OpenC2

302 Sentinel Drive Suite 300, Columbia MD 21046

Information assurance implementation considerations

Michael Larmie and Patrick Izzie @ G2 Inc.

2016

Contents

[Introduction 2](#_Toc453679895)

[Preface: 2](#_Toc453679896)

[Focus 3](#_Toc453679897)

[Threat Landscape 3](#_Toc453679898)

[Operating Environment 3](#_Toc453679899)

[Security Implications of Language Structure 3](#_Toc453679900)

[Security Topics 4](#_Toc453679901)

[Authentication 4](#_Toc453679902)

[Authorization 4](#_Toc453679903)

[Accounting 5](#_Toc453679904)

[Auditing 5](#_Toc453679905)

[Non-Repudiation 5](#_Toc453679906)

[Integrity 6](#_Toc453679907)

[Confidentiality 6](#_Toc453679908)

[Encryption 7](#_Toc453679909)

[Cryptographic Hash Functions 8](#_Toc453679910)

[Verb Whitelisting 9](#_Toc453679911)

[Architecture & Implementation Considerations 9](#_Toc453679912)

[Peer-to-Peer 9](#_Toc453679913)

[(Multi) Hub and Spoke 9](#_Toc453679914)

[Strict Type Enforcement 10](#_Toc453679915)

[Integration with Configuration Management 10](#_Toc453679916)

[Out of Band Management (OOBM) 10](#_Toc453679917)

[Domain Federation 11](#_Toc453679918)

[Certification and Accreditation (C&A) – Regulatory Compliance 11](#_Toc453679919)

# Introduction

- U.S. military, government and commercial IT networks face constant cyberattack from both criminal and state-sponsored adversaries. Current IT security response practices are reactive in nature and typically cannot be executed in cyber relevant time. To stay ahead of increasingly sophisticated, stealthy and dangerous threats, defenders must move beyond traditional static defenses to exploit the natural advantages of their IT systems and expertise.

.

OpenC2 proposes a standardized command set which spans the range of common actions expressed at a high level. These high level commands are then specified and/or modified in terms of the context, the concerns and assets of the entity executing the commands. Each enterprise or enclave provides its own integration via the specific sensors and actuators available, and so can specify the common command set to reflect the implementations of which it is capable.

## 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;
* Well-designed and well-built information technology products based on state-of-the-practice hardware, firmware, and software development processes;
* Sound systems/security engineering principles and practices to effectively integrate information technology products into organizational information systems;
* Sound security practices that are well documented and seamlessly integrated into the training requirements and daily routines of organizational personnel with security responsibilities;
* 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.

From an engineering viewpoint, information security is just one of many required operational capabilities for information systems that support organizational mission/business processes— capabilities that must be funded by organizations throughout the system development life cycle in order to achieve mission/business success.

# Focus

* Emphasis placed on separation of problem space and solution space in context to OpenC2 implementations.
* Focus of OpenC2 as the language, not the cyber defense system
* Introduce concept that a poorly designed/implemented OpenC2 may enrich an environment for potential adversaries

## Threat Landscape

* Control planes of all systems/networks is potentially lucrative target given the access control planes necessarily have.
* CND systems are also potentially lucrative targets as part of subverting access controls and detection.
* Many attacks focus on protocol/language exploitation based on lax enforcement and poor implementation.

## Operating Environment

* Mixed environments.
* Mixed device sophistication.
* Inheritance of legacy devices and architectures.
* Varied regulatory and compliance landscape.
* .

## Security Implications of Language Structure

* To the greatest extent practical, implementations should leverage pre-existing standards and protocols. Modifications to the OpenC2 syntax should only be considered if:
  + desired effect not achievable using message stack
  + applicable to all targets and actuators
  + applicable to all contexts
* Security analysis must account for possibility of specifier being ignored
* Enduring, enclave-wide context-specific security requirements can be articulated as policy, rather than sent with each command

# Security Topics

## Authentication

The process of verifying the identity or other attributes claimed by or assumed of an entity (user, process, or device),In the overall operations of OpenC2 in context of machine-to-machine, the systems need to securely authenticate to verify it is the authorized system involved in any interaction 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 Internet of things (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 network device that has an OpenC2 component needs strong machine authentication. There are many challenges to find the right authentication model that can support a machine-to-machine communication method at extremely fast rates of operating.

