Open Command and Control (OpenC2) Information Assurance Implementation Considerations

Version 1.0 – Release Candidate

15 September 2016

Revision History

Revision	Date	Description
0.1	6/30/2016	Original issue
0.2	07/08/2016	Restructured section layout, added topics, removed tutorial material
0.2a	07/12/2016	Reformatted to OpenC2 standard, further restructured layout and revised material
0.2b	07/13/2016	Added tables of References and Acronyms, revised Authorization and Policy sections
0.2c	07/14/2016	Addressed reviewer comments
0.3	07/29/2016	Addressed reviewer comments, revised content to match audience and purpose
0.4	08/10/2016	Addressed reviewer comments
1.0 - RC	09/15/2016	Updated version to 1.0 – Release Candidate

TABLE OF CONTENTS

1. INTRODUCTION

OpenC2 is a language that enables the coordination and execution of command and control of cyber defense components between domains and within a domain. OpenC2 is used in conjunction with the emerging strategy of Integrated Defensive Cyber Operations (IDCO). Generically, this is the concept of enabling coordinated and automated cyber defense responses to improve the effectiveness and timeliness of reactions. The OpenC2 language provides a universally understood syntax and command set that can be contextualized for specific environments. OpenC2, will cover the range of common actions expressed at a high level which are then interpreted in terms of the context of the entity executing the commands.

IDCO implementations provides a powerful capability that needs to be protected, controlled, and monitored to reduce the risk of misuse, maintain its availability, and determine the effectiveness of response actions. This paper addresses the considerations and trade spaces involved in providing adequate Information Assurance (IA, aka cybersecurity) of C2 within a specific operational environment.

1.1 Preface

Achieving adequate information security for organizations, mission/business processes, and information systems is a multifaceted undertaking that requires:

- Clearly articulated security requirements and security specifications
- Sound systems/security engineering principles and practices to effectively integrate information technology products into organizational information systems
- Continuous monitoring of organizations and information systems to determine the ongoing effectiveness of deployed security controls, changes in information systems and environments of operation, and compliance with legislation, directives, policies, and standards and
- Information security planning and system development life cycle management.

Incorporation of the OpenC2 language into the broader ICDO ecosystem increases the complexities confronted by solution developers. The totality of this challenge calls attention to the need for a disciplined Systems Engineering (SE) approach incorporated with the Information Systems Security Engineering (ISSE) process to produce a holistic and secure solution. One of the cornerstone ideals from the ISSE process is to separate the problem space from the solution space. The problem space defines what the system will do, while the solution space defines how the system will solve the problems. This paper introduces fundamental security problem spaces for consideration; the specific solutions are left to developers for implementation, the intent is to define the considerations and trade spaces for selecting and designing them.

1.2 Audience

This document is intended for use by security systems engineers and system architects that are implementing cyber defense capabilities using the OpenC2 language. This document may also be used by product developers as a starting point for identifying IA considerations but is not intended to provide design and implementation level information. In either case, the reader is assumed to have an understanding of the system security engineering process (how threats and vulnerabilities correspond to risk and impact assessments and selection of security controls/requirements) and of the basic security services and security mechanisms.

1.3 Purpose

The general purpose is to provide architects, developers, and implementers with an overview of the security topics that should be addressed for implementing a C2 capability and relationship to architecture constructs and other considerations affecting the deployment of OpenC2. Given the state of development and adoption of OpenC2 (and advanced cyber defense capabilities in general) this document serves multiple purposes including:

A. Determining security needs leading to identification of security controls and requirements for either a system or a product implementing OpenC2 based cyber defense

B. Assessing existing security services and infrastructure to determine how well they satisfy the IA needs of OpenC2 and identifying areas where augmentation or compensating controls may be needed (e.g., stronger authentication using different certificates)

C. Informing decisions about product, protocol, and infrastructure selections that either have implemented OpenC2 or are capable of supporting its use (e.g., a messaging system)

1.4 Content

This document describes the risks (unique to C2 and common to all information systems) to OpenC2 based cyber defense capabilities, the consequences of successful attacks, and the security services needed to protect these capabilities. It is intended to provide the reader with an understanding of what IA considerations may be different for cyber defense C2 and which are common. This is an overview with the expectation that readers will then use more detailed documentation for tailoring specific security controls or making design implementation decisions.

