OpenC2

BIWEEKLY FORUM MEETING

08 Dec 2016

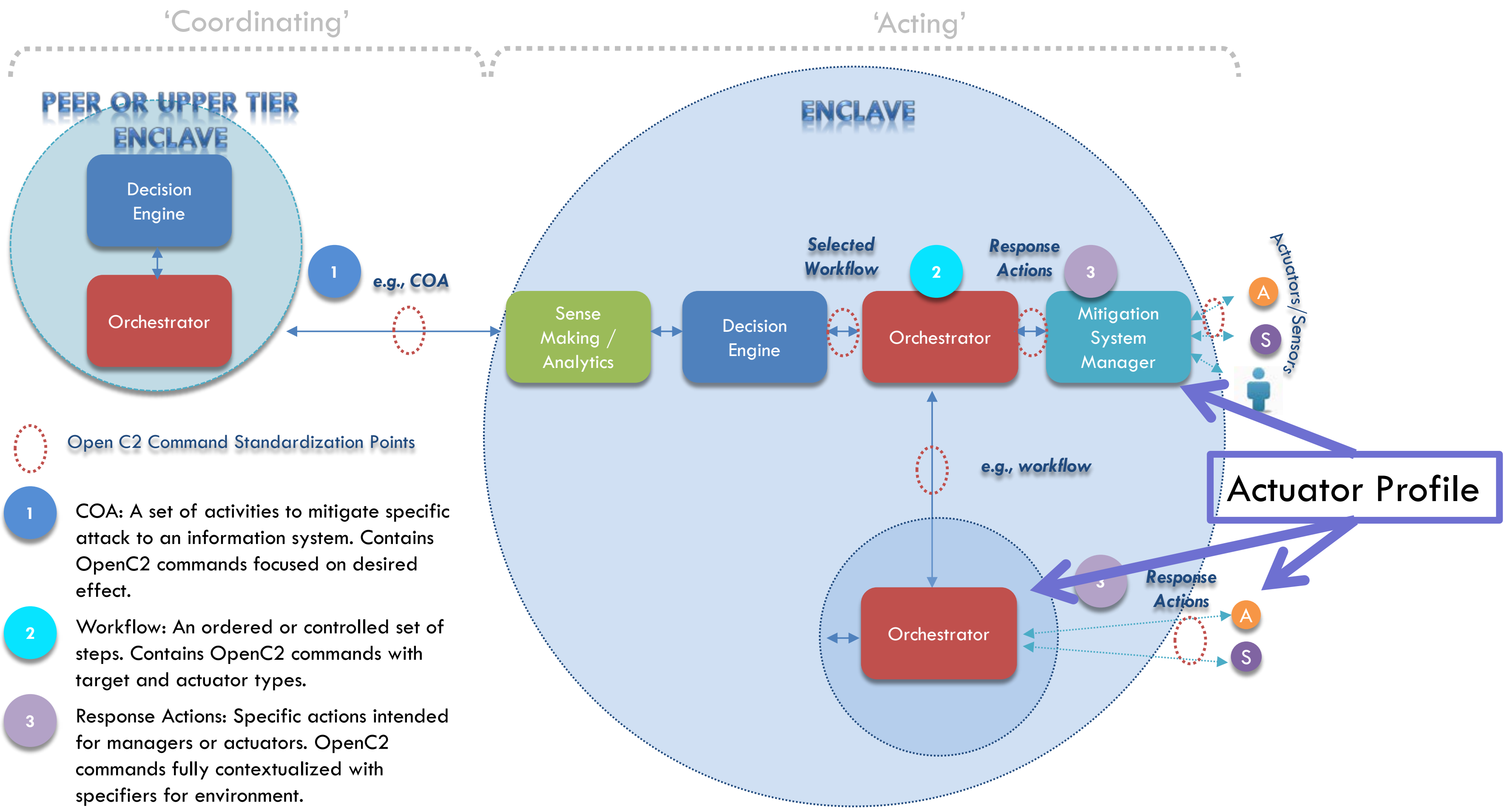
Agenda

- Review Agenda Topics
- STIX 2.X and OpenC2 as the COA field Status Update
- Actuator Profiles
  - Implementation Scope
  - Actuator Data Model
- Target Types
  - Target Specifiers
  - Namespaces
- Standard Business Topics
  - Path to standardization
  - Specifications (e.g., shape and numbering)
  - Issue review
  - STIX/OpenC2 specification document
- New Topic
  - Digital Policy Management and OpenC2
  - Message Fabric
  - Sharing Analytics
  - Authentication/Authorization

STIX 2.X and OpenC2

- WebEx occurred on Wednesday Nov. 30
  - OpenC2
    - Provided ‘Overview’
    - Proposed OpenC2 to populate STIX COA Field
  - STIX
    - Asked about OASIS membership
    - Will consider the use of OpenC2 as MTI for the COA field
- OpenC2 to provide briefing at the STIX Face to Face

Who Implements an Actuator Profile?



