Matrikon®
Powering Interoperability.

Best Practices for Securing OPC Classic Applications

January 2022

LIMITATIONS

Matrikon has made its best effort to prepare this manual. Matrikon makes no representation or warranties of any kind with regard to the completeness or accuracy of the contents herein and accepts no liability of any kind including without limitation warranties of merchantable quality, satisfactory quality, merchantability, and fitness for a particular purpose on those arising by law, statute, usage of trade, course of dealing or otherwise. Matrikon shall not be liable for any losses or damages of any kind caused or alleged to be caused directly or indirectly from this manual.

About This Guide

Scope

This document provides information on Matrikon recommended best practices for securing OPC Classic servers and applications. These practices make use of features that are specific to OPC applications, as well as the existing Windows security framework. Implementation details for these practices are not included in this document. Where available, links to information or specific tools for implementing these practices have been provided.

Intended Audience

Matrikon provides this document for the use of system administrators, systems integrators, operators, managers, users, and any others responsible for

- The installation and configuration of Matrikon OPC software
- The installation and configuration of 3rd party OPC software
- Commissioning and maintaining systems that employ OPC applications
- Troubleshooting systems and applications that employ OPC applications

Document Terminology

Table 1 provides an explanation of the acronyms and abbreviations used in this manual.

Table 1- Document terminology

Term/Abbreviation	Description
ACL	Access Control List
COM	Component Object Model. A method for organizing software, specifying how to build applications that can be dynamically interchanged.
CVE	Common Vulnerabilities and Exposure. A system that provides a method for publicly sharing information on cybersecurity vulnerabilities and exposures.
DCOM	Distributed Component Object Model.
DCOMCNFG	Windows DCOM Configuration Utility, part of Component Services. Used to modify the ACL of COM objects.
gMSA	Group Managed Service Account
GPO	Group Policy Object, a domain-level security policy
LSP	Local Security Policy
MSA	Managed Service Account
OPC	OPC Classic, a communication standard based on Microsoft COM/DCOM. Refer to https://opcfoundation.org for more information.
OPC UA	OPC Unified Architecture, a communication standard currently based on TCP technologies. Refer to https://opcfoundation.org for more information.
OS	Operating System

Matrikon®
Powering Interoperability.

Contents

Purpose

This document describes the recommended best practices for configuring Windows security on platforms hosting OPC applications. These best practices are recommended by Matrikon as measures to ensure the confidentiality, integrity, and availability of the OPC applications and the systems on which they are installed. They are not meant to supersede or replace those security measures dictated by our customers' site security policies and protocols. In cases where there is a conflict between the two, the customers' site security policies should be implemented. It should also be noted that some of these settings may be managed by the organization's IT department and may be unavailable for modification.

Introduction

OPC Classic (**OPC**) applications are used globally as a key connectivity standard for process control systems. This makes them an attractive target for bad actors attempting unauthorized access, or worse, to these systems. Securing OPC communications provides protection from unauthorized users seeking to

- access data from process control systems,
- corrupt data from process control systems, or
- interfere with the availability of process control systems data or devices.

All OPC applications are built on the Microsoft Component Object Model (**COM**) infrastructure and are therefore considered COM components. As such, they can only run on a Windows platform. The Distributed Component Object Model (**DCOM**) extends COM functionality to multi-user and network communication scenarios. All COM objects are constrained by the Windows DCOM security framework.

Besides DCOM security mechanisms, a COM client's ability to connect to servers is also affected by;

- firewalls,
- local security policies (**LSP**),
- Group Policy Objects (**GPO**),
- authentication requirements, and
- application identities.

In short, anything that affects security on a Windows platform can potentially affect OPC connectivity and communication.

Server Registration

DESCRIPTION

Upon installation, all Windows programs are configured in the registry to run as either

- an interactive program which is launched from a user session, or

- a **Windows service** which starts with the operating system and runs in a non-interactive session.

