Open Command and Control (OpenC2) Language Description Document

Version 1.0 – Release Candidate 3

17 February 2017

FOREWORD

The Open Command and Control Forum (OpenC2 or the Forum) supports the cyber defense community of interest by developing and promoting the adoption of the OpenC2 language, data models, prototype implementations, and reference material that addresses the command and control of cyber defense components, technologies, and systems.

This Forum serves developers, users, and the entire cybersecurity ecosystem by providing a set of shared resources to expand the use of standardized command and control for cyber defense activities, to enable technology vendors building orchestration and cyber response technologies, and to assist developers in producing response technologies that can be readily used in coordinated responses. The goal of the Forum is to provide and open and collaborative environment and to present its findings and artifacts to recognized standards bodies for the standardization of the command and control language.

This document represents the outcome of collaboration between technology vendors, government agencies, and academia on the topic of command and control for cyber defensive measures. We gratefully acknowledge their contributions to the definition of the OpenC2 language. As we exercise the language in reference implementations and in real-world operations, we expect to continue to refine the language to ensure its suitability to support machine-to-machine command and control communications in response to cyber threats in cyber-relevant time.

Visit openc2.org for other on-line resources.

TABLE OF CONTENTS

1. Introduction

Cyberattacks are increasingly more sophisticated, less expensive to execute, dynamic, and automated. Current cyber defense products are typically integrated in a unique or proprietary manner and statically configured. As a result, upgrading or otherwise modifying tightly integrated, proprietary cyber defense's functional blocks is resource intensive; cannot be realized within a cyber-relevant timeframe; and the upgrades may degrade the overall performance of the system.

Future cyber defenses against current and pending attacks require the integration of new or upgraded functional capabilities, the coordination of responses across domains, synchronization of response mechanisms, and deployment of automated actions in cyber relevant time.

Standardization of the languages, including lexicons, syntaxes, and encodings, used within the interfaces and protocols necessary for machine-to-machine command and control communications in cyber relevant time will enable cyber defense system flexibility, interoperability, and responsiveness in cyber relevant time.

1.1 Purpose

The purpose of the Open Command and Control (OpenC2) Language Description Document is to define a lexicon language and semantics at a level of abstraction that will enable the coordination and execution of command and control of cyber defense components between and within networks. It is expected that the OpenC2 language will define profiles (i.e.., applicable commands, applicable values) by community groups for specific cyber defense functions such as Software Defined Networking, Firewall, routing.

1.2 Scope

The scope of this document is to create a lexicon of actions and define the semantics, syntax and other aspects of a language that will couple an action with the target of the actions, and the entities that execute the actions. The document also defines an extensible syntax to accommodate attributes that further specify the targets, and modify actions to support a wide range of operational environments.

Other aspects of OpenC2, such as implementation considerations, further refinement of the lexicon to accommodate specific cyber defense functions, encoding of commands for machine to machine communications and reference implementations will be addressed in other artifacts. These other efforts will be consistent with this language description.

The definition of a language such as OpenC2 is necessary but insufficient to enable future cyber defenses. OpenC2 commands can be carried within any number of constructs (e.g., STIX, workflows, playbooks, API's). In addition, OpenC2 is designed to be flexible, agnostic of external protocols that provide services such as transport, authentication, key management and other services. Cyber defense implementations must consider and will require other protocols and security services.

1.3 Intended Audience

This OpenC2 Language Description Document is intended for organizations investigating the implementation of automated pre-approved cyber defensive measures as well as academia and industry partners involved with the development and integration of security orchestration, network components or services, endpoint security applications, and security services for cyber defenses.

1.4 Document Overview

Section 1, Introduction, describes the impetus for the OpenC2 language and lays out the purpose, scope, and intended audience of the document.

Section 2, Background, describes the design principles for the language and how the language can be contextualized for different operating environments.

Section 3, OpenC2 Language, describes the abstract syntax and the basic building blocks of the language. It also further specifies the vocabulary for actions, universal modifiers, action specific modifiers and a default namespace for targets and target specifiers..

Section 4, Example OpenC2 Usage, provides examples of OpenC2 command constructs. For each action, the supported targets, actuators, and action specific modifiers are identified and example usages are provided.

Section 5, Example OpenC2 Use Case, depicts an example use case for mitigating an evil domain. The use case shows the OpenC2 commands that could be used to mitigate the attacks or vulnerabilities and where they could be applied.

Appendix A, Example OpenC2 Commands, contains example OpenC2 commands organized in tables by OpenC2 action. These example commands were based on use cases provided by government agencies, critical infrastructure, industry (e.g., security orchestrator, actuator, and sensor) and academia.

2. Background

2.1 Design Principles

OpenC2 can be implemented in a variety of systems to perform the secure delivery and management of command and control messages in a context-specific way. OpenC2 commands are vendor neutral and message fabric agnostic, thus can be incorporated in different architectures and environments (such as connection oriented, connectionless, pub-sub, hub and spoke, etc.).

OpenC2 was designed to have a concise set of commands are extensible in order to provide context specific details. Conciseness ensures minimal overhead to meet possible latency and overhead constraints while extensions enable greater utility and flexibility.

There is an underlying assumption that issuing OpenC2 commands are event-driven and that an action is warranted. OpenC2 was designed to focus on the actions that are to be executed in order to thwart an attack, mitigate some vulnerability or otherwise address a threat. The exchange of indicators, rationale for the decision to act and/or threat information sharing are beyond the scope of OpenC2 and left to other standards such as STIX, TAXII etc.

The actual performance and efficacy of OpenC2 will be implementation-specific and will require the incorporation of other technologies. The OpenC2 design principles include the following:

- Support cyber relevant response time for coordination and response actions.
- Be infrastructure, architecture, and vendor agnostic.
- Support multiple levels of abstraction, necessary to permit the contextualization of commands for a wide variety of operating environments.
- Permit commands to be invoked that are either tasking/response actions or notifications.
    - Tasking/response actions result in a state change.
    - Notifications require supporting analytics/decision processes.

- Provide an extensible syntax to accommodate different types of actions, targets, and actuators (e.g., sensor, endpoint, network device, human) at varying levels of specificity. .
- Ensure the OpenC2 command is independent of a message construct that provides transport, identifies priority/ quality of service, and supports security attributes.

By design, OpenC2 is dependent upon but agnostic of the transport infrastructure and message fabric. Confidentiality, integrity, availability and authentication must be identified and provisioned by the message fabric.

Traditional command and control implementations utilize complete, self-standing constructs. OpenC2 decouples the actions from the targets of the actions and from the recipients of the commands. An OpenC2 command is not complete until an action is paired with a target, providing the command context for the action. This enables the OpenC2 language to be more concise, yet still support the entire C2 space. This characteristic of OpenC2 also permits a more flexible and extensible approach to accommodate future technologies and varying network environments.

2.2 OpenC2 and Deployment Environments

OpenC2 is defined at a level of abstraction such that an inter-domain tasking or coordination effort can be described without requiring in depth knowledge of the recipient network's components, but through the use of specifiers and modifiers, enough detail can be appended to carry out specific tasks on particular devices to support intra-domain command and control.

This level of abstraction permits end to end applicability of OpenC2. As depicted in Figure 2-1, an OpenC2 command is sent to enable coordination or send a high level tasking from the peer or upper tier enclave. An OpenC2 command received by an enclave will trigger events within the enclave to annotate the command with context specific information so that specific devices within the enclave can respond appropriately. This allows the enclave to take advantage of this context-specific knowledge to interpret and appropriately execute OpenC2 commands .

Each network contextualizes an OpenC2 action for the specific sensors and actuators within its environment so it can further specify the command to reflect the

implementations of which it is capable. Context-specific modifiers provide an ability to further specify the action while enabling the set of actions to remain tightly constrained. This minimizes the overhead, permits further contextualization of the OpenC2 commands for specific environments, and thereby enables flexibility and extensibility.

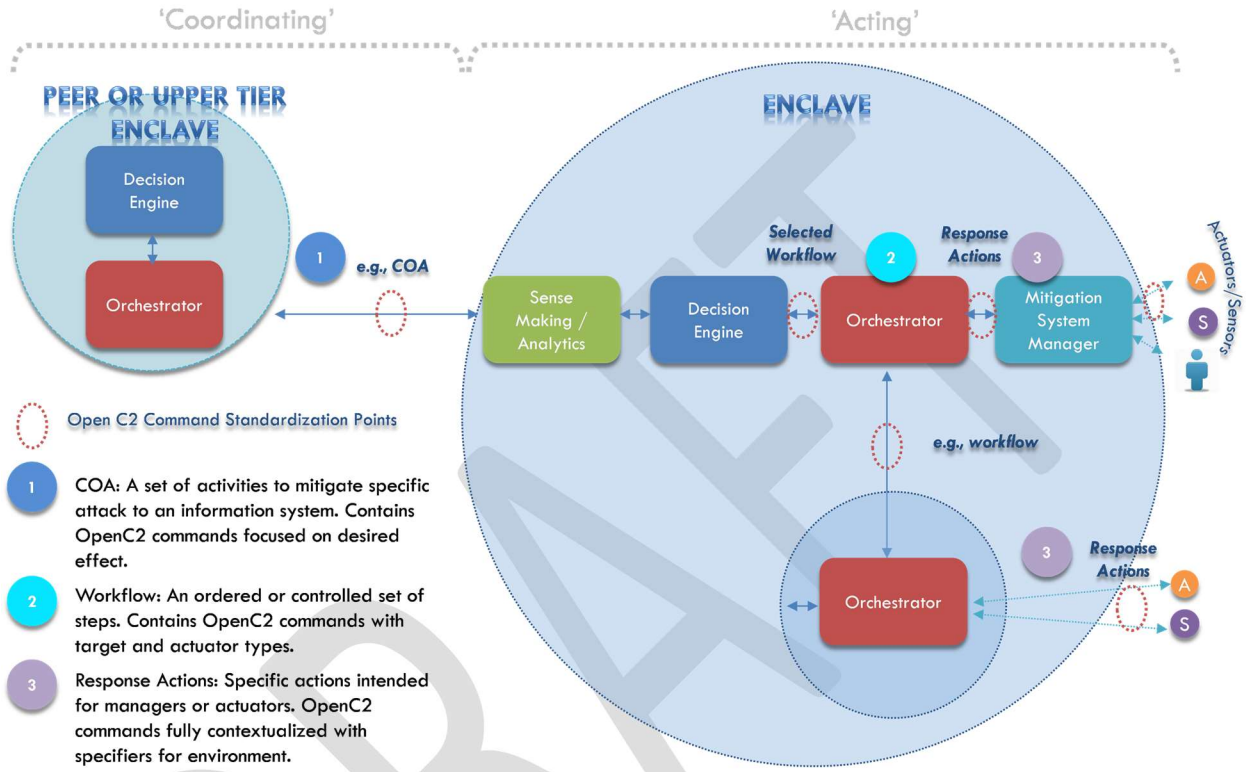

Figure 2-1. OpenC2 Deployment Environments

For example, an organization may have executed a series of actions to protect against a particular attack that was signaled by an external indicator (such as a STIX message). In order to elicit a consistent response across an organization (whether hierarchical or peer to peer), a complex course of action can be constructed and shared. The use of standardized OpenC2 commands will be more precise and more quickly actionable than a set of recommended steps within a text document (e.g., flash), which must be parsed, analyzed, and interpreted, prior to execution. Standardizing OpenC2 commands helps to ensure a more uniform response at enterprises/enclaves that reflects enterprise-wide level decisions.

3. OpenC2 Language

3.1 Overview

The OpenC2 language is designed at a level of abstraction high enough such that it enables persistence as technologies advance and is implementation agnostic, but enough precision so that the need for specifiers and modifiers is limited.

