable triggering on whether the value of a single object attribute field has changed.
id	QName	1..1	The id attribute specifies a unique ID for this Object Attribute.
idref	QName	1..1	The idref attribute specifies a unique ID reference for this Object Attribute.
pattern_type	Common: PatternTypeEnum	1..1	This attribute is optional and defines the type of pattern used if one is specified for the Object Attribute. This is applicable only if the Condition attribute is set to 'FitsPattern'.
regex_syntax	Common: RegexSyntaxEnum	1..1	This attribute is optional and defines the syntax format used for a regular expression, if one is specified for the Object Attribute. This is applicable only if the Condition attribute is set to 'FitsPattern'.
start_range	Common:RangeValueType	1..1	This attribute is optional and defines the starting range for the element. This is applicable only if the Condition attribute is set to 'IsInRange' or 'IsNotInRange'.
trend	boolean	1..1	This attribute is optional and conveys a targeted observation pattern of the nature of any trend in the associated object attribute value. This attribute

			would be leveraged within a pattern observable triggering on the matching of a specified trend in the value of a single object attribute field.
value_set	string	1..1	This attribute is optional and defines a set of values, using commas as delimiters, that the element may have. Ex: value1,value2,value3.

4.2.120 ConditionTypeEnum

ConditionTypeEnum is a (non-exhaustive) enumeration of condition types.

Restriction base: string

Enumeration Value	Description
Equals	Specifies the equality or = condition.
DoesNotEqual	Specifies the "does not equal" or != condition.
Contains	Specifies the "contains" condition.
DoesNotContain	Specifies the "does not contain" condition.
StartsWith	Specifies the "starts with" condition.
EndsWith	Specifies the "ends with" condition.
GreaterThan	Specifies the "greater than" condition.
GreaterThanOrEqual	Specifies the "greater than or equal to" condition.
LessThan	Specifies the "less than" condition.
LessThanOrEqual	Specifies the "less than or equal" condition.
IsInRange	Specifies the condition that a value is in range.
IsNotInRange	Specifies the condition that a value is not in range.
IsInSet	Specifies the condition that a value is in a given set.
IsNotInSet	Specifies the condition that a value is not in a given set.
FitsPattern	Specifies the condition that a value fits a given pattern.
BitwiseAnd	Specifies the condition of bitwise AND.
BitwiseOr	Specifies the condition of bitwise OR.

4.2.121 DatatypeEnum

DataTypeEnum is a (non-exhaustive) enumeration of data types.

Restriction base: string

Enumeration Value	Description
String	Specifies the String datatype as it applies to the W3C standard. See http://www.w3.org/TR/xmlschema-2/#string for more information.
Int	Specifies the Int datatype as it applies to the W3C standard for int. See http://www.w3.org/TR/xmlschema-2/#int for more information.
Float	Specifies the Float datatype as it apples to the W3C standard. See http://www.w3.org/TR/xmlschema-2/#float for more information.
IPv4 Address	Specifies an IPV4 address in dotted decimal form. CIDR notation is also accepted.
IPv6 Address	Specifies an IPV6 address, which is represented by eight groups of 16-bit hexadecimal values separated by colons (:) in the form a:b:c:d:e:f:g:h. CIDR notation is also accepted.
Host Name	Specifies a host name. For compatability reasons, this could be any string. Even so, it is best to use the proper notation for the given host type. For example, web hostnames should be written as fully qualified hostnames in practice.
MAC Address	Specifies a MAC address, which is represented by six groups of 2 hexdecimal digits, separated by hyphens (-) or colons (:) in transmission order.

Domain Name	Specifies a domain name, which is represented by a series of labels concatenated with dots comforming to the rules in RFC 1035, RFC 1123, and RFC 2181.
URI	Specifies a Uniform Resource Identifier, which identifies a name or resource and can act as a URL or URN.
Date	Specifies a date, which is usually in the form yyyy-mm--dd as it applies to the W3C standard. See http://www.w3.org/TR/xmlschema-2/#date for more information.
PositiveInteger	Specifies a positive integer in the infinite set {1,2,...} as it applies to the W3C standard. See http://www.w3.org/TR/xmlschema-2/#positiveInteger for more information.
UnsignedInt	Specifies an unsigned integer, which is a nonnegative integer in the set {0,1,2,...,4294967295} as it applies to the W3C standard. See http://www.w3.org/TR/xmlschema-2/#unsignedInt for more information.
DateTime	Specifies a date in full format including both date and time as it applies to the W3C standard. See http://www.w3.org/TR/xmlschema-2/#dateTime for more information.
Time	Specifies a time as it applies to the W3C standard. See http://www.w3.org/TR/xmlschema-2/#time for more information.
Boolean	Specifies a boolean value in the set {true,false,1,0} as it applies to the W3C standard. See http://www.w3.org/TR/xmlschema-2/#boolean for more information.
Name	Specifies a name (which represents XML Names) as it applies to the W3C standard. See http://www.w3.org/TR/xmlschema-2/#Name and http://www.w3.org/TR/2000/WD-xml-2e-20000814#dt-name for more information.
Long	Specifies a long integer, which is an integer whose maximum value is 9223372036854775807 and minimum value is -9223372036854775808 as it applies to the W3C standard. See http://www.w3.org/TR/xmlschema-2/#long for more information.
UnsignedLong	Specifies an unsigned long integer, which is an integer whose maximum value is 18446744073709551615 and minimum value is 0 as it applies to the W3C standard. See http://www.w3.org/TR/xmlschema-2/#unsignedLong for more information.
Duration	Specifies a length of time in the extended format PnYn MnDTnH nMnS, where nY represents the number of years, nM the number of months, nD the number of days, 'T' is the date/time separator, nH the number of hours, nM the number of minutes and nS the number of seconds, as it applies to the W3 standard. See http://www.w3.org/TR/xmlschema-2/#duration for more information.
Double	Specifies a decimal of datatype double as it is patterned after the IEEE double-precision 64-bit floating point type (IEEE 754-1985) and as it applies to the W3C standard. See http://www.w3.org/TR/xmlschema-2/#double for more information.
TimeZone	Specifies a timezone in UTC notation (UTC+number).
NonNegativeInteger	Specifies a non-negative integer in the infinite set {0,1,2,...} as it applies to the W3C standard. See http://www.w3.org/TR/xmlschema-2/#nonNegativeInteger for more information.
hexBinary	Specifies arbitrary hex-encoded binary data as it applies to the W3C standard. See http://www.w3.org/TR/xmlschema-2/#hexBinary for more information.
AnyURI	Specifies a Uniform Resource Identifier Reference (URI) as it applies to the W3C standard and to RFC 2396, as amended by RFC 2732. See http://www.w3.org/TR/xmlschema-2/#anyURI for more information.
Octal	Specifies arbitrary octal (base-8) encoded data.

Binary	Specifies arbitrary binary encoded data.
BinHex	Specifies arbitrary data encoded in the Mac OS-originated BinHex format.
Subnet Mask	Specifies a subnet mask in IPv4 or IPv6 notation.
UUID/GUID	Specifies a globally/universally unique ID represented as a 32-character hexadecimal string. See ISO/IEC 11578:1996 Information technology -- Open Systems Interconnection -- Remote Procedure Call - http://www.iso.ch/cate/d2229.html
Collection	Specifies data represented as a container of multiple data of a shared elemental type.
CVE#	Specifies a CVE#, expressed as CVE- appended by a four-digit integer, as in CVE-3333.
CWE#	Specifies a CWE#, expressed as CWE- appended by an integer.
CAPEC#	Specifies a CAPEC#, expressed as CAPEC- appended by an integer.
CCE#	Specifies a CCE#, expressed as CCE- appended by an integer.
CPE Name	Specifies a CPE Name. See http://cpe.mitre.org/specification/archive/version2.0/cpe-specification_2.0.pdf for more information.
Base64Binary	Specifies base64-encoded arbitrary binary data as it applies to the W3C standard. See http://www.w3.org/TR/xmlschema-2/#base64Binary for more information.

4.2.122 **EmptyStringType**

The EmptyStringType simple type is a restriction of the built-in string simpleType. The only allowed string is the empty string with a length of zero. This type is used by certain elements to allow empty content when non-string data is accepted.

4.2.123 **PatternTypeEnum**

The PatternTypeEnum type is a non-exhaustive enumeration of potentially relevant pattern types.

Restriction base: string

Enumeration Value	Description
Regex	Specifies the regular expression pattern type.
Binary	Specifies the binary (bit operations) pattern type.
XPath	Specifies the XPath expression pattern type.

4.2.124 **RegexSyntaxEnum**

The RegexSyntaxEnum type is a non-exhaustive enumeration of Regular Expression (Regex) syntaxes.

Restriction base: string

Enumeration Value	Description
POSIX BRE	Specifies that the Regex follows POSIX Basic Regular Expression rules.
POSIX ERE	Specifies that the Regex follows POSIX Extended Regular Expression rules.
Perl 5.x	Specifies that the Regex follows the rules in the Perl 5.x. programming language.
JGSoft	Specifies that the Regex follows the rules in the JGSoft Regex engine.
.NET	Specifies that the Regex follows the rules in .NET programming languages.
Java	Specifies that the Regex follows the rules in the Java programming language.
PCRE	Specifies that the Regex follows Perl Compatible Regular Expression rules.
ECMA	Specifies that the Regex follows the rules of the ECMA (Javascript) standard. See http://www.ecma-international.org/publications/standards/Ecma-262.htm
Python	Specifies that the Regex follows the rules in the Python programming language.

Ruby	Specifies that the Regex follows the rules in the Ruby programming language.
Tcl ARE	Specifies that the Regex follows TCL Advanced Regular Expression rules.
GNU BRE	Specifies that the Regex follows GNU Basic Regular Expression rules.
GNU ERE	Specifies that the Regex follows GNU Extended Regular Expression rules.
XML	Specifies that the Regex follows the rules in the XML programming language.
XPath	Specifies that the Regex follows the rules according to an XPath.

4.2.125 RangeValueType

The RangeValueType simple type is a union of datatypes applicable for use in specifiying a value range.

Union Type
int
double
float
date
dateTime
long
unsignedLong
unsignedInt
nonNegativeInteger

4.2.126 ExtractedFeaturesType

The ExtractedFeaturesType is a complex type representing a description of features extracted from an object such as a file.

Property	Type	Mult	Description
Strings	Common:ExtractedStringsType	0..1	This field enables description of a set of static strings extracted from a raw cyber object.
Imports	Common:ImportsType	0..1	This field enables description of a set of references to external resources imported by a raw cyber object.
Functions	Common:FunctionsType	0..1	This field enables description of a set of references to functions called by a raw cyber object.
Code_Snippets	Common:CodeSnippetsType	0..1	This field enables description of a set of code snippets extracted from a raw cyber object.

4.2.127 ExtractedStringsType

The ExtractedStringsType type is intended as container for strings extracted from CybOX objects.

Property	Type	Mult	Description
String	Common:ExtractedStringType	1..∞	This field enables description of a single static string extracted from a raw cyber object.

4.2.128 ExtractedStringType

The ExtractedStringType type is intended as container a single string extracted from a CybOX object.

Property	Type	Mult	Description
encoding	Common: CharacterEncodingEnum	1..1	The encoding attribute refers to the encoding method used for the string extracted from the

			CybOX object.
String_Value	Common: StringObject AttributeType	0..1	The String_Value element specifies the actual value of the string extracted from the CybOX object.
Hashes	Common:HashListType	0..1	The Hashes element is used to include any hash values computed using the string extracted from the CybOX object as input.
Address	Common: HexBinary ObjectAttributeType	0..1	The Address element specifies the location or offset of the specified string in the CybOX objects.
Length	Common: PositiveIntegerObject AttributeType	0..1	The Length element specifies the length, in characters, of the string extracted from the CybOX object.
Language	Common: StringObject AttributeType	0..1	The Language element specifies the language the string is written in, e.g. English.
English_Translation	Common: StringObject AttributeType	0..1	The English_Translation element specifies the English translation of the string, if it is not written in English.

4.2.129 CharacterEncodingEnum

CharacterEncodingEnum is a (non-exhaustive) enumeration of character encodings.

Restriction base: string

Enumeration Value	Description
ANSI	Indicates ANSI encoding for the string extracted from the CybOX object.
Unicode	Indicates Unicode encoding for the string extracted from the CybOX object.
Other	Indicates a different encoding than those listed for the string extracted from the CybOX object.

4.2.130 ImportsType

The ImportsType is intended to represent an extracted list of imports specified within a CybOX object.

Property	Type	Mult	Description
Import	Common: StringObject AttributeType	1..∞	This field enables description of a single reference to an external resource imported by a raw cyber object.

4.2.131 FunctionsType

The FunctionsType is intended to represent an extracted list of functions leveraged within a CybOX object.

Property	Type	Mult	Description
Function	Common: StringObject AttributeType	1..∞	This field enables description of a single reference to a function called by a raw cyber object.

4.2.132 CodeSnippetsType

The CodeSnippetsType is intended to represent an set of code snippets extracted from within a CybOX object.

Property	Type	Mult	Description

Code_Snippet	CodeObj:CodeObjectType	1..∞	This field enables description of a single code snippet extracted from a raw cyber object.

4.2.133 ByteRunsType

The ByteRunsType is used for representing a list of byte runs from within a raw object.

Property	Type	Mult	Description
Byte_Run	Common:ByteRunType	1..∞	The Byte_Run element contains a single byte run from the raw object.

4.2.134 ByteRunType

The ByteRunType is used for representing a single byte run from within a raw object.

Property	Type	Mult	Description
Offset	Common:IntegerObjectAttributeType	0..1	The Offset element specifies the offset of the beginning of the byte run as measured from the beginning of the object.
File_System_Offset	Common:IntegerObjectAttributeType	0..1	The File_System_Offset element is relevant only for byte runs of files in forensic analysis.It specifies the offset of the beginning of the byte run as measured from the beginning of the relevant file system.
Image_Offset	Common:IntegerObjectAttributeType	0..1	The Image_Offset element is provided for forensic analysis purposes and specifies the offset of the beginning of the byte run as measured from the beginning of the relevant forensic image.
Length	Common:IntegerObjectAttributeType	0..1	The Length element specifies the number of bytes in the byte run.
Hashes	Common:HashListType	0..1	The Hashes element contains computed hash values for this the data in this byte run.
Byte_Run_Data	anyType	0..1	The Byte_Run_Data element contains a raw dump of the byte run data, typically enclosed within an XML CDATA section.

4.2.135 HashListType

The HashListType type is used for representing a list of hash values.

Property	Type	Mult	Description
Hash	Common:HashType	1..∞	The Hash element specifies a single calculated hash value.

4.2.136 HashValueType

The HashValueType is used for specifying the resulting value from a hash calculation.

Property	Type	Mult	Description
Simple_Hash_Value	Common:SimpleHashValueType	0..1	The Simple_Hash_Value element specifies a single result value of a basic cryptograhic hash function outputing a single hexbinary hash value.

Fuzzy_Hash_Value	Common:FuzzyHashValueType	0..1	The Fuzzy_Hash_Value element specifies a single result value of a cryptograhic fuzzy hash function outputing a single complex string based hash value. (e.g. SSDEEP's Block1hash:Block2hash format).

4.2.137 SimpleHashValueType (extends Common:HexBinaryObjectAttributeType)

The SimpleHashValueType is used for characterizing the output of basic cryptograhic hash functions outputing a single hexbinary hash value.

4.2.138 FuzzyHashValueType (extends Common:StringObjectAttributeType)

The FuzzyHashValueType is used for characterizing the output of cryptograhic fuzzy hash functions outputing a single complex string based hash value.