BEST PRACTICES

OPC servers should be run as services unless the server specifically requires running in an interactive session to function properly. This requirement is usually included in the server documentation.

Information on Windows services can be found here.

User Accounts

DESCRIPTION

Every program running on a Windows machine is associated with an identity, a Windows account that the program uses to access the system resources it requires. Upon installation, many OPC applications use the System account by default.

BEST PRACTICES

Matrikon recommends that the following accounts **not** be used for OPC application identities;

- The **System account** as it is a privileged account with more authority in the system than is normally required by OPC applications. Many control system applications deny access to the System account for this reason.
- **Any interactive user accounts**, as these may have permission sets that conflict with what is required by the OPC application.

Matrikon further recommends that service accounts be created specifically for use by the OPC applications and should not be used by other applications. As the requirements of the OPC applications will vary, the following types of service accounts should be created;

- **Local administrator**. This level of authority in a service account should only be used where the application requires administrative privileges to function properly.
- **Local user**
- **Domain user**

These accounts should be created as Group Managed Service Accounts (**gMSA**) with the following attributes;

- Not listed in **Deny log on as a service** in the LSP. This allows the account to log on as a *service* in Windows.
- Listed in **Deny log on locally** in the LSP. No interactive user can log in using this account.
- Listed in **Deny log on as a batch job** in the LSP. This account cannot be used to start scheduled jobs.

Information on creating and using MSAs and gMSAs can be found here.

Information on configuring LSPs can be found here.

Default Passwords

DESCRIPTION

Upon installation, OPC applications may create default passwords used to access application features.

BEST PRACTICES

Default passwords should be changed, even if the features they protect are not implemented. This ensures that default passwords, which are widely known, cannot be exploited by unauthorized users trying to access customers' systems.

Run as Administrator

DESCRIPTION

Using this option allows an interactive application to use local administrative privileges, even if the user that launched the application is a standard user. It permits the application to access system areas that would be otherwise unauthorized. A user requires access to an administrator's credentials to use this option.

BEST PRACTICES

- This option **should** be used for installation and configuration of the OPC software.
- This option **should not** be used during operational run time.

DCOM Security Settings

Modifying DCOM Settings

DESCRIPTION

These settings determine how OPC applications interact with other entities in the Windows environment. These settings include:

- Which identities have which level of access to the COM (OPC) object.
- The identity the OPC server will use to access other system resources.
- When authentication is required during communications

Access to the COM object is constrained by an Access Control List (**ACL**). DCOM maintains an ACL for all COM objects on the machine.

BEST PRACTICES

The recommended method for modifying the security on a COM object, including the DCOM ACLs, is **DCOMCNFG**, a graphic configuration utility that is part of Windows' Component Services.

Client DCOM permissions

DESCRIPTION

Client permissions are set using the COM function call **CoInitializeSecurity**. This function can be called only once per instance; subsequent calls will not be executed and will return an error. If the client calls this function, the settings are based on the parameters included in the call. This is the preferred method for setting client permissions and is used by most clients.

If the client does not call this function, the OS will call it on the application's behalf, based on the settings in the **Default DCOM settings**. Where the client does not set its own security, administrators must ensure that the Default DCOM settings are properly configured. When configuring these settings, it is important to remember that they are System-Wide defaults and will affect all COM objects on the machine that use them.

BEST PRACTICES

When configuring these Default DCOM settings for OPC client use,

- ensure that the **Default Authentication Level** is set to **Packet Integrity**,
- add only the identities that are used by the OPC servers, and
- do not remove any identities already configured.

Instructions for setting these permissions can be found here.

Server DCOM permissions

DESCRIPTION

Although it is possible for developers to configure OPC server security programmatically, the DCOMCNFG utility is normally used to set server permissions. This tool allows for more flexibility in setting permissions for OPC servers that differ from those used by other COM objects on the system.

During the installation of Matrikon OPC servers, the configuration of the server permissions can be selected to include access for the following users:

- Everyone
- Interactive
- Network
- System

Refer to the section on Commissioning and Troubleshooting for more information on these settings.

BEST PRACTICES

When setting OPC server permissions, use the Custom settings

- to avoid reliance on the Default DCOM settings, and

- to ensure that access to the servers is restricted to authorized users with an operational requirement to access the server and the data it exposes.

In configuring the custom settings:

- Set the **Authentication Level** to **Packet Integrity**.
- In the **Launch and Activation**, and **Access** permissions;
    - remove the *Everyone*, *Interactive*, and *Network* users.
    - remove the *System* account unless it is specifically required for the server to operate correctly.
    - add only those identities with a requirement to access the data in the server. This is usually restricted to the identities of the client applications.
- Configure the **Identity** of the server as one of the accounts described in the User Accounts section. The *Launching* and *Interactive* users should not be used as they are known to cause operational issues with OPC applications.
- **Configuration** permissions should be modified to include only administrators authorized to make changes in the overall system configuration.

These permissions can be modified using the **DCOMCNFG** utility. For instructions on setting these permissions, refer to Microsoft Docs.

Additional OPC security considerations

OPC Security Specification implementation

DESCRIPTION

Regulatory requirements or site security policies may mandate that a higher level of security be applied to OPC server applications. For such cases, the OPC Security specification provides vendors of OPC software the opportunity to implement security directly on the server object. This security does not override the DCOM security settings in the registry. If, for example, a user is denied access to the server in DCOM, server security cannot be used to grant access.

This server security configuration is normally secured by a password.

BEST PRACTICES

Where a higher level of security is required on OPC servers, OPC servers that support the OPC security specification should be selected.

If this security is implemented in the server but not required or used, the default password should be changed during installation. Default passwords are widely known and present an easily exploitable attack surface for unauthorized users.

Commissioning and troubleshooting

DESCRIPTION

During these stages of an application life cycle, the system is not normally in production mode and access is restricted to those administrators carrying out associated tasks. The focus is on establishing communication and assessing proper functionality. Application security often places constraints on the servers that work against these purposes.

BEST PRACTICES

During the Commissioning and Troubleshooting stages

- Set the Launch and Activation, and Access permissions in the Custom DCOM settings for the server to include the *Everyone*, *Interactive*, *Network*, and *System* accounts. This removes security as a potential cause of the connectivity issues
- Upon completion of the testing or commissioning stage, return the security settings to the recommended Custom configuration.

User Management

DESCRIPTION

All versions of Windows support the creation of **User Groups** at both the local and domain level. Permissions applied to a User Group affect all identities configured as members of that group.

BEST PRACTICES

If many users require access to one or more OPC servers, segregating the users into appropriate User Groups (example: based on specific roles or areas of responsibility) is recommended.

Use of User Groups simplifies configuration and ongoing management of server access permissions because

- Consistent access options are applied to each member in a given User Group
- Users can be easily added and removed from User Groups without having to edit their individual permissions
- Access permission changes for a given User Group are instantly applied to all of its members, which eliminates potential human error if changes had to be made on a per-user basis

Configuration Files

DESCRIPTION

Client and server applications use configuration files to

- store configuration information,
- enable fast switching between configuration states, and
- enable easy re-configuration of the application.

Although not a requirement under the specification, most OPC applications do implement this functionality.

BEST PRACTICES

To maintain security for the configuration files and the applications they support, these files should be stored in a secure location in the directory structure. Access to the files should be limited to the identity of the client or server application to which they belong and authorized system administrators.

System Considerations

These are beyond the scope of the security measures implemented directly on the server and are subject to local site security policies and system design constraints.

Updates and patching

DESCRIPTION

Microsoft and application vendors regularly release security updates, patches, and service packs. These help ensure that

- known security issues are mitigated,
- bug fixes are implemented, and
- new or enhanced features are added.

BEST PRACTICES

Prior to installation, any system or application update should be assessed and tested to ensure that operational or security issues are not introduced due to the update.