3.2 Abstract Syntax

Conceptually, an OpenC2 command has the following form:

```
(
        ACTION = <ACTION_TYPE>,
        TARGET (
                type = <data-model>:<TARGET_TYPE>,
                <target-specifier>
        ),
        ACTUATOR (
                type = <data-model>:<ACTUATOR_TYPE>,
                <actuator-specifier>
        ),
        MODIFIERS (
                <list-of-modifiers>
        )
)
```

Fields denoted with angle brackets ("<>") are replaced with the appropriate details. Some of the fields are considered optional. The table below describes these fields semantically and whether they are required, optional or ignored in certain situations. Actual encoding will leverage pre-existing conventions and notations such as XML, JSON, TLV or others. .

The following table contains the description of the fields that can be contained in an OpenC2 command.

Table 3-1. OpenC2 Command Field Descriptions

Field	Description
ACTION	Required. The task or activity to be performed (i.e., the 'verb').
data-model	Required. The data model for the TARGET.
TARGET	Required. The object of the action. The ACTION is performed on the TARGET.
type	Required. The TARGET type will be defined within the context of a namespace.
target-specifier	Optional. The specifier further describes a specific target, a list of targets, or a class of targets.
ACTUATOR	Optional. The subject of the action. The ACTUATOR executes the ACTION on the TARGET.
type	Required if the actuator is included, otherwise not applicable. The ACTUATOR type will be defined within the context of a namespace.
data-model	Required if the actuator is included, otherwise not applicable. The data model for the ACTUATOR.
actuator-specifier	Optional if the actuator is included, otherwise not applicable. The specifier further describes a specific actuator, a list of actuators, or a class of actuators.
MODIFIERS (<list-of-modifiers>)	Optional. Provide additional information about the action such as date/time, periodicity, duration, and location.

There are cases where an ACTION and TARGET are sufficient to complete the command, especially in the case of inter-domain commands where the method or approach to complete or execute the action can be determined within the receiving domain/enclave.

The majority of commands within an enclave will have an ACTION, TARGET and ACTUATOR. Inclusion of the ACTUATOR provides additional context for the command as a whole and enables precision. .

Specifiers for TARGETs and ACTUATORs are optional and can be used to provide context specific information that could be used to reflect the local environment, policies, and operational conditions within an enterprise/enclave. Specifiers can call out a specific target/actuator, a list of targets/actuators, or a class of targets/actuators.

Modifiers to the ACTION are optional and are used to provide effect based context to the ACTION. Modifiers are further discussed in Section 3.2.5.

Table 3-2 illustrates the use of specifiers and modifiers to extend the range of OpenC2 commands to cover the higher level 'strategic' commands to the unambiguous enclave-specific use case. This provides greater flexibility to the language and allows the OpenC2 actions to be further contextualized for the mission environment. The table below provides some examples of the different levels of specificity achievable in an OpenC2 command.

Table 3-2. OpenC2 Syntax Flexibility Examples

Description	Action	Target	Actuator	Modifier
		Target-Specifier	Actuator-Specifier	
Block traffic to/from specific IP address(es) [effects-based, no actuator specified]; suitable for inter-domain coordination	DENY	Network Connection		
		Source and/or Destination IP Address(es)		
Block traffic at all network devices	DENY	Network Connection	Network (any devices)	

Description	Action	Target	Actuator	Modifier
		Target-Specifier	Actuator-Specifier	
[specify actuator class]; suitable for inter-domain coordination or as a command to an orchestration engine which further contextualizes to the enclave's environment		Source and/or Destination IP Address(es)		
Block traffic at network routers [specify type of network device actuator]; suitable within an enclave	DENY	Network Connection	Network.router	
		Source and/or Destination IP Address	(optional)	
Block traffic at specific network router; [specify identity of network router]; suitable within an enclave	DENY	Network Connection	Network.router	
		Source and/or Destination IP Address	Router identity	
Block access to bad external IP by null routing; [specify method of performing action]; suitable within an enclave	DENY	Network Connection	Network.router	Method= blackhole
		Source and/or Destination IP Address	(optional)	

3.2.1 Action

All OpenC2 commands start with an ACTION which indicates the type of command to perform such as gather and convey information, control activities and devices, and control permissions and access. The range of options and potential impact on the information

system associated with a particular ACTION is a function of the ACTUATOR. For cases that involve multiple options for an ACTION, modifiers may be used.

Refer to Section 3.3 for the list of ACTIONs and their definitions and usage.

3.2.2 Target

All OpenC2 commands include a TARGET. The TARGET is the object of the ACTION (or alternatively, the ACTION is performed on the TARGET). Targets include objects such as network connections, URLs, hashes, IP addresses, files, processes, fully qualified domain names etc. .

3.2.3 Actuator

An ACTUATOR[1] is the entity that puts command and control into motion or action. The ACTUATOR is the subject of the ACTION which performs the ACTION on the TARGET. There are varying levels of abstraction and functionality for an ACTUATOR ranging from a specific sensor to an entire system or even system of systems.

The source of a command may need to communicate an action that must be taken against a target, but will not necessarily have knowledge of the cyber defense technologies deployed in other enclaves so the inclusion of an actuator is optional within an OpenC2 command. As a command is propagated through the system and context specific information is gained, the command can appended with an actuator and appropriate specifiers.

There will be only one ACTUATOR type per OpenC2 command. The actuator namespace is specified in the OpenC2 profiles. .

[1] Some academic circles model all cyber defense components as sensors and/or actuators. It is acknowledged that OpenC2 will be used for C2 of sensors as well, but in the interest of being concise within this document, actuators encompass sensors.

3.2.4 Specifiers

"Specifiers" are used to identify specific individual or groups of targets or actuators. Table 3-3 illustrates how the commands are appended with specifiers as context specific details become available. The actuator specifiers presented in Table 3-3 are for illustrative purposes. The actual specifiers are defined in the appropriate actuator profiles.

Table 3-3. Example Usage of Specifiers

Description	Action	Target	Actuator	Modifier
		Target-Specifier	Actuator-Specifier	
Block malicious URL	DENY	URI/URL		
		Value Condition = Equals		
Quarantine Artifact with particular byte string	QUARANTINE	Artifact		
		Condition = Contains		
Block access to external IP address by null routing at specific network routers	DENY	Network Connection	Network router	
		Condition = Contains	Manufacturer, Model, Serial Number Value = 123	

3.2.5 Modifiers

"Modifiers" provide additional precision about the action such as time, periodicity, duration, or other details on what is to be done. Modifiers can denote the when, where, and how aspects of an action. The modifier can also be used to convey the need for acknowledgement or additional status information about the execution of an action. Modifiers are similar to specifiers in that they can provide additional context specific

details, and are intended to provide additional details for action/actuator pairs. A modifier may be “actuator-specific”, “action-specific”, or “universal” depending on the applicability of the modifier within the language.

Actuator-specific modifiers are described in Actuator Profiles. Action-specific are described in Section 4. Universal modifiers are described in the following table.

Table 3-4. Example Usage of Modifiers

Description	Action	Target	Actuator	Modifier
		Target-Specifier	Actuator-Specifier	
Shutdown a system, immediate	STOP	Device	endpoint	method = immediate
		Device Object Type	(optional)	
Start Process with Delay	START	Process	endpoint	Delay = duration
		Process Object Type	(optional)	
Quarantine a device	CONTAIN	Device	network	where (network segment, vlan)
		Device Object Type	(optional)	
Block access to suspicious external IP address by redirecting external DNS queries to an internal DNS server	DENY	Network Connection	DNS Server	method = sinkhole
		Network Connection Object Type		

3.3 Actions

This section defines the set of OpenC2 actions grouped by their general activity. The following table summarizes the definition of the OpenC2 actions. Subsequent sections will identify the appropriate targets for each action and the appropriate actuators for the action target pair. Further details will be defined in the actuator profiles. .

- Actions that Control Information:
  These actions are used to gather information needed to determine the current state or enhance cyber situational awareness. These actions typically do not impact the state of the target and are normally not detectable by external observers.

- Actions that Control Permissions:
  These actions are used to control permissions and manage accesses.

- Actions that Control Activities/Devices:
  These actions are used to control the state or the activity of a system, a process, a connection, a host, or a device (e.g., endpoint, sensor, actuator). The actions are used to execute tasks, adjust configurations, set and update parameters, and modify attributes.

- Sensor-Related Actions:
  These actions are used to control the activities of a sensor in terms of how to collect and provide the sensor data.

- Effects-Based Actions:
  Effects-based actions are at a higher level of abstraction for purposes of communicating a desired impact rather than a command to execute specific tasks within an enclave. This level of abstraction enables coordinated actions between enclaves, while permitting a local enclave to optimize its workflow for its specific environment.

  Implementation of an effects-based action requires that the recipient enclave has a decision making capability because an effects-based action permits multiple possible responses.

- Response and Alert:
  RESPONSE is used to provide data requested as a result of an action. The RESPONSE message will contain the requested data and have a reference to the action that

initiated the response. ALERT is used to signal the occurrence of an event or error. It is an unsolicited message that does not reference a previously issued action.

Table 3-5. Summary of Action Definitions

Actions that Control 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 is used to find an object either physically, logically, functionally, or by organization. This action enables one to tell where in the system an event or trigger occurred.
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.
NOTIFY	The NOTIFY action is used to set an entity's alerting preferences.
Actions that Control Permissions	
DENY	The DENY action is used to prevent a certain event or action from completion, such as preventing a flow from reaching a destination (e.g., block) or preventing access.
CONTAIN	The CONTAIN action stipulates the isolation of a file or process or entity such that it cannot modify or access assets or processes that support the business and/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 process, application, system or some other 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.
CANCEL	The CANCEL action invalidates a previously issued action.
SET	The SET action changes a value, configuration, or state of a managed entity within an IT system.
UPDATE	The UPDATE action instructs the component to retrieve and process a 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.
THROTTLE	The THROTTLE action adjusts the throughput of a data flow.
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 Actions	
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 Actions	
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. Mitigate implies that the impacts to the enclave's operations should be minimized while addressing the issue.
REMEDIATE	The REMEDIATE action tasks the recipient enclave to eliminate the vulnerability or attack point. Remediate implies that addressing the issue is paramount.
Response and Alert	
RESPONSE	RESPONSE is used to provide any data requested as a result of an action. RESPONSE can be used to signal the acknowledgement of an action, provide the status of an action along with additional information related to the requested action, or signal the completion of the action. The recipient of the RESPONSE can be the original requester of the action or to another recipient(s) designated in the modifier of the action.
ALERT	ALERT is used to signal the occurrence of an event.

3.4 Target Vocabulary

The TARGET is the object of the ACTION (or alternatively, the ACTION is performed on the TARGET). OpenC2 defines a default TARGET Data Model to support all of the actions. It is derived largely on the STIX Cyber Observables v2.x.

In addition to the default TARGET Data Model, the OpenC2 syntax can support any other data model. To differentiate alternative data models, a data model prefix is used to qualify the target type. The default target data model will prefix "openc2:" to the target type. The implementer will need to supply a unique data model prefix for non-standard target types. It is the responsibility of the implementer to ensure that there are no namespace collisions

when using alternative data models. Refer to the following table for a summary of the OpenC2 TARGET Data Models.

Table 3-6. Target Data Model

Type	Description	Options
data-model	Used to uniquely identify a set of target types so there is no ambiguity; defines the context in which target types are defined.	Choice of: ● openc2 ● <external-ref>

Targets include objects such as network connections, URLs, hashes, IP addresses, files, processes, and domains. Refer to the following table for a summary of the supported OpenC2 TARGETs in the default TARGET Data Model.