4.2.139 FuzzyHashStructureType

The FuzzyHashStructureType is used for characterizing the internal components of a cryptograhic fuzzy hash algorithmic calculation.

Property	Type	Mult	Description
Block_Size	Common:IntegerObjectAttributeType	0..1	The Block_Size element is optional and specifies the calculated block size for this fuzzy hash calculation.
Block_Hash	Common:FuzzyHashBlockType	0..1	The Block_Hash element is optional and enables specification of the elemental components utilized for a fuzzy hash calcuation on the hashed object utilizing Block_Size to calculate trigger points.

4.2.140 FuzzyHashBlockType

The FuzzyHashBlockType is used for characterizing the internal components of a single block in a cryptograhic fuzzy hash algorithmic calculation.

Property	Type	Mult	Description
Block_Hash_Value	Common:HashValueType	0..1	The Block_Hash_Value element is optional and specifies a fuzzy hash calculation result value for this Block.
Segment_Count	Common:IntegerObjectAttributeType	0..1	The Segment_Count element is optional and specifies the number of segments identified and utlized within this fuzzy hash calculation.
Segments	Common:HashSegmentsType	0..1	The Segments element is optional and specifies the set of segments identified and utlized within this fuzzy hash calculation.

4.2.141 HashSegmentsType

The HashSegmentsType is used for characterizing the internal components of a set of trigger point-delimited segments in a cryptograhic fuzzy hash algorithmic calculation.

Property	Type	Mult	Description
Segment	Common:HashSegmentType	1..∞	The Segment element is optional and specifies a single segment identified and utlized within this fuzzy hash calculation.

4.2.142 HashSegmentType

The HashSegmentType is used for characterizing the internal components of a single trigger point-delimited segment in a cryptograhic fuzzy hash algorithmic calculation.

Property	Type	Mult	Description
Trigger_Point	Common: HexBinary ObjectAttributeType	0..1	The Trigger_point element is optional and specifies the offset within the hashed object of the trigger point for this segment.
Segment_Hash	Common:HashValueType	0..1	The Segment_Hash element is optional and specifies a calculated hash value for this segment.
Raw_Segment_Content	anyType	0..1	The Raw_Segment_Content element is optional and contains the raw content of this segment of the hashed object.

4.2.143 HashType

The HashType type is intended to characterize hash values.

Property	Type	Mult	Description
Type	Common:HashNameType	0..1	The type element specifies the type of hash used in the Hash_Value element.
Other_Type	Common: StringObject AttributeType	0..1	The Other_Type element is used to specify the type of hash used in the Hash_Value element if a non-standard hashing algorithm, one which cannot be specified using the Type element, is used.
Simple_Hash_Value	Common: SimpleHashValueType	0..1	The Simple_Hash_Value element specifies a single result value of a basic cryptograhic hash function outputing a single hexbinary hash value.
Fuzzy_Hash_Value	Common: FuzzyHashValueType	0..1	The Fuzzy_Hash_Value element specifies a single result value of a cryptograhic fuzzy hash function outputing a single complex string based hash value. (e.g. SSDEEP's Block1hash:Block2hash format).
Fuzzy_Hash_Structure	Common: FuzzyHashStructureType	0..∞	The Fuzzy_Hash_Structure element is optional and enables the characterization of the key internal components of a fuzzy hash calculation with a given block size.

4.2.144 HashNameType (restriction Common:BaseObjectAttributeType)

HashNameType specifies the name of hashing algorithms, via a union of the HashNameEnum type and the atomic xs:string type. Its base type is the CybOX Core BaseObjectAttributeType, for permitting complex (i.e. regular-expression based) specifications.

Data restrictions: Common:HashNameEnum, string

Property	Type	Mult	Description
datatype	Common:DatatypeEnum	1..1	This attribute is optional and specifies the expected type for the value of the specified element.

4.2.145 HashNameEnum

HashNameEnum is a (non-exhaustive) enumeration of hashing algorithm names.

Restriction base: string

Enumeration Value	Description
MD5	The MD5 value describes the MD5 hashing algorithm.
MD6	The MD6 value describes the MD6 hashing algorithm.
SHA1	The SHA1 value describes the SHA1 hashing algorithm.
SHA256	The SHA256 value describes the SHA256 hashing algorithm.
SSDEEP	The SSDEEP value describes the SSDEEP hashing algorithm.
Other	The Other value describes a different hashing algorithm than those listed.

4.2.146 StructuredTextType

The StructuredTextType is a complex type representing a generalized structure for capturing structured textual information such as descriptions of things.

Property	Type	Mult	Description
Block	Common:Block	1..1	Block is a Structured_Text element consisting of one of Text_Title, Text, Code_Example_Language, or Code followed by another Block element. Structured_Text elements help define whitespace and text segments.

4.2.147 StructuredTextGroup

The StructuredTextGroup is a grouping of common fields representing a generalized structure for capturing structured textual information such as descriptions of things.

Property	Type	Mult	Description
Text_Title	string	0..∞	Presentation Element: This element is used to definebold-faced title for a subsequent block of text.
Text	string	0..∞	Presentation Element: This element is used to define a paragraph of text.
Code_Example_Language	`Common:LanguageTypeEnum`	0..∞	Presentation Element: This element is used to identify the programming language being used in the following block of Code
Code	string	0..∞	Presentation Element: This element is used to define a line of code.
Comment	string	0..∞	Presentation Element: This element is used to define a comment in code.
Images	`Common:ImagesType`	0..1	Presentation Element: This element is used to define a set of images.

4.2.148 ImagesType

The ImagesType specifies a set of images.

Property	Type	Mult	Description
Image	`Common:ImageType`	1..∞	Presentation Element: This element is used to define an image.

4.2.149 ImageType

The ImageType specifies an image.

Property	Type	Mult	Description

Image_Location	string	1..1	This element provides the location of the image file.
Image_Title	string	1..1	This element provides a title for the image.

4.2.150 BlockNatureEnum

The BlockNatureEnum is a (non-exhaustive) enumeration of characterizations of the nature for a given Block.

Restriction base: string

Enumeration Value	Description
Good_Code	Identifies good code within the Block.
Bad_Code	Identifies bad code within the Block.
Mitigation_Code	Identifies mitigation code within the Block.
Attack	Identifies attack code within the Block.
Result	Identifies code that specifies a result within the Block.
List	Identifies code that specifies a list within the Block.

4.2.151 ReferenceListType

The ReferencesListType contains one or more Reference elements, each of which provide further reading and insight into the item. This should be filled out as appropriate.

Property	Type	Mult	Description
Reference	Common:ReferenceType	1..∞	Each Reference subelement should provide a single source from which more information and deeper insight can be obtained, such as a research paper or an excerpt from a publication. Multiple Reference subelements can exist. The sole attribute of this element is the id. The id is optional and translates to a preceding footnote below the context notes if the author of the entry wants to cite a reference. Not all subelements need to be completed, since some are designed for web references and others are designed for book references. The fields Reference_Author and Reference_Title should be filled out for all references if possible. Reference_Section and Reference_Date can be included for either book references or online references. Reference_Edition, Reference_Publication, Reference_Publisher, and Reference_PubDate are intended for book references, however they can be included where appropriate for other types of references. Reference_Link is intended for web references, however it can be included for book references as well if applicable.

4.2.152 ReferenceType

The ReferenceType is a complex type representing a single reference to a source of information.

Property	Type	Mult	Description

reference_id	string	1..1	The id attribute is optional and is used as a mechanism for citing text in the entry. If an id is provided, it is placed between brackets and precedes this reference and the matching id should be used inside of the text for the entry itself where this reference is applicable. All reference ids assigned within an entry must be unique.
Reference_Description	`Common:` `StructuredTextType`	0..1	This element provides a description of the reference.
Reference_Author	string	0..∞	This element identifies an individual author of the material being referenced. It is not required, but may be repeated sequentially in order to identify multiple authors for a single piece of material.
Reference_Title	string	0..1	This element identifies the title of the material being referenced. It is not required if the material does not have a title.
Reference_Section	string	0..1	This element is intended to provide a means of identifying the exact location of the material inside of the publication source, such as the relevant pages of a research paper, the appropriate chapters from a book, etc. This is useful for both book references and internet references.
Reference_Edition	string	0..1	This element identifies the edition of the material being referenced in the event that multiple editions of the material exist. This will usually only be useful for book references.
Reference_Publication	string	0..1	This element identifies the publication source of the reference material, if one exists.
Reference_Publisher	string	0..1	This element identifies the publisher of the reference material, if one exists.
Reference_Date	date	0..1	This element identifies the date when the reference was included in the entry. This provides the reader with a time line for when the material in the reference, usually the link, was valid. The date should be of the format YYYY-MM-DD.
Reference_PubDate	string	0..1	This field describes the date when the reference was published YYYY.
Reference_Link	string	0..1	This element should hold the URL for the material being referenced, if one exists. This should always be used for web references, and may optionally be used for book and other publication references.

4.2.153 LanguageTypeEnum

The LanguageType is a simple type representing the specification of a relevant programming language.

Restriction base: string

Enumeration Value	Description
C	Specifies the C programming language.
C++	Specifies the C++ programming language.
C#	Specifies the C# programming language.
Java	Specifies the Java programming language.

JSP	Specifies the JSP programming language.
Javascript	Specifies the Javascript programming language.
ASP.NET	Specifies the ASP.NET programming language.
SQL	Specifies the SQL programming language.
Python	Specifies the Python programming language.
Perl	Specifies the Perl programming language.
PHP	Specifies the PHP programming language.
SOAP	Specifies the SOAP programming language.
Ruby	Specifies the Ruby programming language.
Shell	Specifies shell code regardless of type.
PseudoCode	Specifies pseudocode.
.NET	Specifies the .NET programming language.
Assembly	Specifies assembly code.
XML	Specifies the XML programming language.
HTML	Specifies the HTML programming language.

4.2.154 DataSegmentType

The DataSegmentType is intended to provide a relatively abstract way of characterizing data segments that may be written/read/transmitted or otherwise utilized in actions or behaviors.

Property	Type	Mult	Description
id	QName	1..1	The id attribute specifies a unique id for this data segment.
Data_Format	Common:DataFormatEnum	0..1	The Data_Format element refers to the type of data contained in the Data_Segment element.
Data_Size	Common:DataSizeType	0..1	The Data_Size element contains the size of the data contained in this element.
Data_Segment	Common:StringObjectAttributeType	0..1	The Data_Segment element contains the actual segment of data being characterized.
Offset	Common:IntegerObjectAttributeType	0..1	The Offset element allows for the specification of where to start searching for the specified data segment in an object, in bytes.
Search_Distance	Common:IntegerObjectAttributeType	0..1	The Search_Distance element specifies how far into an object should be ignored, in bytes, before starting to search for the specified data segment relative to the end of the previous data segment.
Search_Within	Common:IntegerObjectAttributeType	0..1	The Search_Within element specifies that at most N bytes are between data segments in related objects.

4.2.155 DataFormatEnum

The DataFormatEnum is a (non-exhaustive) enumeration of data formats.

Restriction base: string

Enumeration Value	Description
Binary	Specifies binary data.
Hexadecimal	Specifies hexadecimal data.
Text	Specifies text.
Other	Specifies any other type of data from the ones listed.

4.2.156 DataSizeType (extends Common:StringObjectAttributeType)

The DataSizeType specifies the size of the data segment.

Property	Type	Mult	Description
units	Common: DataSizeUnitsEnum	1..1	This attribute represents the Units used in the object size element. Possible values are: Bytes, Kilobytes, Megabytes.

4.2.157 DataSizeUnitsEnum

The DataSizeUnitsEnum is a (non-exhaustive) enumeration of data size units.

Restriction base: string

Enumeration Value	Description
Bytes	Specifies an object size in Bytes.
Kilobytes	Specifies an object size in Kilobytes.
Megabytes	Specifies an object size in Megabytes.

4.2.158 CPESpecificationType

CPESpecificationType is a modularized data type intended for providing a consistent approach to uniquely specifying the identity of a specific platform using the Common Platform Enumeration (CPE) naming standard. http://cpe.mitre.org/

Property	Type	Mult	Description
CPE_Name	Common:CPENameType	1..1	The CPE_Name element contains the CPE Name value for the relevant platform. A CPE Name is a percent-encoded URI with each name starting with the prefix (the URI scheme name) "cpe:". The remainder of the name consists of colon separated values representing the CPE part, vendor, product, version, update, edition and language (i.e. cpe:/{part} : {vendor} : {product} : {version} : {update} : {edition} : {language}).
Title	Common:CPETitleType	0..1	The Title field contains the plain language descriptive title of the relevant platform.
Meta_Item_Metadata	Common: MetaItemMetadataType	0..1	The meta-item-metadata element aggregates the descriptive metadata for this CPE Name instance.

4.2.159 CPENameType (extends Common:StringObjectAttributeType)

The CPENameType contains the CPE Name value for the relevant platform.

Property	Type	Mult	Description
xmlns_value	string	1..1	The xmlns_value attribute contains the XML namespace descriptor for the CPE namespace relevant to this CPE Name use.

4.2.160 MetaItemMetadataType

The MetaItemMetadataType element aggregates the descriptive metadata for a CPE Name instance.

Property	Type	Mult	Description
Modification_Date	Common:	0..1	The modification-date element specifies the last

	DateTimeObject AttributeType		time that any CPE property has been modified.
NVD_ID	Common: UnsignedInteger ObjectAttributeType	0..1	The nvd-id element contains the NVD specific unique identifier for a CPE. This is provided as a long-term identifier that can be used to map different versions of CPE syntax to a CPE with the same meaning. This is not a replacement of a CPEName. Use of a CPEName is still the standard ID naming scheme for CPE 2.x.
Status	Common: StringObject AttributeType	0..1	The status element contains the internal NVD status of a CPE.
XMLNS_Meta	Common: StringObject AttributeType	0..1	The xmlns-meta element contains the XML CPE metadata namespace descriptor for the CPE namespace relevant to this CPE Name use.

4.2.161 CPETitleType (extends Common:StringObjectAttributeType)

The CPETitleType contains the plain language descriptive title of the relevant platform.

Property	Type	Mult	Description
lang	string	1..1	This field holds a shortform descriptor for the language that the Title field is expressed in. Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility. See RFC 3066 at http://www.ietf.org/rfc/rfc3066.txt and the IANA registry at http://www.iana.org/assignments/lang-tag-apps.htm for further information. The union allows for the 'un-declaration' of xml:lang with the empty string.

4.2.162 MetadataType

The MetadataType is intended as mechanism to capture any non-context-specific metadata

Property	Type	Mult	Description
type	string	1..1	This field specifies the type of name of a single metadata field.
Value	string	0..1	This field specifies the value of name of a single metadata field.
SubDatum	Common:MetadataType	0..∞	This field uses recursion of the MetadataType specify subdatum structures for this metadata field.

4.2.163 EnvironmentVariableListType

The EnvironmentVariableListType type is used for representing a list of environment variables.

Property	Type	Mult	Description
Environment_Variable	Common: EnvironmentVariableType	1..∞	The Environment_Variable element is used for representing environment variables using a name/value pair.

4.2.164 EnvironmentVariableType

The EnvironmentVariableType type is used for representing environment variables using a name/value pair.

Property	Type	Mult	Description
Name	Common: StringObjectAttributeType	1..1	The Name element specifies the name of the environment variable.
Value	Common: StringObjectAttributeType	0..1	The Value element specifies the value of the environment variable.

4.2.165 DigitalSignatureInfoType

The DigitalSignatureInfoType type is used as a way to represent some of the basic information about a digital signature.