1.5 Document Organization

The Introduction section describes the audience, purpose, content, and organization of this document.

The Background section describes the operational and threat environments and the security categorization for OpenC2 solutions.

The Security Services section discusses each service applicable to OpenC2 and the particular areas of concern.

The Architecture and Topology Considerations section discusses how aspects of network topology, messaging services, and other communications related design choices affect or are affected by security.

The Policy Considerations section discusses the types of policy rules that may need to be implemented and gives some examples of approaches.

The Other Implementation Considerations section discusses other aspects of the design decision space with security ramifications.

2. BACKGROUND

2.1 Operating Environment

OpenC2 itself is designed to be a concise language and syntax that is extensible and flexible. It can be used in a wide range of operational environments from networks of high performance, well connected systems to those with low bandwidth and limited processing or connectivity. The differences in capabilities and connectivity will affect the attack surface, the approach to providing security services, and which implementation choices are available.

As users increasingly adopt cloud service, support mobility, and move towards the Internet of Things (IoT), the makeup of networked systems will become increasingly heterogeneous. This mixed environment of devices will also come with varied degrees of capabilities and varying options for connected or disconnected use. Coupled with the adoption of emerging technologies is the continued support of legacy devices and architectures. The range of capabilities and technologies may limit the choices available for securing OpenC2 implementations.

2.2 Threat Landscape

There are two threat target areas to address when considering the OpenC2 security design: threats to the user networks/systems being defended and threats directed at OpenC2 itself. OpenC2 is a key enabler of improved cyber defense capabilities – it provides the C2 for time critical response actions.

- Cyber defense systems are particularly high-value targets in order to disable detection/response capabilities and also to allow further attacks
- The implementation and operation as well as the design functionality of security services determine the actual operational effectiveness, poorly implemented or incorrectly configured security mechanisms can fail to achieve desired security goals
- If OpenC2 traffic is communicated in-band with defended network user traffic:
  - It will be subject to the same threats as user traffic, may be delayed or blocked by high volume user traffic, will be affected by legitimate Cyber defense actions such as blocking, or may be subject to targeted attacks
  - It will leverage the same IA protections implemented for user traffic

The threats targeted at user networks and systems in general are not the focus of this paper and are assumed to be understood (or will be analyzed) by the implementer. The mission of OpenC2 is to enable responses to address those threats; however, to the extent that OpenC2 traffic and processing share resources with the user networks/systems, those same threats (and mitigations) will be applicable.

The primary focus in this paper will be on threats directed against the C2 of cyber defense. There would be great value to an attacker to turn off the sensors, breach the defenses, disable responsive actions, and potentially use the cyber defense tools to attack the network. There would be both direct/ first order effects and indirect/second and higher order effects since disabling cyber defenses would allow further attacks. Examples of malicious attacks and impacts include:

- If the attacker can see what response actions are being directed, they can modify their attack tactics appropriately or even force mission-impairing decisions.
- If they can prevent or delay C2 actions, they can reduce the effectiveness of cyber defenses and increase their window of opportunity.
- In the worst case, if the attacker can compromise integrity, authentication and/or authorization, they can subvert the defensive capabilities to their advantage or use the network defenses to compromise or deny the system being protected.

Threats to the networks and systems being defended: The connected world has evolved from fixed enterprise networks with well-defined defensive perimeters to a more dynamic environment including mobile devices, externally controlled services, and a rapidly expanding (both numbers and types of connected devices) ecosystem referred to as the Internet of Things. As both the ubiquity and complexity of connected systems increase, the corresponding attack surface presented by these systems increases and will require cyber defenses that can adapt to and address this. These defenses now include all the components within the network acting as sensor and/or actuators, not just perimeter defenses and dedicated components. OpenC2 may be used to communicate with and control any of these devices.

If OpenC2 traffic is carried in-band with user and other traffic, then it is subject to the same threats (as a minimum, see following discussion) and will leverage the same defenses as the other traffic. Even if the OpenC2 traffic is segregated using logical or cryptographic separation, the underlying physical resources may still be subject to common attacks (and other threats) that will affect OpenC2.

