OpenC2 Course Of Action Specification Version 1.0

Working Draft 01

March 2016

Chairs:

Jyoti Verma (jyoverma@cisco.com), [Cisco Systems Inc.](http://www.cisco.com),

Ron Rudman (rjrudman@mitre.org), [MITRE](http://www.mitre.org)

Editors:

Alex Foley ([alexander.foley@bankofamerica.com](mailto:alexander.foley@bankofamerica.com)),

Applebaum, Andy ([aapplebaum@mitre.org](mailto:aapplebaum@mitre.org)),

Jim Ronayne ([jkronay@nsa.gov](mailto:jkronay@nsa.gov)),

Brule, Joseph M (jmbrule@radium.ncsc.mil),

Joyce Fai ([joyce.fai@gd-ms.com](mailto:joyce.fai@gd-ms.com)),

Juan Gonzalez (juam.m.gonzalez@hq.dhs.gov),

Matt Shabat (Matthew.Shabat@HQ.DHS.GOV),

Olga Livingston (olga.livingston@hq.dhs.gov),

Sourabh Satish (sourabh@phantom.us),

Moss, Mark B. (Mark.Moss@jhuapl.edu),

Romano, Jason (Jason.Romano@gd-ms.com),

Peterson, Joan B (jbpeter@nsa.gov)

Additional artifacts:

This prose specification is one component of a Work Product that also includes:

· XML schema: openC2\_stix\_COA.xsd

· JSON schema: openC2\_stix\_COA.json

· Other parts :

Related work:

This specification is related to:

· OpenC2 specification - <http://openc2.org>

Declared XML namespaces:

· The specification defines the following namespace

*"http://stix.mitre.org/extensions/OpenC2StructuredCOA-1"*

Abstract:

OpenC2 defines a language at a level of abstraction that will enable command and control of cyber defense components with enough generality to provide flexibility in the implementations of devices and accommodate future products. This specification defines a representation of the OpenC2 command using STIX Course Of action

Table of Contents

1 Introduction................................................................................................................................ 3

1.1 Terminology........................................................................................................................... 3

2 Section Title.............................................................................................................................. 4

2.1 Level 2 section title................................................................................................................ 4

2.1.1 Level 3 section title......................................................................................................... 4

2.1.1.1 Level 4 section title is usually deepest for Table of Contents.............................................. 4

Appendix A. Acknowledgments........................................................................................................ 5

Appendix B. Example Title.............................................................................................................. 6

B.1 Subsidiary section.................................................................................................................. 6

B.1.1 Sub-subsidiary section..................................................................................................... 6

B.1.1.1 Sub-sub-subsidiary section............................................................................................. 6

Appendix C. Revision History............................................................................................................ 7

# **1** **Introduction**

An OpenC2 command has the following syntax:

<ACTION> (

TARGET (

type = <TARGET\_TYPE>,

[<target-specifier>]

),

[ACTUATOR (

type = <ACTUATOR\_TYPE>,

[<actuator-specifier>]

)],

[<modifiers>]

)

Fields denoted with angle brackets (“<>”) are replaced with the appropriate details. Fields denoted with square brackets (“[]”) are optional. This specification defines a representation of the above command as a STIX Course Of Action.

The action accepts the following fields:

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| **ACTION**  **TARGET** |  | The task or activity to be performed (i.e., the ‘verb’) |
| **type** | target | The object of the action. The ACTION is performed on the TARGET. |
| **target-specifier** | various | Optional. The specifier further describes a specific target, a list of targets, or a class of targets. |
| **ACTUATOR** |  | Optional. |
| **type** | actuator | The subject of the action. The ACTUATOR executes the ACTION on the TARGET. |
| **actuator-specifier** | various | Optional. The specifier further describes a specific actuator, a list of actuators, or a class of actuators. |
| ***modifiers*** |  | Optional. Provide additional information about the action such as time, periodicity, duration, and location. |

## **1.1 Terminology**

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 [RFC2119].

.

# **2** **Design Considerations**

We considered 2 options while designing the specification and narrowed down on Option 1 for version 1.0.