Property	Type	Mult	Description
signature_exists	boolean	1..1	Specifies whether the digital signature exists.
signature_verified	boolean	1..1	Specifies if the digital signature is verified.
Certificate_Issuer	Common: StringObjectAttributeType	0..1	The certificate issuer of the digital signature.
Certificate_Subject	Common: StringObjectAttributeType	0..1	The certificate subject of the digital signature.
Signature_Description	Common: StringObjectAttributeType	0..1	A description of the digital signature.

4.2.166 SIDType (restriction Common:BaseObjectAttributeType)

SIDType specifies Windows Security ID (SID) types via a union of the SIDTypeEnum type and the atomic xs:string type. Its base type is the CybOX Core BaseObjectAttributeType, for permitting complex (i.e. regular-expression based) specifications.

Data restrictions: Common:SIDTypeEnum, string

Property	Type	Mult	Description
datatype	Common:DatatypeEnum	1..1	This attribute is optional and specifies the expected type for the value of the specified element.

4.2.167 SIDTypeEnum

The SIDTypeEnum type is an enumeration of Windows Security ID (SID) types. These correspond to the values specified by the SID_NAME_USE enumeration--see http://msdn.microsoft.com/en-us/library/windows/desktop/aa379601(v=vs.85).aspx for more information.

Restriction base: string

Enumeration Value	Description
SidTypeUser	Indicates a SID of type User.
SidTypeGroup	Indicates a SID of type Group.
SidTypeDomain	Indicates a SID of type Domain.
SidTypeAlias	Indicates a SID of type Alias.
SidTypeWellKnownGroup	Indicates a SID for a well-known group.
SidTypeDeletedAccount	Indicates a SID for a deleted account.
SidTypeInvalid	Indicates an invalid SID.
SidTypeUnknown	Indicates a SID of unknown type.

SidTypeComputer	Indicates a SID for a computer.
SidTypeLabel	Indicates a mandatory integrity label SID.

4.2.168 **FrequencyTypeEnum**

The FrequencyType is a simple type representing the characterization of how frequently a given event/condition occurs.

Restriction base: string

Enumeration Value	Description
Often	Specifies that a condition occurs often.
Sometimes	Specifies that a condition occurs sometimes.
Rarely	Specifies that a condition occurs rarely.

5 Language Representations & Example Content

5.1 XML

The XML Representation for the CybOX Language Data Model is documented via a series of XML Schemas.[4] These schemas describe how the information presented in this Specification is formatted and represented as XML Documents. Please refer to the appropriate Schema for more information about a specific XML representation.

CybOX Core Schema

http://cybox.mitre.org/XMLSchema/cybox_core_v1.0(draft).xsd

CybOX Common_Types Schema

http://cybox.mitre.org/XMLSchema/cybox_common_types_v1.0(draft).xsd

Defined Objects

For a complete list of the XML Schemas for the CybOX defined objects, see the CybOX Language Defined Objects Specfication

The complete listing of XML representation resources can be found on the CybOX website.[5]

5.2 Validation Requirements

All XML content written against the XML Schema implementation of the CybOX Language MUST be XML Schema valid as defined in the XML Schemas associated with the XML Schema implementation of the CybOX Language.

5.3 Example Content

---

[4] XML Schema Part 0: Primer Second Edition http://www.w3.org/TR/xmlschema-0/

[5] See the CybOX Language documentation at: http://cybox.mitre.org/language/index.html

5.3.1 Simple Examples

5.3.1.1 Single URL

```xml
<?xml version="1.0" encoding="UTF-8"?>
<cybox:Observables
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:cybox="http://cybox.mitre.org/cybox_v1"
    xmlns:common="http://cybox.mitre.org/Common_v1"
    xmlns:URIObj="http://cybox.mitre.org/objects#URIObject"
    xsi:schemaLocation="http://cybox.mitre.org/cybox_v1
            http://cybox.mitre.org/XMLSchema/cybox_core_v1.0(draft).xsd
            http://cybox.mitre.org/objects#URIObject
            http://cybox.mitre.org/XMLSchema/objects/URI/URI_Object_1.1.xsd"
    cybox_major_version="1" cybox_minor_version="0(draft)">
    <cybox:Observable>
        <!-- Observable for a single URL -->
        <cybox:Stateful_Measure>
            <cybox:Object id="cybox:A1" type="URI">
                <cybox:Defined_Object xsi:type="URIObj:URIObjectType">
                    <URIObj:Value condition="Equals"
datatype="AnyURI">www.sample1.com/index.html</URIObj:Value>
                </cybox:Defined_Object>
            </cybox:Object>
        </cybox:Stateful_Measure>
    </cybox:Observable>
</cybox:Observables>
```

5.3.1.2 Observable pattern for a file with one of a set of three MD5 hashes

```xml
<?xml version="1.0" encoding="UTF-8"?>
<cybox:Observables
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:cybox="http://cybox.mitre.org/cybox_v1"
    xmlns:common="http://cybox.mitre.org/Common_v1"
    xmlns:FileObj="http://cybox.mitre.org/objects#FileObject"
    xsi:schemaLocation="http://cybox.mitre.org/XMLSchema/cybox_v1
            http://cybox.mitre.org/XMLSchema/cybox_core_v1.0(draft).xsd
            http://cybox.mitre.org/objects#FileObject
            http://cybox.mitre.org/XMLSchema/objects/File/File_Object_1.2.xsd"
    cybox_major_version="1" cybox_minor_version="0(draft)">
    <cybox:Observable>
        <cybox:Stateful_Measure>
            <cybox:Object id="cybox:A1" type="File">
                <cybox:Defined_Object xsi:type="FileObj:FileObjectType">
                    <FileObj:Hashes>
                        <common:Hash>
                            <common:Type datatype="String">MD5</common:Type>
                            <common:Simple_Hash_Value condition="IsInSet"
value_set="4EC0027BEF4D7E1786A04D021FA8A67F,
21F0027ACF4D9017861B1D021FA8CF76,2B4D027BEF4D7E1786A04D021FA8CC01"
datatype="hexBinary"></common:Simple_Hash_Value>
                        </common:Hash>
                    </FileObj:Hashes>
                </cybox:Defined_Object>
            </cybox:Object>
        </cybox:Stateful_Measure>
    </cybox:Observable>
</cybox:Observables>
```

5.3.1.3 File with basic information including multiple hashes

```
<?xml version="1.0" encoding="UTF-8"?>
<cybox:Observables
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:cybox="http://cybox.mitre.org/cybox_v1"
    xmlns:common="http://cybox.mitre.org/Common_v1"
    xmlns:FileObj="http://cybox.mitre.org/objects#FileObject"
    xsi:schemaLocation="http://cybox.mitre.org/XMLSchema/cybox_v1
                            http://cybox.mitre.org/XMLSchema/cybox_core_v1.0(draft).xsd
                            http://cybox.mitre.org/objects#FileObject
                            http://cybox.mitre.org/XMLSchema/objects/File/File_Object_1.2.xsd"
    cybox_major_version="1" cybox_minor_version="0(draft)">
    <cybox:Observable>
        <!-- Observable for a file with name, path, MD5 hash, SHA1 hash, SHA256 hash and size in bytes
utilizing the base File_Object-->
        <cybox:Stateful_Measure>
            <cybox:Object id="cybox:A1" type="File">
                <cybox:Defined_Object xsi:type="FileObj:FileObjectType">
                    <FileObj:File_Name datatype="String">notepad.exe</FileObj:File_Name>
                    <FileObj:File_Path datatype="String">C:\Temp</FileObj:File_Path>
                    <FileObj:Size_In_Bytes datatype="UnsignedLong">273845</FileObj:Size_In_Bytes>
                    <FileObj:Hashes>
                        <common:Hash>
                            <common:Type datatype="String">MD5</common:Type>
                            <common:Simple_Hash_Value condition="Equals"
datatype="hexBinary">59a7078444ee3c862e4c08b601ed7e01</common:Simple_Hash_Value>
                        </common:Hash>
                        <common:Hash>
                            <common:Type datatype="String">SHA1</common:Type>
                            <common:Simple_Hash_Value condition="Equals"
datatype="hexBinary">98e969b49ff2aedf66b94eb82c54b916f1a634cd</common:Simple_Hash_Value>
                        </common:Hash>
                        <common:Hash>
                            <common:Type datatype="String">SHA256</common:Type>
                            <common:Simple_Hash_Value condition="Equals"
datatype="hexBinary">1706c7cd14a5c9bbf674b21f9c4f873ac04b7a6f1f2202cd0c5977c48968d188</com
mon:Simple_Hash_Value>
                        </common:Hash>
                    </FileObj:Hashes>
                </cybox:Defined_Object>
            </cybox:Object>
        </cybox:Stateful_Measure>
    </cybox:Observable>
</cybox:Observables>
```

5.3.1.4 Create File Action

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<cybox:Observables
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:cybox="http://cybox.mitre.org/cybox_v1"
    xmlns:common="http://cybox.mitre.org/Common_v1"
    xmlns:FileObj="http://cybox.mitre.org/objects#FileObject"
    xsi:schemaLocation="http://cybox.mitre.org/XMLSchema/cybox_v1
                                http://cybox.mitre.org/XMLSchema/cybox_core_v1.0(draft).xsd
                                http://cybox.mitre.org/objects#FileObject
                                http://cybox.mitre.org/XMLSchema/objects/File/File_Object_1.2.xsd"
cybox_major_version="1" cybox_minor_version="0(draft)">
    <cybox:Observable>
        <cybox:Event>
            <cybox:Actions>
                <cybox:Action id="cybox:Action_1" type="Create" action_status="Success" context="Host"
timestamp="09:22:00.0Z">
                    <cybox:Action_Name>
                        <cybox:Defined_Name>Create File</cybox:Defined_Name>
                    </cybox:Action_Name>
                    <cybox:Associated_Objects>
                        <cybox:Associated_Object id="cybox:Object_1" type="File" object_state="Exists"
association_type="Affected">
                            <cybox:Defined_Object xsi:type="FileObj:FileObjectType">
                                <FileObj:File_Name>foobar.dll</FileObj:File_Name>
                                <FileObj:File_Path>C:\Windows\system32</FileObj:File_Path>
                                <FileObj:Hashes>
                                    <common:Hash>
                                        <common:Type datatype="String">MD5</common:Type>
                                        <common:Simple_Hash_Value
datatype="hexBinary">6E48C348D742A931EC2CE90ABD7DAC6A</common:Simple_Hash_Value>
                                    </common:Hash>
                                </FileObj:Hashes>
                            </cybox:Defined_Object>
                        </cybox:Associated_Object>
                    </cybox:Associated_Objects>
                </cybox:Action>
            </cybox:Actions>
        </cybox:Event>
    </cybox:Observable>
</cybox:Observables>
```

5.3.1.5 Simple Email

```
<?xml version="1.0" encoding="UTF-8"?>
<cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:cybox="http://cybox.mitre.org/cybox_v1"
    xmlns:common="http://cybox.mitre.org/Common_v1"
    xmlns:AddrObj="http://cybox.mitre.org/objects#AddressObject"
    xmlns:URIObj="http://cybox.mitre.org/objects#URIObject"
    xmlns:FileObj="http://cybox.mitre.org/objects#FileObject"
    xmlns:EmailMessageObj="http://cybox.mitre.org/XMLSchema/objects#EmailMessageObject"
    xsi:schemaLocation="http://cybox.mitre.org/Common_v1
                                http://cybox.mitre.org/XMLSchema/cybox_core_v1.0(draft).xsd
                                http://cybox.mitre.org/objects#FileObject
                                http://cybox.mitre.org/XMLSchema/objects/File/File_Object_1.2.xsd
                                http://cybox.mitre.org/objects#EmailMessageObject
```

```
            http://cybox.mitre.org/XMLSchema/objects/Email_Message/Email_Message_Object_1.1.xsd"
    cybox_major_version="1" cybox_minor_version="0(draft)">
    <cybox:Observable>
        <cybox:Stateful_Measure>
            <cybox:Object id="cybox:A1" type="Email Message">
                <cybox:Defined_Object xsi:type="EmailMessageObj:EmailMessageObjectType">
                    <EmailMessageObj:Header>
                        <EmailMessageObj:To>
                            <EmailMessageObj:Recipient category="e-mail"><AddrObj:Address_Value
datatype="String">victim1@target.com</AddrObj:Address_Value></EmailMessageObj:Recipient>
                            <EmailMessageObj:Recipient category="e-mail"><AddrObj:Address_Value
datatype="String">victim2@target.com</AddrObj:Address_Value></EmailMessageObj:Recipient>
                        </EmailMessageObj:To>
                        <EmailMessageObj:From category="e-mail">
                            <AddrObj:Address_Value
datatype="String">attacker@example.com</AddrObj:Address_Value>
                        </EmailMessageObj:From>
                        <EmailMessageObj:Subject datatype="String">New modifications to the
specification</EmailMessageObj:Subject>
                    </EmailMessageObj:Header>
                </cybox:Defined_Object>
                <cybox:Related_Objects>
                    <cybox:Related_Object idref="cybox:A2" relationship="Received_From"/>
                </cybox:Related_Objects>
            </cybox:Object>
        </cybox:Stateful_Measure>
    </cybox:Observable>
    <cybox:Observable>
        <cybox:Stateful_Measure>
            <cybox:Object id="cybox:A2" type="IP Address">
                <cybox:Defined_Object xsi:type="AddrObj:AddressObjectType" category="ipv4-addr"
is_source="true">
                    <AddrObj:Address_Value datatype="String">192.168.1.1</AddrObj:Address_Value>
                </cybox:Defined_Object>
            </cybox:Object>
        </cybox:Stateful_Measure>
    </cybox:Observable>
</cybox:Observables>
```

5.3.1.6 Simple email with simple file attachment

```
<?xml version="1.0" encoding="UTF-8"?>
<cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:cybox="http://cybox.mitre.org/cybox_v1"
    xmlns:common="http://cybox.mitre.org/Common_v1"
    xmlns:AddrObj="http://cybox.mitre.org/objects#AddressObject"
    xmlns:URIObj="http://cybox.mitre.org/objects#URIObject"
    xmlns:FileObj="http://cybox.mitre.org/objects#FileObject"
    xmlns:EmailMessageObj="http://cybox.mitre.org/XMLSchema/objects#EmailMessageObject"
    xsi:schemaLocation="http://cybox.mitre.org/Common_v1
                                http://cybox.mitre.org/XMLSchema/cybox_core_v1.0(draft).xsd
                                http://cybox.mitre.org/objects#FileObject
                                http://cybox.mitre.org/XMLSchema/objects/File/File_Object_1.2.xsd
                                http://cybox.mitre.org/objects#EmailMessageObject
```

```xml
    http://cybox.mitre.org/XMLSchema/objects/Email_Message/Email_Message_Object_1.1.xsd"
    cybox_major_version="1" cybox_minor_version="0(draft)">
    <cybox:Observable>
        <cybox:Stateful_Measure>
            <cybox:Object id="A1" type="Email Message">
                <cybox:Defined_Object xsi:type="EmailMessageObj:EmailMessageObjectType">
                    <EmailMessageObj:Attachments>
                        <EmailMessageObj:File>
                            <FileObj:File_Name datatype="String">FooBar Specification (critical
revision).doc</FileObj:File_Name>
                            <FileObj:Hashes>
                                <common:Hash>
                                    <common:Simple_Hash_Value
datatype="hexBinary">4EC0027BEF4D7E1786A04D021FA8A67F</common:Simple_Hash_Value>
                                </common:Hash>
                            </FileObj:Hashes>
                        </EmailMessageObj:File>
                    </EmailMessageObj:Attachments>
                    <EmailMessageObj:Header>
                        <EmailMessageObj:To>
                            <EmailMessageObj:Recipient category="e-mail"><AddrObj:Address_Value
datatype="String">victim1@target.com</AddrObj:Address_Value></EmailMessageObj:Recipient>
                            <EmailMessageObj:Recipient category="e-mail"><AddrObj:Address_Value
datatype="String">victim2@target.com</AddrObj:Address_Value></EmailMessageObj:Recipient>
                        </EmailMessageObj:To>
                        <EmailMessageObj:From category="e-mail">
                            <AddrObj:Address_Value
datatype="String">attacker@example.com</AddrObj:Address_Value>
                        </EmailMessageObj:From>
                        <EmailMessageObj:Subject datatype="String">New modifications to the
specification</EmailMessageObj:Subject>
                    </EmailMessageObj:Header>
                </cybox:Defined_Object>
            </cybox:Object>
        </cybox:Stateful_Measure>
    </cybox:Observable>
</cybox:Observables>
```

5.3.1.7 Observable pattern for a URL matching one of three values utilizing IsInSet

