OPC Security White Paper #2
OPC Exposed

byres research
intrinsically secure

po box 178
#5 – 7217 Lantzville rd
lantzville, bc
canada v0r 2h0
office 250.390.1333
fax 250.390.3899
www.byressecurity.com

Digital Bond

suite 130
1580 sawgrass corp pkwy
sunrise, FL 33323
office 954.315.4633
www.digitalbond.com

PREPARED BY:
Digital Bond
British Columbia Institute of Technology
Byres Research

November 13, 2007
OPC Security WP 2 (Version 1-3c).doc

Revision History

Revision	Date	Authors	Details
0.7	May 15, 2006	E. Byres, M Franz,	Draft internal review version
1.0	May 31, 2006	E. Byres, J. Carter, M Franz	Draft for controlled public review
1.1	August 31, 2006	E. Byres, M. Franz	2nd Draft for controlled public review
1.2	February 9, 2007	E. Byres, D. Peterson	3rd Draft for controlled public review
1.3	May 16, 2007	E. Byres, D. Peterson	Public Release Version
1.3a	June 8, 2007		Typo fixed in Section 2.5.4 and added required vulnerability
1.3b	August 27, 2007		Minor grammatical errors corrected
1.3c	November 13, 2007		Grammatical error corrected in

Acknowledgements

The Group for Advanced Information Technology (GAIT) at the British Columbia Institute of Technology (BCIT), Digital Bond, and Byres Research would like to thank all the vendors and end users that generously supported our efforts through numerous interviews and by providing us with documents that could only be described as extremely sensitive. Unfortunately we can not name you for obvious security reasons, but we appreciate your time, trust and encouragement.

Several people stood out in their contributions and advice for this document that we would like to acknowledge. First are Bill Cotter of MSMUG and Chip Lee of ISA - we thank you for all your help in making the user surveys possible. We would also like to thank Ralph Langner for providing the four example scenarios for this report and lots of useful information on OPC vulnerabilities.

Finally we would like to thank Evan Hand, formerly of Kraft Foods Limited, for his vision and support. Without him, this project never would have been possible.

Disclaimer

Deployment or application of any of the opinions, suggestions or configuration included in this report are the sole responsibility of the reader and are offered without warrantee of any kind by the authors.

Table of Contents

Executive Summary

In recent years, Supervisory Control and Data Acquisition (SCADA), process control and industrial manufacturing systems have increasingly relied on commercial Information Technologies (IT) such as Ethernet™, Transmission Control Protocol/Internet Protocol (TCP/IP) and Windows® for both critical and non-critical communications. This has made the interfacing of industrial control equipment much easier, but has resulted in significantly less isolation from the outside world, resulting in the increased risk of cyber-based attacks impacting industrial production and human safety.

Nowhere is this benefit/risk combination more pronounced than the wide-spread adoption of OLE for Process Control (OPC). OPC is increasingly being 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 applications has proven to be a challenge for most engineers and technicians. While OPC is an open protocol with the specifications freely available, engineers must wade through a large amount of very detailed information to answer even the most basic OPC security questions.

To address this need for security guidance on OPC deployment, a joint research team with staff from BCIT, Byres Research and Digital Bond were commissioned by Kraft Foods Inc. to investigate current practices for OPC security. The results of this study were then used to create three white papers that:

1. Provide an overview of OPC Technology and how it is actually deployed in industry
2. Outline the risks and vulnerabilities incurred in deploying OPC in a control environment
3. Summarizes current good practices for securing OPC applications running on Windows-based hosts.

The white paper you are now reading is the second of the three. In it we detail the vulnerabilities typically found in OPC hosts, based on OPC's current architecture (such as the use of DCOM) and the typical underlying operating system. We also investigated common misconfiguration vulnerabilities found in OPC server or client computers both at the operating system and OPC application level. Finally, using the vulnerabilities uncovered, we discuss four possible risk scenarios for OPC-based attacks.

This small sample of scenarios suggests several interesting conclusions. First, they highlight the fact that attacking OPC deployments does not require special skills or esoteric process controls knowledge. All the tools and

information needed to carry out attacks can be downloaded from the Internet.

The second conclusion is that two core vulnerabilities, namely excessively open firewalls and overly permissive DCOM access rights, lay at the heart of many scenarios. If either vulnerability is addressed, then the chance of these scenarios occurring is significantly reduced. What is especially interesting is that these vulnerabilities could be considered within the control of the knowledgeable OPC end user.