Threats to OpenC2 traffic and processing: The threats, attacks, vulnerabilities, and impacts to a given OpenC2 implementation should be analyzed with a focus on the goals of the attacker and the resulting impacts since these will be different from a standard user analysis. Four categories of threat sources should be addressed:

- Malicious Adversaries (external or insider) – this is the main focus of the rest of this section
- Non-malicious users – users making mistakes, especially privileged users such as administrators or managers of OpenC2 systems, can cause major lapses in cyber defense unintentionally – consider this in identifying requirements for training, on line help, least privilege controls, and audit
- Structural threats – failures of hardware and software can especially affect availability and part of OpenC2's purpose is to ensure availability of the user networks and services so robustness and redundancy need to be considered, also consider failures that impair defenses
- Environmental threats – disasters and infrastructure failures may need to addressed and accommodated depending on the mission needs of the defended networks

Malicious adversaries may use any form of attack, these are some primary examples.

Passive Attacks – An attacker may monitor traffic ranging from simple traffic analysis (is there a change in the volume of OpenC2 traffic) to eavesdropping on the contents of the messages to see what was detected, what actions are being taken, and the specific targets. This information will let the attacker know if their active attacks have been detected and how the system responds. Their active attacks can then be revised to avoid detection or to trigger a known response. In the latter case, the attacker can use knowledge of the response strategy to cause the system to unnecessarily deny services to users.

Active Attacks, Externally Initiated – An attacker may try to manipulate the OpenC2 traffic by deleting, delaying, or replaying legitimate messages. They may also attempt to modify the contents of a message or masquerade as an OpenC2 manager and issue bogus messages. If any of these attacks succeed, the attacker can disrupt or disable responses to other attacks and can cause the defensive capabilities to impede legitimate operations. Successfully subverting defenses can allow more intrusive attacks.

Insider Attacks (Malicious users) – An insider, especially a privileged user, may be able to more effectively perform any of the passive and active attacks already mentioned plus can act as a legitimate user to perform other actions. These actions could include misconfiguring devices, changing policy rules, issuing malicious commands from authorized sources, and even turning systems off.

Supply Chain or Distribution Attacks - A vendor, transporter, developer, or installer may modify the software or hardware used for OpenC2 based functions. The modification may introduce an exploitable vulnerability, disable a critical function, or cause failure under specific conditions. Even if the attack is just substitution of a counterfeit component, the behavior may be different and cause problems.

The goals and impacts of attacks directed against OpenC2 need to be understood in order to prevent, detect, and respond to them. This may build upon common capabilities but may need specialization. For instance, the authentication of OpenC2 manager may need to be stronger than other usages and the audit analysis rules may need expansion to look for OpenC2 specific actions.

2.3 Security Categorization of OpenC2 Information and Information Systems

Implementers will face a varied and changing regulatory and compliance landscape and may use different system security engineering methodologies. While this document is not specific to an industry segment or methodology, we will use some concepts from the NIST Risk Management Framework for illustration.

If is the implementer is following the Risk Management Framework (NIST SP 800-37), the first step is categorization of the information system and information processed using applicable guidelines (e.g., FIPS 199/SP 800-60 Vol. 2, ISO/IEC 27001 and 27002, or corporate). The resulting impact analysis results can then be used to select security controls from NIST SP 800-53. When assessing the Confidentiality, Integrity, and Availability (CIA) impact of OpenC2 information, key factors are the types of missions/operations and the impact assessments of the user networks being defended. Cyber defenses provide protections and responses that support availability (such as preventing Denial of Service), confidentiality (such as stopping data exfiltration), and integrity (such as blocking malware which could subvert system integrity and thereby any other security protections) of user data and services.

The CIA of OpenC2 itself is based on providing the defenses and responses for protected networks:

- The availability impact for OpenC2 is at least as high as the availability of the systems and services being defended, i.e., the cyber defenses should be available to ensure the continued operation and availability of the user network and devices.
- The integrity of OpenC2 message content is critical to correctly performing the cyber defense functions so typically the impact of integrity is as high as or higher than the integrity of the systems and services being defended.
- The confidentiality of OpenC2 message content is important to performing the cyber defense functions without tipping off the attacker. The impact of confidentiality is typically as high as the systems and services being defended, however in some cases the confidentiality impact of the OpenC2 message content could be lower than the system being protected.