```xml
<?xml version="1.0" encoding="UTF-8"?>
<cybox:Observables
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:cybox="http://cybox.mitre.org/cybox_v1"
    xmlns:common="http://cybox.mitre.org/Common_v1"
    xmlns:URIObj="http://cybox.mitre.org/objects#URIObject"
    xsi:schemaLocation="http://cybox.mitre.org/cybox_v1
                                http://cybox.mitre.org/XMLSchema/cybox_core_v1.0(draft).xsd
                                http://cybox.mitre.org/objects#URIObject
                                http://cybox.mitre.org/XMLSchema/objects/URI/URI_Object_1.1.xsd"
    cybox_major_version="1" cybox_minor_version="0(draft)">
    <cybox:Observable id="cybox:Obs1">
        <!-- Observable for any single URL matching one of three URLs utilizing IsInSet-->
        <cybox:Stateful_Measure>
```

```
            <cybox:Object id="A1" type="URI">
                <cybox:Defined_Object xsi:type="URIObj:URIObjectType">
                    <URIObj:Value condition="IsInSet" value_set="www.sample1.com/index.html,
sample2.com/login.html, dev.sample3.com/index/kb.html" datatype="AnyURI"/>
                </cybox:Defined_Object>
            </cybox:Object>
        </cybox:Stateful_Measure>
    </cybox:Observable>
</cybox:Observables>
```

5.3.1.8 Observable pattern for a URL matching one of three values utilizing logical OR composition

```
<?xml version="1.0" encoding="UTF-8"?>
<cybox:Observables
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:cybox="http://cybox.mitre.org/cybox_v1"
    xmlns:common="http://cybox.mitre.org/Common_v1"
    xmlns:URIObj="http://cybox.mitre.org/objects#URIObject"
    xsi:schemaLocation="http://cybox.mitre.org/cybox_v1
                                http://cybox.mitre.org/XMLSchema/cybox_core_v1.0(draft).xsd
                                http://cybox.mitre.org/objects#URIObject
                                http://cybox.mitre.org/XMLSchema/objects/URI/URI_Object_1.1.xsd"
    cybox_major_version="1" cybox_minor_version="0(draft)">
    <cybox:Observable id="cybox:Obs4">
        <!-- Observable for any single URL matching one of three URLs utilizing logical composition -->
        <cybox:Observable_Composition operator="OR">
            <cybox:Observable id="cybox:Obs1">
                <cybox:Stateful_Measure>
                    <cybox:Object id="cybox:A1" type="URI">
                        <cybox:Defined_Object xsi:type="URIObj:URIObjectType">
                            <URIObj:Value condition="Equals"
datatype="AnyURI">www.sample1.com/index.html</URIObj:Value>
                        </cybox:Defined_Object>
                    </cybox:Object>
                </cybox:Stateful_Measure>
            </cybox:Observable>
            <cybox:Observable id="cybox:Obs2">
                <cybox:Stateful_Measure>
                    <cybox:Object id="cybox:A2" type="URI">
                        <cybox:Defined_Object xsi:type="URIObj:URIObjectType">
                            <URIObj:Value condition="Equals"
datatype="AnyURI">sample2.com/login.html</URIObj:Value>
                        </cybox:Defined_Object>
                    </cybox:Object>
                </cybox:Stateful_Measure>
            </cybox:Observable>
            <cybox:Observable id="cybox:Obs3">
                <cybox:Stateful_Measure>
                    <cybox:Object id="cybox:A3" type="URI">
                        <cybox:Defined_Object xsi:type="URIObj:URIObjectType">
                            <URIObj:Value condition="Equals"
datatype="AnyURI">dev.sample3.com/index/kb.html</URIObj:Value>
                        </cybox:Defined_Object>
                    </cybox:Object>
                </cybox:Stateful_Measure>
            </cybox:Observable>
        </cybox:Observable_Composition>
    </cybox:Observable>
</cybox:Observables>
```

5.3.1.9 Observable pattern for a URL matching one of three values utilizing logical OR composition and Object pooling

```
<?xml version="1.0" encoding="UTF-8"?>
<cybox:Observables
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:cybox="http://cybox.mitre.org/cybox_v1"
    xmlns:common="http://cybox.mitre.org/Common_v1"
    xmlns:URIObj="http://cybox.mitre.org/objects#URIObject"
    xsi:schemaLocation="http://cybox.mitre.org/cybox_v1
                            http://cybox.mitre.org/XMLSchema/cybox_core_v1.0(draft).xsd
                            http://cybox.mitre.org/objects#URIObject
                            http://cybox.mitre.org/XMLSchema/objects/URI/URI_Object_1.1.xsd"
    cybox_major_version="1" cybox_minor_version="0(draft)">
    <cybox:Observable id="cybox:Obs4">
        <!-- Observable for any single URL matching one of three URLs utilizing logical composition and
Object Pools-->
        <cybox:Observable_Composition operator="OR">
            <cybox:Observable id="cybox:Obs1">
                <cybox:Stateful_Measure>
                    <cybox:Object idref="cybox:A1" type="URI"/>
                </cybox:Stateful_Measure>
            </cybox:Observable>
            <cybox:Observable id="cybox:Obs2">
                <cybox:Stateful_Measure>
                    <cybox:Object idref="cybox:A2" type="URI"/>
                </cybox:Stateful_Measure>
            </cybox:Observable>
            <cybox:Observable id="cybox:Obs3">
                <cybox:Stateful_Measure>
                    <cybox:Object idref="cybox:A3" type="URI"/>
                </cybox:Stateful_Measure>
            </cybox:Observable>
        </cybox:Observable_Composition>
    </cybox:Observable>
    <cybox:Pools>
        <cybox:Object_Pool>
            <cybox:Object id="cybox:A1" type="URI">
                <cybox:Defined_Object xsi:type="URIObj:URIObjectType">
                    <URIObj:Value condition="Equals"
datatype="AnyURI">www.sample1.com/index.html</URIObj:Value>
                </cybox:Defined_Object>
            </cybox:Object>
            <cybox:Object id="cybox:A2" type="URI">
                <cybox:Defined_Object xsi:type="URIObj:URIObjectType">
                    <URIObj:Value condition="Equals"
datatype="AnyURI">sample2.com/login.html</URIObj:Value>
                </cybox:Defined_Object>
            </cybox:Object>
            <cybox:Object id="cybox:A3" type="URI">
                <cybox:Defined_Object xsi:type="URIObj:URIObjectType">
                    <URIObj:Value condition="Equals"
datatype="AnyURI">dev.sample3.com/index/kb.html</URIObj:Value>
```

```
                </cybox:Defined_Object>
            </cybox:Object>
        </cybox:Object_Pool>
    </cybox:Pools>
</cybox:Observables>
```

5.3.2 **Complex Example**

The following complex example is derived from observable data from a real-world attack campaign observed in the wild during March, 2012. This campaign is known by many names but Iran-Oil is likely its most common name of reference.

5.3.2.1 Iran-Oil example as only static observable Stateful Measures

```
<?xml version="1.0" encoding="UTF-8"?>
<cybox:Observables
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:cybox="http://cybox.mitre.org/cybox_v1"
    xmlns:common="http://cybox.mitre.org/Common_v1"
    xmlns:AddrObj="http://cybox.mitre.org/objects#AddressObject"
    xmlns:URIObj="http://cybox.mitre.org/objects#URIObject"
    xmlns:FileObj="http://cybox.mitre.org/objects#FileObject"
    xmlns:EmailMessageObj="http://cybox.mitre.org/XMLSchema/objects#EmailMessageObject"
    xsi:schemaLocation="http://cybox.mitre.org/Common_v1
                                http://cybox.mitre.org/XMLSchema/cybox_core_v1.0(draft).xsd
                                http://cybox.mitre.org/objects#URIObject
                                http://cybox.mitre.org/XMLSchema/objects/URI/URI_Object_1.1.xsd
                                http://cybox.mitre.org/objects#FileObject
                                http://cybox.mitre.org/XMLSchema/objects/File/File_Object_1.2.xsd
                                http://cybox.mitre.org/objects#EmailMessageObject

    http://cybox.mitre.org/XMLSchema/objects/Email_Message/Email_Message_Object_1.1.xsd"
    cybox_major_version="1" cybox_minor_version="0(draft)">

<!-- This collection of observables were observed as part of the widespread "Iran-Oil" (among many other
names used) attack campaign in March 2012 -->

    <cybox:Observable id="cybox:guid-guid-1a937ec2-90ab-4e0e-a37c-db9b2e66a58e">
        <!-- "Iran-Oil" attack campaign email message with raw header-->
        <cybox:Stateful_Measure>
            <cybox:Object id="cybox:guid-51359587-f201-4383-b032-5a64522fcd7d" type="Email
Message">
                <cybox:Defined_Object xsi:type="EmailMessageObj:EmailMessageObjectType">
                    <EmailMessageObj:Attachments><EmailMessageObj:File object_reference="cybox:guid-
49d31c13-8d7b-4528-b8d6-ce8ed0d43ad7"/></EmailMessageObj:Attachments>
                    <EmailMessageObj:Header>
                        <EmailMessageObj:To><EmailMessageObj:Recipient category="e-
mail"><AddrObj:Address_Value
datatype="String">william.abnett@gmail.com</AddrObj:Address_Value></EmailMessageObj:Recipient><
/EmailMessageObj:To>
                        <EmailMessageObj:From category="e-mail"><AddrObj:Address_Value
datatype="String">wmorrison89@gmail.com</AddrObj:Address_Value></EmailMessageObj:From>
                        <EmailMessageObj:Subject datatype="String">Iran's Oil and Nuclear
Situation</EmailMessageObj:Subject>
```

```
                        <EmailMessageObj:Date datatype="DateTime">2012-03-
02T07:42:24Z</EmailMessageObj:Date>
                    </EmailMessageObj:Header>
                    <EmailMessageObj:Raw_Header datatype="String"><![CDATA[
                        Return-Path: <wmorrison89@gmail.com>
Received-SPF: pass (google.com: domain of wmorrison89@gmail.com designates
10.236.185.4 as permitted sender) client-ip=10.236.185.4;
Authentication-Results: mr.google.com; spf=pass (google.com: domain of
wmorrison89@gmail.com designates 10.236.185.4 as permitted sender)
smtp.mail=wmorrison89@gmail.com; dkim=pass header.i=wmorrison89@gmail.com
Received: from mr.google.com ([10.236.185.4]) by 10.236.185.4 with SMTP
id t4mr5301660yhm.129.1330692273662 (num_hops = 1); Fri, 02 Mar 2012
04:44:33 -0800 (PST)
MIME-Version: 1.0
Received: by 10.236.185.4 with SMTP id t4mr4236541yhm.129.1330692265380;
Fri,
02 Mar 2012 04:44:25 -0800 (PST)
Received: by 10.147.35.14 with HTTP; Fri, 2 Mar 2012 04:44:24 -0800 (PST)
In-Reply-To:
<CADY6HTa-jmaqmtVyyT-nLz6reztNjcs-617wL4bt9YBOGu+h4w@mail.gmail.com>
References:
<CADY6HTa-jmaqmtVyyT-nLz6reztNjcs-617wL4bt9YBOGu+h4w@mail.gmail.com>
Date: Fri, 2 Mar 2012 07:44:24 -0500
Message-ID:
<CADY6HTZ6oopY5v6WkYU81YcSQw3X124CK_Fx4jhnhUiU3Y9z6A@mail.gmail.com>
Subject: Iran's Oil and Nuclear Situation
From: william abnett <wmorrison89@gmail.com>
To: william.abnett <william.abnett@gmail.com>
Content-Type: multipart/mixed; boundary="20cf303f67fac8928804ba41efd5"
                    ]]></EmailMessageObj:Raw_Header>
                </cybox:Defined_Object>
            </cybox:Object>
        </cybox:Stateful_Measure>
    </cybox:Observable>

    <cybox:Observable id="cybox:guid-cybox:35f04c28-5fd2-4d72-8aae-2ad04ee1811f">
        <!-- Iran-Oil corrupted .doc file-->
        <cybox:Stateful_Measure>
            <cybox:Object id="cybox:guid-49d31c13-8d7b-4528-b8d6-ce8ed0d43ad7" type="File">
                <cybox:Description><common:Text>The word document contains flash, which downloads a
corrupted mp4 file. The mp4 file itself is not anything special but an 0C filled (22kb) mp4 file with a valid
mp4 header.</common:Text></cybox:Description>
                <cybox:Defined_Object xsi:type="FileObj:FileObjectType">
                    <FileObj:File_Name datatype="String">Iran's Oil and Nuclear
Situation.doc</FileObj:File_Name>
                    <FileObj:Size_In_Bytes datatype="UnsignedLong">106604</FileObj:Size_In_Bytes>
                    <FileObj:Hashes><common:Hash><common:Type
datatype="String">MD5</common:Type><common:Simple_Hash_Value condition="Equals"
datatype="hexBinary">E92A4FC283EB2802AD6D0E24C7FCC857</common:Simple_Hash_Value></co
mmon:Hash></FileObj:Hashes>
                </cybox:Defined_Object>
            </cybox:Object>
        </cybox:Stateful_Measure>
    </cybox:Observable>

    <cybox:Observable id="cybox:guid-f005fbc6-7427-43ea-8e1e-9a341836f76b">
```

```
        <!-- Iran-Oil invalid .mp4 downloader file-->
        <cybox:Stateful_Measure>
            <cybox:Object id="cybox:guid-8b463e0d-cc16-4036-950e-5eeb09bc51aa" type="File">
                <cybox:Description><common:Text>This mp4 file causes memory corruption and code
execution via heap-spraying code injection.</common:Text></cybox:Description>
                <cybox:Defined_Object xsi:type="FileObj:FileObjectType">
                    <FileObj:File_Name datatype="String">test.mp4</FileObj:File_Name>
                    <FileObj:Size_In_Bytes datatype="UnsignedLong">22384</FileObj:Size_In_Bytes>
                    <FileObj:Hashes><common:Hash><common:Type
datatype="String">MD5</common:Type><common:Simple_Hash_Value condition="Equals"
datatype="hexBinary">8933598C8B1FA5E493497B11C48DA4F2</common:Simple_Hash_Value></com
mon:Hash></FileObj:Hashes>
                </cybox:Defined_Object>
                <cybox:Related_Objects>
                    <cybox:Related_Object idref="cybox:guid-49d31c13-8d7b-4528-b8d6-ce8ed0d43ad7"
type="File" relationship="Downloaded_By"/>
                    <cybox:Related_Object idref="cybox:guid-61041b8b-0c15-48a0-ac5f-b49488788010"
type="URI" relationship="Downloaded_From"/>
                </cybox:Related_Objects>
            </cybox:Object>
        </cybox:Stateful_Measure>
    </cybox:Observable>