Finally, since the typical OPC host configuration is strongly influenced by the guidance provided by the software vendor, we discuss the quality of installation utilities and guidance provided to end-users by the OPC vendor community. In general we find that the guidance from vendors on OPC security could be significantly improved.

The good news is that there are operating system hardening practices that are well proven in the IT security community which we believe can be adopted by the controls community to significantly reduce these risks. In addition there are a number of DCOM specific security settings that can also be applied by the knowledgeable end-user. We will discuss these solutions in detail in our final report in this series, *OPC Security White Paper #3 – Hardening Guidelines for OPC Hosts*.

1 Introduction

This report is the second of three white papers outlining the findings from a study on OPC security conducted by Byres Research, Digital Bond and the British Columbia Institute of Technology (BCIT). The objective of this study was to create a series of simple, authoritative white papers that summarized current good practices for securing OPC client and server applications running on Windows-based hosts. The full study is divided into three Good Practice Guides for Securing OPC as follows:

- **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.
- **OPC Security White Paper #2 – OPC Exposed:** What are the risks and vulnerabilities incurred in deploying OPC in a control environment?
- **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?

All three white papers are intended to be read and understood by IT administrators and control systems technicians who have no formal background in either Windows programming or security analysis.

1.1 The Issues

In recent years, Supervisory Control and Data Acquisition (SCADA), process control and industrial manufacturing systems have increasingly relied on commercial information technologies (IT) such as Ethernet™, TCP/IP and Windows® for both critical and non-critical communications. The use of these common protocols and operating systems has made the interfacing of industrial control equipment much easier, but there is now significantly less isolation from the outside world. Unless the controls engineer takes specific steps to secure the control system, network security problems from the Enterprise Network (EN) and the world at large will be passed onto the SCADA and Process Control Network (PCN), putting industrial production and human safety at risk.

The wide-spread adoption of OLE for Process Control (OPC) standards for interfacing systems on both the plant floor and the business network is a classic example of both the benefits and risks of adopting IT technologies in the control world. OPC is an industrial standard based on the Microsoft Distributed Component Object Model (DCOM) interface of the RPC (Remote Procedure Call) service. Due to its perceived vendor-neutral position in the industrial controls market, OPC is being increasingly used to interconnect

Human Machine Interface (HMI) workstations, data historians and other servers on the control network with enterprise databases, ERP systems and other business-oriented software. Furthermore, since most vendors support OPC, it is often thought of as the one of the few universal protocols in the industrial controls world, adding to its widespread appeal.

Many readers will be aware that the OPC Foundation is developing a new version of OPC (called OPC Unified Architecture or OPC-UA) that is based on protocols other than DCOM[1]. This is in conjunction with Microsoft's goal of retiring DCOM in favour of the more secure .NET and service-oriented architectures. Once most OPC applications do make this migration from the DCOM-based architecture to NET-based architecture, industry will have the opportunity for far better security when it comes to OPC, but also a new set of risks.

Unfortunately, based on our experience in the industry, it may be a number of years before many companies actually convert their systems. So since DCOM-based OPC is what is on the plant floor today and will continue to see use for years to come, we focused our investigation on how to secure this type of OPC.

Our initial research showed two main areas of security concern for OPC deployments. The first (and most often quoted in the popular press) is that the underlying protocols DCOM and RPC can be very vulnerable to attack. In fact, viruses and worms from the IT world may be increasingly focusing on the underlying RPC/DCOM protocols used by OPC, 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 and against the Windows LSASS (Local Security Authority Subsystem Service). These seem to be the current favorites for virus and worm writers, and we expect this trend to continue."*[2]

At the same time, news of the vulnerabilities in OPC are starting to reach the mainstream press, as seen in the March 2007 eWeek article entitled "*Hole Found in Protocol Handling Vital National Infrastructure*"[3]. Thus, the use of OPC connectivity in control systems and servers leads to the possibility of DCOM-based protocol attacks, disrupting control systems operations.