Table 3-7. Summary of Supported Targets

Target Type	Description	Target Specifier
openc2:artifact	The Artifact Object permits capturing an array of bytes (8-bits), as a base64-encoded string or linking to a file-like payload.	mime_type : string, payload_bin : binary, url : string, hashes : hashes-type
openc2:command	The Command Object represents and OpenC2 command.	id : command-ref
openc2:device	The Device Object	description: string,

Target Type	Description	Target Specifier
	represents the properties of a hardware device.	device_type: string, manufacturer: string, model : string, serial_number : string, firmware_version : string
openc2:directory	The Directory Object represents the properties common to a file system directory.	path : string, path_enc : string, created : timestamp, modified : timestamp, accessed : timestamp, contains_refs : list of type object-ref
openc2:disk	The Disk Object represents a disk drive.	disk_name : string, disk_size : integer, free_space : integer, partition_list : list of type disk-partition type : string
openc2:disk-partition	The Disk Partition Object represents a single partition of a disk drive.	created : timestamp, device_name : string, mount_point : string, partition_id : string, partition_length : integer, partition_offset : integer, space_left : integer, space_used : integer, total_space : integer, type : string
openc2:domain-name	The Domain Name represents the properties of a network domain name.	value : string, resolves_to_refs : list of type object-ref
openc2:email-addr	The Email Address	value : string,

Target Type	Description	Target Specifier
	Object represents a single email address.	display_name : string, belongs_to_ref : object-ref
openc2:email-message	The Email Message Object represents an instance of an email message, corresponding to the internet message format described in RFC 5322 and related RFCs.	is_multipart : boolean, date : timestamp, content_type : string, from_ref : object-ref, sender_ref : object-ref, to_refs : list of type object-ref, cc_refs : list of type object-ref, bcc_refs : list of type object-ref, subject : string, received_lines : list of type string, additional_header_fields : dictionary, body : string, body_multipart : list of type mime-part-type, raw_email_ref : object-ref
openc2:file	The File Object represents the properties of a file.	extensions : dictionary, hashes : hashes-type, size : integer, name : string, name_enc : string, magic_number_hex : hex, mime_type : string, created : timestamp, modified : timestamp, accessed : timestamp, parent_directory_ref : object-ref, is_encrypted : boolean, encryption_algorithm : open-vocab, decryption_key : string, contains_refs : list of type object-ref, content_ref: object-ref

Target Type	Description	Target Specifier
openc2:ipv4-addr	The IPv4 Address Object represents one or more IPv4 addresses expressed using CIDR notation.	value : string, resolves_to_refs : list of type object-ref, belongs_to_refs : list of type object-ref
openc2:ipv6-addr	The IPv6 Address Object represents one or more IPv6 addresses expressed using CIDR notation.	value : string, resolves_to_refs : list of type object-ref, belongs_to_refs : list of type object-ref
openc2:mac-addr	The MAC Address Object represents a single Media Access Control (MAC) address.	value : string
openc2:memory	The Memory Object represents memory objects.	hashes : list of type string, name : string, memory_source : string, region_size : integer, block_type : string, region_start_address : string, region_end_address : string, extracted_features : string
openc2:network-traffic	The Network Traffic Object represents arbitrary network traffic that originates from a source and is	extensions : dictionary, start : timestamp, end : timestamp, is_active : boolean, src_ref : object-ref, dst_ref : object-ref, src_port : integer,

Target Type	Description	Target Specifier
	addressed to a destination.	dst_port : integer, protocols : list of type string, src_byte_count : integer, dst_byte_count : integer, src_packets : integer, dst_packets : integer, ipfix : dictionary, src_payload_ref : object-ref, dst_payload_ref : object-ref, encapsulates_refs : list of type object-ref, encapsulated_by_ref : object-ref
openc2:openc2	The OpenC2 Object is a subset of the Artifact Object that represents an Actuator's OpenC2 supported capabilities.	value : string, attributes : list of type string, search : string
openc2:process	The Process Object represents common properties of an instance of a computer program as executed on an operating system.	extensions : dictionary, is_hidden : boolean, pid : integer, name : string, created : timestamp, cwd : string, arguments : list of type string, encironment_variables : dictionary, opened_connection_refs : list of type object-ref, creator_user_ref : object-ref, binary_ref : object-ref, parent_ref : object-ref, child_refs : list of type object-ref

Target Type	Description	Target Specifier
openc2:software	The Software Object represents high-level properties associated with software, including software products.	name : string, cpe : string, language : string, vendor : string, version : string
openc2:url	The URL Object represents the properties of a uniform resource locator (URL).	value : string
openc2:user-account	The User Account Object represents an instance of any type of user account, including but not limited to operating system, device, messaging service, and social media platform accounts.	extensions : dictionary, user_id : string, account_login : string, account_type : open-vocab, display_name : string, is_service_account : boolean, is_privileged : boolean, can_escalate_privs : boolean, is_disabled : boolean, account_created : timestamp, account_expires : timestamp, password_last_changed : timestamp, account_first_login : timestamp, account_last_login : timestamp
openc2:user-session	The User Session Object represents a user session.	effective_group : string, effective_group_id : string, effective_user : string, effective_user_id : string, login_time : timestamp, logout_time : timestamp

Target Type	Description	Target Specifier
openc2:volume	The Volume Object represents a generic drive volume.	name : string, device_path : string, file_system_type : string, total_allocation_units : integer, sectors_per_allocation_unit : integer, bytes_per_sector : integer, actual_available_allocation_units : integer, creation_time : timestamp, file_system_flag_list : list of type string, serial_number : string
openc2:windows-registry-key	The Registry Key Object represents the properties of a Windows registry key.	key : string, values : list of type windows-registry-value-type, modified : timestamp, creator_user_ref : object-ref, number_of_subkeys : integer
openc2:x509-certificate	The X509 Certificate Object represents the properties of an X.509 certificate, as defined by ITU recommendation X.509.	is_self_signed : boolean, hashes : hashes-type, version : string, serial_number : string, signature_algorithm : string, issuer : string, validity_not_before : timestamp, validity_not_after : timestamp, subject : string, subject_public_key_algorithm : string, subject_public_key_modulus : string, subject_public_key_exponent : integer, x509_v3_extensions : x509-v3-extensions-type

3.5 Actuator Vocabulary

An ACTUATOR is the entity that puts command and control into motion or action. The ACTUATOR executes the ACTION on the TARGET. The ACTUATOR data model is defined in one or more *actuator profiles* where an actuator profile is a document that defines actions that are mandatory to implement, optional and the appropriate actuator specifiers and the actuator specific modifiers. The data model identifies which actuator profile is being referenced. The actuator profiles referenced in this document are for illustrative purposes.

In addition to the default ACTUATOR Data Model, the OpenC2 syntax can support any other data model. To differentiate alternative data models, a data model prefix is used to qualify the actuator type. The default actuator data model will prefix "openc2:" to the actuator type. The implementer will need to supply a unique data model prefix for non-standard actuator types. It is the responsibility of the implementer to ensure that there are no namespace collisions when using alternative data models. Refer to the following table for a summary of the OpenC2 ACTUATOR Data Models.

Table 3-8. Actuator Data Model

Type	Description	Options
data-model	Used to uniquely identify a set of actuator types so there is no ambiguity; defines the context in which target types are defined.	Choice of: ● openc2 ● <external-ref> ●

Table 3-9. List of Functional Actuators???

Actuator Type	Description
endpoint	Endpoint Device
endpoint-workstation	

Actuator Type	Description
endpoint-server	
network	Network Platform
network-firewall	
network-router	
network-proxy	
network-sensor	
network-hips	
network-sense-making	
process	Services/Processes
process-anti-virus-scanner	
process-aaa-service	
process-virtualization-service	
process-sandbox	
process-email-service	
process-directory-service	
process-remediation-service	
process-location-service	

3.6 Modifier Vocabulary

Modifiers provide additional information about the action such as time, periodicity, duration, and location. Modifiers can denote the when, where, and how aspects of an action. The modifier can also be used to convey the need for additional status information

about the execution of an action such as a response is required. The requested status/information will be carried in a RESPONSE. Refer to Section 4.6.

Modifiers are similar to specifiers in that they can provide additional context specific details for an action. Modifiers that are applicable to any action are referred to as 'universal modifiers' and are presented in table 3-10. Modifiers that are applicable to a particular action , regardless of the actuator are referred to as , 'Action-specific' and are are identified in the sections detailing out each action. Modifiers that are only applicable to an action for a particular actuator are referred to as 'Actuator Specific' and are defined within the actuator profiles.

The following table lists the set of universal modifiers that are applicable to all types of actions.

Table 3-10. Summary of Universal Modifiers

Modifier	Type	Description	Target Applicability
context	string	A reference that provides context for the action.	All
datetime	date-time (RFC 3339)	The specific date/time to initiate the action.	All
delay	duration (RFC 3339)	The time to wait before performing the action.	All
duration	duration (RFC 3339)	The period of time that an action is valid.	All
id	command_id	The unique identifier for the action.	All
response	ack, status	Indicate the type of response required for the action.	All
respond-to	string	The location where the	All

Modifier	Type	Description	Target Applicability
		response should be sent.	

4. EXAMPLE OpenC2 USAGE

This section provides examples of OpenC2 commands that correspond to each OpenC2 action and its applicable targets. This section also defines any action specific modifiers. The purpose of this section is to provide sample commands that are consistent with the syntax defined in this document and to illustrate the flexibility of the OpenC2 language. Additional examples are presented in in Appendix A.

4.1 Actions that Control Information

These actions are used to gather information needed to further determine courses of action or assess the effectiveness of courses of action. These actions can be used to support data enrichment use cases and maintain situational awareness. These actions typically do not impact the state of the target and are normally not detectable by external observers.

4.1.1 SCAN

The SCAN action is the systematic examination of some aspect of the entity or its environment in order to obtain information.

This action can be used to command the characterization of an environment (e.g., perform network, port, or vulnerability scanning) or to look for a specific occurrence of an object (e.g., file, IP, process). SCAN commands are distinct from the QUERY in that SCAN implies an analytic while a QUERY implies a routine retrieval of data.

Table. Supported Targets and Actuators: SCAN

Target Type	Actuator Type
openc2:device openc2:disk openc2:disk-partition openc2:domain-name openc2:email-message openc2:file openc2:ipv4-addr openc2:memory openc2:network-traffic openc2:process openc2:software openc2:url openc2:user-account openc2:user-session	network-sensor

Target Type	Actuator Type
openc2:volume	

Table. Modifiers: SCAN

Modifier	Type	Description	Target Applicability
method	enumeration: non-authenticated, authenticated	Optional. When there is more than one way to perform the action, the method can be specified, if necessary.	All
search	cve, patch, vendor bulletin, signature	Required. The search criteria for performing the scan.	All

Below is a sample of OpenC2 commands to perform a SCAN of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: SCAN

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Scan a device for vulnerabilities	scan	openc2:device (as required)	network-sensor (optional)	search = CVE

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
2	Scan email messages for malware	scan	openc2:email-message (as required)	network-sensor (optional)	search = malware signature
3	Scan network traffic for malicious activities	scan	openc2:network-traffic (as required)	network-sensor (optional)	search = network signature

4.1.2 LOCATE

The LOCATE action is used to find an object either physically, logically, functionally, or by organization. This action enables one to tell where in the system an event or trigger occurred.

This action is used for example to enable one to tell where in the system an event or trigger occurred, confirm that an asset is appropriately deployed, or ascertain details regarding a rogue device.

