Leveraging OPC UA to Help Meet Site Security Requirements

By: Tony Paine

Introduction

Today's industrial control systems leverage the latest OPC Unified Architecture (UA) standards, which provide cross-platform interoperability between software applications and devices. This allows for the vendor-independent exchange of real-time, alarm and condition, historical, and many other types of data. Due to the criticality of Industrial Control Systems (ICS) and the desire to make process and business information available to anyone, anywhere, and at any time, the information that is exchanged between interested parties must meet the security requirements of the site. The OPC UA standards are designed to meet these security requirements while maintaining the level of flexibility and control that site administrators expect.

Security Objectives

Before we discuss how OPC UA can meet site security requirements, we will review the security objectives that apply to industrial control systems. These objectives, along with a brief description, are as follows:

- **Auditability:** This provides traceability on all actions performed within a system, with the goal of logging which user performed what action and when—as well as any attempts that have been made to compromise the system.
- **Authentication:** An ICS is comprised of hardware, software applications, and users. Each component must be able to prove its identity in order to be considered a trusted party.
- **Authorization:** Even though a party may be trusted, components should be given the minimum access needed to perform their functions. This role based access will include the ability to read, write, and/or perform actions that execute a task.
- **Availability:** This ensures the ICS is fully operational by limiting factors that may impact its execution.

- **Confidentiality:** The information exchanged between trusted parties must be hidden from those that are not trusted or have no reason to view the information. This requires that the sender and receiver of information must be able to encrypt and decrypt the data they exchange, based on an agreed upon algorithm to which only they are privy.
- **Integrity:** The information exchanged between trusted parties must not be modifiable. The information received must be the same information that was initially sent.

"It is imperative that we identify the threats that may compromise security."

Security Threats

In order to meet these objectives, it is imperative that we identify the threats that may compromise security. Though not comprehensive of all possible threats, some that have plagued ICS over the years are as follows:

- **Compromising User Credentials:** This occurs when an attacker is able to assume the identity of a user by obtaining their username, password, or other credentials either by guessing or through physical or electronic means.
- **Eavesdropping:** An unauthorized party is able to intercept confidential information for personal gain or to leverage in future security attacks.
- **Malformed Messages:** By sending abnormal communications traffic to an application or device, the receiver may perform inappropriate tasks or burden itself with unnecessary processing that may limit its availability to authorized parties.
- **Message Alteration/Spoofing:** An attacker manipulates or forges a message between applications and devices in an effort to perform unauthorized tasks under the identity of an authorized party.
- **Message Flooding:** An attacker targets an application or device by sending it very frequent and large amounts of communications traffic with the goal of taking the receiver offline, thereby impacting its availability to others through denial of service.
- **Message Replay:** By capturing authenticated messages and resending them at some time in the future, an attacker is able to perform valid operations at inappropriate times. They may also deceive the users of the system that operations are running normally, while it is actually being compromised.
- **Profiling:** This occurs when an attacker applies knowledge about security vulnerabilities that exist within a particular version of an application or device. These vulnerabilities may be announced by the vendor in an effort to make the public aware that previous flaws have been resolved.
- **Session Hijacking:** This occurs when an attacker is able to inject itself in between running applications and/or devices and take over the session from an authorized party.

By implementing a security strategy for a site, an administrator can thwart these threats and achieve the security objectives needed to protect critical infrastructure.

Site Security

Most sites will incorporate a Cyber Security Management System (CSMS) to address security-related requirements. These requirements may range from the adoption of security policies around physical and electronic boundaries, auditing, and preventive and response procedures. In order to address the threats discussed earlier, a security risk assessment will be initiated and appropriate security measures will be implemented. A good implementation will follow a "defense–in-depth" strategy, where there will be multiple layers of protection. This is necessary because there are no one-size-fits-all solutions that will protect against all security threats. Instead, many security threat-specific appliances will be deployed to protect a site. This may include a combination of firewalls, intrusion detection/prevention systems (IDS/IPS), patch-management systems, and IT rules for what is allowed and what is not allowed within the context of the system.

Through its flexible security model, OPC UA can adapt to a site's CSMS by allowing the administrator complete control on how communications are setup and managed. OPC UA's client/server architecture also bodes well with a defense-in-depth strategy, as UA-aware applications can act as an intermediary between different layers within a site and limit the amount of information that can be exposed or manipulated.