---

[1] See Whitepaper #1, *Section 5.7: OPC Unified Architecture* for more information on OPC-UA.
[2] Bruce Schneier, "Attack Trends" QUEUE Magazine, Association of Computing Machinery, June 2005
[3] Lisa Vaas, "Hole Found in Protocol Handling Vital National Infrastructure" eWeek, http://www.eweek.com/article2/0,1759,2107265,00.asp, March 23, 2007

Despite all these concerns, it is our belief that the most serious issue for OPC is that configuring OPC applications securely has proven to be a major challenge for most engineers and technicians. Even though OPC is an open protocol with the specifications freely available, users must wade through a large amount of very detailed information to answer even basic security questions. There is little direct guidance on securing OPC, and our research indicates that much of what is available may actually be ineffective or misguided.

All things considered, there is little doubt that some clear advice would be very useful for the control engineer on how best to secure currently deployed, COM/DCOM-based OPC systems. This series of white papers aims to help fill that gap for the end-user.

1.2 Organization of OPC White Paper Series

As noted earlier, this is the second of three white papers outlining the findings and recommendations from a study on OPC security. In White Paper #1 we reviewed the OPC specifications, focusing on details that are relevant from a security point of view and might be useful to users wishing to understand the risks of OPC deployments. We then described the real-world operation of OPC applications, identifying components that need to be understood to harden hosts running OPC client and server applications. In White Paper #3, our final white paper, we will use this information to give the OPC end-user a series of practical recommendations they can draw on to secure their OPC host machines.

Before one can provide security recommendations, it is important to clearly define the security risks faced when deploying OPC. Thus in this second white paper we define a set of vulnerabilities and possible threats to OPC hosts, based on OPC's current architecture (i.e. the use of DCOM). We also look at common misconfiguration vulnerabilities found in OPC server or client computers both at the operating system and OPC application level. Finally, since the typical OPC host configuration is strongly influenced by the guidance provided by the software vendor, we looked at the quality of configuration utilities and guidance provided to end-users by the OPC vendor community.

1.3 Study Methodology

Developing the findings and recommendations for all three of the white papers required the following four-phase approach to the study:

1. Data Gathering

    - Conducting user surveys and collecting information on OPC deployments in order to get a representative sample of how actual

OPC deployments were configured in the field by our target audience.

- Reviewing OPC Foundation and vendor configuration guidelines.
- Conducting a literature search for OPC-related papers and guidelines.

2. Ascertaining potential threats and vulnerabilities in OPC systems
   - Identifying what operating system configuration issues exist in typical OPC deployments.
   - Identifying what OPC, RPC and DCOM issues exist in typical OPC deployments.
3. Creating recommendations for mitigating potential threats and vulnerabilities
   - Determining what could be done to secure the underlying operation system without impacting the OPC functionality.
   - Determining what could be done to secure RPC/DCOM components in an OPC host.
   - Determining OPC-specific client and server security configurations.
4. Testing the security recommendations
   - Lab testing all recommendations in a typical OPC environment and modifying our recommendations accordingly.

1.4 Limitations of this Study

It is important to understand that this report is not intended to be a formal security analysis of OPC or DCOM, but instead is a set of observations and practices that will help end-users secure their OPC systems. As well, this report is focused only on securing the host computers that are running OPC. Securing the network OPC operates over is an interesting and important area of research, but is beyond the scope of this report. A follow-on study is planned to investigate these network security aspects and consider solutions for OPC/DCOM in the network infrastructure, including firewall rule-sets and analysis of third party OPC tunnelling solutions.

Finally, we cannot guarantee that following our recommendations will result in a completely secure configuration. Nor can we guarantee that these

recommendations will work in all situations; some modifications may be required for individual OPC client and server applications or Microsoft Windows network deployments. However, we are confident that using these guidelines will result in more secure systems as compared to the typical default application and operating system settings we have seen in our investigations.

2 Threats & Vulnerabilities for OPC Host Systems

To assess the security risks in deploying OPC, it is important to define the set of inherent vulnerabilities, the threats that may exploit them and the possible impact of compromised software components. Only when we understand these threats can we begin to determine what are reasonable methods for defending against them.

For the purpose of this study, we defined a *vulnerability* as a design, implementation or configuration flaw that could result in the loss of confidentiality, integrity or availability of either the OPC application or the underlying operating system. Vulnerabilities may occur at all layers of the system, starting with the operating system and progressing upwards to the actual configuration of the OPC application.

Closely related to vulnerabilities, we define *threats* as exploits, tools, or human agents that could compromise the security of an OPC application or the underlying operating system. In terms of impact we focus on the effects of a compromise, including impacts such as denial of service (DoS), unauthorized alteration of data and possible negative effects on the physical process being controlled.

