Homeland Security

INL/EXT-05-01005

Security Implications of OPC, OLE, DCOM, and RPC in Control Systems

January 2006

Prepared by Idaho National Laboratory

Control Systems Security Center

INL/EXT-05-0005
Rev. 0

Security Implications of OPC, OLE, DCOM, and RPC in Control Systems

January 2006

Idaho National Laboratory

Idaho Falls, Idaho 83415

Prepared for the
U.S. Department of Homeland Security
Under DOE Idaho Operations Office
Contract DE-AC07-99ID13727

Control Systems Security Center

Security Implications of OPC, OLE, DCOM, and RPC in Control Systems

INL/EXT-05-01005, Rev. 0

January 2006

Approved by:

Raymond Fink, Author Control Systems Security Center Idaho National Laboratory	2/2/06 Date
Fred Cowart, Program Manager Control Systems Security Center Idaho National Laboratory	2/2/06 Date
Rita Wells, Interim Program Lead SCADA/Power Systems Idaho National Laboratory	2/2/06 Date

EXECUTIVE SUMMARY

OPC is a collection of software programming standards and interfaces used in the process control industry. It is intended to provide open connectivity and vendor equipment interoperability. The use of OPC technology simplifies the development of control systems that integrate components from multiple vendors and support multiple control protocols. OPC-compliant products are available from most control system vendors, and are widely used in the process control industry.

OPC was originally known as OLE for Process Control; the first standards for OPC were based on underlying services in the Microsoft Windows computing environment. These underlying services (OLE [Object Linking and Embedding], DCOM [Distributed Component Object Model], and RPC [Remote Procedure Call]) have been the source of many severe security vulnerabilities. It is not feasible to automatically apply vendor patches and service packs to mitigate these vulnerabilities in a control systems environment. Control systems using the original OPC data access technology can thus inherit the vulnerabilities associated with these services.

Current OPC standardization efforts are moving away from the original focus on Microsoft protocols, with a distinct trend toward web-based protocols that are independent of any particular operating system. However, the installed base of OPC equipment consists mainly of legacy implementations of the OLE for Process Control protocols.

CONTENTS

1. TECHNOLOGY BACKGROUND: WHAT IS IT AND WHY DO WE CARE? 1

2. OPC IN THE CONTROL SYSTEMS MARKET 2

3. POTENTIAL FOR VULNERABILITIES IN OPC 3

   3.1 VULNERABILITIES IN UNDERLYING SERVICES 3

   3.2 ISSUES SPECIFIC TO OPC 4

4. CONCLUSION 5

ACRONYMS

API	Application Programming Interface
COM	Component Object Model
DCS	Distributed Control System
DCOM	Distributed COM
HMI	Human Machine Interface
OLE	Object Linking and Embedding
OPC	OLE for Process Control
PLC	Programmable Logic Controller
RPC	Remote Procedure Call
RTU	Remote Terminal Unit
SCADA	Supervisory Control and Data Acquisition

SECURITY IMPLICATIONS OF OPC, OLE, DCOM, AND RPC IN CONTROL SYSTEMS

The purpose of this white paper is to describe the use of OPC, OLE (Object Linking and Embedding), DCOM (Distributed Component Object Model), and RPC (Remote Procedure Call) technologies in Supervisory Control and Data Acquisition (SCADA) and process control systems, and examine the implications of their use from a computer security perspective.

OPC is a collection of standards and interfaces that are specific to process control, intended to provide open connectivity and vendor equipment interoperability. To quote the OPC Foundation:

> *OPC technology can eliminate expensive custom interfaces and drivers traditionally required for moving information easily around the enterprise. It promotes interoperability, including amongst different computing solutions and platforms both horizontally and vertically in the enterprise. It therefore cuts costs, speeds development and promotes increased operating efficiency.*[a]

Current OPC standardization efforts are moving away from the original focus on Microsoft protocols, with a distinct trend toward web-based protocols that are independent of any particular operating system. Most currently installed OPC equipment remains based on technologies specific to the Microsoft Windows environment. This paper will focus on those technologies and protocols.

1. TECHNOLOGY BACKGROUND: WHAT IS IT AND WHY DO WE CARE?

OPC was originally known as OLE for Process Control, as the first OPC standards were based on Microsoft's OLE technology. For the purposes of this paper, the history of this technology begins with the Component Object Model (COM). This was Microsoft's approach to distributed object technology—implementing object-oriented interfaces across different processes or different computers. COM is a client-server technology that provides a set of interfaces allowing clients and servers to communicate within the same computer.