## Authorization

Following authentication, a user must gain authorization for doing certain tasks. After logging on to a component, such as an orchestrator, the user account may try to issue actions. The authorization process should be in place to determine whether the user has the authority to issue such actions. Authorization is the process of enforcing policies: determining what types of qualities of activities, resources, or services/actions a user is permitted. Usually authorization occurs within the context of authentication. Once a user has been authenticated, they may be authorized for different types of actions depending on the policy assigned. In the context of OpenC2, this policy enforcement is necessary because even though a certain device can execute any action it is tasked to perform, some actions should not be executed at only certain times, or not at all as they could lead to network or device compromise.

There are actions within the OpenC2 language that can be grouped by their general activity. Each group of actions may need to have some level of authorization to allow such actions to be performed. One set of actions that control permissions and accesses are a desired area of interest of an attacker that can use these actions for his advantages. The OpenC2 commands are DENY, CONTAIN, ALLOW, which either of these used in the wrong context would lead to network compromise. Based on the security issues with authentication, the same concerns are with command authorization. OpenC2 syntax does not have inherited user schemas or even user to command mapped permissions that allow or not allow the ACTION to be performed.

Consider the following:

STOP (

[target (type=process, [id=123345])],

[actuator (type=endpoint ,[id=9876t6])],

[immediate]

)

The STOP construct would stop a process on an endpoint with an immediate action when the message is received. If this was a mission critical device that received the command without authorization, the network could be rendered inoperable.

## Accounting

Accounting will be necessary for a multitude of activity to improve communication paths and actions within an OpenC2 environment. Accounting will provide the ability to measure resources a user, a device or a subsystem consumes . This could include the amount of system time or amount of messages has sent or received during a communication session. Accounting is carried out by logging of session statistics and usage information and is used for authorization control, 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.

## Auditing

Audit trails are necessary in machine-to-machine communications. Audit trails maintain a record of system activity by system or application processes and by user activity. 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:

* Logins (successful, unsuccessful)
* Logouts
* Remote System Accesses
* System Application manipulation (start, stop, restart, change / modified status)
* File Opens, closes, and renames, deletions, modifications.
* Changes in privileges or security attributes (e.g. a change in a network service label or a user’s permission)

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 might prevent security violations from occurring.

## Non-Repudiation

Nonrepudiation is the assurance that someone cannot deny something. Typically, nonrepudiation refers to the ability to ensure that a party to a contract or a communication cannot deny the authenticity of their signature on a document or the sending of a message that they originated.

A repudiation attack happens when an application or system does not adopt controls to properly track and log users' actions, thus permitting malicious manipulation or forging the identification of new actions. This attack can be used to change the authoring information of actions executed by a malicious user in order to log wrong data to log files. Its usage can be extended to general data manipulation in the name of others, in a similar manner as spoofing mail messages. If this attack takes place, the data stored on log files can be considered invalid or misleading.

Managerial control of enforcing confidentiality throughout an OpenC2 deployment architecture along with the implementation control considerations of adding encryption could be possible to enhance the operational effectiveness of OpenC2 usage and provide a level of technical assurance of message authenticity.

## Integrity

Integrity of the messages ensures that there is some form of proof that information has not been modified either accidentally or deliberately by non-authorized entities. Integrity insures that all resources in the system are always available in a “high availability” form i.e. that the content, meaning, structure and function of each resource are always available and preserved in its correct form. It should be noted that the definition of integrity also implies availability.

Without message integrity there are several security risks that are prevalent and currently used by attackers, applicable to OpenC2, which include:

* Man in the Middle – ability to intercept the messages while in transit and possible use the information gathered for reconnaissance, and message tampering
* Forgery of the message – changing the content of the message so a different action is received.
* Replay attack - Even a valid message may present a danger if it is utilized in a "replay attack". i.e. it is sent multiple times to the server to make it repeat the requested operation. This may be achieved by capturing an entire message, even if it is sufficiently protected against tampering, since it is the message itself that is used for attack now. Usual means to protect against replayed messages is either using unique identifiers (nonce) on messages and keeping track of processed ones, or using a relatively short validity time window.

## Confidentiality

Confidentiality means that the contents of the information processed, stored, or transmitted are protected. That means the existence of a piece of information , its sources, its recipients, its contents, the meaning of structure of system resources dealing with the piece of information (message) must not be made available or disclosed in any way to unauthorized entities. Confidentiality protections may apply to the entire message being processed, or only to certain parts of it.

Applying encryption requires conducting an extensive setup work, since the communicating parties now have to be aware of which keys they can trust, deal with certificate and key validation, and know which keys should be used for communication.