In this study we focused on two distinct types of vulnerabilities:

- *High risk operating system vulnerabilities* - common platform or operating system vulnerabilities that could adversely affect the security of the OPC application. Since this is such a broad topic, we will further focus on a subset of the operating system threats and vulnerabilities that are most critical to the security of the OPC servers.

- *Inherent OPC vulnerabilities* - weaknesses due to the design of the protocols underlying the current OPC specifications as well as implementation choices made by vendors, such as DCOM configuration settings.

Since OPC security (particularly authentication and authorization) is so tightly bound to operating system security, this division may seem somewhat awkward. In many cases, both the problems and solutions are related. However, by approaching the problem from two different angles — in other words, by considering the ways that a poorly configured OPC application could adversely affect the security of the underlying operating system and vice-versa — we not only make the problem more manageable, but it allows easier and more logical selection of countermeasures.

2.1 Underlying System Vulnerabilities on OPC Hosts

To date, the most common threats to control systems have been indirect, targeting the underlying operating system or network infrastructure rather than the control system devices and protocols themselves. Generally, control system applications or devices have suffered more as collateral damage from malware rather than from directed attacks. In other words, compromising the host computer's operating system and its related applications is the common attack path for an attacker to take and will likely continue to be in the foreseeable future.

Thus we start our list of "OPC Host Vulnerabilities" by describing the most critical operating system exposures that need to be addressed by OPC administrators. Since OPC is almost exclusively run on a Windows platform some of these vulnerabilities will be Windows specific, but many apply to other operating systems as well. Finally, since Microsoft has significantly improved the overall security posture of its operating system in recent releases (XP SP2 and Server 2003 in particular), many of these vulnerabilities are more critical for older OPC systems running on Windows 2000 or NT4.

Most of these problems are well known among IT administrators, but less understood in the control system world. Furthermore, due to their potential impact, these issues are important enough for us to revisit within the context of OPC Host security.

For each of the following vulnerabilities, we describe the root cause, the consequences of exploitation, and in some cases, introduce high level remediation techniques. Specific guidance to address each of these issues is provided in White Paper #3.

2.1.1 Unnecessary System Services

One of the basic security principles is that a single-purpose device with a limited set of tightly controlled functions is far easier to secure than a multipurpose system. This is not only because the multipurpose system provides more services that can be potentially exploited, but also because it is a single point of failure.

Since in most cases OPC servers will directly monitor and control process data, it follows that the computer containing the OPC server should not also be used as a file, print, or web server. For example, if a web server is needed to allow employee access to OPC data, it is far better to have a separate web server host that retrieves data from the OPC server host (via a mechanism such as ODBC or OPC-XML). Using a web server that is integrated into the same OPC host that has direct access to controllers or I/O devices is asking for trouble.

Older versions of Windows NT/2000 Server enabled many unnecessary services by default, but Windows XP/2003 does a much better job of reducing these potential vectors for attacks. As noted in White Paper #1, OPC servers and DCOM have surprisingly few dependencies on other Windows services, so there is little need for many of the default Windows services such as File and Print Sharing or NetBIOS over TCP. Thus these services should be disabled unless there is a pressing need for them in the control strategy.

Although the practice is not widespread, some IT application vendors not only ensure that their application exposes the smallest possible attack surface, but also disable unnecessary functionality and tighten access controls during the installation process. We would like to see all OPC vendors and automation software vendors consider adding hardening scripts to their products. This will minimize unnecessary system services enabled after a successful installation.

2.1.2 System Enumeration and Profiling

Regardless of whether the threat agent is a human or an automated piece of malware, targets must be discovered before they can be attacked. Unless prior knowledge of the system is available to the attacker, reconnaissance is typically the first step in any attack sequence. Most reconnaissance efforts consist of active probes of the targeted network or system to gain information about its identity, capabilities and potential weaknesses. System profiling takes advantage of the fact that, by default, most applications and services provide excessive information to unauthenticated parties. Windows and OPC are no exception.

There are a number of well known tools or tools suites that allow the attacker to collect this useful information. We will describe some of the most common below.

2.1.2.1 Traffic Sniffing

Once an attacker has identified a given device, computer or an "interesting" process control network, it is possible to gather information about the victim based on passive or indirect means that are unlikely to be detected on the targeted system even if monitoring and logging is in place. This is typically accomplished through simple traffic capture tools that can analyze the nature of traffic to and from a target device.