The responsible ISSE will provide analysis and guidance with respect to the security categorization, however ultimately, the categorization of OpenC2 information and those information systems impact is made by the Authorization Official. Typically, the OpenC2 systems that primarily perform cyber defense would be categorized as high Integrity and availability impact, particularly when protecting an operational network. If this is a higher impact rating than the defended user systems, then additional controls, enhancements and higher strengths or assurance of mechanisms may be required for OpenC2. End user systems and other information systems that have an OpenC2 component but primarily perform other functions than cyber defense should be categorized based on their primary function but consideration should be given to the OpenC2 component security controls.

3. SECURITY SERVICES

3.1 Confidentiality

Confidentiality is important to OpenC2 message content to prevent an attacker from seeing what kinds of response actions are being taken or seeing the specific targets of actions. Knowledge of either could aid the attacker in manipulating or circumventing cyber defenses. Confidentiality protections may apply to the entire message being processed, or only to certain parts of it. Since confidentiality protection is not integral to OpenC2 language structure, the options for partial protection will probably be at the level of whole body of the message versus full message protection including header and body, i.e., it would not normally be possible to selectively protect fields within the message.

3.2 Integrity

Both data and system integrity need to be addressed in OpenC2 implementations.

Data integrity is extremely important - the contents of a C2 message should not be modifiable without detection. Replay and out of sequences attacks also need to be addressed. Message integrity must always be paired with source authentication.

System integrity including software/application integrity is also critical to OpenC2 security. If a system including system and application software is not in a compliant, stable configuration then its actions cannot be trusted.

3.3 Availability

Assuring availability can be very difficult if the OpenC2 message traffic is carried in–band with the user traffic. Even if the C2 traffic is logically or cryptographically isolated, it may still share physical resources (systems or network segments) with the user network and be vulnerable to outages at that level. Means to determine reachability or presence of devices may be required. Also, approaches to addressing intermittent connectivity and actions upon reconnection should be addressed.

Use of out-of-band management networks, where possible, can be engineered to provide better support for high availability.

3.4 Authentication

Authentication areas to address include verifying the identity and associated attributes claimed by or assumed of an entity (user, process, or device), and verifying the source and integrity of data. Since the goal of ICDO is to speed up response time, C2 will be automated as much as possible. In the consequent machine-to-machine exchanges, the systems need to securely authenticate that authorized system are involved and not a rogue entity. With the increasing number of Internet-enabled devices, reliable machine authentication is crucial to allow secure communication in automated network environments. In the IoT scenario, almost any imaginable entity or object may be made addressable and able to exchange data over the network. It is important to realize that each access point is a potential intrusion point. Each device that issues OpenC2 commands may need to apply and all users need to be able to validate strong machine authentication. In any architecture deployment, consider the appropriate levels and types of authentication for managers and actuators. There are also aspects of identity and credential management that need to be addressed: uniqueness of name space, identification of device type and instance, provisioning of credentials (typically digital certificates), means to verify trust chain and current status of credentials, means to revoke credentials, and session management. There are many challenges to find the right authentication model that can support a machine-to-machine communication method depending on the range of device and network capabilities in the operating environment.

3.5 Authorization and Access Control

Coupled with user or device authentication, a requesting entity must have authorization before executing certain tasks. Authorization is the process of enforcing policies: determining what types of actions on a resource or service are permitted for this requester. Once a user has been authenticated, they may be authorized for different types of actions depending on the policy assigned. The authorization should be role or attribute based to avoid the problems of maintaining an identity based access control list.

The policy rules may include conditional aspects such as time of day or operational status of network to prevent actions from adversely affecting missions. In these cases, it is important to determine if the requester has knowledge of the conditions and can self-impose the policy rules or whether the policy needs to be enforced at (or near) the resource.

With respect to controlling the environment and keeping commands in sync with allowed permissions and commands, another consideration for implementation is to map a controlled list of OpenC2 commands that are authorized and not authorized to various actuators.   There are actions within the OpenC2 language that can be grouped by their general activity. Each group of actions may need to have specific authorization policy rules to allow such actions to be performed.   For example, the set of actions that control permissions and accesses should be strictly limited.   The OpenC2 commands (e.g., DENY, CONTAIN, ALLOW, etc.) that affect network operations and defenses directly would have a different set of authorized users.