    <cybox:Observable id="cybox:guid-b63c8bd4-e9c6-4e5a-b012-040f81dcc699">
        <!-- URL from which malicious .mp4 file was downloaded-->
        <cybox:Stateful_Measure>
            <cybox:Object id="cybox:guid-61041b8b-0c15-48a0-ac5f-b49488788010" type="URI">
                <cybox:Defined_Object xsi:type="URIObj:URIObjectType" type="URL">
                    <URIObj:Value datatype="AnyURI"
condition="Equals">http://208.115.230.76/test.mp4</URIObj:Value>
                </cybox:Defined_Object>
            </cybox:Object>
        </cybox:Stateful_Measure>
    </cybox:Observable>

    <cybox:Observable id="cybox:guid-210f18f3-3874-4f9a-861d-71b328be90c6">
        <!-- Iran-Oil .exe Trojan file-->
        <cybox:Stateful_Measure>
            <cybox:Object id="cybox:guid-b7e0bc39-f519-4878-8fb0-5902554efe1c" type="File">
                <cybox:Description><common:Text>The file (us.exe MD5:
FD1BE09E499E8E380424B3835FC973A8 4861440 bytes) is created in the logged in user %Temp%
directory. The size of the embedded file is 22.5 KB (23040 bytes) and the size of the created us.exe is
4.63MB. It is an odd discrepancy until you look at the file and it looks like the code is repeated over and
over - 211 times. The file resource section indicates the file is meant to look like a java updater, which is
always larger than 22.5KB and that would explain all this padding, which is done at the time when the file
is being written to the disk.</common:Text></cybox:Description>
                <cybox:Defined_Object xsi:type="FileObj:FileObjectType">
                    <FileObj:File_Name datatype="String">us.exe</FileObj:File_Name>
                    <FileObj:File_Path datatype="String">%Temp%</FileObj:File_Path>
                    <FileObj:Size_In_Bytes datatype="UnsignedLong">4861440</FileObj:Size_In_Bytes>
                    <FileObj:Hashes><common:Hash><common:Type
datatype="String">MD5</common:Type><common:Simple_Hash_Value condition="Equals"
datatype="hexBinary">FD1BE09E499E8E380424B3835FC973A8</common:Simple_Hash_Value></com
mon:Hash></FileObj:Hashes>
                </cybox:Defined_Object>
                <cybox:Related_Objects>
```

```
                    <cybox:Related_Object idref="cybox:guid-8b463e0d-cc16-4036-950e-5eeb09bc51aa"
type="File" relationship="Created_By"/>
                    <!-- The trojan connects to the following set of URLs/IPs for C&C -->
                    <cybox:Related_Object idref="cybox:guid-41b220d8-4c45-48de-9d08-30d661b2dc8e"
type="URI" relationship="Connected_To"/>
                    <cybox:Related_Object idref="cybox:guid-61aa225b-90ef-415c-8bbd-a17282e457c9"
type="IP Address" relationship="Connected_To"/>
                    <cybox:Related_Object idref="cybox:guid-568db11e-39ee-43d7-83d8-032bdec3801a"
type="URI" relationship="Connected_To"/>
                    <cybox:Related_Object idref="cybox:guid-80bea4d1-0e70-4a03-a54f-e40373bf94f1"
type="IP Address" relationship="Connected_To"/>
                    <cybox:Related_Object idref="cybox:guid-af7cb3b6-d70b-4b3b-b24f-7cfad739710f"
type="URI" relationship="Connected_To"/>
                    <cybox:Related_Object idref="cybox:guid-5ceb9d54-24e2-4627-948d-6b92ac81962a"
type="IP Address" relationship="Connected_To"/>
                </cybox:Related_Objects>
            </cybox:Object>
        </cybox:Stateful_Measure>
    </cybox:Observable>

    <cybox:Observable id="cybox:guid-dee72b3e-82fb-4319-bfcc-007e3cf930e8">
        <!-- Iran-Oil core embedded .exe Trojan file-->
        <cybox:Stateful_Measure>
            <cybox:Object id="cybox:guid-bed1ff22-08e8-4e04-b7ac-908b5271176f" type="File">
                <cybox:Defined_Object xsi:type="FileObj:FileObjectType">
                    <FileObj:File_Name datatype="String">us-embedded.exe</FileObj:File_Name>
                    <FileObj:Size_In_Bytes datatype="UnsignedLong">23040</FileObj:Size_In_Bytes>
                    <FileObj:Hashes><common:Hash><common:Type
datatype="String">MD5</common:Type><common:Simple_Hash_Value condition="Equals"
datatype="hexBinary">CB3DCDE34FD9FF0E19381D99B02F9692</common:Simple_Hash_Value></co
mmon:Hash></FileObj:Hashes>
                </cybox:Defined_Object>
                <cybox:Related_Objects>
                    <cybox:Related_Object idref="cybox:guid-b7e0bc39-f519-4878-8fb0-5902554efe1c"
type="File" relationship="Contained_Within"/>
                </cybox:Related_Objects>
            </cybox:Object>
        </cybox:Stateful_Measure>
    </cybox:Observable>

    <!-- The next six Observables represent the 3 different URL/IP pairs of C&C servers that the trojan
communicates with-->
    <cybox:Observable id="cybox:guid-066cef51-c886-432e-9a22-a17f57f3f3f2">
        <!-- One of three Command and Control URLs-->
        <cybox:Stateful_Measure>
            <cybox:Object id="cybox:guid-41b220d8-4c45-48de-9d08-30d661b2dc8e" type="URI">
                <cybox:Defined_Object xsi:type="URIObj:URIObjectType" type="URL">
                    <URIObj:Value datatype="AnyURI"
condition="Equals">www.documents.myPicture.info</URIObj:Value>
                </cybox:Defined_Object>
                <cybox:Related_Objects>
                    <cybox:Related_Object  idref="cybox:guid-61aa225b-90ef-415c-8bbd-a17282e457c9"
type="IP Address" relationship="Resolved_To"/>
                </cybox:Related_Objects>
            </cybox:Object>
        </cybox:Stateful_Measure>
    </cybox:Observable>
    <cybox:Observable id="cybox:guid-4e05804c-f552-44e1-9793-ff4bb7f88f9c">
        <!-- One of three Command and Control IPs-->
        <cybox:Stateful_Measure>
            <cybox:Object id="cybox:guid-61aa225b-90ef-415c-8bbd-a17282e457c9" type="IP Address">
                <cybox:Defined_Object xsi:type="AddrObj:AddressObjectType" category="ipv4-addr">
                    <AddrObj:Address_Value datatype="String"
condition="Equals">199.192.156.134</AddrObj:Address_Value>
                </cybox:Defined_Object>
            </cybox:Object>
        </cybox:Stateful_Measure>
    </cybox:Observable>

    <cybox:Observable id="cybox:guid-75ce59ad-1f01-4eae-9ecc-0b22c4c24ce7">
        <!-- One of three Command and Control URLs-->
        <cybox:Stateful_Measure>
            <cybox:Object id="cybox:guid-568db11e-39ee-43d7-83d8-032bdec3801a" type="URI">
                <cybox:Defined_Object xsi:type="URIObj:URIObjectType" type="URL">
                    <URIObj:Value datatype="AnyURI"
condition="Equals">documents.myPicture.info</URIObj:Value>
                </cybox:Defined_Object>
                <cybox:Related_Objects>
                    <cybox:Related_Object idref="cybox:guid-80bea4d1-0e70-4a03-a54f-e40373bf94f1"
type="IP Address" relationship="Resolved_To"/>
                </cybox:Related_Objects>
            </cybox:Object>
        </cybox:Stateful_Measure>
    </cybox:Observable>
    <cybox:Observable id="cybox:guid-1ea53b14-8fe9-467b-a298-62d9684e797d">
        <!-- One of three Command and Control IPs-->
        <cybox:Stateful_Measure>
            <cybox:Object id="cybox:guid-80bea4d1-0e70-4a03-a54f-e40373bf94f1" type="IP Address">
                <cybox:Defined_Object xsi:type="AddrObj:AddressObjectType" category="ipv4-addr">
                    <AddrObj:Address_Value datatype="String"
condition="Equals">199.192.156.134</AddrObj:Address_Value>
                </cybox:Defined_Object>
            </cybox:Object>
        </cybox:Stateful_Measure>
    </cybox:Observable>

    <cybox:Observable id="cybox:guid-f6c8ee75-ee7e-4490-bd5d-0661d0db7264">
        <!-- One of three Command and Control URLs-->
        <cybox:Stateful_Measure>
            <cybox:Object id="cybox:guid-af7cb3b6-d70b-4b3b-b24f-7cfad739710f" type="URI">
                <cybox:Defined_Object xsi:type="URIObj:URIObjectType" type="URL">
                    <URIObj:Value datatype="AnyURI"
condition="Equals">ftp.documents.myPicture.info</URIObj:Value>
                </cybox:Defined_Object>
                <cybox:Related_Objects>
                    <cybox:Related_Object idref="cybox:guid-5ceb9d54-24e2-4627-948d-6b92ac81962a"
type="IP Address" relationship="Resolved_To"/>
                </cybox:Related_Objects>
            </cybox:Object>
        </cybox:Stateful_Measure>
    </cybox:Observable>
    <cybox:Observable id="cybox:guid-c78c0a83-6d14-45f8-827f-f758f0cd11ea">
```

```
        <!-- One of three Command and Control IPs-->
        <cybox:Stateful_Measure>
            <cybox:Object id="cybox:guid-5ceb9d54-24e2-4627-948d-6b92ac81962a" type="IP Address">
                <cybox:Defined_Object xsi:type="AddrObj:AddressObjectType" category="ipv4-addr">
                    <AddrObj:Address_Value datatype="String"
condition="Equals">199.192.156.134</AddrObj:Address_Value>
                </cybox:Defined_Object>
            </cybox:Object>
        </cybox:Stateful_Measure>
    </cybox:Observable>

    <cybox:Observable id="cybox:guid-47d6a950-884d-46b5-9938-ac5555065a81">
        <!-- This composed observable defines a pattern that is true if the observed email exists AND the
malicious .doc file exists AND the downloader .mp4 file exists AND the trojan .exe exists AND all three of
the C&C IP addresses are seen-->
        <!-- This yields a very tight filter that will have very low false positives but could miss almost any
variation of the attack elements-->
        <cybox:Observable_Composition operator="AND">
            <!-- "Iran-Oil" attack campaign email message with raw header-->
            <cybox:Observable idref="cybox:guid-1a937ec2-90ab-4e0e-a37c-db9b2e66a58e"/>
            <!-- Iran-Oil corrupted .doc file-->
            <cybox:Observable idref="cybox:guid-35f04c28-5fd2-4d72-8aae-2ad04ee1811f"/>
            <!-- Iran-Oil invalid .mp4 downloader file-->
            <cybox:Observable idref="cybox:guid-f005fbc6-7427-43ea-8e1e-9a341836f76b"/>
            <!-- Iran-Oil .exe Trojan file-->
            <cybox:Observable idref="cybox:guid-210f18f3-3874-4f9a-861d-71b328be90c6"/>
            <!-- The three Command and Control IPs-->
            <cybox:Observable idref="cybox:guid-4e05804c-f552-44e1-9793-ff4bb7f88f9c"/>
            <cybox:Observable idref="cybox:guid-1ea53b14-8fe9-467b-a298-62d9684e797d"/>
            <cybox:Observable idref="cybox:guid-c78c0a83-6d14-45f8-827f-f758f0cd11ea"/>
        </cybox:Observable_Composition>
    </cybox:Observable>

    <cybox:Observable id="cybox:guid-94b0aa45-065e-486f-acaf-2d8e793f525e">
        <!-- This composed observable defines a pattern that is true if the observed email exists OR the
malicious .doc file exists OR the downloader .mp4 file exists OR the trojan .exe exists OR any of the three
C&C IP addresses are seen-->
        <!-- This yields a very loose filter that could have false positives but could catch numerous potential
variations of the attack elements-->
        <cybox:Observable_Composition operator="OR">
            <!-- "Iran-Oil" attack campaign email message with raw header-->
            <cybox:Observable idref="cybox:guid-1a937ec2-90ab-4e0e-a37c-db9b2e66a58e"/>
            <!-- Iran-Oil corrupted .doc file-->
            <cybox:Observable idref="cybox:guid-35f04c28-5fd2-4d72-8aae-2ad04ee1811f"/>
            <!-- Iran-Oil invalid .mp4 downloader file-->
            <cybox:Observable idref="cybox:guid-f005fbc6-7427-43ea-8e1e-9a341836f76b"/>
            <!-- Iran-Oil .exe Trojan file-->
            <cybox:Observable idref="cybox:guid-210f18f3-3874-4f9a-861d-71b328be90c6"/>
            <!-- The three Command and Control IPs-->
            <cybox:Observable idref="cybox:guid-4e05804c-f552-44e1-9793-ff4bb7f88f9c"/>
            <cybox:Observable idref="cybox:guid-1ea53b14-8fe9-467b-a298-62d9684e797d"/>
            <cybox:Observable idref="cybox:guid-c78c0a83-6d14-45f8-827f-f758f0cd11ea"/>
        </cybox:Observable_Composition>
    </cybox:Observable>
    <!-- CybOX enables a wide myriad of other potential observable pattern variations at the logical
composition level or utilizing patterns at the Object attribute level including Regex all of which allow the
user to define an almost infinitely variable set of patterns and filters -->

