Version 0.7 (Official) – Release Notes

The Version 0.7 release of CybOX includes a significant number of changes to improve the expressiveness of the schema to better support the broad set of use cases it targets (malware characterization, indicator and incident data sharing, event management, etc.), to improve simplicity and efficiency, and to mature descriptive formality for reference and implementation by other standards and users.

Summary of changes from CybOX_v0.62 to CybOX_v0.7:

- **Annotated all Attributes, Elements & Types throughout the core schema and most of the defined object schemas. A formal specification is in the works.**
- **Greatly increased the expressiveness of object attributes enabling a variety of condition specifications including use of regex. The new structure enables specification not only of attribute value "equal to" something but also conditions such as: contains, does not contain, start with, ends with, greater than, greater than or equal to, less than, less than or equal to, is in range, is not in range, is in set, is not in set, fits pattern (enabling use of regex, xpath or any other pattern syntax), bitwise and, bitwise or.**
- **Greatly enhanced the expressiveness of the Action construct to more fully characterize cyber actions for use cases such as malware characterization and event management. Details are in the bullets below.**
- **Added the capability to describe Measure_Source content directly to Event, Action and Object constructs to enable description of the producing/observing source for each of these constructs.**
- **Flattened out the core schema in several places through restructuring for expressiveness, yielding a simpler and more efficient schema.**
- **For attributes and elements with enumerated value restrictions, the inline restrictions for enumerated values have been replaced with enumerated restriction types offering greater flexibility and maintainability.**
- **Added global Pools at the highest level of the core schema for Events, Actions, Objects and Attributes to enable potential reduction of redundancy and referential efficiencies.**

Detailed changes from CybOX_v0.62 to CybOX_v0.7:

- **Annotated all Attributes, Elements & Types throughout the core schema. A formal specification is in the works.**
- **All Object Attributes are now derived from a common base type with the following attibutes: ID, IDREF, Datatype, Condition, PatternType, RegexSyntax, StartRange, EndRange, ValueSet. This new attribute structure greatly increases the expressiveness of object attributes in v0.7 over v0.62. The new structure enables specification not only of attribute value "equal to" something but also conditions such as: contains, does not contain, start with, ends with, greater than, greater than or equal to, less than, less than or equal to, is in range, is not in**

range, is in set, is not in set, fits pattern (enabling use of regex, xpath or any other pattern syntax), bitwise and, bitwise or.

- **Greatly enhanced the expressiveness of the Action construct. Details are in the bullets below.**
- **Flattened out the core schema by removing the Measure, Measure_Source and Pattern elements under Observable and placing Stateful_Measure and Event together as a choice directly under the Observable element.**
- **Added the Measure_Source content (using the MeasureSourceType) directly to Event (Producer-Observer), Action (Discovery_Method) and Object (Discovery_Method) constructs to enable description of the producing/observing source for each of these constructs.**
- **Flattened the core schema by removing the primary Choice under the Observable element separating an observable specification from a recursive reference of Observable. Also removed the Sequence on the recursive Observable branch and moved just the recursive Observable element up under the same Choice with Stateful_Measure and Event. This simplified the schema while making it more flexible and expressive (enabling full observable description of composite observables).**
- **For xs:String type attributes and elements with enumerated value restrictions, the inline restriction for enumerated values have been replaced with enumerated restriction types (e.g. OperatorTypeEnum, ObjectTypeEnum, SourceTypeEnum, ToolTypeEnum).**
- **Added optional global Pools at the highest level of the core schema for Events, Actions, Objects and Attributes to enable potential reduction of redundancy and referential efficiencies.**
- Simplified ObservableType into a straightforward global complex type without any unnecessary extension.
- A richly expressive Tools construct has been added to Measure_SourceType to enable richer description of tools involved in the measure source.
- The following attributes have been added to Measure_SourceType for increased expressiveness: ToolType, AnalysisType, AnalysisMethod, InformationSourceType.
- Enabled the capture of multiple Actions under Event rather than just a single Action.
- Added ID and IDREF attributes to Event for better identification and reference capability.
- The following attributes have been added to Action for increased expressiveness: Type, ordinal_position, Action_Status, Context, Network_Protocol, timestamp, anyAttribute.
- The Name attribute under Action has been moved into an element and can capture either a Defined_Name from an enumerated restriction of values (including a wide range of practical complex actions) or non-standardized Undefined_Name.
- A Description element has been added under Action.
- An Action_Arguments construct has been added under Action to capture arguments utilized by actions. Like Action_Name, the action arguments can capture either an Argument_Name-Defined from an enumerated restriction of values or non-standardized Argument_Name-Undefined.
- The simple Object multiplicity under Action in CybOX_v0.62 has been replaced with an Associated_Objects element containing a multiplicity of Associated_Objects which extend the

basic Object_Type to include an attribute specifying an AssociationType (potential values of Initiating, Affected, Utilized or Returned) and an Action-Pertinent_Object_Attributes element that enables specification of which attributes of the associated Object are pertinent to the given Action.