Windows networks are particularly chatty and divulge information about end hosts through broadcast traffic that is visible to all members of the LAN. Any information sent in NetBIOS name service broadcasts can be captured and harvested for attacks.

2.1.2.2 Domain Name System (DNS) Queries

Informational gathering methods using DNS are commonly considered attacks originating from the Internet, but similar information can be retrieved by querying internal DNS servers if systems do not deploy a "split DNS."

Since it is a common practice to include functional information, operating system type, or the owner's name in DNS address records, DNS queries can be a valuable source of information. By either using tools to resolve a range of host IP addresses or by directly querying a DNS server with a zone transfer, an attacker can often quickly identify critical nodes such as an OPC server.

If name resolution is enabled then the names of internal systems can be obtained by simply conducting a ping sweep of the network. While this may seem innocent enough, it can often be used for highly inappropriate purposes. For example, recently one of the authors used this technique during a site security assessment to quickly identify the Ethernet-Serial Gateways and IP video cameras used in a physical security system by noting the presence of "security" in their host name. For a real attacker a simple next step might have been to launch a denial of service attack against these devices in order to disable the site's video surveillance capabilities.

2.1.2.3 Active Directory/LDAP Queries

Depending on the configuration, authorized domain members may be able to extract a significant amount of information about other domain members by sending Lightweight Directory Access Protocol (LDAP) queries to a domain controller. If the "Guest" user is enabled (a requirement for legacy NT4 services) even unauthenticated users may be able to gain access to this information.

2.1.2.4 TCP/UDP Scanning

Following passive information gathering techniques, attackers typically begin actively probing a device's TCP or User Datagram Protocol (UDP) services to determine whether common applications are running. This is typically known as "port scanning" and can be extremely effective in locating basic services. For example, a standard port scan allows one to determine the probable presence of DCOM by checking whether TCP Port 135 is open and listening. Since this does not provide any information about OPC or even whether an OPC server is present on a given host, attackers may also send valid messages to the application layer to identify specific services.

2.1.2.5 NetBIOS Enumeration - Domains and Workgroups

There are a number of built-in tools on any Windows system that will allow an attacker to determine which domains or workgroups are present on a given network or host. In Figure 2-1, we notice that the administrator's username

(OPCADMIN) and the workgroup name (OPCGROUP) are revealed by a Network Basic Input Output System (NetBIOS) Name Service scan.

If NetBIOS cannot be disabled, turning off the Alerter and Messenger services can prevent disclosure of this information (Note: starting with Windows XP Service Pack 2, both of these services are set to Disabled by default).

```
# nbtscan -v 192.168.68.60
Doing NBT name scan for addresses from
192.168.68.60

NetBIOS Name Table for Host 192.168.68.60:

Name             Service          Type
----------------------------------------
CLEANW2KSP4    <00>             UNIQUE
CLEANW2KSP4    <20>             UNIQUE
OPCGROUP       <00>              GROUP
OPCGROUP       <1e>              GROUP
CLEANW2KSP4    <03>             UNIQUE
OPCADMIN       <03>             UNIQUE
```

Figure 2-1: NetBIOS Name Service Scan

2.1.2.6 SMB Enumeration Using Anonymous Login

Built into the Common Internet File System (CIFS)/Server Message Block (SMB) protocol are APIs that allow extensive information about servers to be discovered, even to unauthenticated users. Since Windows NT/2000 allows anonymous connections by default (but not XP/2003, unless it is enabled), these older operating systems are particularly susceptible to this reconnaissance attack. The following is some of the information that can be harvested using this technique:

- **Account Information**: Using the anonymous login an attacker can make queries to identify userid, username, account description, login and password activity using operating system utilities and free security tools. Based on this information, attackers can then begin the search for weak passwords.

- **Policy Settings**: Besides account information, policy attributes can be determined that allow an attacker to discover when a given account requires a strong password or has account lockout enabled.

- **File Shares**: Armed with a list of accessible shares, an attacker can begin to browse the file system and attempt to retrieve interesting

files and use writeable directories to upload tools and gain interactive system access. Depending on the permissions, it may also be possible to modify OPC configuration files.

- **Domain Trust Relationships:** Although OPC is most commonly used within a single domain, unauthenticated users can identify domain trust relationships that not only can be used to identify additional attack vectors against the OPC server, but also to allow the attacker to launch attacks from a compromised server to other systems.