3.6 Accountability

Authentication is also the basis for associating the requester with the actions requested, the authorization decision (allow or deny), and the actions taken. The authenticated identity of the actor along with the action is captured in the audit logs and provides traceability to the responsible party.

3.7 Non-Repudiation

Nonrepudiation may be required if there is a requirement to formally prove the issuance or receipt of a C2 message, however any non-repudiation requirement should be evaluated critically due to impacts on the processing, availability and delays. This level of security may not be required in a closed system where source authentication and logged receipt events are sufficient evidence of who sent and who received messages. Non repudiation implementations may require a third party acting as a notary or signature-based message authentication resulting in additional costs in terms of processing, communications, and invoking third part services for the commands and responses. A time stamping service will typically also be required. The third party will time stamp OpenC2 messages and certify proof of issuance and delivery. This will add dependencies and overhead to the system.

3.8 Auditing

Audit trails are necessary in any secure system but have specific considerations in machine-to-machine communications. In conjunction with appropriate tools and procedures, audit trails can provide a means to help accomplish several security-related objectives, including individual accountability, reconstruction of events, intrusion detection, and problem identification. Typical events include:

- Authentication exchange between each component (manager, actuator, and end points)
- Message generated, message sent, message received
- Action taken/allowed or request denied
- Success or failure of any OpenC2 exchange.
- Configuration changes

Actions and the following results that are the direct result of OpenC2 should be recorded and analyzed for security areas such as forensics, secure implementation, security architecture of impact changes within the environment, and completion of such tasks. This type of auditing provides the essential ingredients for early detection of actions which violate security policy.

3.9 Metrics Collection and Analysis

Collecting metrics will be necessary for a multitude of activities to assess performance and improve effectiveness of actions within an OpenC2 environment. Implementations should provide the ability to measure resources a user or system component (e.g., sensors and actuators) consumes. This could include the amount of system time or amount of messages has sent or received during a session. This can be accomplished by logging of session statistics and usage information and is used for trend analysis, resource utilization, performance, and capacity planning. Overall all of these are important data captures to improve the configuration and deployment of OpenC2 components and a verification that intended operations are working as intended.

4. ARCHITECTURE & TOPOLOGY CONSIDERATIONS

The networking architecture, topology, methods, and technology all have implications for and may also be constrained by OpenC2 use and security. The topology and communications modes supporting OpenC2 traffic will affect the ability and approaches to achieving robustness, providing redundancy, and meeting responsiveness goals. Ideally, OpenC2 traffic should be quickly and reliably delivered to all intended recipients with some guarantee or confirmation of both delivery and action taken but this may not be practical or consistent with the risk management decisions.

It will not always be possible to achieve these goals due to constraints of available or legacy networking and systems, mobility/connectedness of devices, effects of attacks or outages in the network, and management /cost factors. The factors derived from the goals that do need to be evaluated in the various topologies and schemes described below are:

- Timeliness of delivery: synchronous or asynchronous communications, latency (if applicable)
- Priority or Quality of Service (QoS) capabilities to circumvent traffic congestion
- Separation of OpenC2 traffic or communities within a shared resource including use of Virtual Private Networks (VPNs), Virtual Local Area Networks (VLANs), and cryptographic separation of communities noting that the techniques used may preclude use of QoS routing and access to common services
- Guaranteed or best efforts delivery of messaging
- Pushed or pulled information and notification/periodicity to initiate a pull
- A-priori knowledge of all recipients versus broadcast/multicast with appropriate recipients recognizing messages that apply to them, correspondence of managers to actuators (does actuator A only respond to manager B or can any manager within the group command it?)
- Acknowledgement/confirmation of message delivery and responses indicating actions taken
- Handling of disconnected devices or devices that do not acknowledge or respond, detection of presence

The planning should include considerations for cyber resiliency or being able to continue to perform critical missions while the network is under attack. This may include congestion/flooding, temporary loss of some systems or network segments, loss of trust (or ability to determine it) in some systems, and the presence of attacking systems or persistent attacks within the network. Note that reconstitution of networks that have been fragmented or destroyed and restoral of service are not specifically covered in this document. Disaster recovery and continuity of operations planning should include considerations for the role of OpenC2/cyber defense and reconstitution/restoral of those services.