Table. Supported Targets and Actuators: LOCATE

Target Type	Actuator Type
openc2:device openc2:file openc2:ipv4-addr openc2:user-account	process-location-service

Table. Modifiers: LOCATE

Modifier	Type	Description	Target Applicability
None to Date			

Below is a sample of OpenC2 commands to perform a LOCATE of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: LOCATE

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Geolocate a device	locate	openc2:device (as required)	process-location-service (optional)	
2	Get location of an IP address	locate	openc2:ipv4-addr (as required)	process-location-service (optional)	

4.1.3 QUERY

The QUERY action initiates a single request for information.

QUERY, like SCAN, is used to find out more information about the system or its environment. In the case of QUERY, however, it is an isolated or specific information request, rather than a broadly scoped scan or on-going check. QUERY is used to retrieve data that is already present in a database or data store, while SCAN implies a more thorough examination and identification of anomalies (relative to a known good state). The response to a query is typically (but not necessarily) conveyed within the command and control channel.

The target for QUERY is usually openc2:artifact. The target-specifier describes the search criteria for the information request.

A special target for QUERY is openc2:openc2 which signifies a request for an actuator's OpenC2 capabilities (i.e., a list of supported actions, targets). If not target-specifier is included in the request then the full report of the actuator's capabilities should be provided. A response could be filtered for a particular capability by providing details in the target-specifier.

Table. Supported Targets and Actuators: QUERY

Target Type	Actuator Type
openc2:artifact openc2:openc2	endpoint network-firewall network-router process-directory-service

Table. Modifiers: QUERY

Modifier	Type	Description	Target Applicability
response		Where and how to direct the response to the query.	All

Below is a sample of OpenC2 commands to perform a QUERY of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: QUERY

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	List all network connections	query	openc2:artifact (as required)	network-router (optional)	response
2	List running processes on a machine	query	openc2:artifact (as required)	endpoint (optional)	response
3	Request an Actuator's supported OpenC2 capabilities	query	openc2:openc2 (as required)	network-firewall (optional)	response

4.1.4 REPORT

The REPORT action tasks an entity to provide information to a designated recipient of the information.

The REPORT action is used to request an actuator/sensor to provide certain information. Along with the REPORT action and the type of information being requested, the recipient of the information must be specified in the command. The response to a REPORT action is typically (but not necessarily) conveyed outside of the command and control channel.

Table. Supported Targets and Actuators: REPORT

Target Type	Actuator Type
openc2:artifact	

Table. Modifiers: REPORT

Modifier	Type	Description	Target Applicability
frequency	duration (RFC 3339)	Optional. The frequency at which to perform the action. The value is the requested time between execution events.	All
report-to	openc2:ipv4-addr, openc2:ipv6-addr	Required. This modifier identifies where to send the report.	All

Below is a sample of OpenC2 commands to perform a REPORT of targets, utilizing actuators at different levels of specificity,

qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: REPORT

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Produce and send a report	report	openc2:artifact (as required)		report-to

4.1.5 NOTIFY

The NOTIFY action is used to direct an entity to send information to another entity.

NOTIFY is distinct from REPORT in that NOTIFY is used for time sensitive event notification and carries a sense of persistence.

Table. Supported Targets and Actuators: NOTIFY

Target Type	Actuator Type
openc2:process openc2:user-account	endpoint-server process-email-service

Table. Modifiers: NOTIFY

Modifier	Type	Description	Target Applicability
frequency	duration (RFC 3339)	Optional. The frequency at which to perform the action. The value is the requested time between execution events.	All
message		The intended message to notify the target.	All

Below is a sample of OpenC2 commands to perform a NOTIFY of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: NOTIFY

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Notify security officer to report compliance with change of configuration	notify	openc2:user-account (as required)	process-email-service (optional)	message
2	Send a command to notify an external enclave	notify	openc2:process (as required)		message = acknowledge

4.2 Actions that Control Permissions

These actions are used to control permissions and accesses.

4.2.1 DENY

The DENY action is used to prevent a certain event or action from completion, such as preventing a flow from reaching a destination (e.g., block) or preventing access.

DENY is a superset of current terms such as BLOCK (network perimeter devices) and DENY (user, access to system, access to files).

Table. Supported Targets and Actuators: DENY

Target Type	Actuator Type
openc2:device	endpoint
openc2:network-traffic	network-firewall
openc2:process	network-proxy
openc2:software	network-router
openc2:url	process
openc2:user-account	process-aaa-service

Table. Modifiers: DENY

Modifier	Type	Description	Target Applicability
method	enumeration: acl, blackhole, sinkhole, blacklist, whitelist	Optional. When there is more than one way to perform the action, the method can be specified, if necessary.	openc2:network-traffic, openc2:product
where	enumeration: internal, perimeter	Optional. The general location within the enclave to perform the DENY action.	openc2:network-traffic

Below is a sample of OpenC2 commands to perform a DENY of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: DENY

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Block traffic to/from specific IP address; suitable for coordinating across multiple enclaves and allowing enclaves to determine most appropriate response	deny	openc2:network-traffic (as required)		
2	Block traffic to/from specific IP address at all network firewalls	deny	openc2:network-traffic (as required)	network-firewall (optional)	

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
3	Block traffic at the network routers	deny	openc2:network-traffic (as required)	network-router (optional)	
4	Block network traffic inside the enclave	deny	openc2:network-traffic (as required)		where = internal
5	Block network traffic at the perimeter	deny	openc2:network-traffic (as required)		where = perimeter
6	Block network traffic by ACL	deny	openc2:network-traffic (as required)	network-router (optional)	method = acl
7	Block access to a bad external IP address by null routing at the network routers.	deny	openc2:network-traffic (as required)	network-router (optional)	method = blackhole

4.2.2 CONTAIN

The CONTAIN action stipulates the isolation of a file or process or entity such that it cannot modify or access assets or processes that support the business and/or operations of the enclave.

The CONTAIN action is a superset of currently used terms such as ISOLATE, QUARANTINE or SANDBOX.

Table. Supported Targets and Actuators: CONTAIN

Target Type	Actuator Type
openc2:device openc2:file openc2:network-traffic openc2:process openc2:user-account	endpoint network

Table. Modifiers: CONTAIN

Modifier	Type	Description	Target Applicability
where		Optional. The general location within the enclave to contain the target.	openc2:device, openc2:file, openc2:network-traffic, openc2:process, openc2:user-account

Below is a sample of OpenC2 commands to perform a CONTAIN of targets, utilizing actuators at different levels of specificity,

qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: CONTAIN

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Quarantine a file, general	contain	openc2:file (as required)		
2	Quarantine a file	contain	openc2:file (as required)	endpoint (optional)	where
3	Contain a user or group, general	contain	openc2:user-account (as required)		
4	Contain network traffic to a honeynet, general	contain	openc2:network-traffic (as required)		

4.2.3 ALLOW

The ALLOW action permits the access to or execution of a target.

An ALLOW action is typically associated with something that was previously denied (e.g., deny, contain).

Table. Supported Targets and Actuators: ALLOW

Target Type	Actuator Type
openc2:device openc2:file openc2:network-traffic openc2:process openc2:software openc2:url openc2:user-account	endpoint network network-firewall network-proxy network-router process process-aaa-service

Table. Modifiers: ALLOW

Modifier	Type	Description	Target Applicability
delay	duration (RFC 3339)	Optional. The time to wait before performing the action.	openc2:device, openc2:user-account
permissions		Optional. Specific permissions to be granted to the user.	openc2:user-account

Modifier	Type	Description	Target Applicability
where	enumeration: internal, perimeter	Optional. The general location within the enclave to perform the DENY action.	openc2:network-traffic

Below is a sample of OpenC2 commands to perform a ALLOW of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: ALLOW

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Unblock traffic to/from specific IP address; suitable for coordinating across multiple enclaves and allowing enclaves to determine most appropriate response	allow	openc2:network-traffic (as required)		
2	Unblock traffic to/from specific IP address at all network firewalls	allow	openc2:network-traffic (as required)	network-firewall (optional)	
3	Unblock traffic at the network routers	allow	openc2:network-traffic (as required)	network-router (optional)	

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
4	Unblock network traffic inside the enclave	allow	openc2:network-traffic (as required)		where = internal
5	Delay Machine Authentication	allow	openc2:device (as required)	process-aaa-server (optional)	delay = <duration>
6	Unquarantine a file	allow	openc2:file (as required)	endpoint (optional)	

4.3 Actions that Control Activities/Devices

These actions are used to execute some task, adjust configurations, set and update parameters etc. These actions typically change the state of the system.

4.3.1 START

The START action initiates a process, application, system or some other activity.

Table. Supported Targets and Actuators: START

Target Type	Actuator Type
openc2:device openc2:disk-partition openc2:process openc2:software	endpoint network process-virtualization-service

Table. Modifiers: START

Modifier	Type	Description	Target Applicability
delay	duration (RFC 3339)	Optional. The time to wait before performing the action.	All
method	enumeration: spawn		openc2:process

Below is a sample of OpenC2 commands to perform a START of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: START

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Start Process, general	start	openc2:process (as required)		
2	Start Process	start	openc2:process (as required)	endpoint (optional)	
3	Start Process with Delay	start	openc2:process (as required)	endpoint (optional)	delay
4	Spawn Process	start	openc2:process (as required)	endpoint (optional)	method = spawn

4.3.2 STOP

The STOP action halts a system or ends an activity.

The STOP OpenC2 action is used to convey terms in current use such as shutdown, kill, and terminate. The STOP action has nuances and options associated with it that are ACTUATOR specific. In the case where more than one type of STOP action is applicable for a particular target and actuator, if practical, the default implementation of STOP should be a graceful shutdown. Action modifiers are used to indicate immediate or atypical STOP actions.

Table. Supported Targets and Actuators: STOP

Target Type	Actuator Type
openc2:device openc2:disk-partition openc2:process openc2:user-account openc2:user-session	endpoint network process-aaa-service process-virtualization-service

Table. Modifiers: STOP

Modifier	Type	Description	Target Applicability
method	enumeration: graceful, immediate	Optional. When there is more than one way to perform the action, the method can be specified, if necessary.	All

Below is a sample of OpenC2 commands to perform a STOP of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: STOP

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Shutdown a system	stop	openc2:device (as required)	endpoint (optional)	[method = graceful]
2	Shutdown a system, immediate	stop	openc2:device (as required)	endpoint (optional)	method = immediate
3	Logoff User: Logoff all the sessions of a particular user from the machine	stop	openc2:user-account (as required)	endpoint (optional)	[method = graceful]
4	Stop a vm	stop	openc2:process (as required)	process-virtualization-service (optional)	[method = graceful]

4.3.3 RESTART

The RESTART action conducts a STOP of a system or an activity followed by a START of a system or an activity.

A RESTART implies a graceful shutdown, maintenance of state, and a new configuration.

Table. Supported Targets and Actuators: RESTART

Target Type	Actuator Type
openc2:device openc2:process	endpoint process-virtualization-service

Table. Modifiers: RESTART

Modifier	Type	Description	Target Applicability
delay	duration (RFC 3339)	Optional. The time to wait before performing the action.	All
frequency	duration (RFC 3339)	Optional. The frequency at which to perform the action. The value is the requested time between execution events.	All
options		Additional options that specify how to restart	All

Below is a sample of OpenC2 commands to perform a RESTART of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: RESTART

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Restart device (system)	restart	openc2:device (as required)		
2	Restart device (system) with different OS	restart	openc2:device (as required)		options, e.g., OS
3	Restart VM	restart	openc2:process (as required)	process-virtualization-service (optional)	

4.3.4 PAUSE

The PAUSE action ceases a system or activity while maintaining state.

A PAUSE remains in effect until a RESUME is issued, unless the PAUSE action is accompanied by modifier for a time-interval.