DCOM is COM distributed across different computers. That is, a client program object can request services from server program objects on other computers in a network. COM and DCOM were originally Microsoft-specific technologies. The analogous or competing technology in the UNIX world is CORBA (Common Object Request Broker Architecture). DCOM implementations are also available for major UNIX platforms.

---

a. http://www.opcfoundation.org

DCOM uses the RPC protocol to request services from COM servers on different computers. RPC includes capabilities for Microsoft message queueing (MSMQ) and asynchronous communications.

COM and DCOM are the underlying protocols normally used to implement OLE. OLE is a Microsoft technology for compound documents. A compound document is something like a display desktop that can contain visual and information objects of all kinds: text, calendars, animations, sound, motion video, 3-D, continually updated news, controls, and so forth. Each desktop object is an independent program entity that can interact with a user and also communicate with other objects on the desktop.[b] For a simple example, consider a Microsoft Word document that contains an embedded Excel spreadsheet object. Through the use of OLE, the user can edit the Excel spreadsheet without leaving Microsoft Word: spreadsheet commands are passed through an OLE interface to an Excel OLE automation server. If instead of a Word document, a programmer wanted to write a custom program that included an Excel spreadsheet embedded in the user interface, the programmer simply uses the standard OLE interfaces provided by Excel.

The utility of OLE for control system development is apparent. Conceptually, the Human-Machine Interface (HMI) display is a compound document that contains individual display objects for specific Programmable Logic Controllers (PLCs), Remote Terminal Units (RTUs), etc. These end devices are represented by COM server objects that know the specifics of how to communicate with the end device. By embedding these objects in the display and programming to their standard Application Programming Interfaces (APIs), parameter display information can easily be obtained, control actions can be passed from the HMI to the end devices, endpoint alarms can be received, etc., yet the developer does not need to write specific interfaces to each and every type of device (and protocol) on the system. Thus, a SCADA system HMI can easily communicate with a wide variety of devices from different manufacturers. Further, the use of OLE interfaces is extremely common and well-supported in the world of Microsoft application developers. It is easy to implement OLE automation using standardized tools such as Visual Basic and Visual C++.

2. OPC IN THE CONTROL SYSTEMS MARKET

The OPC Foundation (http://www.opcfoundation.org) is an industry organization dedicated to ensuring interoperability in automation by creating and maintaining open specifications that standardize the communication of acquired process data, alarm and event records, historical data, and batch data to multi-vendor enterprise systems and between production devices. Production devices include sensors, instruments, PLCs, RTUs, Distributed Control Systems (DCS), HMIs, historians, trending subsystems, alarm subsystems, and more as used in the process industry, in manufacturing, and in acquiring and transporting oil, gas, and minerals.

b. This description is taken from http://searchwin2000.techtarget.com/sDefinition/0,,sid1_gci214126,00.html

The original OPC specifications addressed movement of real-time data from PLCs, DCS, and other control devices to HMIs and other display clients. The OPC Foundation has since developed specifications for handling alarm and event data, access to data historian functions, server-to-server protocols, etc. There is work in progress to develop specifications for web services, Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), and others. These latter efforts reflect the trend to web-based architectures that are less dependent on a specific operating system environment.

As of October 2005, the OPC Foundation had more than 360 member companies. According to a survey article in Control Engineering magazine,[c] 53% of HMI system installations use OPC for communications. All of the major HMI vendors include support for OPC protocols. In addition to Original Equipment Manufacturer (OEM) equipment vendors, there are third-party companies in the business of providing OPC interfaces. The largest third-party developer, MatrikonOPC (http://www.matrikonopc.com), has more than 500 specific OPC interface products.

3. POTENTIAL FOR VULNERABILITIES IN OPC

3.1 VULNERABILITIES IN UNDERLYING SERVICES

As discussed earlier, the original OPC data access standards are founded on base capabilities provided by OLE, COM and DCOM, and RPC services. These services are commonly used features in a Microsoft Windows enterprise environment. As such, they are also common targets for security attacks. Table 1 lists some of the more severe vulnerabilities involving RPC and DCOM services.

Table 1. Selected RPC and DCOM vulnerabilities.