4.1 In Band Cyber Defense C2

If OpenC2 traffic is carried in-band with user and other traffic, then it is subject to the same threats (plus the threats against cyber defense C2) and will leverage the same defenses as the other traffic. Even if the OpenC2 traffic is segregated using logical or cryptographic separation, the underlying physical resources may still be subject to common attacks (and other threats) that will affect OpenC2.

Besides being subject to the same external threats as the other traffic, the implementer also needs to consider:

- Resource contention issues: C2 traffic may be delayed or blocked by high volumes of user traffic or reductions in network capacity or connectivity
- Intended cyber defense actions: The same blocking or filtering of traffic mean to stop an external attacker may affect C2 traffic flow as well, e.g., external monitoring feeds could be cut off
- Targeted attacks against cyber defense C2: The attacker may specifically attempt to single out C2 traffic for intercept, modification, or denial of service or other attack

4.2 Out of Band Cyber Defense C2

Out-of-Band management involves the use of a dedicated channel for managing network devices. This allows the network operator to establish trust boundaries in accessing the management function to apply it to network resources. It also can be used to ensure management connectivity (including the ability to determine the status of any network component) independent of the status of in-band network components. Out of Band Management (OOBM) is a common best practice with renewed focus based on the evolving threat landscape.

C2 systems are prime objectives of adversaries and OOBM can provide another layer in the defense-in-depth model. The effectiveness of this layering or separation depends on how OOBM is implemented and secured. There should be a much lower attack surface since general users would not have access to this channel. Also security policies, generally, will restrict or prohibit connection to the OOBM through access control lists or other access methods. In practice though, implementations may have prioritized administrator access (including remote access) and chosen weaker security. For example, implementers may have left back-door access in place so that disastrous failures can rapidly be fixed. To address these types of issues, a security plan should be implemented and enforced, focusing in these areas, which will enhance the entire security architecture of the enterprise:

- Definitions of vulnerabilities and risks of out of band access for OpenC2
- Review security architecture for mitigating those risks
- Proper balance between security and the need for timely out-of-band-access during critical events
- Systems of processes, equipment and technologies that provide, wherever required for OpenC2, integrity, confidentiality, and/or non-repudiation for out of band access.

4.3 Relation of Topology and Communications to Security Services

There are relationships between network topology/messaging modes and security implementations that need to be addressed including:

- Encryption and source authentication for point to point versus multicast or broadcast communications
- Encryption for synchronous, real-time versus asynchronous communications – channel/VPN or message based encryption and source authentication
- Support for use of digital signatures and accompanying access to credential validation services and sources of role and attribute information such as directory servers
- Key management schemes to support group versus point-point
- Authorization/policy enforcement in the path between manager and actuator especially to account for additional constraints on time and conditions of activation
- Traffic overhead for encryption, digital signatures, and other additional 'headers' or encapsulation especially when multiple layers are involved such as an encrypted message within a VPN
- Traffic load for broadcast, repetition, acknowledgements and means to throttle or prioritize traffic when resources are limited
- Traffic load for audit and metrics collection and means to throttle or prioritize

4.3.1 Point to Point/Peer-to-Peer

This is the simplest case for encryption (allowing for use of modern keying techniques to establish session keys, for example) and is the most direct model. A manager, for example, is connected to a specific actuator and all traffic exchanged over that connection is clearly for one of these two components. Obvious disadvantages are the need to establish and manage a lot of different connections for the range of communicating entities, having to know beforehand what component connections are needed, and potentially replicating the same message multiple times.

4.3.2 Full Mesh or Bussed

This can be accomplished with multicast or broadcast protocols and would result in all actuators (and other components) receiving a message sent from one source. It would generally be more difficult to employ VPNs in this case so the use of group keying or VLANs may be appropriate. It the messages are individually encrypted and further protection or separation is not required, and then channel encryption may not be needed. This communications mode can be more robust since multiple components can easily convey messages to multiple actuators and can listen to see if other managers are still online. There is an implication that an actuator would be able to recognize which messages apply and may also be required to reconcile getting messages from multiple managers (and determine whether they are duplicates, conflict, both apply, and if there is a sequence). There is also a larger problem with trying to introduce authorization/policy enforcement in the path between manager and actuator.