Table. Supported Targets and Actuators: PAUSE

Target Type	Actuator Type
openc2:device openc2:process	endpoint process-virtualization-service

Table. Modifiers: PAUSE

Modifier	Type	Description	Target Applicability
duration	duration (RFC 3339)	Optional. The time to wait until returning to the previous state.	All
method	enumeration: sleep, hibernate, suspend	Optional. When there is more than one way to perform the action, the method can be specified, if necessary.	All

Below is a sample of OpenC2 commands to perform a PAUSE of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: PAUSE

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Pause device (system)	pause	openc2:device (as required)		[method = sleep]
2	Hibernate device (system)	pause	openc2:device (as required)		method = hibernate
3	Pause VM	pause	openc2:process (as required)	process-virtualization-service (optional)	
4	Pause a system or VM for a specified duration	pause	openc2:process (as required)		duration = <DURATION>

4.3.5 RESUME

The RESUME action starts a system or activity from a paused state.

RESUME is only meaningful after a PAUSE command.

Table. Supported Targets and Actuators: RESUME

Target Type	Actuator Type
openc2:device openc2:process	endpoint process-virtualization-service

Table. Modifiers: RESUME

Modifier	Type	Description	Target Applicability
None to Date			

Below is a sample of OpenC2 commands to perform a RESUME of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: RESUME

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Resume device (system)	resume	openc2:device (as required)		
2	Resume VM	resume	openc2:process (as required)	process-virtualization-service (optional)	

4.3.6 CANCEL

The CANCEL action invalidates a previously issued action.

CANCEL must be associated with a previously issued command through the "command-ref" modifier. This action is intended to stop an action that has not initiated or completed and is not intended to undo a completed action and return to a previous state. It can set the validity period to immediately end or it could define a future duration for which the action is valid.

Table. Supported Targets and Actuators: CANCEL

Target Type	Actuator Type
openc2:command	endpoint network process

Table. Modifiers: CANCEL

Modifier	Type	Description	Target Applicability
command-ref	command_id	The reference to the associated command that is to be cancelled.	openc2:Command
duration	duration (RFC 3339)	Optional. The period of time that an action is valid. If not present, the CANCEL operation should occur immediately.	openc2:Command

Below is a sample of OpenC2 commands to perform a CANCEL of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: CANCEL

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Cancel a previously issued command	cancel	openc2:command (as required)		command-ref = command reference
2	Cancel a previously issued command, directed to a specific actuator (endpoint)	cancel	openc2:command (as required)	endpoint (optional)	command-ref = command reference

4.3.7 SET

The SET action changes a value, configuration, or state of a managed entity within an IT system.

Typically this action is specified by a configuration item such as a sensor setting or privilege level and the command will have specifiers. SET commands are intended for specific individual changes to the entity and the parameters are communicated in the C2 channel.

Table. Supported Targets and Actuators: SET

Target Type	Actuator Type
openc2:artifact openc2:file openc2:process openc2:user-account openc2:windows-registry-key	endpoint-workstation network-firewall network-hips network-router network-sensor process-directory-service

Table. Modifiers: SET

Modifier	Type	Description	Target Applicability
set-to		The value to set the target to.	All

Below is a sample of OpenC2 commands to perform a SET of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: SET

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Set registry key value	set	openc2:windows-registry-key (as required)	endpoint-workstation (optional)	set-to
2	Set file permissions	set	openc2:file (as required)	process-directory-service (optional)	set-to
3	Set user rights	set	openc2:user-account (as required)	process-directory-service (optional)	set-to

4.3.8 UPDATE

The UPDATE action instructs the component to retrieve, install, process, and operate in accordance with a software update, reconfiguration, or some other update.

The settings, files, patches associated with an UPDATE action are typically retrieved out of band from the control channel. It is incumbent upon the OpenC2 compliant devices to include implementation details such as save, reboot, restart.

Table. Supported Targets and Actuators: UPDATE

Target Type	Actuator Type
openc2:artifact openc2:file openc2:software openc2:windows-registry-key	endpoint network-sensor process-anti-virus-scanner

Table. Modifiers: UPDATE

Modifier	Type	Description	Target Applicability
frequency	duration (RFC 3339)	Optional. The frequency at which to perform the action. The value is the requested time between execution events.	All
source		The source of the updated information.	All

Below is a sample of OpenC2 commands to perform a UPDATE of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: UPDATE

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Install software	update	openc2:software (as required)	endpoint (optional)	
2	Install patch	update	openc2:software (as required)	endpoint (optional)	
3	Update signature file (anti-virus)	update	openc2:artifact (as required)	process-anti-virus-scanner (optional)	

4.3.9 MOVE

The MOVE action changes the location of a file, subnet, network, or, process.

MOVE is distinct from CONTAIN in that CONTAIN implies a desired effect of isolation and MOVE supports the more general case.

Table. Supported Targets and Actuators: MOVE

Target Type	Actuator Type
openc2:artifact openc2:file	

Table. Modifiers: MOVE

Modifier	Type	Description	Target Applicability
move-to	location	The location to move to	All

Below is a sample of OpenC2 commands to perform a MOVE of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: MOVE

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Move file/directory	move	openc2:file (as required)		move-to

4.3.10 REDIRECT

The REDIRECT action changes the flow of traffic to a particular destination other than its original intended destination.

The REDIRECT action includes the case of bypassing an intermediate point. REDIRECT is distinct from MOVE in that it encompasses the entire flow rather than a single instance, item or object. MOVE supports the more atomic case.

Table. Supported Targets and Actuators: REDIRECT

Target Type	Actuator Type
openc2:network-traffic openc2:url	network-router

Table. Modifiers: REDIRECT

Modifier	Type	Description	Target Applicability
where		Optional. The location within the enclave to redirect the target. "where = null" will cancel previous redirection actions.	All

Below is a sample of OpenC2 commands to perform a REDIRECT of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: REDIRECT

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Redirect traffic to a honeypot; suitable for coordinating across multiple enclaves and allowing enclaves to determine most appropriate response	redirect	openc2:network-traffic (as required)		where
2	Redirect traffic to a honeypot at a specific router	redirect	openc2:network-traffic (as required)	network-router	where
3	Cancel traffic redirection; suitable for coordinating across multiple enclaves and allowing enclaves to determine most appropriate response	redirect	openc2:network-traffic (as required)		where = null

4.3.11 DELETE

The DELETE action removes data and files.

Table. Supported Targets and Actuators: DELETE

Target Type	Actuator Type
openc2:artifact openc2:email-message openc2:file	endpoint network-firewall process-email-service

Table. Modifiers: DELETE

Modifier	Type	Description	Target Applicability
None to Date			

Below is a sample of OpenC2 commands to perform a DELETE of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: DELETE

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Delete file, inter-enclave	delete	openc2:file (as required)		
2	Delete file, within an enclave	delete	openc2:file (as required)	endpoint (optional)	
3	Delete email, inter-enclave	delete	openc2:email-message (as required)		
4	Delete email from exchange server	delete	openc2:email-message (as required)	process-email-service (optional)	

4.3.12 SNAPSHOT

The SNAPSHOT action records and stores the state of a target at an instant in time.

Table. Supported Targets and Actuators: SNAPSHOT

Target Type	Actuator Type
openc2:process	process-virtualization-service

Table. Modifiers: SNAPSHOT

Modifier	Type	Description	Target Applicability
None to Date			

Below is a sample of OpenC2 commands to perform a SNAPSHOT of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: SNAPSHOT

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Take a snapshot of a VM	snapshot	openc2:process	process-virtualization-service	

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
			(as required)	(optional)	

4.3.13 DETONATE

The DETONATE action executes and observes the behavior of a target (e.g., file, hyperlink) in a manner that is isolated from assets that support the business or operations of the enclave.

DETONATE is distinct from CONTAIN in that DETONATE includes an execution and analytic component rather than just isolation.

Table. Supported Targets and Actuators: DETONATE

Target Type	Actuator Type
openc2:file openc2:url	process-sandbox

Table. Modifiers: DETONATE

Modifier	Type	Description	Target Applicability
None to Date			

Below is a sample of OpenC2 commands to perform a DETONATE of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: DETONATE

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Acting sends the URL to be analyzed in a sandbox.	detonate	openc2:url (as required)	process-sandbox (optional)	
2	Acting sends the file to the Sandbox for detonation analysis.	detonate	openc2:file (as required)	process-sandbox (optional)	

4.3.14 RESTORE

The RESTORE action deletes and/or replaces files, settings, or attributes to return the system to an identical or similar known state.

The RESTORE could impact the whole system or return the state of an application or program to its previous state.

Table. Supported Targets and Actuators: RESTORE

Target Type	Actuator Type
openc2:device	process-remediation-service

Table. Modifiers: RESTORE

Modifier	Type	Description	Target Applicability
restore-point		Required. The specific restore point to restore to.	All

Below is a sample of OpenC2 commands to perform a RESTORE of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: RESTORE

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Restore a device to a known restore point.	restore	openc2:device (as required)	process-remediation-service (optional)	restore-point

4.3.15 SAVE

The SAVE action commits data or system state to memory.

Table. Supported Targets and Actuators: SAVE

Target Type	Actuator Type
openc2:email-message openc2:file openc2:network-traffic	endpoint network-router process-email-service

Table. Modifiers: SAVE

Modifier	Type	Description	Target Applicability
save-to	location	The location to save to.	All

Below is a sample of OpenC2 commands to perform a SAVE of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: SAVE

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Save data	save	openc2:file (as required)	endpoint (optional)	save-to
2	Save an email message	save	openc2:email-message (as required)	process-email-service (optional)	save-to
3	Save a raw network packet	save	openc2:network-traffic (as required)	network-router (optional)	save-to

4.3.16 THROTTLE

The THROTTLE action adjusts the throughput of a data flow.

Table. Supported Targets and Actuators: THROTTLE

Target Type	Actuator Type
openc2:network-traffic	network-router

Table. Modifiers: THROTTLE

Modifier	Type	Description	Target Applicability
None to Date			

Below is a sample of OpenC2 commands to perform a THROTTLE of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: THROTTLE

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Limit bandwidth	throttle	openc2:network-traffic	network-router	

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
			(as required)	(optional)	

4.3.17 DELAY

The DELAY action stops or holds up an activity or data transmittal.

The period of time for the delay can be specified in a modifier to the DELAY action.

Table. Supported Targets and Actuators: DELAY

Target Type	Actuator Type
openc2:network-traffic	

Table. Modifiers: DELAY

Modifier	Type	Description	Target Applicability
delay	duration (RFC 3339)	Required. The time delay to add to a network connection.	All

Below is a sample of OpenC2 commands to perform a DELAY of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: DELAY

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Delay all traffic	delay	openc2:network-traffic (as required)		delay

4.3.18 SUBSTITUTE

The SUBSTITUTE action replaces all or part of the data, content or payload in the least detectable manner.

SUBSTITUTE is used in cases where an attack is to be impeded or thwarted in an undetectable manner.

Table. Supported Targets and Actuators: SUBSTITUTE

Target Type	Actuator Type
openc2:file openc2:network-traffic	endpoint network-router

Table. Modifiers: SUBSTITUTE

Modifier	Type	Description	Target Applicability
options		Additional options that specify what to replace and replace with what.	All

Below is a sample of OpenC2 commands to perform a SUBSTITUTE of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: SUBSTITUTE

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Overwrite data	substitute	openc2:file (as required)	endpoint (optional)	options
2	Substitute traffic	substitute	openc2:network-traffic (as required)	network-router (optional)	options

4.3.19 COPY

The COPY action duplicates a file or data flow.