2.1.2.7 RPC Endpoint Discovery

Information about RPC endpoints is available by default. However, based on our testing, Windows or Linux RPC scanning tools could not identify the presence of the OPC Program IDs. This situation may be changing rapidly as a number of OPC scan tools have been noted on the Internet in the past few months.

2.1.2.8 Network Management Applications

The Simple Network Management Protocol (SNMP) is a commonly used method for managing and monitoring a variety of network devices including switches, routers, servers, and even some PLCs. Although SNMP is not enabled by default on most Windows server installations, if it is enabled and a predictable community string is in place, the Microsoft MIB (Management Information Base) exposes a significant amount of information that could be used to compromise the confidentiality (and possibly the integrity) of an OPC Server.

The information that can be gained by browsing the SNMP MIB includes: running services, file share names and paths, usernames, domain names, hardware information, and more. All of this could provide useful information to exploit an OPC Server or the operating system. Besides SNMP, other system management software, such as Microsoft Terminal services, if not adequately secured, may result in additional exposures to OPC servers.

2.1.3 Password Vulnerabilities

Given OPC's reliance on Microsoft authentication mechanisms, weak passwords are among the most critical vulnerabilities that can undermine the security of an OPC server. The poor selection of passwords is the tipping point for the security of any application, operating system, or device. This is especially true for OPC/DCOM authentication which uses local or domain/Active Directory credentials for authentication.

For ease of administration, it may make sense to use the term “OPC” as part of the username or group name, but “OPC”, the organization, vendor, or

product name, or process control information should never be used in the actual passwords. Having strong, difficult to guess passwords are especially critical since security techniques that limit password guessing attacks (such as account lockout based on failed authentication attempts) may be inappropriate for the industrial controls environment. Further guidance on password selection and management can be obtained from papers such as the paper "*Password Memorability and Security: Empirical Results*"[4]

2.1.4 Inadequate Logging

By default, Windows 2000/XP auditing settings do not record DCOM connection requests, SMB logins, or attempts to access system objects. Unless these settings are enabled, it is impossible to determine whether access violations have occurred, or the source of the intrusion. In White Paper #3 we define a Windows audit configuration for OPC servers designed to mitigate this vulnerability.

2.1.5 Patching and Updates

The past several years have made users and vendors keenly aware of the need to patch operating systems and applications. This topic is only included for the sake of completeness, and we assume that most OPC users and vendors have developed effective patching procedures. For those readers who do not currently have a good patch management process in place we suggest contacting your control system vendor or referencing the GAO report "*Information Security: Agencies Face Challenges in Implementing Effective Software Patch Management Processes*"[5], and the Edison Electric Institute's "*Patch management Strategies for the Electric Sector*".[6]

2.1.6 Use of Weak Authentication Mechanisms

Although NT4 Service Pack 4 (SP4) and Windows 2000/XP provide robust authentication and authorization mechanisms, for compatibility reasons insecure authentication mechanisms are still supported by clients and server as a default setting. Windows 2000 still accepts LanMan (LM) and NT LanMan (NTLM) authentication mechanisms, both of which have well-known weaknesses and are susceptible to quick offline attacks.

For example, a rogue server or other client-side attacks can exploit the default support of weak hashes to capture authentication credentials of

---

[4] Jeff Yan, Alan, Ross, Alasdair. "Password Memorability and Security: Empirical Results," IEEE Security and Privacy, vol.02, no.5, pp. 25-31, September-October 2004

[5] "Information Security: Agencies Face Challenges in Implementing Effective Software Patch Management Processes", GAO Report GAO-04-816T, US General Accounting Office, June 02, 2004

[6] "Patch management Strategies for the Electric Sector", White Paper, Edison Electric Institute –IT Security Working Group, March 2004

otherwise secure machines. In this type of attack, malicious HyperText Markup Language (HTML)/ Uniform Resource Locator (URL) is sent through email or a compromised web site can redirect a user to the rogue SMB server which is configured to support only weak hashes and can be used to gather credentials. For this reason we recommend that NTLMv2 authentication be required for all OPC communications.

2.1.7 Remote Registry Browsing

As the storehouse of configuration information, the Windows registry is a likely target of any attack, whether local or remote. Remote read-only access to the registry can divulge a significant amount of information that is useful for an attacker. In the case of OPC, it would allow an attacker to identify active OPC servers and other COM objects that are installed. This information will be disclosed even if anonymous logins are disabled.