"To mitigate against security threats, OPC UA has a multi-tier design that consists of an application layer, communications layer, and a transport layer."

OPC UA Security Architecture

To mitigate against security threats, OPC UA has a multi-tier design that consists of an application layer, communications layer, and a transport layer.

The majority of OPC functionality is handled within the context of the application layer. This is where clients and servers process UA information—operations like reading, writing, and browsing items. It is also where UA provides management for authentication and user authorization through the concept of a session between a client and server instances. Each session communicates its information over a secure channel that is handled by the communications layer.

The secure channel that makes up the communications layer provides the functionality that allows for application authentication, confidentiality, and integrity. It does this by utilizing an appropriate level of encryption and decryption to maintain confidentiality of communication messages, by signing messages to ensure the integrity of data, and by utilizing digital certificates that provide application authentication. The resulting secured data is then passed on to the transport layer for further processing.

The transport layer handles the actual sending and receiving of the data over a communications infrastructure. The transport mechanism used (such as OPC UA Binary or XML Web Services via HTTP) impacts the implementation of the secured channel managed by the communications layer.

How It Works

Every OPC UA application has a unique digital (X.509) certificate assigned to it per installation that is referred to as an application instance certificate. This certificate is created by the application at the time of installation, but can be overwritten with another certificate as deemed appropriate by a site administrator. This certificate is comprised of a public key that can be shared with other trusted parties, as well as a private key that is only known to the application instance. These keys vary in size, where the longer the key, the harder it is for a third party to guess.

"Every OPC UA application has a unique digital (X.509) certificate assigned to it per installation that is referred to as an application instance certificate."

When a client application connects to a server, it creates a secured channel. This process requires the client and server to exchange public keys for further communications. Only if an administrator has configured the client and server to trust each other's certificates will the secured channel be established. This procedure provides application authentication.

In order to provide the identity of the user of an application, the client will next create a session that leverages the secure channel for communications. Applications may utilize this user information to limit or restrict access to certain operations, thus providing a level of user authorization.

From here on out, the client will encrypt all communications it sends to the server with the server's public key, and will sign each message with its own private key. Upon receiving a message, the server will test the integrity of the message by validating the signature against the client's public key and decrypt the message with its own private key. This ensures that all messages remain confidential and are not tampered with.

Mitigation of Threats

By now we should be able to see how OPC UA's secure channel/session model protects against eavesdropping, message spoofing, message alteration, session hijacking, and the electronic detection of user credentials through its use of public/private key signing and encryption to ensure the authentication, confidentiality, and integrity of communications. For some of the other threats, we need additional counter measures that are handled by the design, specification, and recommendations of OPC UA.

OPC UA limits what actions a client can perform on a server before it is authenticated. Clients are limited to obtaining the security and connection

requirements of a server and creating a secure channel. In the case of security and connection requirements, this information seldom changes after deployment and will require little processing time on the server. For the more processor-intensive secure channel creation, servers should monitor for repeated failed secure channel creation requests and intentionally delay the handling of future requests to minimize the impact of a potential attack. Servers should also allow administrators to limit the number of concurrent connections it will handle at any given time. These steps will prevent against message flooding. An additional benefit of limiting what an unauthenticated party can accomplish is that any known security vulnerabilities that could be compromised are limited, which minimizes the server profiling that could be done by an attacker.

"OPC UA provides the industry with interoperability between software-based applications and hardware appliances from various vendors."

Each message that is exchanged contains a session ID, secure channel ID, request ID, timestamp, and sequence numbers. Since these messages are not modifiable, applications can validate these values to ensure that an attacker has not captured a message to replay at some point in the future. Clients and servers also validate each message to ensure it is of the proper form. Together, this eliminates the concerns around message replay and reception of malformed messages.

Summary

OPC UA provides the industry with interoperability between software-based applications and hardware appliances from various vendors. This interoperability allows for the exchange of information that is critical to any industrial control system, as well as for the overall business. In order to make optimal operating and business decisions, the ability to obtain information from anywhere in the world is pivotal. Since this will require data to be transmitted over public domains, it must be done securely in order to protect the authenticity, integrity, and confidentiality of information. OPC UA's adoption of today's widespread IT principles and techniques makes this possible.