Table. Supported Targets and Actuators: COPY

Target Type	Actuator Type
openc2:disk-partition openc2:file openc2:memory openc2:network-traffic	

Table. Modifiers: COPY

Modifier	Type	Description	Target Applicability
copy-to	location	The location to copy to.	All

Below is a sample of OpenC2 commands to perform a COPY of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: COPY

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Copy a file	copy	openc2:file (as required)		copy-to
2	Copy network traffic	copy	openc2:network-traffic (as required)		copy-to

4.3.20 SYNC

The SYNC action synchronizes a sensor or actuator with other system components.

Table. Supported Targets and Actuators: SYNC

Target Type	Actuator Type
openc2:device	endpoint

Table. Modifiers: SYNC

Modifier	Type	Description	Target Applicability
frequency	duration (RFC 3339)	Optional. The frequency at which to perform the action. The value is the requested time between execution events.	All

Below is a sample of OpenC2 commands to perform a SYNC of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: SYNC

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Synchronize an endpoint sensor or actuator to another device	sync	openc2:device (as required)	endpoint (optional)	

4.4 Sensor-Related Actions

These actions are used to control the activities of a sensor in terms of how to collect and provide the sensor data.

4.4.1 DISTILL

The DISTILL action tasks the sensor to send a summary or abstraction of the sensing information instead of the raw data feed.

The DISTILL action reduces the amount of sensor data. The means of reduction or filtering is indicated by a specifier.

Table. Supported Targets and Actuators: DISTILL

Target Type	Actuator Type
openc2:network-traffic	network-sensor

Table. Modifiers: DISTILL

Modifier	Type	Description	Target Applicability
None to Date			

Below is a sample of OpenC2 commands to perform a DISTILL of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: DISTILL

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Filter	distill	openc2:network-traffic (as required)	network-sensor	

4.4.2 AUGMENT

The AUGMENT action tasks the sensor to do a level of preprocessing or sense making prior to sending the sensor data.

The means of augmentation and the source of additional data are indicated by a specifier.

Table. Supported Targets and Actuators: AUGMENT

Target Type	Actuator Type
openc2:network-traffic	network-sensor

Table. Modifiers: AUGMENT

Modifier	Type	Description	Target Applicability
method	enumeration	The specific augmentation function to perform on the network traffic.	openc2:network-traffic

Below is a sample of OpenC2 commands to perform a AUGMENT of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: AUGMENT

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Preprocess network traffic, inter-enclave	augment	openc2:network-traffic (as required)		method
2	Preprocess network traffic, within an enclave	augment	openc2:network-traffic (as required)	network-sensor (optional)	method

4.5 Effects-Based Actions

Effects-based actions are at a higher level of abstraction and focus on the desired impact rather than a command to execute specific tasks within an enclave. These actions enable the coordination actions, while permitting a local enclave to execute actions in accordance with its local policies and/or capabilities. .

Implementation of an effects-based action requires that the recipient enclave has a decision making capability because an effects-based action permits multiple possible responses.

4.5.1 INVESTIGATE

The INVESTIGATE action tasks the recipient enclave to aggregate and report information as it pertains to an anomaly.

Examples of actions resulting from a received INVESTIGATE OpenC2 command could include scan multiple machines, quarantine an endpoint, or detonate a file. These actions are determined by the enclave based on the results of sense-making/analytics and decision-making based on operational constraints and mission needs.

Table. Supported Targets and Actuators: INVESTIGATE

Target Type	Actuator Type
openc2:device openc2:domain-name openc2:email-message openc2:file openc2:ipv4-addr	

Target Type	Actuator Type
openc2:network-traffic openc2:process openc2:software openc2:x509-certificate	

Table. Modifiers: INVESTIGATE

Modifier	Type	Description	Target Applicability
report-to	openc2:ipv4-addr, openc2:ipv6-addr	Optional. If requested, this modifier identifies where to report the results of the investigation.	All

Below is a sample of OpenC2 commands to perform a INVESTIGATE of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: INVESTIGATE

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Investigate the specified IP address for malicious activities	investigate	openc2:ipv4-addr		[report-to]

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
			(as required)		
2	Investigate the specified device	investigate	openc2:device (as required)		[report-to]
3	Investigate the specified domain	investigate	openc2:domain-name (as required)		[report-to]
4	Investigate the specified email message	investigate	openc2:email-message (as required)		[report-to]
5	Investigate the specified file(s)	investigate	openc2:file (as required)		[report-to]
6	Investigate the specified hostname	investigate	openc2:domain-name (as required)		[report-to]

4.5.2 MITIGATE

The MITIGATE action tasks the recipient enclave to circumvent the problem without necessarily eliminating the vulnerability or attack point.
Mitigate implies that the impacts to the enclave's operations should be minimized while addressing the issue.

Examples of actions resulting from a received MITIGATE OpenC2 command could include deny a URL or process, scan, redirect traffic to honeypot, or move.

Table. Supported Targets and Actuators: MITIGATE

Target Type	Actuator Type
openc2:device openc2:domain-name openc2:email-message openc2:file openc2:ipv4-addr openc2:network-traffic openc2:process openc2:software openc2:x509-certificate	

Table. Modifiers: MITIGATE

Modifier	Type	Description	Target Applicability
None to Date			

Below is a sample of OpenC2 commands to perform a MITIGATE of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: MITIGATE

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Mitigate the specified malicious IP address	mitigate	openc2:ipv4-addr (as required)		[report-to]
2	Mitigate the specified infected device	mitigate	openc2:device (as required)		[report-to]
3	Mitigate the specified malicious email message	mitigate	openc2:email-message (as required)		[report-to]

4.5.3 REMEDIATE

The REMEDIATE action tasks the recipient enclave to eliminate the vulnerability or attack point.
Remediate implies that addressing the issue is paramount.

Examples of actions resulting from a received REMEDIATE OpenC2 command could include contain/quarantine to a VLAN, set authorizations, redirect URL to quarantine portal, get new configuration, or update patches.

Table. Supported Targets and Actuators: REMEDIATE

Target Type	Actuator Type
openc2:device openc2:domain-name openc2:email-message openc2:file openc2:ipv4-addr openc2:network-traffic openc2:process openc2:software openc2:x509-certificate	

Table. Modifiers: REMEDIATE

Modifier	Type	Description	Target Applicability
None to Date			

Below is a sample of OpenC2 commands to perform a REMEDIATE of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: REMEDIATE

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Remediate the specified malicious email message	remediate	openc2:email-message (as required)		[report-to]
2	Remediate the specified infected hostname	remediate	openc2:domain-name (as required)		[report-to]

4.6 Response and Alert

RESPONSE is used to provide data requested as a result of an action. The RESPONSE message will contain the requested data and have a reference to the action that initiated the response. ALERT is used to signal the occurrence of an event or error. It is an unsolicited message that does not reference a previously issued action.

4.6.1 RESPONSE

RESPONSE is used to provide any data requested as a result of an action. RESPONSE can be used to signal the acknowledgement of an action, provide the status of an action along with additional information related to the requested action, or signal the completion of the action. The recipient of the RESPONSE can be the original requester of the action or to another recipient(s) designated in the modifier of the action.

The RESPONSE action accepts the following modifiers:

Table. Modifiers: RESPONSE

Modifier	Type	Description	Target Applicability
command-ref	command_id	The reference to the associated command that is in response to.	N/A
type	enumeration: acknowledgement, status, query	The type of response.	N/A

Modifier	Type	Description	Target Applicability
value		The value of the response.	N/A

Below is a sample of OpenC2 commands to perform a RESPONSE of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: RESPONSE

	Description	Action	Modifier
1	Acknowledge the receipt of an action	RESPONSE	type = acknowledge, command-ref = command reference
2	Signal completion of an action	RESPONSE	type = status, value = complete, command-ref = command reference
3	Provide the status of an action	RESPONSE	type = status, value = current, command-ref = command reference

4.6.2 ALERT

ALERT is used to signal the occurrence of an event.

Table. Modifiers: ALERT

Modifier	Type	Description	Target Applicability
type	enumeration	The type of alert.	N/A
value		Additional data associated with the alert.	N/A

Below is a sample of OpenC2 commands to perform a ALERT of targets, utilizing actuators at different levels of specificity, qualified by modifiers to the action as appropriate.

Table. Sample of OpenC2 Commands: ALERT

	Description	Action	Modifier
1	An actuator sends an alert as the result of some condition.	ALERT	type, value
2	A sensor sends an alert as the result of some condition.	ALERT	type, value

5. Example OpenC2 Use Case

TBSL

Appendix A. Example OpenC2 Commands

A.1 ALERT

Table. Example OpenC2 Commands: ALERT

	Description	Action	Modifier
1	An actuator sends an alert as the result of some condition.	ALERT	type, value
2	A sensor sends an alert as the result of some condition.	ALERT	type, value

A.2 ALLOW

Table. Example OpenC2 Commands: ALLOW

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Unblock traffic to/from specific IP address; suitable for coordinating across multiple enclaves and allowing enclaves	allow	openc2:network-traffic (as required)		

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
	to determine most appropriate response				
2	Unblock traffic to/from specific IP address at all network firewalls	allow	openc2:network-traffic (as required)	network-firewall (optional)	
3	Unblock traffic at the network routers	allow	openc2:network-traffic (as required)	network-router (optional)	
4	Unblock network traffic inside the enclave	allow	openc2:network-traffic (as required)		where = internal
5	Delay Machine Authentication	allow	openc2:device (as required)	process-aaa-server (optional)	delay = <duration>
6	Unquarantine a file	allow	openc2:file (as required)	endpoint (optional)	
7	Unblock network traffic at the perimeter	allow	openc2:network-traffic (as required)		where = perimeter

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
8	Unblock network traffic at the perimeter routers	allow	openc2:network-traffic (as required)	network-router (specify perimeter routers)	
9	Unblock traffic to/from specific IP address at all endpoints' firewalls	allow	openc2:network-traffic (as required)	process (specify endpoint and firewall application)	
10	Unblock URL (blacklist domain); suitable for coordinating across multiple enclaves and allowing enclaves to determine most appropriate response	allow	openc2:url (as required)		
11	Unblock URL at proxy server	allow	openc2:url (as required)	network-proxy (optional)	
12	Unblock URL at all network firewalls	allow	openc2:url (as required)	network-firewall (optional)	
13	Unblock URL at all endpoint	allow	openc2:url	process	

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
	firewalls		(as required)	(specify endpoint and firewall application)	
14	Unblock URL at all endpoint browsers	allow	openc2:url (as required)	process (optional)	
15	Unblock system application; suitable for coordinating across multiple enclaves and allowing enclaves to determine most appropriate response	allow	openc2:software (as required)		
16	Unblock system application from executing at endpoint with certain characteristics or specific endpoint(s)	allow	openc2:software (as required)	endpoint (specify based on endpoint characteristics)	
17	Authenticate Machine	allow	openc2:device (as required)	process-aaa-server (optional)	
18	Unblock Process	allow	openc2:process	endpoint	

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
			(as required)	(optional)	
19	Unblock Process by Domain	allow	openc2:process (as required)	endpoint (optional)	
20	Authenticate user; suitable for coordinating across multiple enclaves	allow	openc2:user-account (as required)		
21	Delay user authentication; suitable for coordinating across multiple enclaves	allow	openc2:user-account (as required)		delay = <duration>
22	Grant User Access to Specific System	allow	openc2:user-account (as required)	process-aaa-server (optional)	permissions
23	Unquarantine a file, general	allow	openc2:file (as required)		
24	Release a process from isolation, general	allow	openc2:process		

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
			(as required)		
25	Release a process from isolation	allow	openc2:process (as required)	endpoint (optional)	
26	Unquarantine a device, general	allow	openc2:device (as required)		
27	Unquarantine a device	allow	openc2:device (as required)	network (optional)	