Open C2 Command Standardization Points

1 COA: A set of activities to mitigate specific attack to an information system. Contains OpenC2 commands focused on desired effect.

2 Workflow: An ordered or controlled set of steps. Contains OpenC2 commands with target and actuator types.

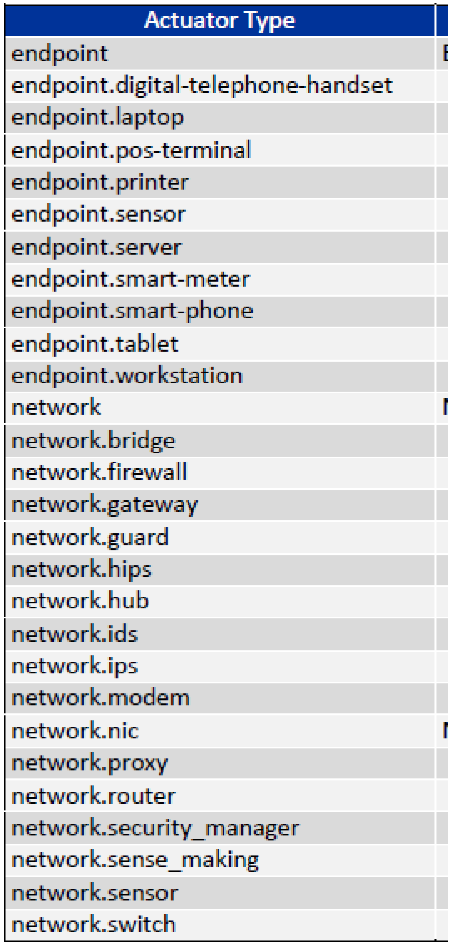
3 Response Actions: Specific actions intended for managers or actuators. OpenC2 commands fully contextualized with specifiers for environment.

Actuator Profile: Firewall

- Document Organization
  - Define scope of profile
  - Define MTI actions
  - Define Actuator specific Modifiers and specifiers
  - Provide JSON encoded sample commands for each action
- Appendix
  - Architecture Diagram
  - Fully implemented use case (encoded in JSON)
- Defines namespace
  - Actuator, and unique specifiers & modifiers
  - Define Response and Alert

Actuator Data Model

- □ Set of Actuator Profiles becomes the Actuator Data Model
- □ Actuator is defined by the supported Actions and Targets and Response behavior
- □ Actuator-Specifiers are defined in Actuator Profiles



Actuator Type
endpoint
endpoint.digital-telephone-handset
endpoint.laptop
endpoint.pos-terminal
endpoint.printer
endpoint.sensor
endpoint.server
endpoint.smart-meter
endpoint.smart-phone
endpoint.tablet
endpoint.workstation
network
network.bridge
network.firewall
network.gateway
network.guard
network.hips
network.hub
network.ids
network.ips
network.modem
network.nic
network.proxy
network.router
network.security_manager
network.sense_making
network.sensor
network.switch

⋮

Actuator not provided

- Inter-enclave exchange
  - Actuator not provided due to lack of knowledge
- Broad execution desired
  - Actuator not provided because multiple actuator types could execute the action

- LDD defines the base, common, behavior
- Actuator Profiles define actuator-specific behavior
- LDD → Actuator Profiles: Hierarchical relationship

Target Specifiers

Embedded Target Specifier Object

```
{
   "action": "deny",
   "target": {
      "type": "network-traffic",
      "network-traffic": {
         "src_ref": {
            "ipv4_addr": {
               "value": "198.51.100.1/32",
            }

         },
         "src-port": 443
      }
   }
}
```

Flattened Target Specifier Fields

```
{
   "action": "deny",
   "target": {
      "type": "network-traffic",
      "src_ref:ipv4_addr:value": "198.51.100.1/32",
      "src-port": 443
   }
}
```

STIX Pattern Grammar

```
{
   "action": "deny",
   "target": {
      "type": "network-traffic",
      "pattern": "[src_ref:ipv4-addr:value = '198.51.100.1/32' AND src-port = 443]"
   }

}
```

Target Specifier Notes

- All three versions represent the same target
- STIX Pattern Grammar:
  - Provides more complex specifiers
    - RegEx-like
    - `ALONGWITH`, `OTHERWISE`, `FOLLOWEDBY`
    - `REPEATS`, `WITHIN` x `SECONDS`
  - Overhead requirements
    - ANTLR (or equivalent) compiler and runtime library
    - Pattern expression evaluation engine
- Indicators/Observations use complex expressions
- Conclusion
  - Should OpenC2 preclude the use of ‘pattern’ grammar?
  - ‘Pattern’ should not be MTI
    - Use case for complex expressions in Commands?
    - Overhead burden on Actuator

Target Namespaces

- cybox: (CybOX 2.1 XML-derived JSON)
  - Used in initial OpenC2 design
  - Deprecate when replacement available
- stix: (STIX 2 native JSON)
  - Need to update LDD content and examples
- openc2:
  - Namespace largely derived from STIX 2
  - Augmented with ‘homegrown’ terms
  - Defined in OpenC2 spec