OpenC2 – STIX COA Subgroup

February 25, 2016

AGENDA

# Logistics

* Meet in off-weeks of the main group
* Identify a Plan B timeslot of our own in case we need it or the main group needs the off-week.

# Objective

* Define a STIX-COA extension that allows STIX users to use OpenC2 to express automated COAs.

# Use Cases

* What are one or more real use cases for extending STIX to include OpenC2?

# Options

* STIX “StructuredCOAType” is an abstract type just made for this purpose.
  + Extend StructuredCOAType to define all the fields of the OpenC2 action construct - action, target, target-specifier, actuator, actuator-specifier and modifier?
  + Extend StructuredCOAType to define action, modifier and use the CourseOfActionType to capture target & target-specifier (Parameter Observable), actuator & actuator-specifier (Information source)?
  + Define an abstract type for OpenC2 actions extending from StructuredCOAType and define extensions for network and endpoint based actions?
* Are there any options we should consider?
* Should we target a version of STIX other than current production (1.2)? [The version of CybOX we assume would be whatever OpenC2 assumes.]
* Should the extension leverage existing vocabularies or define new ones? Should we develop our own vocabularies/enums for targets, actuators etc. or should we extend existing controlled vocabularies from STIX and/or CybOX?

# Deliverables

* What do we need to deliver?
  + XML and/or JSON schema?
  + Documentation: what additional document(s)?
  + Code: Java and/or Python reference implementation?

# Approach

* How should the varied resources of this subgroup be used to meet the objective?
* What more do we need to know before we can get a grasp on a preliminary schedule?

# Further topics for discussion:

* How will actions evolve?
  + Define relationships such as child-of, parent-of, related-to etc.
  + Would the syntax change for the different types of COAs? -  SuggestedCOA, RequestedCOA, COATaken
  + How would a COATaken relate to SuggestedCOA or RequestedCOA?
* Do we need a translator to convert from OpenC2 to the STIX format and vice versa?

Kickoff meeting Feb 25 2016

Option 1

* 1. Extend StructuredCOAType to define all the fields of the OpenC2 action construct - action, target, target-specifier, actuator, actuator-specifier and modifier

**Pros**: Self-contained directly mapping to an OpenC2 command

**Cons**: Could duplicate some of the content from over-arching COAType – example Target will be same as the parameter observable

Possible solution: Reference to data where possible instead of re-defining. Referencing but referencing indicates knowing the id to refer to.

Example:

<incident:Course\_Of\_Action xsi:type="COA:CourseOfActionType" id="Internal Blocking">

                    <COA:Stage>Remedy</COA:Stage>

<coa:Parameter\_Observables cybox\_major\_version=*"2"* cybox\_minor\_version=*"1"* cybox\_update\_version=*"0"*>

<cybox:Observable id=*"example:Observable-e04425e4-60a2-4d91-a9f9-0ca956f19edb"*>

<cybox:Object id=*"example:Address-d5bc7186-319d-44e0-85f4-0b65f59034a3"*>

<cybox:Properties xsi:type=*"AddressObj:AddressObjectType"* category=*"ipv4-addr"*>

<AddressObj:Address\_Value condition=*"Equals"*>10.10.10.10</AddressObj:Address\_Value>

</cybox:Properties>

</cybox:Object>

</cybox:Observable>

</coa:Parameter\_Observables>

                    <COA:Structured\_COA xsi:type="openc2:courseOfAction" id="unique\_id">

                        <COA:action>Deny</COA:action>

<COA:target>

<cybox:Object id=*"example:Address-d5bc7186-319d-44e0-85f4-0b65f59034a3"*>

<cybox:Properties xsi:type=*"AddressObj:AddressObjectType"* category=*"ipv4-addr"*>

<AddressObj:Address\_Value condition=*"Equals"*>10.10.10.10</AddressObj:Address\_Value>

</cybox:Properties>

</cybox:Object>

</COA:target>

                    </COA:Structured\_COA>

                </incident:Course\_Of\_Action>

Option 2

Extend StructuredCOAType to define action, modifier and use the CourseOfActionType to capture target & target-specifier (Parameter Observable), actuator & actuator-specifier (Information source)

**Pros**: Leverages the CourseOfActionType for existing structures and reduces duplication of data

**Cons**: Breaks up the OpenC2 command