## Encryption

Encryption is the conversion of electronic data into another form, called [cipher text](http://searchcio-midmarket.techtarget.com/definition/ciphertext), which cannot be easily understood by anyone except authorized parties.

The primary purpose of encryption is to protect the confidentiality of digital data stored on computer systems and/or transmitted via the network. Modern encryption algorithms play a vital role in the security assurance of IT systems and communications as they can provide not only confidentiality, but also the following key elements of security:

* Authentication: the origin of the message can be verified
* Integrity: proof that the contents of a message have not been changed since it was sent
* Non-repudiation: the sender of the message cannot deny sending the message

Applicable to OpenC2, the syntax is meant to pass content data as a plaintext message. The content message is encrypted using an encryption algorithm and an encryption key. This process generates cipher text that can only be viewed in its original form if decrypted with the correct key. Decryption is simply the inverse of encryption, following the same steps but reversing the order in which the keys are applied. Today's encryption algorithms are divided into two categories: symmetric and asymmetric.

Symmetric-key ciphers use the same key, or secret, for encrypting and decrypting a message or file. The most widely used symmetric-key cipher is AES which was created to protect government classified information. Symmetric-key encryption is much faster than asymmetric encryption, but the sender must exchange the key used to encrypt the data with the recipient before he or she can decrypt it. This requirement to securely distribute and manage large numbers of keys means most cryptographic processes use a symmetric algorithm to efficiently encrypt data, but use an asymmetric algorithm to exchange the secret key.

Asymmetric cryptography, also known as public-key cryptography, uses two different but mathematically linked keys, one [public](http://searchsecurity.techtarget.com/definition/public-key) and one [private](http://searchsecurity.techtarget.com/definition/private-key). The public key can be shared with everyone, whereas the private key must be kept secret. RSA is the most widely used asymmetric algorithm, partly because both the public and the private keys can encrypt a message; the opposite key from the one used to encrypt a message is used to decrypt it. This attribute provides a method of assuring not only confidentiality, but also the integrity, authenticity and non-reputability of electronic communications and data at rest through the use of digital signatures.

Either category could apply to an implementation and use case with OpenC2 however architecture considerations and maintenance of key pairs, speed, and distribution would have to be highly thought out. Another consideration that could be considered to at least obfuscate OpenC2 messages would be with cryptographic hashes.

For any cipher, the most basic method of attack is brute force; trying each key until the right one is found. The length of the key determines the number of possible keys, and hence the feasibility of this type of attack. Encryption strength is directly tied to key size, but as the key size increases so too do the resources required to perform the computation.

Alternative methods of breaking a cipher include side-channel attacks, which don't attack the actual cipher but its implementation. An error in system design or execution can allow such attacks to succeed.

## Cryptographic Hash Functions

A cryptographic hash function plays a somewhat different role than other cryptographic algorithms. Hash functions are widely used in many aspects of security, such as digital signatures and data integrity checks. They take an electronic file, message or block of data and generate a short digital fingerprint of the content called a message digest or hash value. The key properties of a secure cryptographic hash function are:

* Output length is small compared to input
* Computation is fast and efficient for any input
* Any change to input affects lots of output bits
* One-way-value – the input cannot be determined from the output
* Strong collision resistance – two different inputs can’t create the same output

All of these properties would benefit any OpenC2 implementation with limited resources and lack of support for providing full encryption of the messages.

The ciphers in hash functions are built for hashing: they use large keys and blocks, can efficiently change keys every block and have been designed and vetted for resistance to related-key attacks. General-purpose ciphers used for encryption tend to have different design goals. For example, the symmetric-key block cipher AES could also be used for generating hash values, but its key and block sizes make it nontrivial and inefficient.

## Verb Whitelisting

Even though devices have support for a variety of methods of configuration management and operational change configurations, the ability to use such methods at any given time and at any given moment would be considered hazardous to network and device operations. It would be an operational method to form a whitelist of device operational commands and have a technical consideration of allowing certain commands to be used by certain operators and at different times of execution (off-hours, maintenance windows). An implementation model could be developed with the commands and OpenC2 syntax to make best use of creating a best use model of operating the devices autonomously.

For example, many devices such as a firewall are capable to accept commands such as DELETE, MODIFY, ACCEPT, DENY, START, STOP, RESTART however even if such a device is capable of accepting the commands and act upon an action, not all actions should take place especially during production operating hours. Actuators and Target devices will blindly act upon actions, in essence OpenC2 command hierarchy is too much of an enabling system without certain capability checks in place.