4.3.3 Partial Mesh/Star Deployment/Hub and Spoke

This supports the model where some nodes are more likely to issue multicast messages and most other nodes listen and communicate primarily with the hub. A common implementation will use a message broker to handle the dissemination and addressing of messages. This may create a hub and spoke configuration with the message broker at the hub (it could also take the form of a bussed structure). Where there are typically many actuators for relatively few managers, this is a good model for OpenC2 communications. It is important to ensure that the hub systems do not become single points of failure (that would reduce the availability of the entire system) or bottlenecks (that could become overloaded with traffic in introduces delays or losses of messages).

4.3.4 Publish – Subscribe (Pub-Sub)

For OpenC2, it is important to implement this in a push mode where subscribers are either notified or provided messages quickly. Alternatively, subscribers could be configured to poll for new messages frequently but this may not be as effective and will result in significant bandwidth overhead as systems grow in size and complexity.

5. POLICY CONSIDERATIONS

The policy rules to be implemented for OpenC2 use will impact design and security choices. For example, determining whether a manager is allowed to direct any reachable actuator or only certain types and instances is related to role based access control, key management strategies, and communications topology. Determining whether, how, and under what circumstances a local administrator can override a remote command will affect how access control and policy management are implemented. Additional policy rules and enforcement may be necessary because, even though a requester is authorized to execute an action, there may be other constraints such as actions that should not be executed at certain times, or not when certain conditions exist where the action could lead to network or device compromise.

The policy rules may need to include conditional aspects such as accounting for normal operating hours. For example, many devices such as firewalls are capable of accepting commands such as DELETE, MODIFY, ACCEPT, DENY, START, STOP, RESTART; however, not all actions should be allowed to take place especially during production operating hours. Typically devices will act upon receipt of a command and policy enforcement needs to intervene prior to the formation and transmittal. The implementation will have to include considerations for what is known and enforceable at the source (issuer of commands) and what may only be known local to the destination (receiver of commands).

5.1 Source-based Policy Controls

Managers and actuators that are responsible for sending configuration changes should have a validated list of authorized commands and have logic within the interpretation of messages to know if certain commands are not authorized to be executed. A logical mapping of device capabilities and production concerns should be considered when creating the policy rules that control issuance of commands. Policy generations should include an understanding of the devices to be managed, the flexibility of automated C2, and enforcement methods and points.

5.2 Destination-based Policy Controls

It may be possible to form a whitelist of device operational commands and enforce rules allowing certain commands to be used by certain operators and at certain times of execution (e.g., off-hours, maintenance windows). All actuators and other end points would need to able to enforce this policy and to be updated with changes in policy rules or conditional parameters. This would probably require a digital policy management capability to disseminate policy updates and maintain consistency in all the destination devices.

5.3 Application of Context Specific Policy Controls

Policy controls will also need to be enforced when high level tasking is converted to localized, more specific tasking. A high level 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. Local security policy will need to be applied and enforced at this point.

6. OTHER IMPLEMENTATION CONSIDERATIONS

6.1 Use of Non-Private Cloud Services, Service Providers, and Multi-tenancy

There may be portions of user networks, systems, applications, and/or data where the cyber defenses are provided by other organizations or there is some level of shared responsibility. If it is possible to cooperatively share information and coordinate response actions with other organizations, then there special considerations for the security of cross-organizational OpenC2 use. These include:

- Agreement on responsibilities and sharing: the using organization needs clear identification of what is being monitored, what information shared, what defenses in place, what types of response actions can be requested and may be taken, and what information on response plans and actions taken will be shared. This should include determining who has responsibility for deciding when and which patches are applied.
- In multi-tenancy implementations, determine whether all instances of an application are patched at the same time. Also determine whether information on attacks or compromises of other organizations' instances of an application uses will be shared.
- Agreement on policies and security: The two organizations will also need to agree on the security of exchanges including authentication, confidentiality and integrity protections, and authorization/use limitations (e.g., what items of one organization's information are allowed to be shared with other customer organizations of the service provider).
- Federation of identity and credential management systems to allow cross-organizational authentication (see Domain Federation section below).
- Determination of the trust level to place on information received from other organizations. This may require some form of broker or human in the loop to assess OpenC2 messages received from another organization before acting on them as well as determining what to share with the other organization.