Early versions of Windows 2000 allowed remote registry browsing by default. Furthermore, prior to the release of the OPC Server Browser, the registry was always queried by OPC Clients to determine which servers were present. The subsequent release of both Windows XP and OPC Server Browser means that this risk can be now reasonably mitigated.

2.1.8 Local Vulnerabilities

Although our primary focus is remote threats and vulnerabilities, if an attacker can gain access to a local domain, it provides a stepping stone to compromise OPC related files and processes. This is particularly important since the default DCOM and file system permissions are far too permissive, often allowing access to any user or the EVERYONE group on a system.

In order to conduct these attacks, INTERACTIVE access is usually required, which is the default for DCOM servers including OPC. In White Paper #3 we will identify a more appropriate set of object permissions to provide multiple layers of defense, including removing the OPC server process from the INTERACTIVE group.

2.2 OPC Related Vulnerabilities

Building on the vulnerabilities found in the underlying operating system, we now discuss specific flaws in OPC servers and clients. Many of the inherent vulnerabilities in OPC from both architecture and the vendor implementation are sins of omission rather than known design errors.

2.2.1 Use of Historically Insecure Transport

Throughout their history, RPC implementations have had a poor security record on both Windows and non-Windows platforms alike. Since the currently deployed versions of OPC are based on DCOM (which is based on

RPC), the OPC host will be vulnerable to all pre-authentication flaws in RPC, such as those exploited by the Blaster worm in 2003[7]. These in turn could result in the compromise of services, execution of arbitrary code, or denial of service attacks against the OPC host. Only aggressive patching of OPC hosts or blocking of all OPC traffic through firewalls can mitigate this risk.

Adding to the problem is the fact that the default (and most common) configuration for RPC uses dynamic ports, which makes it more difficult to write effective firewall rules. As we will discuss in White Paper #3, this issue can be addressed, but the solution does add additional configuration complexity for the system administrator.

In the future, the importance of these RPC issues should fade as Web services-based OPC implementations (i.e. OPC-UA) are built using type safe programming languages (such as C# and Java) with code access security. Unfortunately at the present time almost all OPC users must still deal with the vulnerabilities inherent in RPC and DCOM.

2.2.2 Lack of Authentication in OPC Server Browser

Configuration guidance from many vendors, as well as the OPC Foundation's configuration guidance for XP-SP2[8], recommends allowing remote Anonymous Login so OPCEnum will work when DCOM Authentication is sent to "None". If a buffer overflow of some type were discovered in the OPC Server Browser code, the consequences could be arbitrary code execution or denial of service against any computer running the OPC Server Browser. Fortunately we are unaware of any such buffer overflow in the OPC Server Browser code at this time.

Certain OPC clients rely on OPCenum.exe to get the CLSID of the OPC server. If OPCEnum will not function in these servers, the OPC client will be unable to determine the servers CLSID and therefore be unable to connect to the OPC server. Several researchers have demonstrated techniques to remotely disable OPCEnum, indicating that this could be used as a possible denial of service attack vector.

2.2.3 Overly Permissive Authorization Policy on OPC Server Browser

Many documents, including the OPC Foundation's XP SP2 Guidance[9], recommend that the "Everyone" group be added to the OPC permissions.

---

[7] http://www.microsoft.com/security/encyclopedia/details.aspx?name=Win32%2fMsblast

[8] OPC Foundation, Using OPC via DCOM with Microsoft Windows XP Service Pack 2, Version 1.1, http://www.opcfoundation.org/DownloadFile.aspx/Using OPC via DCOM with XP SP2 v1.10.pdf?RI=326

[9] To its credit, the OPC Foundation document does include a note stating "*it is often desirable to add these permissions to a smaller subset of users*". While we would have preferred to see this suggestion be included as the main text of the document, it is better than many of vendor documents we inspected.

This violates the basic principle of least privilege. If a local Guest user exists, then by allowing the "Everyone" group to access (and perhaps launch) OPC servers, a guest user without any password would be able to launch OPC servers and access them.

2.2.4 OPC Server and OPC Server Browser Assigned Excessive Privileges

Most of the OPC Servers we analyzed in our study had default installations that ran with SYSTEM level privileges, the most powerful account in Windows. This is a highly risky setting that offers little advantage to the user. In White Paper #3 we will discuss alternatives to this situation.