</cybox:Observables>
```

5.3.2.2 Iran-Oil example as dynamic observable Events

```
<?xml version="1.0" encoding="UTF-8"?>
<cybox:Observables
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:cybox="http://cybox.mitre.org/cybox_v1"
    xmlns:common="http://cybox.mitre.org/Common_v1"
    xmlns:AddrObj="http://cybox.mitre.org/objects#AddressObject"
    xmlns:URIObj="http://cybox.mitre.org/objects#URIObject"
    xmlns:FileObj="http://cybox.mitre.org/objects#FileObject"
    xmlns:EmailMessageObj="http://cybox.mitre.org/XMLSchema/objects#EmailMessageObject"
    xsi:schemaLocation="http://cybox.mitre.org/Common_v1
                                http://cybox.mitre.org/XMLSchema/cybox_core_v1.0(draft).xsd
                                http://cybox.mitre.org/objects#URIObject
                                http://cybox.mitre.org/XMLSchema/objects/URI/URI_Object_1.1.xsd
                                http://cybox.mitre.org/objects#FileObject
                                http://cybox.mitre.org/XMLSchema/objects/File/File_Object_1.2.xsd
                                http://cybox.mitre.org/objects#EmailMessageObject

    http://cybox.mitre.org/XMLSchema/objects/Email_Message/Email_Message_Object_1.1.xsd"
    cybox_major_version="1" cybox_minor_version="0(draft)">
    <!-- This collection of observables were observed as part of the widespread "Iran-Oil" (among many
other names used) attack campaign in March 2012 -->
    <cybox:Observable id="cybox:guid-1a937ec2-90ab-4e0e-a37c-db9b2e66a58e">
        <!-- Receive "Iran-Oil" attack campaign email message -->
        <cybox:Event type="Email Ops">
            <cybox:Description>
                <common:Text>Receive "Iran-Oil" attack campaign email message.</common:Text>
            </cybox:Description>
            <cybox:Actions>
                <cybox:Action type="Receive">
                    <cybox:Associated_Objects>
                        <cybox:Associated_Object id="cybox:guid-51359587-f201-4383-b032-5a64522fcd7d"
type="Email Message" association_type="Returned">
                            <cybox:Defined_Object xsi:type="EmailMessageObj:EmailMessageObjectType">
                                <EmailMessageObj:Attachments>
                                    <EmailMessageObj:File object_reference="cybox:guid-49d31c13-8d7b-
4528-b8d6-ce8ed0d43ad7"/>
                                </EmailMessageObj:Attachments>
                                <EmailMessageObj:Header>
                                    <EmailMessageObj:To><EmailMessageObj:Recipient category="e-mail">
                                        <AddrObj:Address_Value
datatype="String">william.abnett@gmail.com</AddrObj:Address_Value>
                                    </EmailMessageObj:Recipient></EmailMessageObj:To>
                                    <EmailMessageObj:From category="e-mail">
                                        <AddrObj:Address_Value
datatype="String">wmorrison89@gmail.com</AddrObj:Address_Value>
                                    </EmailMessageObj:From>
                                    <EmailMessageObj:Subject datatype="String">Iran's Oil and Nuclear
Situation</EmailMessageObj:Subject>
                                    <EmailMessageObj:Date datatype="DateTime">2012-03-
02T07:42:24Z</EmailMessageObj:Date>
                                </EmailMessageObj:Header>
                                <EmailMessageObj:Raw_Header datatype="String"><![CDATA[
                        Return-Path: <wmorrison89@gmail.com>
Received-SPF: pass (google.com: domain of wmorrison89@gmail.com designates
10.236.185.4 as permitted sender) client-ip=10.236.185.4;
Authentication-Results: mr.google.com; spf=pass (google.com: domain of
wmorrison89@gmail.com designates 10.236.185.4 as permitted sender)
smtp.mail=wmorrison89@gmail.com; dkim=pass header.i=wmorrison89@gmail.com
Received: from mr.google.com ([10.236.185.4]) by 10.236.185.4 with SMTP
id t4mr5301660yhm.129.1330692273662 (num_hops = 1); Fri, 02 Mar 2012
04:44:33 -0800 (PST)
MIME-Version: 1.0
Received: by 10.236.185.4 with SMTP id t4mr4236541yhm.129.1330692265380;
Fri,
02 Mar 2012 04:44:25 -0800 (PST)
Received: by 10.147.35.14 with HTTP; Fri, 2 Mar 2012 04:44:24 -0800 (PST)
In-Reply-To:
<CADY6HTa-jmaqmtVyyT-nLz6reztNjcs-617wL4bt9YBOGu+h4w@mail.gmail.com>
References:
<CADY6HTa-jmaqmtVyyT-nLz6reztNjcs-617wL4bt9YBOGu+h4w@mail.gmail.com>
Date: Fri, 2 Mar 2012 07:44:24 -0500
Message-ID:
<CADY6HTZ6oopY5v6WkYU81YcSQw3X124CK_Fx4jhnhUiU3Y9z6A@mail.gmail.com>
Subject: Iran's Oil and Nuclear Situation
From: william abnett <wmorrison89@gmail.com>
To: william.abnett <william.abnett@gmail.com>
Content-Type: multipart/mixed; boundary="20cf303f67fac8928804ba41efd5"
                    ]]></EmailMessageObj:Raw_Header>
                            </cybox:Defined_Object>
                        </cybox:Associated_Object>
                    </cybox:Associated_Objects>
                </cybox:Action>
            </cybox:Actions>
        </cybox:Event>
    </cybox:Observable>

    <cybox:Observable id="cybox:guid-35f04c28-5fd2-4d72-8aae-2ad04ee1811f">
        <!-- Open Iran-Oil corrupted .doc file-->
        <cybox:Event type="File Ops (CRUD)">
            <cybox:Description>
                <common:Text>Open Iran-Oil corrupted .doc file.</common:Text>
            </cybox:Description>
            <cybox:Actions>
                <cybox:Action type="Open">
                    <cybox:Associated_Objects>
                        <cybox:Associated_Object id="cybox:guid-49d31c13-8d7b-4528-b8d6-
ce8ed0d43ad7" type="File" association_type="Affected">
                            <cybox:Description>
                                <common:Text>The word document contains flash, which downloads a
corrupted mp4 file. The mp4 file itself is not anything special but an 0C filled (22kb) mp4 file with a valid
mp4 header.</common:Text>
                            </cybox:Description>
                            <cybox:Defined_Object xsi:type="FileObj:FileObjectType">
                                <FileObj:File_Name datatype="String">Iran's Oil and Nuclear
Situation.doc</FileObj:File_Name>
                                <FileObj:Size_In_Bytes
datatype="UnsignedLong">106604</FileObj:Size_In_Bytes>
                                <FileObj:Hashes>
                                    <common:Hash>
                                        <common:Type datatype="String">MD5</common:Type>
                                        <common:Simple_Hash_Value condition="Equals"
datatype="hexBinary">E92A4FC283EB2802AD6D0E24C7FCC857</common:Simple_Hash_Value>
                                    </common:Hash>
                                </FileObj:Hashes>
                            </cybox:Defined_Object>
                        </cybox:Associated_Object>
                    </cybox:Associated_Objects>
                </cybox:Action>
            </cybox:Actions>
        </cybox:Event>
    </cybox:Observable>

    <cybox:Observable id="cybox:guid-f005fbc6-7427-43ea-8e1e-9a341836f76b">
        <!-- Download Iran-Oil invalid .mp4 downloader file-->
        <cybox:Event type="File Ops (CRUD)">
            <cybox:Description>
                <common:Text>Download Iran-Oil invalid .mp4 downloader file.</common:Text>
            </cybox:Description>
            <cybox:Actions>
                <cybox:Action type="Download">
                    <cybox:Associated_Objects>
                        <cybox:Associated_Object idref="cybox:guid-49d31c13-8d7b-4528-b8d6-
ce8ed0d43ad7" type="File" association_type="Initiating"/>
                        <cybox:Associated_Object id="cybox:guid-8b463e0d-cc16-4036-950e-
5eeb09bc51aa" type="File" association_type="Affected">
                            <!-- Iran-Oil invalid .mp4 downloader file-->
                            <cybox:Description>
                                <common:Text>This mp4 file causes memory corruption and code execution
via heap-spraying code injection.</common:Text>
                            </cybox:Description>
                            <cybox:Defined_Object xsi:type="FileObj:FileObjectType">
                                <FileObj:File_Name datatype="String">test.mp4</FileObj:File_Name>
                                <FileObj:Size_In_Bytes
datatype="UnsignedLong">22384</FileObj:Size_In_Bytes>
                                <FileObj:Hashes>
                                    <common:Hash>
                                        <common:Type datatype="String">MD5</common:Type>
                                        <common:Simple_Hash_Value condition="Equals"
datatype="hexBinary">8933598C8B1FA5E493497B11C48DA4F2</common:Simple_Hash_Value>
                                    </common:Hash>
                                </FileObj:Hashes>
                            </cybox:Defined_Object>
                            <cybox:Related_Objects>
                                <cybox:Related_Object idref="cybox:guid-49d31c13-8d7b-4528-b8d6-
ce8ed0d43ad7" type="File" relationship="Downloaded_By"/>
                                <cybox:Related_Object idref="cybox:guid-61041b8b-0c15-48a0-ac5f-
b49488788010" type="URI" relationship="Downloaded_From"/>
                            </cybox:Related_Objects>
                        </cybox:Associated_Object>
                        <cybox:Associated_Object id="cybox:guid-61041b8b-0c15-48a0-ac5f-b49488788010"
type="URI" association_type="Utilized">
                            <!-- URL from which malicious .mp4 file was downloaded-->
                            <cybox:Defined_Object xsi:type="URIObj:URIObjectType" type="URL">
                                <URIObj:Value datatype="AnyURI"
condition="Equals">http://208.115.230.76/test.mp4</URIObj:Value>
                            </cybox:Defined_Object>
                        </cybox:Associated_Object>
                    </cybox:Associated_Objects>
                </cybox:Action>
            </cybox:Actions>
        </cybox:Event>
    </cybox:Observable>

    <cybox:Observable id="cybox:guid-210f18f3-3874-4f9a-861d-71b328be90c6">
        <!-- Create Iran-Oil .exe Trojan file-->
        <cybox:Event type="File Ops (CRUD)">
            <cybox:Description>
                <common:Text_Title>Create Iran-Oil .exe Trojan file.</common:Text_Title>
            </cybox:Description>
            <cybox:Actions>
                <cybox:Action type="Create">
                    <cybox:Associated_Objects>
                        <cybox:Associated_Object idref="cybox:guid-8b463e0d-cc16-4036-950e-
5eeb09bc51aa" type="File" association_type="Initiating"/>
                        <cybox:Associated_Object id="cybox:guid-b7e0bc39-f519-4878-8fb0-5902554efe1c"
type="File" association_type="Affected">
                            <cybox:Description>
                                <common:Text>The file (us.exe MD5:
FD1BE09E499E8E380424B3835FC973A8 4861440 bytes) is created in the logged in user %Temp%
directory. The size of the embedded file is 22.5 KB (23040 bytes) and the size of the created us.exe is
4.63MB. It is an odd discrepancy until you look at the file and it looks like the code is repeated over and
over - 211 times. The file resource section indicates the file is meant to look like a java updater, which is
always larger than 22.5KB and that would explain all this padding, which is done at the time when the file
is being written to the disk.</common:Text>
                            </cybox:Description>
                            <cybox:Defined_Object xsi:type="FileObj:FileObjectType">
                                <FileObj:File_Name datatype="String">us.exe</FileObj:File_Name>
                                <FileObj:File_Path datatype="String">%Temp%</FileObj:File_Path>
                                <FileObj:Size_In_Bytes
datatype="UnsignedLong">4861440</FileObj:Size_In_Bytes>
                                <FileObj:Hashes>
                                    <common:Hash>
                                        <common:Type datatype="String">MD5</common:Type>
                                        <common:Simple_Hash_Value condition="Equals"
datatype="hexBinary">FD1BE09E499E8E380424B3835FC973A8</common:Simple_Hash_Value>
                                    </common:Hash>
                                </FileObj:Hashes>
                            </cybox:Defined_Object>
                            <cybox:Related_Objects>
                                <cybox:Related_Object idref="cybox:guid-8b463e0d-cc16-4036-950e-
5eeb09bc51aa" type="File" relationship="Created_By"/>
                                <!-- The trojan connects to the following set of URLs/IPs for C&C -->
                                <cybox:Related_Object idref="cybox:guid-41b220d8-4c45-48de-9d08-
30d661b2dc8e" type="URI" relationship="Connected_To"/>
```

```
                                <cybox:Related_Object idref="cybox:guid-61aa225b-90ef-415c-8bbd-
a17282e457c9" type="IP Address" relationship="Connected_To"/>
                                <cybox:Related_Object idref="cybox:guid-568db11e-39ee-43d7-83d8-
032bdec3801a" type="URI" relationship="Connected_To"/>
                                <cybox:Related_Object idref="cybox:guid-80bea4d1-0e70-4a03-a54f-
e40373bf94f1" type="IP Address" relationship="Connected_To"/>
                                <cybox:Related_Object idref="cybox:guid-af7cb3b6-d70b-4b3b-b24f-
7cfad739710f" type="URI" relationship="Connected_To"/>
                                <cybox:Related_Object idref="cybox:guid-5ceb9d54-24e2-4627-948d-
6b92ac81962a" type="IP Address" relationship="Connected_To"/>
                            </cybox:Related_Objects>
                        </cybox:Associated_Object>
                    </cybox:Associated_Objects>
                </cybox:Action>
            </cybox:Actions>
        </cybox:Event>
    </cybox:Observable>

    <cybox:Observable id="cybox:guid-b650c988-aac7-45ff-967d-9f1e5fc66161">
        <!-- Execute Iran-Oil .exe Trojan file-->
        <cybox:Event type="File Ops (CRUD)">
            <cybox:Description>
                <common:Text>Execute Iran-Oil .exe Trojan file.</common:Text>
            </cybox:Description>
            <cybox:Actions>
                <cybox:Action type="Execute">
                    <cybox:Associated_Objects>
                        <cybox:Associated_Object idref="cybox:guid-8b463e0d-cc16-4036-950e-
5eeb09bc51aa" type="File" association_type="Initiating"/>
                        <cybox:Associated_Object idref="cybox:guid-b7e0bc39-f519-4878-8fb0-
5902554efe1c" type="File" association_type="Affected"/>
                    </cybox:Associated_Objects>
                </cybox:Action>
            </cybox:Actions>
        </cybox:Event>
    </cybox:Observable>


    <cybox:Observable id="cybox:guid-dee72b3e-82fb-4319-bfcc-007e3cf930e8">
        <!-- Iran-Oil core embedded .exe Trojan file-->
        <cybox:Stateful_Measure>
            <cybox:Object id="cybox:guid-bed1ff22-08e8-4e04-b7ac-908b5271176f" type="File">
                <cybox:Defined_Object xsi:type="FileObj:FileObjectType">
                    <FileObj:File_Name datatype="String">us-embedded.exe</FileObj:File_Name>
                    <FileObj:Size_In_Bytes datatype="UnsignedLong">23040</FileObj:Size_In_Bytes>
                    <FileObj:Hashes>
                        <common:Hash>
                            <common:Type datatype="String">MD5</common:Type>
                            <common:Simple_Hash_Value condition="Equals"
datatype="hexBinary">CB3DCDE34FD9FF0E19381D99B02F9692</common:Simple_Hash_Value>
                        </common:Hash>
                    </FileObj:Hashes>
                </cybox:Defined_Object>
                <cybox:Related_Objects>
                    <cybox:Related_Object idref="cybox:guid-b7e0bc39-f519-4878-8fb0-5902554efe1c"
type="File" relationship="Contained_Within"/>
```

```
                </cybox:Related_Objects>
            </cybox:Object>
        </cybox:Stateful_Measure>
    </cybox:Observable>

    <cybox:Observable id="cybox:guid-a24ff8bc-b534-4616-838b-8bbe260a8e8f">
        <!-- Trojan .exe file connects out to C&C URLs/IPs-->
        <cybox:Event type="App Layer Traffic">
            <cybox:Description>
                <common:Text>Trojan .exe file connects out to C2 URLs/IPs.</common:Text>
            </cybox:Description>
            <cybox:Actions>
                <cybox:Action type="Connect">
                    <cybox:Associated_Objects>
                        <cybox:Associated_Object idref="cybox:guid-b7e0bc39-f519-4878-8fb0-
5902554efe1c" type="File" association_type="Initiating"/>
                        <cybox:Associated_Object idref="cybox:guid-41b220d8-4c45-48de-9d08-
30d661b2dc8e" type="URI" association_type="Utilized"/>
                        <cybox:Associated_Object idref="cybox:guid-61aa225b-90ef-415c-8bbd-
a17282e457c9" type="IP Address" association_type="Utilized"/>
                        <cybox:Associated_Object idref="cybox:guid-568db11e-39ee-43d7-83d8-
032bdec3801a" type="URI" association_type="Utilized"/>
                        <cybox:Associated_Object idref="cybox:guid-80bea4d1-0e70-4a03-a54f-
e40373bf94f1" type="IP Address" association_type="Utilized"/>
                        <cybox:Associated_Object idref="cybox:guid-af7cb3b6-d70b-4b3b-b24f-
7cfad739710f" type="URI" association_type="Utilized"/>
                        <cybox:Associated_Object idref="cybox:guid-5ceb9d54-24e2-4627-948d-
6b92ac81962a" type="IP Address" association_type="Utilized"/>
                    </cybox:Associated_Objects>
                </cybox:Action>
            </cybox:Actions>
        </cybox:Event>
    </cybox:Observable>