- A Relationships construct has been added under Action to enable identification/characterization of relationships between Actions.
- The Action_Role attribute under Object has been removed and replaced with the aforementioned more richly expressive AssociationType attribute under Associated_Object.
- The Object_State element has been converted to an attribute under Object.
- An anyAttribute has been added as an attribute under Object to enable flexibility and use of custom data.
- An anyAttribute has been added to the Defined_Object construct under Object to enable flexibility and use of custom data.
- The Custom_Attributes construct under Object now utilizes the standard Attribute_Type instead of a distinct structure.
- The following attributes have been added under Related_Object: Object_State and an anyAttribute.
- The Action_Role attribute has been removed under Related_Object.
- An enumerated list of values has been added for the Relationship attribute under Related_Object.
- A Defined_Effect construct has been added as an abstract type under Object to enable capture of more complex effect data (e.g. state change, data read, data written) related to a particular Action and Object. New types deriving from the Defined_Effect abstract type have been added for several common defined effects (StateChangeEffectType, DataReadEffectType, DataWrittenEffectType, DataSentEffectType, DataReceivedEffectType, ValuesEnumeratedEffectType, PropertiesEnumeratedEffectType, PropertyReadEffectType & SendControlCodeEffectType).

Defined Object Schema Changes:

Disk Object

- Add ‘Free_Space’ element for specifying amount of free space available on the disk
- Added ‘type’ attribute for specifying the type of the disk, e.g. fixed, removable, etc.

DNS Cache Object

- Added ‘DNS_Cache_List_Type’ type for aggregating DNC Cache entries

Library Object

- Removed ‘Init’ element

Memory Object

- Renamed ‘Region_Start’ element to ‘Region_Start_Address’ for clarity

Process Object

- Renamed ‘Port_Type’ type to ‘Network_Connection_Type’ for clarity
- Renamed ‘Port_List_Type’ type to ‘Network_Connection_List_Type’ for clarity
- Renamed ‘Port_State_Enum’ type to ‘Connection_State_Enum’ for clarity

System Object

- Renamed ‘User’ element to ‘Username’ for clarity

UNIX Network Route Object
- Renamed ‘Unix_Network_Route_Entry_Object’ to ‘Unix_Network_Route_Entry_Type’ for consistency

UNIX Process Object
- Removed ‘timestamp’ attribute from ‘Unix_Process_Status_Type’ type to eliminate redundancy

User Account Object
- Moved ‘Password_Required’ element to an attribute for consistency

Windows Driver Object
- Made all elements optional

Windows Executable File Object
- Renamed ‘PE_Info_Type’ type to ‘PE_Attributes_Type’ for clarity

Windows Kernel Object
- Renamed ‘FunctionAddress’ element in ‘SSDT_Entry_Type’ type to ‘Current_Function_Address’ for clarity
- Added ‘Original_Function_Address’ element to ‘SSDT_Entry_Type’ type
- Added ‘Index’ element to ‘SSDT_Entry_Type’ type
- Added ‘Index’ element to ‘IDT_Entry_Type’ type

Windows Process Object
- Added ‘Startup_Info_Type’ type for specifying information contained in the STARTUP_INFO structure for the process
- Added ‘Startup_Info’ element for specifying aforementioned startup info

Windows Registry Object
- Removed ‘Path’ element to eliminate redundancy

Windows Service Object
- Moved ‘Service_DLL_Signature_Exists’ element to an attribute for consistency
- Moved ‘Service_DLL_Signature_Verified’ element to an attribute for consistency

Windows System Object
- Added ‘NetBIOS_Name’ element for specifying the system’s NETBios name
- Added ‘Windows_Directory’ element for specifying the path to the Windows install directory
- Added ‘Windows_System_Directory’ element for specifying the path to the Windows System directory
- Added ‘Windows_Temp_Directory’ element for specifying the path to the Windows temporary files directory

Windows Thread Object
- Added ‘Context’ element for specifying the thread context structure