## **2.1 Option 1**

Extend StructuredCOAType to define all the fields of the OpenC2 action construct - action, target, target-specifiers, actuator, actuator-specifiers and modifiers

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| The entire OpenC2 command is represented in one structure | Since the StructuredCOA is wrapped inside of CourseOfAction and there are some overlapping fields in the CourseOfAction, this may be a cause of some confusion. |
|  |  |
|  |  |

## **2.2 Option 2**

Extend CourseOfActionType and StructuredCOAType to capture OpenC2 constructs using the two structures combined

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| Overlap between the CourseOfActionType and StructuredCOAType will be greatly reduced | The OpenC2 command will be deconstructed into 2 separate structures and re-constitution if desired might be challenging. |
|  |  |
|  |  |

# **3** **Data Model**

## **3.1 Action type vocabulary**

The vocabulary defined here reflects the actions defined by the main OpenC2 spec and are subject to change

|  |  |
| --- | --- |
| **Actions THAT CONTROL PERMISSIONS** | |
| **DENY** | The DENY action prevents a certain event or action from completing. |
| **CONTAIN** | The CONTAIN action stipulates the isolation of a file or process such that it cannot modify or access assets that support the business or operations of the enclave. |
| **ALLOW** | The ALLOW action permits the access to or execution of something. |
| **Actions THAT CONTROL ACTIVITIES/DEVICES** | |
| **START** | The START action initiates a system or an activity. |
| **STOP** | The STOP action halts a system or ends an activity. |
| **RESTART** | The RESTART action conducts a STOP of a system or an activity followed by a START of a system or an activity. |
| **PAUSE** | The PAUSE action ceases a system or activity while maintaining state |
| **RESUME** | The RESUME action starts a system or activity from a paused state. |
| **SET** | The SET action changes a configuration or state of a managed entity within an IT system. |
| **UPDATE** | The UPDATE action instructs the component to retrieve and process a large or comprehensive software update, reconfiguration or some other update |
| **MOVE** | The MOVE action changes the location of a file, subnet, network, or, process. |
| **REDIRECT** | The REDIRECT action changes the flow of traffic to a particular destination other than its original intended destination. |
| **DELETE** | The DELETE action removes data and files. |
| **SNAPSHOT** | The SNAPSHOT action records and stores the state of a target at an instant in time. |
| **DETONATE** | The DETONATE action executes and observes the behavior of an object (e.g., file, hyperlink) in a manner that isolates the object from assets that support the business or operations of the enclave. |
| **RESTORE** | The RESTORE action deletes and/or replaces files, settings, or attributes such that the state of the system is identical to its state at some previous time. |
| **SAVE** | The SAVE action commits data or system state to memory. |
| **MODIFY** | The MODIFY action augments, enhances, transforms, or changes some aspect of a system |
| **THROTTLE** | The THROTTLE action adjusts the throughput of the entire data flow to a different rate |
| **DELAY** | The DELAY action stops or holds up an activity or data transmittal. |
| **SUBSTITUTE** | The SUBSTITUTE action replaces all or part of the data, content or payload in the least detectable manner. |
| **COPY** | The COPY action duplicates a file or data flow. |
| **SYNC** | The SYNC action synchronizes a sensor or actuator with other system components. |
| **SENSOR RELATED ACTION** | |
| **DISTILL** | The DISTILL action tasks the sensor to send a summary or abstraction of the sensing information instead of the raw data feed. |
| **AUGMENT** | The AUGMENT action tasks the sensor to do a level of preprocessing or sense making prior to sending the sensor data. |
| **EFFECTS-BASED ACTION** | |
| **INVESTIGATE** | The INVESTIGATE action tasks the recipient enclave to aggregate and report information as it pertains to an anomaly. |
| **MITIGATE** | The MITIGATE action tasks the recipient enclave to circumvent the problem without necessarily eliminating the vulnerability or attack point. |
| **REMEDIATE** | The REMEDIATE tasks the recipient enclave to eliminate the vulnerability or attack point. |
| **Actions THAT GATHER and CONVEY Information** | |
| **SCAN** | The scan action is the systematic examination of some aspect of the entity or its environment in order to obtain information**.** |
| **LOCATE** | The LOCATE action requests the physical, logical, functional, or organizational location of an entity. |
| **QUERY** | The QUERY action initiates a single request for information. |
| **REPORT** | The REPORT action tasks an entity to provide information to a designated recipient of the information. |
| **GET** | The GET action tasks an entity to retrieve a specific object. |
| **NOTIFY** | The NOTIFY action is used to direct an entity to send information to another entity. |

## **3.2 Actuator type vocabulary**

## **3.3 Modifier vocabulary**

# **4** **External Dependencies**

## **4.1 Cybox Dependency – call out the different cybox objects needed for the various actions**

Appendix A. Acknowledgments

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants: