Tofino Security OPC Foundation

White Paper

Version 1.1
Published September 3, 2010

Securing Your OPC Classic Control System

Contents

Authors

Thomas Burke, President, OPC Foundation

Eric J. Byres, Chief Technology Officer, Byres Security Inc.

Executive Summary

OPC Classic is a software interface technology used to facilitate the transfer of data between different industrial control systems. It is widely used to interconnect Human Machine Interface (HMI) workstations, data historians and other hosts on the control network with enterprise databases, Enterprise Resource Planning (ERP) systems and other business-oriented software. Unfortunately, securely deploying OPC Classic has proven to be a challenge.

This white paper describes two independent techniques for ensuring strong security in systems using OPC Classic technology. This first creates zone-based defenses using OPC-aware firewalls. The second takes advantages of improvement in the Windows operating system to managing OPC accounts and permissions. Both security techniques are available and proven for use in today's control systems.

OPC Classic – The World's Leading Industrial Integration Standard

Formerly known as OLE for Process Control, (where OLE stood for Object Linking and Embedding), OPC Classic was developed in 1996 in response to a demand for standard methods to allow different control systems to interface with each other. Today it has grown to be the world's leading technology for integrating different automation products.

No single industrial communications standard has achieved the widespread acceptance across so many different verticals, industries and equipment manufacturers as OPC Classic. It is used to interconnect an amazing variety of industrial and business systems, ranging from Human Machine Interface (HMI) workstations, Safety Instrumented Systems (SIS) and Distributed Control Systems (DCS) on the plant floor, to enterprise databases, ERP systems and other business-oriented software in the corporate world.

The reason for OPC's wide spread popularity has been simple – it is the only truly universal interface for communicating with diverse industrial devices and applications, regardless of manufacturer, software or protocols used in the control system. Before OPC Classic was developed, developers had to create specific communications drivers for each control system they wished to connect with. For example, one HMI vendor developed over 200 drivers for different DCS and PLC systems in the pre-OPC days. Now they focus their efforts on a single optimized OPC driver for their product, which then communicates to other OPC servers designed and sold by the manufacturers of the other control products.

For the end user, a significant advantage of using OPC is not having to directly deal with the control device's internal architecture. In other words, integration teams can work with named items (or groups of items) across multiple product lines, instead of dealing with raw register locations (e.g. 40020 or N7:2) and data types (e.g. 32-bit integer or IEEE floating point). This allows for an easier job when adding or changing control systems, such as when migrating from a proprietary protocol to an Ethernet-based protocol.

Most engineers soon found using OPC saved significant time during configuration, as compared to using traditional communications technologies. The result of all this is that it is rare to find an industrial facility today that isn't using OPC for some portion of its system integration strategy.

What is OPC Classic Anyway?

The term "OPC Classic" refers to all OPC standards prior to the new OPC-Unified Architecture (OPC-UA) standard. This includes the most popular specifications such as OPC Data Access (OPC DA), OPC Alarms and Events (OPC A&E) and OPC Historical Data Access (OPC HDA).

What sets OPC Classic apart from the emerging OPC-UA standards is that OPC Classic is based on Microsoft's Distributed Component Object Model (DCOM) technology. DCOM in turn, is the culmination of a number of other technologies including Component Object Model (COM) and Object Linking and Embedding (OLE). All these technologies use a network protocol called Remote Procedure Call (RPC) to connect over a typical Ethernet/TCPIP industrial network.

One of the important things to understand about OPC is it is an Application Programming Interface (API) and not an "on the wire" protocol. Instead the underlying DCOM and RPC technologies are the real "protocols". OPC is at a higher level of abstraction than true communications protocols such as Ethernet, TCP/IP or RPC.

All OPC technology is based on a client/server architecture where computers run software that makes them either a client or a server (or both in some cases). The OPC server is a software application that gathers information from a device (such as PLC, DCS or SCADA controller) using that device's native protocols (such as Modbus or PROFIBUS). The server then provides access to this data via COM objects and method calls, allowing multiple OPC clients to indirectly read and write to the field device via the OPC server.

An OPC client is an application that accesses the data held by OPC servers. An HMI package may contain an OPC client that allows it to access data provided by an OPC server application resident on another computer. The HMI package could also act as an OPC server, allowing other OPC clients to access the data it has aggregated either directly from field controllers or from other OPC servers.

To illustrate this client-server architecture, imagine a simple system with three basic components designed for controlling the water level in a tank:

- A Modbus-capable PLC performing the actual control functions
- An OPC platform that contains an OPC server and a Modbus protocol driver
- A HMI for operator access to the control system

The HMI will need to be able to write the set point in the controller, read the current water level, and monitor the controlled output (the pump) and alarms. If the HMI needs to read a value from the PLC, it sends a request via an OPC API call and the server translates this into a Modbus message for communications to the PLC. When the desired information returns from the PLC to the OPC server, it then translates that back to OPC for transmission to the HMI.

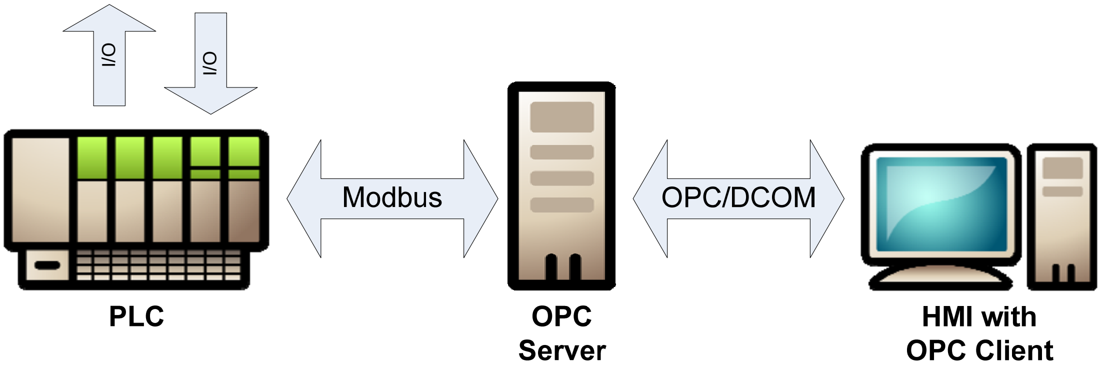


Figure 1: Example of OPC Client-Server Architecture in Tank Level Control

The Security Challenges of OPC Classic

While control system manufacturers, integrators and end users were happily deploying OPC Classic in their plants and factories, security researchers (and the hacking community) were noticing that there were a few serious issues with the standard.

The first issue stems from the fact that the RPC and DCOM protocols were designed before security issues were widely understood. Thus a number of early design decisions were made that make DCOM deployments easy rather than secure. This created one of the most vexing problems in OPC security, namely the problem of dynamic port allocation.

To understand the security risks that dynamic port allocation poses to a control system, it is necessary to understand a bit about TCP (and UDP) ports. These ports are not physical ports like an Ethernet port, but instead are special numbers embedded in every TCP or UDP message to identify the application protocol being carried in the message. For example, Modbus/TCP uses port 502 and HTTP uses port 80. These numbers are registered under the Internet Assigned Numbers Authority (IANA) and are rarely ever changed.

This port number consistency makes firewall rule creation relatively simple – if you want to block all Modbus traffic through the firewall, simply define a rule that blocks all packets containing 502 in the destination port field.

The problem with OPC Classic is that out-of-the-box OPC servers don't use a fixed port number. Instead they dynamically assign a new TCP port number to each executable process serving objects to clients. The OPC clients then discover the port numbers associated with a particular object by connecting to the server and asking what TCP port number they should use for this session. Then they make a new TCP connection to the server using the new port number.

Because OPC servers are free to use any port between 1024 and 65535, OPC becomes very "firewall unfriendly" - configuring an IT firewall to leave such a wide range of ports open presents a serious security hole and is generally considered unacceptable practice. As a result, OPC Classic has been considered by many to be impossible to secure using conventional IT-style firewalls.

The second issue with the use OPC Classic is caused by overly permissive access rights. Because setting up OPC can be a complex process, a number of major vendors make recommendations that leave the end users' OPC security configuration wide open. For example, one PLC vendor recommends that all remote access and launch controls be set for Anonymous Logon. These overly permissive settings allow any individual on any network to run arbitrary OPC services on the OPC computer, a major security risk.

The final issue (but the one most often quoted in the popular press) is that OPC Classic's underlying protocols, namely DCOM and RPC, can be vulnerable to attack. Over the past half decade, viruses and worms from the IT world have increasingly focused on these protocols, as noted in this attack trends discussion:

> *"Over the past few months, the two attack vectors that we saw in volume were against the Windows DCOM (Distributed Component Object Model) interface of the RPC (remote procedure call) service … These seem to be the current favorites for virus and worm writers, and we expect this trend to continue."*[i]

As operating system testing and patching has improved over the past few years, this has become less of an issue, but plenty of worms are still out there looking for a poorly secured OPC system.

Why Security Matters

One might be tempted to wonder if security is even important in a system using OPC, as these systems are rarely connected directly to the Internet. Unfortunately, even if you have a completely isolated system, good security is essential for reliable and safe plant operation.

The Stuxnet worm incidents of July 2010 provided a clear indication that the hacking community is now focusing specifically on industrial automation systems. In the Stuxnet case, a worm propagated through infected USB keys (so Internet connectivity was a not a requirement for infection). Subsequent analysis has shown the worm was designed specifically to target and infect Siemens' HMI, PCS7 and Step7 PLC products. It is capable of logging into and stealing process information and even hiding modifications it might make to PLC programs from users trying to examine the PLC logic.

In a less famous incident directly related to OPC Classic, a major refining complex was infected by the virus W32.Sality in 2009 when a contractor remotely connected to a control system to provide maintenance support. The virus was able to propagate from OPC clients to OPC servers, infecting

[i] Bruce Schneier, "Attack Trends" QUEUE Magazine, Association of Computing Machinery, June 2005

multiple control systems in the facility and causing repeated crashes of key servers. OPC Classic's dynamic port allocation issues complicated the problem, as it made it almost difficult to use firewalls to isolate one control system from another.

Making OPC Classic Secure

The good news is that two of the three security issues faced by OPC Classic, namely excessively open firewalls (caused by dynamic port allocation) and the overly permissive access rights, are within the control of most OPC end users. An analysis conducted by Byres Research, BCIT and Digital Bond for Kraft Foods and the US Government, showed that if either issue is addressed, then the chance of a security event is significantly reduced.

To address the third OPC Classic security issue, namely vulnerabilities in DCOM and RPC, good anti-virus or patching programs are the best answer. These are now standard practice in most control systems and significantly reduce the window of opportunity for viruses and worms to exploit vulnerabilities in these protocols.

If your OPC computers do not have an anti-virus and patch management program in place, we strongly recommend this as an initial step. Providing detailed guidance on anti-virus or patching techniques is beyond the scope of this document, but we have listed a few excellent documents and guidelines in the References section. As well, most major control systems vendors now provide guidance and approved software packages for anti-virus and automated patching deployment.

The other good news for users of OPC Classic is that new technologies have been developed in the past few years that make OPC security much simpler than it was in earlier years. Using the ANSI/ISA-99 security standards as a framework, we will outline two solutions that should be considered in securing OPC systems. Each solution can be implemented independently of the other, so we have listed them below simply in order of ease of deployment.

Creating Zone-Based Defenses with OPC-Aware Firewalls

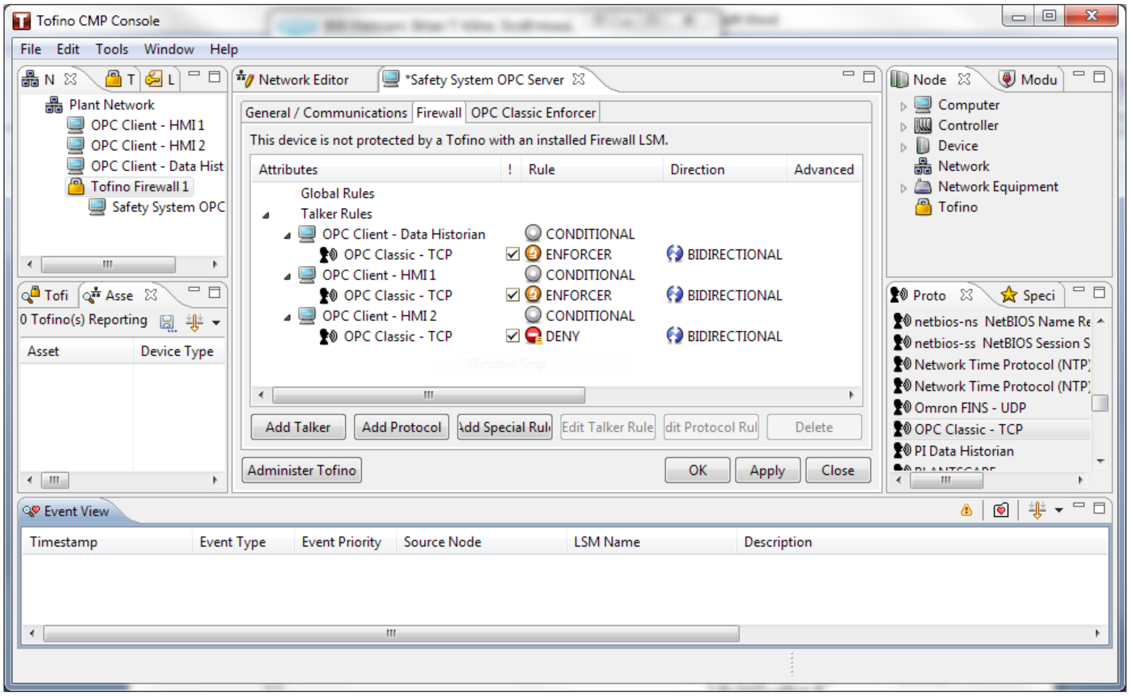
One of the core concepts in the ANSI/ISA-99 standards (and in the soon-to-be-released IEC 62443-2-1 standards) is the use of security zones. Like watertight bulkheads in a ship, dividing a control system into security zones prevents issues in one area from migrating to another area. Between the zones are "conduits" (typically firewalls) that carefully manage all the inter-zone network traffic. For example, if zones had been in place in the refinery noted earlier, the virus would have been contained to the single computer that the contractor infected, rather than impacting systems throughout the entire facility.

Until recently, OPC Classic's dynamic port allocation problem made deploying ANSI/ISA-99 zones almost impossible. Since the OPC server might use any port number between 1024 and 65535, any firewall between clients and servers must also have all those ports open, effectively making the firewall useless.

This issue was partially addressed in 2007 when Microsoft revealed a technique to modify the Windows Registry settings in OPC servers to limit the range of port numbers that are dynamically allocated. This technique is described in detail in the *US-CERT Recommended Practices Guide - Hardening Guidelines for OPC Hosts* and in a Tofino Security White Paper. Unfortunately, this solution can add configuration complexity for the system administrator because each OPC host needs to have its Windows registry adjusted. Furthermore, subsequent testing has indicated that the technique does not work for some poorly behaved OPC Server products.

A simpler solution is to use the OPC-aware firewalls now on the market. Invensys Operations Management, MTL Instruments and Hirschmann/Belden have all released firewalls that can automatically track and manage OPC Classic's dynamic port problem. These units are designed to be dropped into an existing network carrying OPC DA, HDA or A&E traffic, and require no changes to the existing OPC clients and servers.

In the Invensys case, the firewall is completely preconfigured to work with Triconex safety products. In the MTL and Hirschmann cases, the firewall is configured using a simple drag-and-drop editor to select permitted clients and servers. Figure 2 below shows the simple rule configuration process, where the icons representing three OPC clients have been dropped onto the firewall table for the OPC Server. In this example, two of the clients are permitted to communicate with the server, but one (HMI 2) has been denied.



This device is not protected by a Tofino with an installed Firewall LSM.

Attributes	!	Rule	Direction	Advanced
Global Rules				
Talker Rules				
OPC Client - Data Historian		CONDITIONAL		
OPC Classic - TCP	☑	ENFORCER	BIDIRECTIONAL	
OPC Client - HMI 1		CONDITIONAL		
OPC Classic - TCP	☑	ENFORCER	BIDIRECTIONAL	
OPC Client - HMI 2		CONDITIONAL		
OPC Classic - TCP	☑	DENY	BIDIRECTIONAL	

Figure 2: OPC-Aware Firewall Rules Allowing Two Clients to Connect to a Server

In all cases where OPC-aware firewalls are used, the control network appears closed to all but well-formed OPC traffic from the user-approved clients and servers. All the open firewall ports typically required in a system using OPC Classic are gone, improving security significantly with very little effort.

Managing OPC Accounts and Permissions

When Microsoft released Windows XP/SP2 and Windows Server 2003/SP1, it included a number of improvements for managing DCOM services. By adjusting some DCOM options in a computer‘s Component Services application you can

1. Control authentication levels for various OPC actions;
2. Control the location of various OPC actions;
3. Manage the DCOM permissions;
4. Limit protocols used by DCOM/RPC

Below we will provide a summary of the steps needed to secure an OPC sever that is running on Windows XP SP2, Windows Server 2003, Windows Server 2008, Windows Vista or Windows 7. Additional details are available in the US-CERT Recommended Practices document, “*OPC Security White Paper #3–Hardening Guidelines for OPC Hosts*” listed in the References section.

It is important to note that these settings are more complex than the ones noted in the earlier section on firewalls and can negatively impact OPC operations for some products. Thus we highly recommend that you test them in a non-production environment such as a lab or simulator first.

Launching Component Services

There are two main objectives in managing accounts and permissions in an OPC Server. First, we only want to give as much permission as is required, and ideally we want to do that on a per DCOM application basis. For example, if a computer is running three OPC servers, but only one needs to be accessed remotely, only allowing remote access to that one server is the preferred solution. Similarly, if all OPC servers and clients are on a single host, then disable remote access and allowing only local access significantly improves overall security.

Second, we need to create and use different level user accounts for OPC's Launch and Access permissions. In most control environments, the day-to-day operation of OPC-based applications does not require a highly privileged account. On the other hand, the configuration of OPC applications often does. Unfortunately, in many systems we see the highly privileged account settings being the norm, exposing the system to numerous security issues.

To address this, we recommend OPC administrators create two accounts, one for day-to-day operations and one for configuration. First, an account (or better yet an account group) should be created called "opcadmin" that is the only user account used to launch or configure OPC servers. A second account (or account group) called the "opcuser" account can be created and used by users who need only to connect and access running OPC servers.

Once these accounts are in place, we can move to the DCOM Configuration Tool that is found under *Control Panel/Administrative Tools/Component Services as* shown in Figure 3. Once the Component Services application is running, open up "*Component Services*" tab. Within it, click on "*Computers*" and then "*My Computer*".

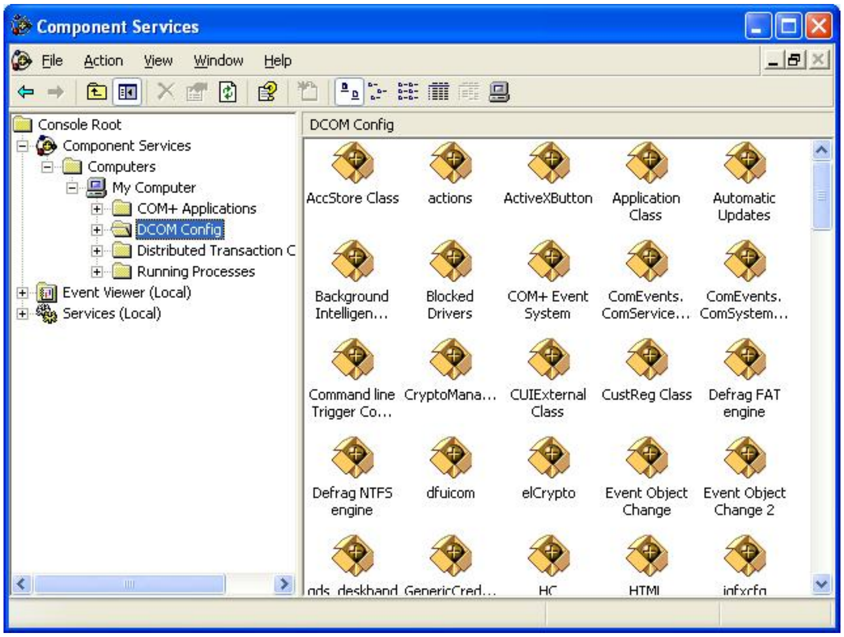


Figure 3: Component Services (DCOM) Configuration Tool

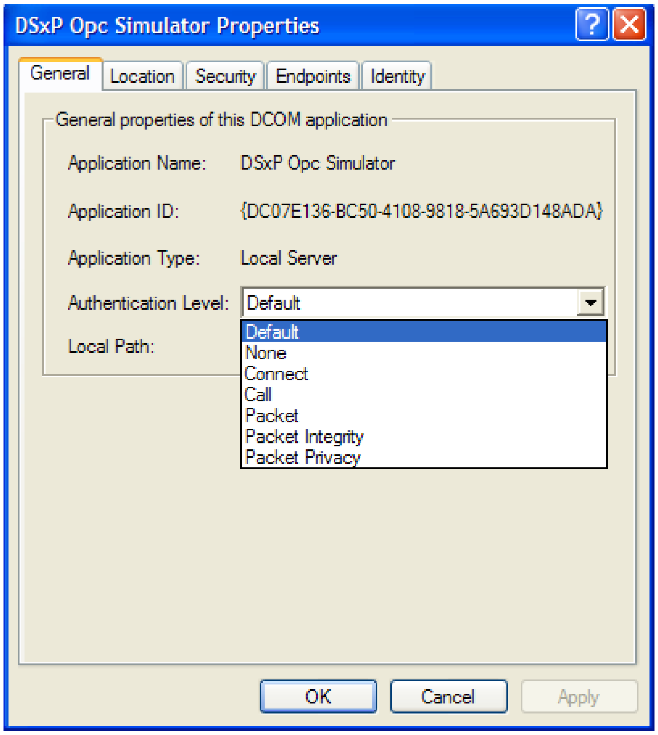
At this point you have two options – you can either configure the default DCOM permissions for all DCOM applications or just the permissions for a specific OPC server application. Right clicking on "*My Computer*" in the "Component Services" screen and choosing "*Properties*" from the menu will let you set the defaults.

On the other hand, if you want to set the permissions for a specific OPC application click on "*DCOM Config*" to get the screen shown in Figure 3. This list will include all the applications on this server that can use DCOM. On the plant floor you are likely to find the OPC servers you are using, but you may have to dig around for them. For the rest of this section we will assume that you are setting the permissions for a specific OPC application.

Controlling the Authentication Level

The first change to make is to the Authentication Level of the OPC server as shown in Figure 4. These Authentication levels determine what authentication is needed for an OPC client to connect, and are defined as follows:

- *Default* - May vary depending upon operating system and obviously the default "*My Computer*" Property settings. Usually it is "None" or "Connect".
- *None* - No authentication.
- *Connect* - Authentication occurs when a connection is made to the server.
- *Call* - The authentication occurs when a RPC call is accepted by the server.
- *Packet* - Authenticates the data on a per-packet basis. All data is authenticated.
- *Packet Integrity* - This authenticates the data that has come from the client, and checks that the data has not been modified.
- *Packet Privacy* - In addition to the checks made by the other authentication methods, this authentication level causes the data to be encrypted.



Application Name: DSxP Opc Simulator

Application ID: {DC07E136-BC50-4108-9818-5A693D148ADA}

Application Type: Local Server

Authentication Level: Default

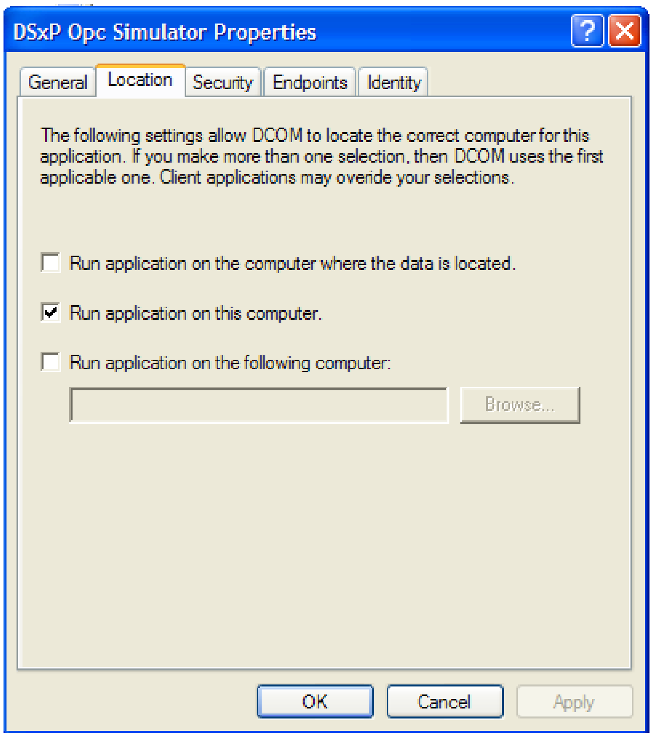
Local Path:

Figure 4: General Configuration Tab for an OPC Server

Select the OPC server you are configuring and in the General Tab, and change authentication to either "*Packet Integrity*" or "*Packet Privacy*". The "*Packet Privacy*" option can be used if data confidentiality is required since it encrypts all traffic and is the most secure option. However it is important to test this offline first as the encryption may impact performance. In most cases "*Packet Integrity*" is sufficient.

Controlling the Location

The “Location” tab lets you configure where the DCOM server can run. Here only the local computer is specified which is the typical situation in most environments as shown in Figure 5.



DSxP Opc Simulator Properties

The following settings allow DCOM to locate the correct computer for this application. If you make more than one selection, then DCOM uses the first applicable one. Client applications may overide your selections.

☐ Run application on the computer where the data is located.

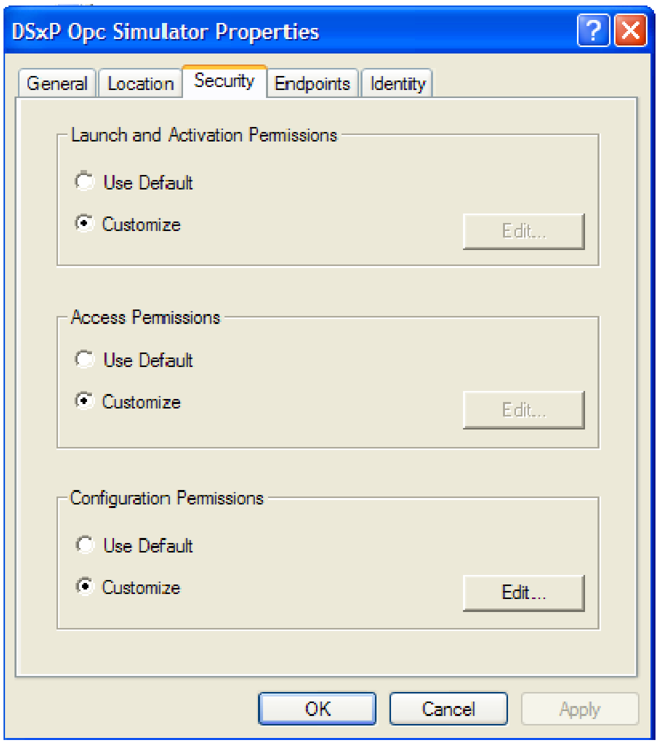
☑ Run application on this computer.

☐ Run application on the following computer:

Figure 5: Location Configuration Tab for an OPC Server

Managing DCOM Permissions

From here we move to the “*Security*” tab which allows you to configure the permissions for the different accounts. COM server applications have three types of permissions, namely Launch permissions, Access permissions and Configuration permissions. Configuration permissions control configuration changes to a DCOM server, while Launch permissions control the authorization to start a DCOM server if the server is not already running. Finally Access permissions control authorization to call a running COM server, and are the least dangerous. These permissions can be further divided into Local and Remote permissions.



DSxP Opc Simulator Properties

Launch and Activation Permissions

○ Use Default

● Customize

Access Permissions

○ Use Default

● Customize

Configuration Permissions

○ Use Default

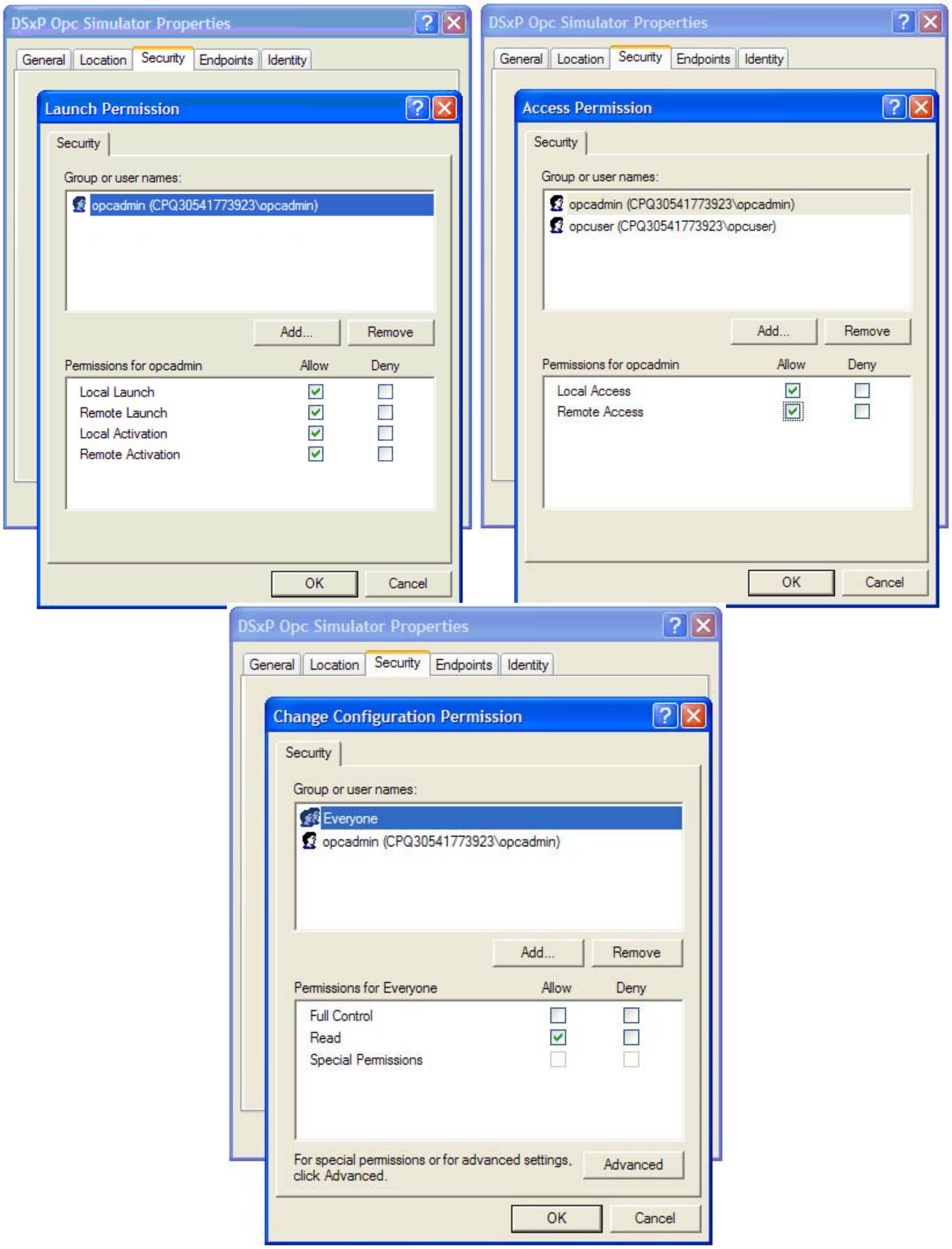
● Customize

Figure 6: Security Configuration Tab for an OPC Server

These permissions control what user accounts can execute which action on an OPC server. For all three options choose *Customize*, then *Edit* and adjust the accounts as follows:

- *Launch Permissions* - Remove all existing entries and add the *opcadmin* account created earlier. (Some servers may also require launch permission for the *opcuser* account.) If a particular OPC server is meant only to be used locally, then remote access to that server can also be disabled.
- *Access Permissions* - Remove all existing entries and add the *opcadmin* and *opcuser* accounts. Again, if a particular OPC server is meant only to be used locally, then remote access to that server can also be disabled.
- *Configuration Permissions* - Remove all existing entries other than the *Everyone* account. Modify *Everyone* to be read-only, and add *opcadmin* with full control.

These settings are shown in Figure 7. As noted above, if the server is only to be used locally (i.e. the clients and servers are all on the same machine) then *Remote* should be turned off.



Launch Permission

Security

Group or user names:

opcadmin (CPQ30541773923\opcadmin)

Permissions for opcadmin	Allow	Deny
Local Launch	☑	□
Remote Launch	☑	□
Local Activation	☑	□
Remote Activation	☑	□

Access Permission

Security

Group or user names:

opcadmin (CPQ30541773923\opcadmin)
opcuser (CPQ30541773923\opcuser)

Permissions for opcadmin	Allow	Deny
Local Access	☑	□
Remote Access	☑	□

Change Configuration Permission

Security

Group or user names:

Everyone
opcadmin (CPQ30541773923\opcadmin)

Permissions for Everyone	Allow	Deny
Full Control	□	□
Read	☑	□
Special Permissions	□	□

For special permissions or for advanced settings, click Advanced.

Figure 7: Launch, Access, and Configuration Permission Tabs for an OPC Server

Limiting RPC Ports and Protocols

The “*Endpoints*” tab allows you to select what protocols and ports can be used by this server. Prior to the development of OPC-aware firewalls, this tab also could be used to limit dynamic port allocation. Unfortunately, not all vendors of OPC products respect the setting of port numbers in this tab, so it was rather problematic. Today this setting should remain at *Default System Protocols*.

Setting the OPC Application's Account

Finally, the "*Identity*" tab lets you configure what user account the DCOM application will run under. Unless specifically required by the vendor of the OPC server, the OPC software should be set to run as the "opcuser" account and not the "opcadmin" account.

Test the New OPC Permissions

As you will have noted from the above, the settings for restricting DCOM account permissions are more complex than the ones needed for OPC-aware firewalls. In addition, making these changes can negatively impact the operation of some OPC products. Thus we highly recommend that you test all the above security settings first in a non-production environment and then again in the actual production system.

Looking Forward - OPC-UA and OPC-XI

Over the past few years, the OPC Foundation has developed two new versions of OPC called OPC-UA and OPC-XI, which are based on protocols other than DCOM. Once most OPC applications make this migration from the DCOM-based architecture to .NET-based architecture, then industry will have a significant opportunity for better security when it comes to OPC.

If your operation has already converted to OPC-UA or XI, we salute you. However, if your company is like most, it will be a while before you can rid your plant of all traces of OPC Classic. With the world facing new and evolving cyber threats, some now directed specifically at industrial control systems, we recommend that all companies take a serious look at improving the security of their OPC Classic systems. The techniques and technologies for better OPC security outlined in this white paper are available and proven. As many companies have discovered, not using them can be costly.

Frequently Asked Questions

What is OPC Classic?

OPC Classic is the new name for all OPC specifications that are based on Microsoft's COM and DCOM technologies. This includes the most popular OPC specifications such as OPC Data Access (OPC DA), OPC Alarms and Events (OPC A&E) and OPC Historical Data Access (OPC HDA).

What is OPC-UA (Unified Architecture)?

OPC-UA is a new specification created by the OPC Foundation to tie together all existing OPC technology using the Microsoft .NET Architecture. OPC-UA replaces COM, DCOM and RPC in favor of two different transports: SOAP/HTTPS and a binary message encoding scheme that operates direct communications on top of TCP.

What are COM, DCOM and RPC?

COM (Component Object Model), DCOM (Distributed Component Object Model) and RPC (Remote Procedure Call) are the actual communications protocols used by OPC Classic to communicate between clients and servers. In most cases, DCOM and RPC use the lower layer Ethernet, IP and TCP protocols to travel between computers.

What is a TCP or UDP Port Number?

The Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP) specify a that 16-bit unsigned integer (i.e. 0 to 65535) should be placed in the packet header to indicate the application which is sending or receiving the message. This is known as the port number.

What is a Firewall?

A firewall is a mechanism used to control and monitor traffic to and from a network for the purpose of protecting devices on the network. It compares the traffic passing through it to a predefined security criteria or policy, discarding messages that do not meet the policy's requirements.

My control system is never connected to the Internet. Am I skill at risk from cyber incidents?

ABSOLUTELY – Studies have shown that only a few attacks on control systems have come directly from the Internet. Most enter the system from either the business network or through secondary pathways such as infected laptops, USB keys, remote access over Virtual Private Networks (VPNs) or modems.

References

For more information on securing systems using OPC Classic, we recommend the following:

ANSI/ISA Security for Industrial Automation and Control Systems Standards

ANSI/ISA-99.00.01-2007 Security for Industrial Automation and Control Systems Part 1: *Terminology, Concepts, and Models*
http://www.isa.org/Template.cfm?Section=Standards2&template=/Ecommerce/ProductDisplay.cfm&ProductID=9661

ANSI/ISA-99.02.01-2009 Security for Industrial Automation and Control Systems: *Establishing an Industrial Automation and Control Systems Security Program*
http://www.isa.org/Template.cfm?Section=Standards&template=/Ecommerce/ProductDisplay.cfm&ProductID=10242

US-CERT Control Systems Security Program (CSSP) Recommended Practice Guides

OPC Security White Paper #1 – Understanding OPC and How it is Used: An introduction to what OPC is, what are its basic components and how is it actually deployed in the real world. http://www.tofinosecurity.com/understanding-opc

OPC Security White Paper #2 – OPC Exposed: What are the risks and vulnerabilities incurred in deploying OPC in a control environment?
http://www.tofinosecurity.com/opc-exposed

OPC Security White Paper #3 – Hardening Guidelines for OPC Hosts: How can a server or workstation running OPC be secured in a simple and effective manner?
www.tofinosecurity.com/opc-hardening

Security Implications of OPC, OLE, DCOM, and RPC in Control Systems
http://www.us-cert.gov/control_systems/practices/Recommended_Practices.html

Microsoft Support Articles

How to configure RPC dynamic port allocation in servers to work with traditional firewalls
http://support.microsoft.com/kb/154596

Configuring the Windows Firewall to allow inbound traffic that uses Dynamic RPC
http://technet.microsoft.com/en-us/library/cc732839%28WS.10%29.aspx

RPC Configuration Tool – a tool that configures a server using Microsoft RPC to listen on specified ports and to use specified subnets for RPC (for Windows Server 2000 only)
http://www.microsoft.com/downloads/details.aspx?FamilyID=0f9cde2f-8632-4da8-ae70-645e1ddaf369&DisplayLang=en

NIST Guidelines for Securing Windows Systems

SP 800-68 Revision 1, Guide to Securing Microsoft Windows XP Systems for IT Professionals
http://csrc.nist.gov/itsec/guidance_WinXP.html

CERN Guidelines for Securing OPC Host Systems

http://itcofe.web.cern.ch/itcofe/Services/OPC/GettingStarted/DCOM/RelatedDocuments/ITCODCOMSettings.pdf

Acknowledgements

The authors of this white paper would like to acknowledge the pioneering security work of the original 2006 OPC Best Practices team, namely John Karsch, Joel Carter, Darren Lissimore, Dale Peterson and Matt Franz. Without their efforts, many of the concepts in this paper would not have been as complete or as well tested. Thank you.