6.2 Strict Type Enforcement and Input Validation

Strict type enforcement and related input validation are essential to avoid some of the most commonly exploited application vulnerabilities. Within an OpenC2 implementation, it may be possible to receive any arbitrary string of bytes and determine if it is a syntactically valid OpenC2 command. The absence of input validation may present opportunities for unintended code to execute on critical network defense systems.

6.3 Integration with Configuration Management

Configuration Management (CM) is the application of sound programing practices to establish and maintain consistency of a product's or system's attributes with its requirements and evolving technical baseline over its life. Considering configuration management during OpenC2 development and deployment will allow for a better organized OpenC2 deployment and help with the overall system ecosystem. Having configuration management will allow a developer/engineer to search the configuration management database for specific Configuration Items (CIs) that someone wants to designate as the different components to be deployed, such as managers, actuators, and end point systems. Configuration management owners would be notified when configuration integrations are changed, for example, if server is updated with a new release of a business application, owners of both the server CI, which would include OpenC2, and the business application CI would receive notices.

The CM process is not administrative-only; when coupled with robust auditing, CM becomes a key component used for the enrichment of operations monitoring data. As the threat landscape evolves, a well-documented CM process will ensure the audit trail does not become obsolete. While CM was once typically only found in Federal Enterprise Architectures, the proliferation of both regulations and security best practices has encouraged the wide-spread adoption of CM into systems of all scale.

6.4 Domain Federation

Federation may be needed if OpenC2 messages will be exchanged between organizations or administrative domains (inter-domain tasking or coordination). For example, exchanging OpenC2 messages with a public cloud service provider still requires a level of authentication and appropriate integrity and confidentiality protections. Unless the using organization and the cloud service provider happen to use the same PKI and namespace for digital certificates, there will be a need to address how the different schemes relate to each other. Policy and agreements will be needed on the level of assurance that is acceptable and what exchanges and accesses are allowed. There are various ways to cross-certify or bridge trust between two PKI systems and a trade decision will be needed on the most effective for each situation.

APPENDIX A ACRONYMS

List of Acronyms

Acronym	Definition
A&A	Assessment and Approval
ABAC	Attribute-Based Access Control
ACL	Access Control List
C2	Command and Control
CC	Common Criteria
CI	Configuration Item
CIA	Confidentiality, Integrity, and Availability
CM	Configuration Management
IA	Information Assurance
IDCO	Integrated Defensive Cyber Operations
IoT	Internet of Things
ISSE	Information Systems Security Engineering
M2M	Machine to Machine
OCSP	Online Certificate Status Protocol
OOBM	Out of Band Management
OpenC2	Open Command and Control
PKI	Public Key Infrastructure
POA&M	Plan of Action and Milestones
QoS	Quality of Service
RBAC	Role-Based Access Control
SDLC	System Development Life Cycle
SE	Systems Engineering
SHA-1	Secure Hash Algorithm – 1
VLAN	Virtual Local Area Network (LAN)
VPN	Virtual Private Network

APPENDIX B LIST OF REFERENCES

Number	Title and Author
CNSSI 1253	Committee on National Security Systems (CNSS) Instruction 1253, Security Categorization and Control Selection for National Security Systems
FIPS 140-2	Federal Information Processing Standard 140-2, Security Requirements for Cryptographic Modules
FIPS 199	Federal Information Processing Standard 199, Standards for Security Categorization of Federal Information and Information Systems)
ISO/IEC 27001	Information Security Management
ISO/IEC 27002	Information Technology – Security Techniques – Code of Practice for Information Security Controls
SP 800-37	National Institute of Standards (NIST) Special Publication 800-37, Guide for Applying the Risk Management Framework to Federal Information Systems: A Security Life Cycle Approach
SP 800-53	NIST Special Publication 800-53, Security and Privacy Controls for Federal Information Systems and Organizations
SP 800-60, Volume 2	NIST Special Publication 800-60, Volume II: Appendices to Guide for Mapping Types of Information and Information Systems to Security Categories