Published Vulnerability	Description
CVE-2002-2077	DCOM information leak in Windows 2000 pre-SP3; remote attacker can obtain sensitive information.
CVE-2003-0352	A RPC DCOM buffer overflow; remote attacker can execute arbitrary code (exploited by the Blaster and Nachi worms).
CVE-2003-0528	A RPCSS DCOM buffer overflow; remote attacker can execute arbitrary code.
CVE-2003-0605	Remote DoS leading to local privilege escalation. Windows 2000 SP3 and SP4. Patched in MS03-039.
CVE-2003-0715	A RPCSS DCOM buffer overflow; remote attacker can execute arbitrary code.

c. http://www.manufacturing.net/ctl/article/CA6255302

Published Vulnerability	Description
CVE-2003-0807	Buffer overflow in COM internet services and RPC over HTTP proxy. Patched in MS04-012.
CVE-2003-0813	RPC DCOM denial of service. Patched in MS04-012. Problems with MS03-039.
CVE-2004-0116	RPCSS DCOM activation; denial of service. Patched in MS04-012.
CVE-2004-0124	The DCOM RPC interface for Microsoft Windows NT 4.0, 2000, XP, and Server 2003 allows remote attackers to cause network communications via an "alter context" call that contains additional data; also known as the "Object Identity Vulnerability." High severity; remotely exploitable. Patched in MS04-012.[d]

These RPC and DCOM vulnerabilities are not specific to control system networks. However, a control system network that is using OPC is vulnerable to these threats. Control system network administrators must mitigate these threats by keeping current with patches and service packs, or applying other security measures.

From the perspective of a knowledgeable attacker, knowing that OPC is being used implies that these underlying services are also present. The attacker could try known attack methods against these services to gain system access.

3.2 ISSUES SPECIFIC TO OPC

For control systems networks, it is typically not feasible to blindly apply vendor service packs and patches. Asset owners must carefully test these to ensure that they do not interfere with specific capabilities that are unique to control systems, as distinct from conventional IT networks. Due to the difficulty of testing and deploying vendor service packs and patches, and the time required for these activities, control systems are therefore more likely to have unpatched vulnerabilities for which there are known exploits.

One example is Windows XP Service Pack 2. If this service pack is installed with default settings, OPC over DCOM will cease to work.[e] Using the default settings, Windows Firewall blocks traffic created by OPC callbacks (where the OPC client becomes a DCOM server, and vice versa). The recommended fix is to add all OPC client and server machines to the exception list, or perhaps turn off the firewall entirely (if appropriate within the network). The firewall must allow incoming Transmission Control Protocol (TCP) port 135, and some COM security

d. http://www.microsoft.com/technet/security/bulletin/MS04-012.mspx

e. http://www.opcfoundation.org/Archive/1adcdca3-f6bb-48ca-8cc7-37403baf6b58/White%20Papers/OPC,%20DCOM%20and%20Security.pdf

settings must be modified from their defaults. Note that a particular installation might have assigned services to port numbers other than the typical defaults; these port numbers might also be blocked by the default firewall settings in Service Pack 2.

Since OPC client applications are inherently distributed applications, they also involve security and authentication mechanisms. They rely on standard Windows RPC mechanisms for security controls. Access Control Lists are used for OLE components in the same manner as for files and directories. As of 1998, "…most OPC client vendors… implemented browsing for remote OPC Servers by using the Registry API's that support access to a remote machine's registry."[f] Allowing remote registry access gives potential attackers an additional means of entry, particularly if the access control lists are not configured for maximum security. There are efforts underway to resolve this security issue by developing alternate browsing mechanisms.

Similarly, *opcenum.exe*, a commonly used tool for browsing available OPC servers on the network, requires that anonymous login be enabled. Some other OPC clients and servers also have this requirement. The use of anonymous login accounts presents an obvious security concern.

4. CONCLUSION

Control systems using OPC technology inherit the vulnerabilities associated with the underlying RPC and DCOM services in Microsoft Windows environments. These services have been a source of many severe vulnerabilities; exploits against unpatched systems are widely available. The system configuration required for OPC usage precludes the automatic application of security patches and service packs for the operating system. Asset owners should remain aware of the vulnerabilities presented by unpatched systems using these services, and ensure that other appropriate security controls are in place when conventional patches cannot be applied.

f. http://www.opcfoundation.org/Archive/1adcdca3-f6bb-48ca-8cc7-37403baf6b58/White%20Papers/OPC,%20DCOM%20and%20Security.pdf