    <!-- The next six Observables represent the 3 different URL/IP pairs of C&C servers that the trojan
communicates with-->
    <cybox:Observable id="cybox:guid-066cef51-c886-432e-9a22-a17f57f3f3f2">
        <!-- One of three Command and Control URLs-->
        <cybox:Stateful_Measure>
            <cybox:Object id="cybox:guid-41b220d8-4c45-48de-9d08-30d661b2dc8e" type="URI">
                <cybox:Defined_Object xsi:type="URIObj:URIObjectType" type="URL">
                    <URIObj:Value datatype="AnyURI"
condition="Equals">www.documents.myPicture.info</URIObj:Value>
                </cybox:Defined_Object>
                <cybox:Related_Objects>
                    <cybox:Related_Object idref="cybox:guid-61aa225b-90ef-415c-8bbd-a17282e457c9"
type="IP Address" relationship="Resolved_To"/>
                </cybox:Related_Objects>
            </cybox:Object>
        </cybox:Stateful_Measure>
    </cybox:Observable>
    <cybox:Observable id="cybox:guid-4e05804c-f552-44e1-9793-ff4bb7f88f9c">
        <!-- One of three Command and Control IPs-->
        <cybox:Stateful_Measure>
            <cybox:Object id="cybox:guid-61aa225b-90ef-415c-8bbd-a17282e457c9" type="IP Address">
                <cybox:Defined_Object xsi:type="AddrObj:AddressObjectType" category="ipv4-addr">
```

```
                    <AddrObj:Address_Value datatype="String"
condition="Equals">199.192.156.134</AddrObj:Address_Value>
                </cybox:Defined_Object>
            </cybox:Object>
        </cybox:Stateful_Measure>
    </cybox:Observable>
    <cybox:Observable id="cybox:guid-75ce59ad-1f01-4eae-9ecc-0b22c4c24ce7">
        <!-- One of three Command and Control URLs-->
        <cybox:Stateful_Measure>
            <cybox:Object id="cybox:guid-568db11e-39ee-43d7-83d8-032bdec3801a" type="URI">
                <cybox:Defined_Object xsi:type="URIObj:URIObjectType" type="URL">
                    <URIObj:Value datatype="AnyURI"
condition="Equals">documents.myPicture.info</URIObj:Value>
                </cybox:Defined_Object>
                <cybox:Related_Objects>
                    <cybox:Related_Object idref="cybox:guid-80bea4d1-0e70-4a03-a54f-e40373bf94f1"
type="IP Address" relationship="Resolved_To"/>
                </cybox:Related_Objects>
            </cybox:Object>
        </cybox:Stateful_Measure>
    </cybox:Observable>
    <cybox:Observable id="cybox:guid-1ea53b14-8fe9-467b-a298-62d9684e797d">
        <!-- One of three Command and Control IPs-->
        <cybox:Stateful_Measure>
            <cybox:Object id="cybox:guid-80bea4d1-0e70-4a03-a54f-e40373bf94f1" type="IP Address">
                <cybox:Defined_Object xsi:type="AddrObj:AddressObjectType" category="ipv4-addr">
                    <AddrObj:Address_Value datatype="String"
condition="Equals">199.192.156.134</AddrObj:Address_Value>
                </cybox:Defined_Object>
            </cybox:Object>
        </cybox:Stateful_Measure>
    </cybox:Observable>
    <cybox:Observable id="cybox:guid-f6c8ee75-ee7e-4490-bd5d-0661d0db7264">
        <!-- One of three Command and Control URLs-->
        <cybox:Stateful_Measure>
            <cybox:Object id="cybox:guid-af7cb3b6-d70b-4b3b-b24f-7cfad739710f" type="URI">
                <cybox:Defined_Object xsi:type="URIObj:URIObjectType" type="URL">
                    <URIObj:Value datatype="AnyURI"
condition="Equals">ftp.documents.myPicture.info</URIObj:Value>
                </cybox:Defined_Object>
                <cybox:Related_Objects>
                    <cybox:Related_Object idref="cybox:guid-5ceb9d54-24e2-4627-948d-6b92ac81962a"
type="IP Address" relationship="Resolved_To"/>
                </cybox:Related_Objects>
            </cybox:Object>
        </cybox:Stateful_Measure>
    </cybox:Observable>
    <cybox:Observable id="cybox:guid-c78c0a83-6d14-45f8-827f-f758f0cd11ea">
        <!-- One of three Command and Control IPs-->
        <cybox:Stateful_Measure>
            <cybox:Object id="cybox:guid-5ceb9d54-24e2-4627-948d-6b92ac81962a" type="IP Address">
                <cybox:Defined_Object xsi:type="AddrObj:AddressObjectType" category="ipv4-addr">
                    <AddrObj:Address_Value datatype="String"
condition="Equals">199.192.156.134</AddrObj:Address_Value>
                </cybox:Defined_Object>
            </cybox:Object>
        </cybox:Stateful_Measure>
    </cybox:Observable>


    <cybox:Observable id="cybox:guid-47d6a950-884d-46b5-9938-ac5555065a81">
        <!-- This composed observable defines a pattern that is true if the receive email event occurs AND the create malicious .doc file event occurs AND the download the downloader .mp4 file event occurs AND the create trojan .exe file event occurs AND the execute trojan .exe file event occurs AND the connect to all three of the C&C URLs/IPs event occurs-->
        <!-- This yields a very tight filter that will have very low false positives but could miss almost any variation of the attack elements-->
        <cybox:Observable_Composition operator="AND">
            <!-- Receive "Iran-Oil" attack campaign email message -->
            <cybox:Observable idref="cybox:guid-1a937ec2-90ab-4e0e-a37c-db9b2e66a58e"/>
            <!-- Open Iran-Oil corrupted .doc file-->
            <cybox:Observable idref="cybox:guid-35f04c28-5fd2-4d72-8aae-2ad04ee1811f"/>
            <!-- Download Iran-Oil invalid .mp4 downloader file-->
            <cybox:Observable idref="cybox:guid-f005fbc6-7427-43ea-8e1e-9a341836f76b"/>
            <!-- Create Iran-Oil .exe Trojan file-->
            <cybox:Observable idref="cybox:guid-210f18f3-3874-4f9a-861d-71b328be90c6"/>
            <!-- Execute Iran-Oil .exe Trojan file-->
            <cybox:Observable idref="cybox:guid-b650c988-aac7-45ff-967d-9f1e5fc66161"/>
            <!-- Trojan .exe file connects out to C&C URLs/IPs-->
            <cybox:Observable idref="cybox:guid-a24ff8bc-b534-4616-838b-8bbe260a8e8f"/>
        </cybox:Observable_Composition>
    </cybox:Observable>

    <cybox:Observable id="cybox:guid-80594430-7567-4402-88a4-05d556b21884">
        <!-- This composed observable defines a pattern that is true if the receive email event occurs OR the create malicious .doc file event occurs OR the download the downloader .mp4 file event occurs OR the create trojan .exe file event occurs OR the execute trojan .exe file event occurs OR the connect to all three of the C&C URLs/IPs event occurs-->
        <!-- This yields a very loose filter that could have false positives but could catch numerous potential variations of the attack elements-->
        <cybox:Observable_Composition operator="OR">
            <!-- Receive "Iran-Oil" attack campaign email message -->
            <cybox:Observable idref="cybox:guid-1a937ec2-90ab-4e0e-a37c-db9b2e66a58e"/>
            <!-- Open Iran-Oil corrupted .doc file-->
            <cybox:Observable idref="cybox:guid-35f04c28-5fd2-4d72-8aae-2ad04ee1811f"/>
            <!-- Download Iran-Oil invalid .mp4 downloader file-->
            <cybox:Observable idref="cybox:guid-f005fbc6-7427-43ea-8e1e-9a341836f76b"/>
            <!-- Create Iran-Oil .exe Trojan file-->
            <cybox:Observable idref="cybox:guid-210f18f3-3874-4f9a-861d-71b328be90c6"/>
            <!-- Execute Iran-Oil .exe Trojan file-->
            <cybox:Observable idref="cybox:guid-b650c988-aac7-45ff-967d-9f1e5fc66161"/>
            <!-- Trojan .exe file connects out to C&C URLs/IPs-->
            <cybox:Observable idref="cybox:guid-a24ff8bc-b534-4616-838b-8bbe260a8e8f"/>
        </cybox:Observable_Composition>
    </cybox:Observable>

    <cybox:Observable id="cybox:guid-7d932074-fded-4056-870e-dd51980501d4">
        <!-- This composed observable defines a pattern that is true if (the receive email event occurs AND the create malicious .doc file event occurs) OR (the download the downloader .mp4 file event occurs AND the create trojan .exe file event occurs AND the execute trojan .exe file event occurs) OR the connect to all three of the C&C URLs/IPs event occurs-->
        <cybox:Observable_Composition operator="OR">
```

```
            <cybox:Observable><cybox:Observable_Composition operator="AND">
                <!-- Receive "Iran-Oil" attack campaign email message -->
                <cybox:Observable idref="cybox:guid-1a937ec2-90ab-4e0e-a37c-db9b2e66a58e"/>
                <!-- Open Iran-Oil corrupted .doc file-->
                <cybox:Observable idref="cybox:guid-35f04c28-5fd2-4d72-8aae-2ad04ee1811f"/>
            </cybox:Observable_Composition></cybox:Observable>
            <cybox:Observable><cybox:Observable_Composition operator="AND">
                <!-- Download Iran-Oil invalid .mp4 downloader file-->
                <cybox:Observable idref="cybox:guid-f005fbc6-7427-43ea-8e1e-9a341836f76b"/>
                <!-- Create Iran-Oil .exe Trojan file-->
                <cybox:Observable idref="cybox:guid-210f18f3-3874-4f9a-861d-71b328be90c6"/>
                <!-- Execute Iran-Oil .exe Trojan file-->
                <cybox:Observable idref="cybox:guid-b650c988-aac7-45ff-967d-9f1e5fc66161"/>
            </cybox:Observable_Composition></cybox:Observable>
            <!-- Trojan .exe file connects out to C&C URLs/IPs-->
            <cybox:Observable idref="cybox:guid-a24ff8bc-b534-4616-838b-8bbe260a8e8f"/>
        </cybox:Observable_Composition>
    </cybox:Observable>