Windows Updates

Updates are commonly released monthly. Where operational or regulatory concerns preclude timely implementation of the updates, a maintenance schedule should be introduced to permit update installation on a regular basis.

Annual Software Maintenance Plans

Subscription-based maintenance plans offer an excellent source for providing and tracking product functionality updates, security patches, and bug fixes. Where available, these plans should be implemented with the same considerations for applicability, regulatory controls, and timing as mentioned previously in this section.

Firewalls

DESCRIPTION

Firewalls provide a secure mechanism for restricting incoming communication to computers and applications in zones where a higher level of security is required. Firewalls can be associated with

- an operating system,
- an anti-virus or system protection suite, or
- a network component such as a router or layer-4 switch.

BEST PRACTICES

Where installed, firewalls should be turned on and application whitelisting enabled. Where whitelisting is ineffective or not available, an OPC Tunnelling application like Matrikon OPC UA Tunneller should be used to enable OPC communication without compromising firewall efficacy.

OPC Tunnelling

DESCRIPTION

OPC Tunnelling is a method of enabling OPC applications to communicate across a network without using the DCOM transport protocol. This is achieved by running an instance of a tunnelling application, like Matrikon OPC UA Tunneller, on the PCs where OPC components are installed. Tunnelling applications use COM to communicate with local OPC components and but employ a non-DCOM-based protocol, such as TCP/IP, between each other. By doing this, the tunnelling applications prevent Windows from invoking the DCOM communication protocol.

Advantages of using an application like Matrikon OPC UA Tunneller (**UAT**) over plain OPC communications include

- UAT replaces use of the DCOM communication protocol with a secure TCP/IP based protocol, enabling;
    - all communications between a client and server via a single port,
    - easy firewall configuration, and
    - simplified routing across domains and network segments.
- UAT provides additional security mechanisms
    - Encryption
    - Server access restriction
- UAT seamlessly integrates with the Matrikon OPC Security Gateway to implement Matrikon Per-User-Per-Tag Security on all installed OPC servers

BEST PRACTICES

Use an OPC tunnelling application to facilitate

- secure network traversal,
- easy security configuration, and
- effective security implementation.

When Matrikon UAT is installed;

- enable encryption using 256 AES encryption
- enable server access restrictions
- use **dcomcnfg** to restrict access to local OPC servers
    - remove all authorized users except the identity used by UAT server-side component
    - deny remote permissions for remaining authorized user(s)

Documentation

DESCRIPTION

Document Everything. *"If it is not written down, it did not happen."* Whether for regulatory compliance, incident investigation, or system maintenance and troubleshooting, a detailed record of what is installed on a system makes it easier to assess and rectify any incidents and their effects on the system.

BEST PRACTICES

Document sets should include;

- system architecture design documents
- as-built drawings of the actual system architecture
- locations of application configuration and log files
- security configuration settings for applications and operating systems
- other relevant documentation not covered by the above list

Update Frequency & Change Logs

Documents should be updated on a regular basis and after any changes are made to the system. Change logs should be kept and updated for all relevant configuration changes.

Applications such as Matrikon OPC Analyzer can be used to gather system information when required. The Microsoft Attack Surface Analyzer can be used to assess the vulnerability level of a system before and after software installation or configuration changes.

Information on the Matrikon OPC Analyzer can be found here.

Information on the Microsoft Attack Surface Analyzer can be found here.

Summary

In many cases, security is not configured on OPC applications as they are deemed to be sufficiently protected by the layers of protection above them in the system infrastructure. Recently, vulnerable attack surfaces within Windows have been discovered. Bad actors, and the tools they employ, have become increasingly more sophisticated, subtly exploiting any vulnerability. These best practices are recommended as “due diligence” efforts to ensure that OPC applications, the data they expose, and the systems they are installed on are protected against these threats.

For additional information

For more information and additional whitepapers on OPC technology, visit www.matrikonopc.com.

For additional information on implementing OPC security, best practices, and resolving issues related to OPC applications, visit www.opcsupport.com.