# Architecture & Implementation Considerations

## Peer-to-Peer

With-in a peer-to-peer architecture and without basic verifications of commands there are a few issues that could arise. Misinterpretation of commands may be the most common threat. The threat appears if the language of OpenC2, say a communications protocol in the simplest case, is not sufficiently verified. If the communication language is used which is more complicated than the messages of a fixed format, the semantic problems of understanding appear. The language implementation in a peer-to-peer architecture may need to be kept as simple as possible.

Misunderstanding of a situation could lead to wrong decision planning and execution stages. It is a threat that may be more prevalent based on the environment that OpenC2 is implemented into is sometimes unknown to the designer (the author of the OpenC2 implementation). The environment in any cyber space network is complex with many different operating systems, software platforms, protocols etc. The best recommendation would be to restrict the environment as much as possible by permitting the OpenC2 components to operate only on known platforms. Situational awareness always needs to be improved and verified, or at least carefully tested with respect to safety of the OpenC2 components behaviors.

Peer-to-peer architectures can present many atypical issues not found on other domain-based systems. Secure connections have different key management interface challenges and basic issue of trust must be addressed differently.

## (Multi) Hub and Spoke

Actuators and Orchestrators in cyber operations and cyber defense can be used most efficiently in multi-tenant formations. Botnets could be an example, if bots are developed as actuators. However, the control in botnets has still remained quite simple. One can expect that multi-tenant systems will become the main form of deployments in cyber operations. In this case, the botnets will negotiate between themselves and will cooperatively create a complex behavior for achieving the general goals stated by a commander known as the orchestrator. As a consequence, the strict control of behavior of every single bot will be weaker. Also, it will be more difficult to foresee all possible cases for decision making. Practically, it will be impossible to verify the outcome of multi-tenant behavior for all situations. It is possible that backdoors and forced destruction will have to be built into the system. Multicast control messages may be needed for emergency cases of the actuator control. Another option could be self-destruction of actuators if loss of contact occurs, i.e. if for some time no command and control messages are received. A specific threat of multi-tenant systems is the formation of unwanted coalitions by actuators. This can happen if the actuators get too much autonomy in decision making. Communication between the actuators will be only partially observable to human controllers in this case. This will require very careful selection of constraints on the behavior of actuators. However, there will never be an absolute guarantee of avoiding a misunderstanding of a situation by a team of actuators. Also, a danger remains that a collection of actuators may behave unintentionally in a harmful way.

Strict Type Enforcement

Strict type enforcement and the related input validation is a requirement. Within an OpenC2 implementation, it must be possible to receive any arbitrary string of bytes and determine if it is a valid OpenC2 command. As long as all the elements in the message are validated against a controlled set of vocabulary, using optional modifiers would not be a problem.

## 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 CI’s that someone wants to designate as the different components to be deployed, such as orchestrators, or actuators, and targets for a release. 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.

## Out of Band Management (OOBM)

The continued exponential growth and increased complexity of IT infrastructure — M2M, Cloud Computing, Internet of Things — drives the need for smarter real-time decision-making, deeper data analysis, and more robust data storage at the network edge. The demand for always-on connectivity and resilience in the face of network failure events also continues to rise. Rapid identification and remediation of connectivity issues between dispersed Internet-connected devices and remote infrastructure is critical. 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 other in-band network components. OOBM is a common best practice with renewed focus based on the threat landscape listed earlier in this document. C2 systems are prime objectives for bad actors and OOBM offers another layer in the defense-in-depth model.

Company security policies, generally, will restrict or prohibit a dedicated connection to the OOBM ports through access control lists or other access methods. In practice though, the requirements of system availability often lead to unsecure access points. Those responsible for keeping devices and networks running may leave back-door access in place on these OOMB ports so that disastrous failures can rapidly 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.

## Domain Federation

A federation trust model is often found in partner organizations with explicit trust and a shared security and/or accreditation boundary.

## Certification and Accreditation (C&A) – Regulatory Compliance

Regulated industries such as public companies, health care providers, financial institutions and other organizations that handle PII and adhere to PCI will have to follow strict policy enforcements in validating security requirements for systems, applications, system software, and other technologies before they are deployed into a production environment. This process is designed to ensure compliance with specifications, regulations, standards and objectives identified during each phase of a System Development Life Cycle (SDLC). With the previous topics discussed around OpenC2, when it comes to an implementation, the overall system architecture has to address fundamental security features (auditing, CIA, AAA, encryption) in order to meet the requirements listed in the various security requirements to pass compliance standards.