A.3 AUGMENT

Table. Example OpenC2 Commands: AUGMENT

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Preprocess network traffic, inter-enclave	augment	openc2:network-traffic (as required)		method
2	Preprocess network traffic, within an enclave	augment	openc2:network-traffic (as required)	network-sensor (optional)	method

A.4 CANCEL

Table. Example OpenC2 Commands: CANCEL

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Cancel a previously issued command	cancel	openc2:command (as required)		command-ref = command reference
2	Cancel a previously issued command, directed to a specific	cancel	openc2:command	endpoint	command-ref = command

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
	actuator (endpoint)		(as required)	(optional)	reference
3	Cancel a previously issued command, directed to a specific actuator (network)	cancel	openc2:command (as required)	network (optional)	command-ref = command reference
4	Cancel a previously issued command, directed to a specific actuator (process)	cancel	openc2:command (as required)	process (optional)	command-ref = command reference

A.5 CONTAIN

Table. Example OpenC2 Commands: CONTAIN

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Quarantine a file, general	contain	openc2:file (as required)		

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
2	Quarantine a file	contain	openc2:file (as required)	endpoint (optional)	where
3	Contain a user or group, general	contain	openc2:user-account (as required)		
4	Contain network traffic to a honeynet, general	contain	openc2:network-traffic (as required)		
5	Isolate a process, general	contain	openc2:process (as required)		
6	Isolate a process	contain	openc2:process (as required)	endpoint (optional)	where
7	Quarantine a device, general	contain	openc2:device (as required)		
8	Quarantine a device	contain	openc2:device	network	where (network

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
			(as required)	(optional)	segment, vlan)
9	Contain a user or group	contain	openc2:user-account (as required)	network (optional)	where
10	Contain network traffic to a honeynet	contain	openc2:network-traffic (as required)	network (optional)	where

A.6 COPY

Table. Example OpenC2 Commands: COPY

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Copy a file	copy	openc2:file (as required)		copy-to

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
2	Copy network traffic	copy	openc2:network-traffic (as required)		copy-to
3	Copy netflow information related to particular ip address	copy	openc2:network-traffic (as required)		copy-to
4	Copy the full contents of a disk partition	copy	openc2:disk-partition (as required)		copy-to
5	Copy the full contents of a system's memory	copy	openc2:memory (as required)		copy-to

A.7 DELAY

Table. Example OpenC2 Commands: DELAY

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Delay all traffic	delay	openc2:network-traffic (as required)		delay

A.8 DELETE

Table. Example OpenC2 Commands: DELETE

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Delete file, inter-enclave	delete	openc2:file (as required)		
2	Delete file, within an enclave	delete	openc2:file (as required)	endpoint (optional)	
3	Delete email, inter-enclave	delete	openc2:email-message		

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
			(as required)		
4	Delete email from exchange server	delete	openc2:email-message (as required)	process-email-service (optional)	
5	Delete firewall rule	delete	openc2:artifact (as required)	network-firewall (optional)	
6	Delete srp	delete	openc2:artifact (as required)		

A.9 DENY

Table. Example OpenC2 Commands: DENY

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Block traffic to/from specific IP address; suitable for coordinating across multiple enclaves and allowing enclaves to determine most appropriate response	deny	openc2:network-traffic (as required)		
2	Block traffic to/from specific IP address at all network firewalls	deny	openc2:network-traffic (as required)	network-firewall (optional)	
3	Block traffic at the network routers	deny	openc2:network-traffic (as required)	network-router (optional)	
4	Block network traffic inside the enclave	deny	openc2:network-traffic (as required)		where = internal
5	Block network traffic at the perimeter	deny	openc2:network-traffic (as required)		where = perimeter
6	Block network traffic by ACL	deny	openc2:network-traffic	network-router	method = acl

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
			(as required)	(optional)	
7	Block access to a bad external IP address by null routing at the network routers.	deny	openc2:network-traffic (as required)	network-router (optional)	method = blackhole
8	Block access to/from suspicious internal IP address by null routing at the network routers	deny	openc2:network-traffic (as required)	network-router (optional)	method = blackhole
9	Block network traffic at the perimeter routers	deny	openc2:network-traffic (as required)	network-router (specify perimeter routers)	
10	Block access to suspicious external IP address by redirecting external DNS queries to an internal DNS server	deny	openc2:network-traffic (as required)		method = sinkhole
11	Block traffic to/from specific IP address at all endpoints' firewalls	deny	openc2:network-traffic (as required)	process (specify endpoint and firewall application)	

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
12	Block malicious URL (blacklist domain); suitable for coordinating across multiple enclaves and allowing enclaves to determine most appropriate response	deny	openc2:url (as required)		
13	Block malicious URL at proxy server	deny	openc2:url (as required)	network-proxy (optional)	
14	Block malicious URL at all network firewalls	deny	openc2:url (as required)	network-firewall (optional)	
15	Block malicious URL at all endpoint firewalls	deny	openc2:url (as required)	process (specify endpoint and firewall application)	
16	Block malicious URL at all endpoint browsers	deny	openc2:url (as required)	process (optional)	
17	Block system application;	deny	openc2:software		

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
	suitable for coordinating across multiple enclaves and allowing enclaves to determine most appropriate response		(as required)		
18	Block system application from executing at endpoint with certain characteristics or specific endpoint(s)	deny	openc2:software (as required)	endpoint (specify based on endpoint characteristics)	
19	Block system application from executing by application white listing	deny	openc2:software (as required)	endpoint (optional)	method = whitelist
20	Deny Device Access (Infected Host)	deny	openc2:device (as required)	process-aaa-server (optional)	
21	Block Process	deny	openc2:process (as required)	endpoint (optional)	
22	Block Process by Domain	deny	openc2:process (as required)	endpoint (optional)	

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
23	Deny user access to the system; suitable for coordinating across multiple enclaves	deny	openc2:user-account (as required)		

A.10 DETONATE

Table. Example OpenC2 Commands: DETONATE

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Acting sends the URL to be analyzed in a sandbox.	detonate	openc2:url (as required)	process-sandbox (optional)	
2	Acting sends the file to the Sandbox for detonation analysis.	detonate	openc2:file (as required)	process-sandbox (optional)	
3	Acting sends the attachments to be analyzed in a sandbox.	detonate	openc2:file	process-sandbox	

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
			(as required)	(optional)	

A.11 DISTILL

Table. Example OpenC2 Commands: DISTILL

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Filter	distill	openc2:network-traffic (as required)	network-sensor	
2	Reduce	distill	openc2:network-traffic (as required)	network-sensor	
3	Flatten	distill	openc2:network-traffic (as required)	network-sensor	

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
4	Specify Block of IP addresses to capture sensing data from	distill	openc2:network-traffic (as required)	network-sensor	

A.12 INVESTIGATE

Table. Example OpenC2 Commands: INVESTIGATE

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Investigate the specified IP address for malicious activities	investigate	openc2:ipv4-addr (as required)		[report-to]
2	Investigate the specified device	investigate	openc2:device (as required)		[report-to]
3	Investigate the specified domain	investigate	openc2:domain-name		[report-to]

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
			(as required)		
4	Investigate the specified email message	investigate	openc2:email-message (as required)		[report-to]
5	Investigate the specified file(s)	investigate	openc2:file (as required)		[report-to]
6	Investigate the specified hostname	investigate	openc2:domain-name (as required)		[report-to]
7	Investigate the specified network traffic	investigate	openc2:network-traffic (as required)		[report-to]
8	Investigate the specified port for malicious activities	investigate	openc2:network-traffic (as required)		[report-to]
9	Investigate the specified process	investigate	openc2:process		[report-to]

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
			(as required)		
10	Investigate the specified software product	investigate	openc2:software (as required)		[report-to]
11	Investigate the specified system	investigate	openc2:device (as required)		[report-to]
12	Investigate the specified certificate	investigate	openc2:x509-certificate (as required)		[report-to]

A.13 LOCATE

Table. Example OpenC2 Commands: LOCATE

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Geolocate a device	locate	openc2:device (as required)	process-location-service (optional)	
2	Get location of an IP address	locate	openc2:ipv4-addr (as required)	process-location-service (optional)	
3	Get location of a user	locate	openc2:user-account (as required)	process-location-service (optional)	
4	Get a logical location of a file	locate	openc2:file (as required)	process-location-service (optional)	

A.14 MITIGATE

Table. Example OpenC2 Commands: MITIGATE

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Mitigate the specified malicious IP address	mitigate	openc2:ipv4-addr (as required)		[report-to]
2	Mitigate the specified infected device	mitigate	openc2:device (as required)		[report-to]
3	Mitigate the specified malicious email message	mitigate	openc2:email-message (as required)		[report-to]
4	Mitigate the specified malicious file(s)	mitigate	openc2:file (as required)		[report-to]
5	Mitigate the specified infected hostname	mitigate	openc2:domain-name (as required)		[report-to]
6	Mitigate the specified malicious network traffic	mitigate	openc2:network-traffic (as required)		[report-to]

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
7	Mitigate the specified malicious process	mitigate	openc2:process (as required)		[report-to]
8	Mitigate the specified malicious software product	mitigate	openc2:software (as required)		[report-to]
9	Mitigate the specified infected system	mitigate	openc2:device (as required)		[report-to]
10	Mitigate the specified compromised certificate	mitigate	openc2:x509-certificate (as required)		[report-to]

A.15 MOVE

Table. Example OpenC2 Commands: MOVE

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Move file/directory	move	openc2:file (as required)		move-to
2	Fork: Copy and redirect data to more than one destination	move	openc2:artifact (as required)		move-to

A.16 NOTIFY

Table. Example OpenC2 Commands: NOTIFY

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Notify security officer to report compliance with change of configuration	notify	openc2:user-account (as required)	process-email-service (optional)	message
2	Send a command to notify an	notify	openc2:process		message =

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
	external enclave		(as required)		acknowledge
3	Send a command to notify an authorized user to request approval	notify	openc2:user-account (as required)	endpoint-server	message

A.17 PAUSE

Table. Example OpenC2 Commands: PAUSE

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Pause device (system)	pause	openc2:device (as required)		[method = sleep]
2	Hibernate device (system)	pause	openc2:device (as required)		method = hibernate

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
3	Pause VM	pause	openc2:process (as required)	process-virtualization-service (optional)	
4	Pause a system or VM for a specified duration	pause	openc2:process (as required)		duration = <DURATION>
5	Pause process	pause	openc2:process (as required)	endpoint (optional)	
6	Pause a process for a specified duration	pause	openc2:process (as required)	endpoint (optional)	duration = <DURATION>

A.18 QUERY

Table. Example OpenC2 Commands: QUERY

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	List all network connections	query	openc2:artifact (as required)	network-router (optional)	response
2	List running processes on a machine	query	openc2:artifact (as required)	endpoint (optional)	response
3	Request an Actuator's supported OpenC2 capabilities	query	openc2:openc2 (as required)	network-firewall (optional)	response
4	Get attributes of a user	query	openc2:artifact (as required)	process-directory-service (optional)	response
5	List all alerts configured on the device	query	openc2:artifact (as required)	endpoint (optional)	response
6	List all endpoint applications/sensors configured on the device	query	openc2:artifact (as required)	endpoint (optional)	response
7	Get current running	query	openc2:artifact	endpoint	response

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
	configuration of the device		(as required)	(optional)	

A.19 REDIRECT

Table. Example OpenC2 Commands: REDIRECT

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Redirect traffic to a honeypot; suitable for coordinating across multiple enclaves and allowing enclaves to determine most appropriate response	redirect	openc2:network-traffic (as required)		where
2	Redirect traffic to a honeypot at a specific router	redirect	openc2:network-traffic (as required)	network-router	where
3	Cancel traffic redirection;	redirect	openc2:network-traffic		where = null