    <!-- CybOX enables a wide myriad of other potential observable pattern variations at the logical
composition level or utilizing patterns at the Object attribute level including Regex all of which allow the
user to define an almost infinitely variable set of patterns and filters -->
</cybox:Observables>
```

Appendix A. Leveraging the CybOX Language Data Model

There are two primary modes for leveraging the CybOX language to define cyber observable content: directly and indirectly.

- Directly leveraging the CybOX language involves simply leveraging a schematic implementation of the language to capture and utilize content.
- Indirectly leveraging the CybOX language involves leveraging a domain-specific language, standard, process or tool which within its own structure imports or includes elements of the CybOX language. Any domain-specific language, standard, process or tool is free to incorporate any relevant portions of the CybOX language via importing or including the appropriate data model types as instantiated in a schematic implementation of the language (e.g. using XML Schema).

  For example:

  - The Common Attack Pattern Enumeration and Classification (CAPEC)[6] can import the entire CybOX language XML Schema implementation from the ObservablesType on down.
  - The Malware Attribute Enumeration and Characterization (MAEC)[7] can import just the CybOX ActionType & ObjectType (along with portions of the CybOX library of common defined objects) to utlize as the foundation of its malware characterization.
  - The Common Event Expression (CEE)[8] can align with and import the CybOX EventType to serve as its broad scope structure for characterizing cyber events.

---

[6] http://capec.mitre.org

[7] http://maec.mitre.org

[8] http://cee.mitre.org

Appendix B. Extending the CybOX Language Data Model

The CybOX Language Data Model defines a set of core capabilities, as described within this Specification document and the accompanying CybOX Language Defined Objects Specification, with numerous extension points. This appendix highlights the opportunities for extension within the CybOX Language. It is particularly important to understand the role of CybOX Defined Object Models within the CybOX Language, as they form a large basis of cyber observable expression and allow CybOX to easily expand to cover new object types or new levels of detailed characterization of existing object types. Additionally, this appendix will raise awareness of some other extension points that have been built into the CybOX Language.

CybOX Defined Object Models

The primary foundation of the cyber observables construct lies in the set of observable objects that exist as stateful measures or are involved in observable actions and events. As such, any language like CybOX providing a practical solution for characterizing cyber observables must include the capability to describe a set of commonly observed objects utilizing a common set of attributes for any given object type. The diversity of the cyber domain however makes such a set of potential objects very large with new objects coming into play over time and differing use cases requiring different objects. To provide effective capability to a diverse set of use cases a cyber observable expression language like CybOX must provide a common library of defined object models for unrestricted use but also must incorporate them into the language in a way that makes it easy for new objects to be added. It must support the addition of new objects by domain-specific use cases independent of CybOX as well as the addition of new objects to the CybOX common library without affecting the rest of the CybOX language as the defined object portions of the language are the likely to experience the highest rate of change over time.

In the CybOX language, these defined object structures are defined in their own Models as described in the Data Model section of the accompanying CybOX Language Defined Objects Specification. The CybOX Defined Object Data Models each provide the necessary constructs for characterizing a comprehensive set of commonly leveraged attributes for any given defined object type. Where possible and appropriate the structure and syntax of these models or portions thereof adhere to relevant existing normative specifications. Due to the nature of uniquely comprehensive coverage of the CybOX language and its targeted support of a broad range of use cases, there exist several instances where the CybOX data models diverge from existing normative specifications through extension, aggregation, restriction or abstraction.

To ensure flexibility and extensibility all defined objects are incorporated into the CybOX language as extensions of the abstract DefinedObjectType which acts as a generalized placeholder in the language for context-specific structures and syntax of the various potential defined object types. Through this mechanism new defined object types can be created or existing types modified with no effect on the core CyboX language or any other non-dependent defined object type. Similarly, any domain-specific use case could create their own new defined object types as extensions of the abstract DefinedObjectType and use them in localized content. Sharing this data with any entities outside their scope may result in a limited ability to parse of validate content for that object type (unless the appropriate model is also shared) but all other portions of the CybOX language should work without issue. Over time, independently created defined object models will be reviewed and, if appropriate, incorporated into the CybOX common defined object library.

The CybOX library of defined object models is designed in an intentionally architected and modular fashion such that more complex or specialized objects can leverage and incorporate existing objects where appropriate. The two most common situations for this sort of incorporation are:

1. Defined objects which require attributes that are themselves more atomic-level defined objects.

   For example, the DNSRecordObjectType could make use of the AddressObjectType and the URIObjectType to describe its associated IP_Address and Domain_Name attributes.

2. Defined objects that are specializations sharing a significant basis with other defined objects.

   For example, the WindowsExecutableFileObjectType could be an extension of the WindowsFileObjectType adding PE-specific attributes and the WindowsFileObjectType could further be an extension of a basic FileObjectType adding Windows specific attributes to the general set of attributes shared by all files.

Other Abstract Types

The same abstract type approach described above for the DefinedObjectType is also leveraged by the CybOX language to enable other points of generalized extension. A short list of these other extension points includes:

- BaseObjectAttributeType

  The BaseObjectAttributeType is an abstract type that acts as a basis for all atomic-level object attribute types and provides the basic capabilities for pattern characterization for a given object attribute. There are a range of extensions of this abstract type provided in the CybOX language for a variety of primitive data types. All leaf attributes for CybOX objects should be of types defined using extensions from the BaseObjectAttributeType.

- DomainSpecificObjectAttributeType

  The DomainSpecificObjectAttributesType is an abstract type placeholder within the CybOX language enabling the inclusion of domain-specific metadata for an object through the use of a custom type defined as an extension of this base abstract type. This enables domains utilizing CybOX such as malware analysis or forensics to incorporate non-generalized object metadata from their domains into CybOX objects.

- DefinedEffectType

  The DefinedEffectType is an abstract placeholder for various predefined Object Effect types (e.g. DataReadEffect, ValuesEnumeratedEffect or StateChangeEffect) that can be instantiated in its place through extension of the DefinedEffectType. This mechanism enables the specification of a broad range of types of potential complex action effects on Objects. The set of Defined Effect types (extending the DefinedEffectType) are maintained as part of the core CybOX language.

- PersonnelType

  The PersonnelType is an abstracted data type to standardize the description of sets of personnel.

- ToolSpecificDataType

  The ToolSpecificDataType is an abstract type placeholder within the CybOX language enabling the inclusion of metadata for a specific type of tool through the use of a custom type defined as an extension of this base abstract type.

- IndicatorType

  The IndicatorType is an abstract type placeholder within the CybOX language enabling the inclusion of varying specifications for indicators contributing to this cyber observation. Externally defined indicator structures can be defined through the use of a custom type defined as an extension of this base abstract type.

- **FileObjectType**
  - FileAttributeType

    The FileAttributeType type specifies a native attribute of a file. Since native attributes are platform-specific, it is defined here as an abstract type.

  - FilePermissionsType

    The FilePermissionsType specifies the native permissions of a file. Since this is a platform-specific attribute, it is defined here as an abstract type and then implemented in any platform-specific derived CybOX file objects.

- **ProcessObjectType**
  - ProcessStatusType

    The ProcessStatusType is used for specifying the status of a running or terminated process. Since this property is platform-specific, it is created here as an abstract type and then used in the platform-specific process CybOX objects.

- **UserAccountObjectType**
  - PrivilegeType

    The PrivilegeType specifies a specific privilege that a user has. This is an abstract type since user privileges are OS-specific, and is extended as needed in the derived CybOX objects.

  - GroupType

    The GroupType specifies a group that a user account belongs to. This is an abstract type since group IDs are OS-specific, and is extended as needed in the derived CybOX objects.

- **VolumeObjectType**
    - VolumeOptionsType

      The VolumeOptionsType specifies the particular options set for the volume. This is an abstract type since volume options are OS-specific, and is extended by the related OS-specific CybOX volume objects.

Generalized Extension Mechanisms

To support domain-specific attribute adornment on key components, CybOX provides an open attribute wildcard extension mechanism as part of DefinedObjectType, ActionType and ObjectType.

CybOX provides a generalized data structure named MetadataType that can be used to capture any sort of custom metadata structure via a field/value tuple and recursion.

Fundamental Extension

The most basic, simple and broadly applicable extension mechanism is via domain-specific extension of any of the modular and layered set of native CybOX types.

Appendix C. Normative References

[1] W3C Recommendation for Hex-Encoded Binary Data
http://www.w3.org/TR/xmlSchema-2/#hexBinary

[2] W3C Recommendation for Base64-Encoded Binary Data
http://www.w3.org/TR/xmlschema-2/#base64Binary

[3] W3C Recommendation for Boolean Data
http://www.w3.org/TR/xmlSchema-2/#boolean

[4] W3C Recommendation for Integer Data
http://www.w3.org/TR/xmlSchema-2/#integer

[5] W3C Recommendation for Unsigned Integer Data
http://www.w3.org/TR/xmlschema-2/#unsignedInt

[6] W3C Recommendation for Non-Negative Integer Data
http://www.w3.org/TR/xmlschema-2/#nonNegativeInteger

[7] W3C Recommendation for Positive Integer Data
http://www.w3.org/TR/xmlschema-2/#positiveInteger

[8] W3C Recommendation for Long Data
http://www.w3.org/TR/xmlschema-2/#long

[9] W3C Recommendation for Unsigned Long Data
http://www.w3.org/TR/xmlschema-2/#unsignedLong

[10] W3C Recommendation for Double Data
http://www.w3.org/TR/xmlschema-2/#double

[11] W3C Recommendation for Float Data
http://www.w3.org/TR/xmlSchema-2/#float

[12] W3C Recommendation for Time Data
http://www.w3.org/TR/xmlschema-2/#time

[13] W3C Recommendation for Date Data
http://www.w3.org/TR/xmlschema-2/#date

[14] W3C Recommendation for DateTime Data
http://www.w3.org/TR/xmlschema-2/#dateTime

[15] W3C Recommendation for Duration Data
http://www.w3.org/TR/xmlschema-2/#duration

[16] W3C Recommendation for String Data

http://www.w3.org/TR/xmlSchema-2/#string

[17] W3C Recommendation for QName Data
http://www.w3.org/TR/xmlschema-2/#QName

[18] W3C Recommendation for URI Data
http://www.w3.org/TR/xmlschema-2/#anyURI

RFC 2461: Neighbor Discovery for IP Version 6 (IPv6)
http://www.ietf.org/rfc/rfc2461.txt

RFC 4861: Neighbor Discovery for IP Version 6 (IPv6)
http://www.ietf.org/rfc/rfc4861.txt

RFC 791: Internet Protocol
http://www.ietf.org/rfc/rfc791.txt

RFC 2474: Differentiated Services Field
http://www.ietf.org/rfc/rfc2474.txt

RFC 3168: Explicit Congestion Notification
http://www.ietf.org/rfc/rfc3168.txt

RFC 3692: Experimental and Testing Numbers
http://www.ietf.org/rfc/rfc3692.txt

RFC 3513: Internet Protocol Version 6 (IPv6) Addressing Architecture
http://www.ietf.org/rfc/rfc3513.txt

RFC 2460: Internet Protocol Version 6 (IPv6) Specification
http://www.ietf.org/rfc/rfc2460.txt

RFC 2402: IP Authentication Header
http://www.ietf.org/rfc/rfc2402.txt

RFC 2406: IP Encapsulating Security Payload (ESP)
http://www.ietf.org/rfc/rfc2406.txt

RFC 1347: TCP and UDP with Bigger Addresses (TUBA)
http://www.ietf.org/rfc/rfc1347.txt

RFC 4443: Internet Control Message Protocol (ICMPv6) for IPv6 Specification
http://www.ietf.org/rfc/rfc4443.txt

RFC 2463: Internet Control Message Protocol (ICMPv6) for IPv6 Specification
http://www.ietf.org/rfc/rfc2463.txt

RFC 768: User Datagram Protocol
http://www.ietf.org/rfc/rfc768.txt

RFC 791: Internet Protocol

http://www.ietf.org/rfc/rfc791.txt

RFC 792: Internet Control Message Protocol (ICMP)
http://www.ietf.org/rfc/rfc792.txt

RFC 793: Transmission Control Protocol
http://www.ietf.org/rfc/rfc793.txt

RFC 826: Ethernet Address Resolution Protocol
http://tools.ietf.org/html/rfc826

RFC 903: A Reverse Address Resolution Protocol
http://www.ietf.org/rfc/rfc903.txt

RFC 1219: On the Assignment of Subnet Numbers
http://www.ietf.org/rfc/rfc1219.txt

RFC 1349: Type of Service in the Internet Protocol Suite
http://www.ietf.org/rfc/rfc1349.txt

RFC 5101: Specification of the IPFIX Protocol
http://www.ietf.org/rfc/rfc5101.txt

RFC 5102: Information Model for IP Flow Information Export
http://www.ietf.org/rfc/rfc5102.txt

RFC 3954: Cisco Systems Netflow Services Export Version 9
http://www.ietf.org/rfc/rfc3954.txt

RFC 821: Simple Mail Transfer Protocol
http://www.ietf.org/rfc/rfc821.txt

RFC 2076:Common Internet Message Headers
http://www.ietf.org/rfc/rfc2076.txt

RFC 822: Standard for the Format of Arpa Internet Text Messages
http://www.ietf.org/rfc/rfc822.txt

RFC 2822: Internet Message Format
http://www.ietf.org/rfc/rfc2822.txt

RFC 1035: Domain Names – Implementation and Specification
http://www.ietf.org/rfc/rfc1035.txt

RFC 3597: Handling of Unknown DNS Resource Record (RR) Types
http://www.ietf.org/rfc/rfc3597.txt

RFC 1058: Routing Information Protocol
http://www.ietf.org/rfc/rfc1058.txt

RFC 147: The Definition of a Socket

http://www.ietf.org/rfc/rfc147.txt

RFC 1122: Requirements for Internet Hosts –Communication Layers
http://www.ietf.org/rfc/rfc1122.txt

RFC 2396: Uniform Resource Identifiers (URI): Generic Syntax
http://www.ietf.org/rfc/rfc2396.txt

RFC 1518: An Architecture for IP Address Allocation with CIDR
http://www.ietf.org/rfc/rfc1518.txt

RFC 5070: The Incident Object Description Exchange Format
http://www.ietf.org/rfc/rfc5070.txt

RFC 5901: Extensions to the IODEF-Document Class for Reporting Phishing
http://www.ietf.org/rfc/rfc5901.txt

X.509
http://www.itu.int/rec/T-REC-X.509/en

<Semaphore.h>
http://pubs.opengroup.org/onlinepubs/007904975/basedefs/semaphore.h.html

<Socket.h>
http://pubs.opengroup.org/onlinepubs/009695399/basedefs/sys/socket.h.html

route(8) – Linux man page
http://linux.die.net/man/8/route

mount(8) – Linux man page
http://linux.die.net/man/8/mount

Event Schema
http://msdn.microsoft.com/en-us/library/aa385201(v=vs.85).aspx

Event Objects
http://msdn.microsoft.com/en-us/library/windows/desktop/ms682655(v=vs.85).aspx

Mutex Objects
http://msdn.microsoft.com/en-us/library/windows/desktop/ms684266(v=vs.85).aspx

Semaphore Objects
http://msdn.microsoft.com/en-us/library/windows/desktop/ms685129(v=vs.85).aspx

Waitable Timer Objects
http://msdn.microsoft.com/en-us/library/windows/desktop/ms687012(v=vs.85).aspx

An In-depth Look into the Win32 Portable Executable File Format
http://msdn.microsoft.com/en-us/magazine/cc301805.aspx

File Management Reference

http://msdn.microsoft.com/en-us/library/aa364233(v=vs.85).aspx

Mailslots
http://msdn.microsoft.com/en-us/library/windows/desktop/aa365576(v=vs.85).aspx

Network Share Functions
http://msdn.microsoft.com/en-us/library/windows/desktop/bb525391(v=vs.85).aspx

Named Pipes
http://msdn.microsoft.com/en-us/library/windows/desktop/aa365590(v=vs.85).aspx

Registry
http://msdn.microsoft.com/en-us/library/windows/desktop/ms724871(v=vs.85).aspx

Services
http://msdn.microsoft.com/en-us/library/windows/desktop/ms685141(v=vs.85).aspx

GFlags
http://msdn.microsoft.com/en-us/library/windows/hardware/ff549557(v=vs.85).aspx

System Restore
http://msdn.microsoft.com/en-us/library/windows/desktop/dd408121(v=vs.85).aspx

Task Scheduler
http://msdn.microsoft.com/en-us/library/windows/desktop/aa383614(v=vs.85).aspx

Process and Thread Reference
http://msdn.microsoft.com/en-us/library/windows/desktop/ms684852(v=vs.85).aspx

Volume Object
http://msdn.microsoft.com/en-us/library/windows/desktop/aa383970(v=vs.85).aspx

Memory Management
http://msdn.microsoft.com/en-us/library/windows/desktop/aa366779(v=vs.85).aspx

Task Scheduler
http://msdn.microsoft.com/en-us/library/windows/desktop/aa383614(v=vs.85).aspx

Windows Driver Development
http://msdn.microsoft.com/en-us/library/windows/hardware/ff557573(v=vs.85).aspx

Appendix D. Changelog

Appendix E. Acronyms

ACL	Access Control List
API	Application Programming Interface
APC	Asynchronous Procedure Calls
ARE	Advanced Regular Expression
ARP	Address Resolution Protocol
ARM	Acorn RISC Machine
API	Application Programming Interface
AS	Autonomous System
ASN	Autonomous System Number
ASP	Active Server Pages
ATM	Asynchronous Transfer Mode
BIOS	Basic Input/Output System
BCC	Blind Carbon Copy
BRE	Basic Regular Expression
CAPEC	Common Attack Pattern Enumeration and Classification
CC	Carbon Copy
CCE	Common Configuration Enumeration
CDS	Content Delivery System
CIDR	Classless Inter-Domain Routing
CLR	Common Language Runtime
CPE	Common Platform Enumeration
CVE	Common Vulnerabilities and Exposures
CWE	Common Weakness Enumeration
CybOX	Cyber Observable eXpression
DBMS	Database Management System
DHCP	Dynamic Host Configuration Protocol
DHS	Department of Homeland Security
DLL	Dynamically Linked Library
DNS	Domain Name System
DST	Daylight Savings Time
ECMA	European Computer Manufacturers Association
EP	Entry Point
ERE	Extended Regular Expression
EVR	Epoch, version, and release
FTP	File Transfer Protocol
FQDN	Fully Qualified Domain Name
FQN	Fully Qualified Name
GNU	GNU's Not Unix!
GUI	Graphical User Interface
GUID	Globally Unique Identifier
HIDS	Host Intrusion Detection System
HIPS	Host Intrusion Prevention System
HTML	HyperText Markup Language
HTTP	Hyper Text Transfer Protocol
IAVM	Information Assurance Vulnerability Management

ICMP	Internet Control Message Protocol
ID	Identifier
IDT	Interrupt Descriptor Table
IETF	Internet Engineering Task Force
IMAP	Internet Message Access Protocol
INODE	Index Node
IP	Internet Protocol
IPFIX	Internet Protocol Flow Information Export
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
IRP	Interrupt Request Packet
IPC	Inter-Process Communication
JSP	Java Server Pages
KVM	Keyboard Video Mouse
MAC	Media Access Control
MIB	Management Information Base
MIPS	Microprocessor without Interlocked Pipeline Stages
MUTEX	MUTual Exclusion
MSS	Maximum Segment Size
NAC	Network Access Control
NDP	Network Discovery Protocol
NETBEUI	NetBIOS Extended User Interface
NETBIOS	Network Basic Input/Ouput System
NIDS	Network Intrusion Detection System
NIPS	Network Intrustion Prevention System
NIST	National Institute of Standards and Technology
NSA	National Security Agency
OS	Operating System
PCRE	Perl-Compatible Regular Expression
PE	Portable Executable
PEID	Portable Executable Identifier
PID	Process Identifier
POP	Post Office Protocol
POSIX	Portable Operating System Interface
PHP	PHP HyperText Processor
PPC	PowerPC
RARP	Reverse Address Resolution Protocol
RDF	Resource Description Framework
RFC	Request For Comment
RISC	Reduced Instruction Set
RSA	Ron Rivest, Adi Shamir, and Leonard Adleman
RUID	Real User ID
RVA	Relative Virtual Address
SID	Security Identifier
SIM	Security Information Management
SMB	Server Message Block
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol

SO	Socket Option
SOAP	Simple Object Access Protocol
SPARC	Scalable Processor ARChitecture
SSDT	System Service Dispatch Table
SQL	Structured Query Language
TCP/IP	Transmission Control Protocol/Internet Protocol
TLS	Thread Local Storage
TOS	Type of Service
TTL	Time To Live
UDP	User Datagram Protocol
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URN	Uniform Resource Name
USB	Universal Serial Bus
UUID	Universally Unique Identifier
VLAN	Virtual Lan
VM	Virtual Machine
W3C	World Wide Web Consortium
XOR	Exclusive OR
XML	eXtensible Markup Language
XSD	XML Schema Document