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
	suitable for coordinating across multiple enclaves and allowing enclaves to determine most appropriate response		(as required)		
4	Cancel traffic redirection at a specific router	redirect	openc2:network-traffic (as required)	network-router	where = null
5	In order to investigate a suspicious user/endpoint, an investigator would want to issue a 'redirect' command so that the endpoint's traffic is redirected to an intrusion detection system where alerts will be fired as signatures are matched	redirect	openc2:network-traffic (as required)		
6	In order to enable self-remediation of a user's endpoint, the investigator would want to redirect all URLs to a quarantine portal so that remediation services can be accessed (URL redirection for self-service remediation)	redirect	openc2:url (as required)	network-router	where

A.20 REMEDIATE

Table. Example OpenC2 Commands: REMEDIATE

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Remediate the specified malicious email message	remediate	openc2:email-message (as required)		[report-to]
2	Remediate the specified infected hostname	remediate	openc2:domain-name (as required)		[report-to]
3	Remediate the specified malicious IP address	remediate	openc2:ipv4-addr (as required)		[report-to]
4	Remediate the specified infected device	remediate	openc2:device (as required)		[report-to]
5	Remediate the specified	remediate	openc2:file		[report-to]

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
	malicious file(s)		(as required)		
6	Remediate the specified malicious network traffic	remediate	openc2:network-traffic (as required)		[report-to]
7	Remediate the specified malicious process	remediate	openc2:process (as required)		[report-to]
8	Remediate the specified malicious software product	remediate	openc2:software (as required)		[report-to]
9	Remediate the specified infected system	remediate	openc2:software (as required)		[report-to]
10	Remediate the specified compromised certificate	remediate	openc2:x509-certificate (as required)		[report-to]

A.21 REPORT

Table. Example OpenC2 Commands: REPORT

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Produce and send a report	report	openc2:artifact (as required)		report-to

A.22 RESPONSE

Table. Example OpenC2 Commands: RESPONSE

	Description	Action	Modifier
1	Acknowledge the receipt of an action	RESPONSE	type = acknowledge, command-ref = command reference
2	Signal completion of an action	RESPONSE	type = status, value = complete,

	Description	Action	Modifier
			command-ref = command reference
3	Provide the status of an action	RESPONSE	type = status, value = current, command-ref = command reference

A.23 RESTART

Table. Example OpenC2 Commands: RESTART

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Restart device (system)	restart	openc2:device (as required)		
2	Restart device (system) with different OS	restart	openc2:device		options, e.g., OS

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
			(as required)		
3	Restart VM	restart	openc2:process (as required)	process-virtualization-service (optional)	
4	Restart process	restart	openc2:process (as required)	endpoint (optional)	

A.24 RESTORE

Table. Example OpenC2 Commands: RESTORE

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Restore a device to a known restore point.	restore	openc2:device (as required)	process-remediation-service	restore-point

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
				(optional)	

A.25 RESUME

Table. Example OpenC2 Commands: RESUME

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Resume device (system)	resume	openc2:device (as required)		
2	Resume VM	resume	openc2:process (as required)	process-virtualization-service (optional)	
3	Resume process	resume	openc2:process (as required)	endpoint (optional)	

A.26 SAVE

Table. Example OpenC2 Commands: SAVE

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Save data	save	openc2:file (as required)	endpoint (optional)	save-to
2	Save an email message	save	openc2:email-message (as required)	process-email-service (optional)	save-to
3	Save a raw network packet	save	openc2:network-traffic (as required)	network-router (optional)	save-to

A.27 SCAN

Table. Example OpenC2 Commands: SCAN

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Scan a device for vulnerabilities	scan	openc2:device (as required)	network-sensor (optional)	search = CVE
2	Scan email messages for malware	scan	openc2:email-message (as required)	network-sensor (optional)	search = malware signature
3	Scan network traffic for malicious activities	scan	openc2:network-traffic (as required)	network-sensor (optional)	search = network signature
4	Scan a disk for vulnerabilities	scan	openc2:disk (as required)	network-sensor (optional)	search
5	Scan a disk partition for malware	scan	openc2:disk-partition (as required)	network-sensor (optional)	search
6	Scan a domain for malicious activities	scan	openc2:domain-name	network-sensor	search

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
			(as required)	(optional)	
7	Scan files for malware	scan	openc2:file (as required)	network-sensor (optional)	search
8	Scan memory for malicious activities	scan	openc2:memory (as required)	network-sensor (optional)	search
9	Scan network packets for malicious activities	scan	openc2:network-traffic (as required)	network-sensor (optional)	search
10	Scan a subnet for vulnerabilities or malicious activities	scan	openc2:ipv4-addr (as required)	network-sensor (optional)	search
11	Scan a process for malicious activities	scan	openc2:process (as required)	network-sensor (optional)	search
12	Scan a software product for vulnerabilities or malicious activities	scan	openc2:software (as required)	network-sensor (optional)	search

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
13	Scan a system for vulnerabilities or malicious activities	scan	openc2:device (as required)	network-sensor (optional)	search
14	Scan a URL for malicious activities	scan	openc2:url (as required)	network-sensor (optional)	search
15	Scan a user for malicious activities	scan	openc2:user-account (as required)	network-sensor (optional)	search
16	Scan a user session for vulnerabilities or malicious activities	scan	openc2:user-session (as required)	network-sensor (optional)	search
17	Scan a volume for malware	scan	openc2:volume (as required)	network-sensor (optional)	search

A.28 SET

Table. Example OpenC2 Commands: SET

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Set registry key value	set	openc2:windows-registry-key (as required)	endpoint-workstation (optional)	set-to
2	Set file permissions	set	openc2:file (as required)	process-directory-service (optional)	set-to
3	Set user rights	set	openc2:user-account (as required)	process-directory-service (optional)	set-to
4	Set password policy	set	openc2:artifact (as required)	process-directory-service (optional)	set-to
5	Set auditing policy	set	openc2:artifact (as required)		set-to
6	Set registry permissions	set	openc2:windows-registry-key	endpoint-	set-to

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
			(as required)	workstation (optional)	
7	Set service permissions	set	openc2:process (as required)		set-to
8	Set group policy (computer, user)	set	openc2:artifact (as required)	endpoint-workstation (optional)	set-to
9	Set user settings (remediate per user instead of per computer)	set	openc2:user-account (as required)	process-directory-service (optional)	set-to
10	Change a specific value in a config file	set	openc2:artifact (as required)	(optional)	set-to
11	Change firewall rule	set	openc2:artifact (as required)	network-firewall (optional)	set-to

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
12	Change HIPS rule	set	openc2:artifact (as required)	network-hips (optional)	set-to
13	Change network device rule	set	openc2:artifact (as required)	network-router (optional)	set-to
14	Acting quarantines the infected Host by commanding Directory Services to set the Host's security group. (No return requested.)	set	openc2:artifact (as required)	process-directory-service (optional)	set-to
15	Acting commands Directory Services to return the Host to the active group. (No return requested.)	set	openc2:artifact (as required)	process-directory-service (optional)	set-to
16	[Alternative] Mitigation Manager sends an OpenC2 command containing the configuration update (signatures)	set	openc2:artifact (as required)	network-sensor (optional)	set-to
17	Change system ou	set	openc2:artifact		set-to

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
			(as required)		
18	Set system attribute	set	openc2:artifact (as required)		set-to
19	Set/reset password	set	openc2:user-account (as required)		set-to
20	Change machine settings	set	openc2:artifact (as required)		set-to
21	Change desktop settings	set	openc2:artifact (as required)		set-to
22	Change device IP	set	openc2:artifact (as required)		set-to
23	Change device MAC	set	openc2:artifact (as required)		set-to

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
24	Change sensor sample rate	set	openc2:artifact (as required)		set-to
25	Limit connections to process	set	openc2:artifact (as required)		set-to

A.29 SNAPSHOT

Table. Example OpenC2 Commands: SNAPSHOT

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Take a snapshot of a VM	snapshot	openc2:process (as required)	process-virtualization-service (optional)	

A.30 START

Table. Example OpenC2 Commands: START

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Start Process, general	start	openc2:process (as required)		
2	Start Process	start	openc2:process (as required)	endpoint (optional)	
3	Start Process with Delay	start	openc2:process (as required)	endpoint (optional)	delay
4	Spawn Process	start	openc2:process (as required)	endpoint (optional)	method = spawn
5	Execute Command	start	openc2:process (as required)	endpoint (optional)	

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
6	Start an Application	start	openc2:software (as required)	endpoint (optional)	
7	Start a device	start	openc2:device (as required)	network (optional)	
8	Start a virtual machine	start	openc2:process (as required)	process-virtualization-service (optional)	
9	Activates the system partitions of a machine	start	openc2:disk-partition (as required)	endpoint (optional)	

A.31 STOP

Table. Example OpenC2 Commands: STOP

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Shutdown a system	stop	openc2:device (as required)	endpoint (optional)	[method = graceful]
2	Shutdown a system, immediate	stop	openc2:device (as required)	endpoint (optional)	method = immediate
3	Logoff User: Logoff all the sessions of a particular user from the machine	stop	openc2:user-account (as required)	endpoint (optional)	[method = graceful]
4	Stop a vm	stop	openc2:process (as required)	process-virtualization-service (optional)	[method = graceful]
5	Terminate a process, general	stop	openc2:process (as required)		
6	Terminate a process	stop	openc2:process (as required)	endpoint (optional)	
7	Stop service	stop	openc2:process	endpoint	

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
			(as required)	(optional)	
8	Terminate a session	stop	openc2:user-session (as required)	process-aaa-server (optional)	
9	Shutdown a system, general	stop	openc2:device (as required)		
10	Disable Device	stop	openc2:device (as required)	network (optional)	method = disable
11	Deactivate Partition: Deactivates the system partitions of a machine. Disallows booting from the specified partition	stop	openc2:disk-partition (as required)	endpoint (optional)	
12	Logoff User: Logoff all the sessions of a particular user, general	stop	openc2:user-account (as required)		
13	Logoff User: Logoff all the sessions of a particular user	stop	openc2:user-account	endpoint	method = immediate

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
	from the machine, immediate		(as required)	(optional)	
14	Stop a vm, immediate	stop	openc2:process (as required)	process-virtualization-service (optional)	method = immediate

A.32 SUBSTITUTE

Table. Example OpenC2 Commands: SUBSTITUTE

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Overwrite data	substitute	openc2:file (as required)	endpoint (optional)	options
2	Substitute traffic	substitute	openc2:network-traffic (as required)	network-router (optional)	options

A.33 SYNC

Table. Example OpenC2 Commands: SYNC

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Synchronize an endpoint sensor or actuator to another device	sync	openc2:device (as required)	endpoint (optional)	

A.34 THROTTLE

Table. Example OpenC2 Commands: THROTTLE

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Limit bandwidth	throttle	openc2:network-traffic	network-router	

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
			(as required)	(optional)	

A.35 UPDATE

Table. Example OpenC2 Commands: UPDATE

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
1	Install software	update	openc2:software (as required)	endpoint (optional)	
2	Install patch	update	openc2:software (as required)	endpoint (optional)	
3	Update signature file (anti-virus)	update	openc2:artifact (as required)	process-anti-virus-scanner (optional)	

	Description	Action	Target Target-Specifier	Actuator Actuator-Specifier	Modifier
4	Update sensor's signatures	update	openc2:artifact (as required)	network-sensor (optional)	
5	Load machine settings	update	openc2:artifact (as required)	endpoint (optional)	
6	Synchronize machine	update	openc2:artifact (as required)	endpoint (optional)	
7	Update registry	update	openc2:windows-registry-key (as required)	endpoint (optional)	
8	Load file	update	openc2:file (as required)	endpoint (optional)	