2.2.5 Unnecessary Protocol Support for OPC Server Browser

Default protocols transports for OPCEnum.exe include not only TCP/IP, but Sequenced Packet Exchange (SPX), NetBIOS Extended User Interface (NetBEUI), and Connection Oriented NetBIOS over InterNetwork packet Exchange (IPX). Some vendors even go so far as to recommend enabling COM Internet Services, an HTTP transport for DCOM.

The fact is that new vulnerabilities in overlooked protocols or applications are frequently discovered. Since there is little reason to support the other legacy protocols at this time, they should be disabled and only TCP/IP allowed. For example, there is an interesting note in Matrikon documentation[10] on how the use of datagram-based protocols may expose the servers to memory leaks. It turns out that there is a memory leak problem with older Windows systems that use UDP for DCOM[11]. Since few systems really need to support this protocol, it is far better if disabled.

2.2.6 Lack of Integrity of OPC Communications

The default DCOM settings do not provide message integrity checking. If the underlying network infrastructure is compromised and the attacker can sniff and inject traffic, it is likely that rogue messages could be injected once the client and server have authenticated during the initial connection establishment. A number of SMB "man in the middle" tools and techniques are available and it is likely that these could be modified or enhanced to conduct man in the middle attacks against OPC communication. For these reasons we recommend that if at all possible, Packet Integrity should be enabled as discussed in White Paper #3.

2.2.7 Lack of Confidentiality of OPC Traffic

Although DCOM supports message encryption, none of the OPC vendors we reviewed recommended enabling Packet Privacy for their OPC Server or the

[10] MODBUS/TCP OPC Server, MatrikonOPC Inc.
[11] http://support.microsoft.com/kb/294710/en-us

OPC Server Browser. However, some vendors recommend VPN tunnelling as a means of providing secure off-site access. The primary issue with Packet Privacy is that for some applications, the encryption may be too processor intensive unless hardware acceleration cards are installed to offload cryptographic tasks. In most configurations that we investigated in the lab, encryption did not cause a significant degradation of performance.

2.2.8 COM Internet Services Reliance on IIS

To overcome known issues with DCOM and firewalls, HTTP can be used as a transport alternative to tunnelling DCOM over a single TCP port. While a few companies reported using this technique, it is relatively rare and we generally do not recommend it. The COM Internet Services requires the deployment of Microsoft Internet Information Server (IIS) and thus introduces additional overhead and potentially a different set of vulnerabilities. Although the network security posture of IIS has improved, we believe that most of the benefits of this method are outweighed by the difficulty of hardening this complex, feature-rich application that itself has had numerous vulnerabilities.

2.2.9 OPC Security Configuration Lacks Fine Grained Access Control

Unfortunately there is no standardized method of having per-item access control for read-only versus read write access. Some OPC server vendors implement item-based security; however most only implement server-level permissions – i.e. they may have a server that only allows read-only access, and a different server that allows both reading and writing to tags.

2.3 Security Considerations for Specific OPC Specifications

2.3.1 Security Considerations for OPC-DA

OPC-DA provides an attacker with the ability to not only gain unauthorized read access to process control data, but to also modify data points on devices. Apart from providing erroneous values, attackers could inject messages into the client application and conceivably launch more subtle attacks by manipulating time stamps and quality metrics. Depending on requirements of the OPC client application, OPC-DA servers may be especially susceptible to denial of service attacks at multiple layers of the protocol stack.

2.3.2 Security Considerations for OPC A&E

OPC-AE will typically provide an attacker with less ability to directly control the physical process. However it provides direct access to another weak link, namely the human operator. If an attacker can degrade the operator's ability to view, respond, and acknowledge critical system events, other attacks may go unnoticed, or worse, the operator may take inappropriate action that negatively impacts the physical process. For example,

overloading an operator with alarms may undermine his trust in the system. It is also likely that the complexity of OPC A&E servers, given their use of callbacks and state machines, could make them more prone to implementation flaws than other OPC servers.

2.3.3 Security Considerations for OPC-HDA

The compromise of an OPC-HDA server will not likely have the immediate consequences of impacts on other OPC applications. However there still exist risks such as the disclosure of sensitive business information or the modification of data that is required for compliance purposes.

2.3.4 Security Considerations for OPC-DX

As OPC-DX communication is used for OPC server to server data exchange, and is often used to provide multi-vendor control system interoperability. The sensitivity of this data exchange can vary greatly based on whether the data is used to make control decisions.

2.3.5 Security Considerations for OPC XML-DA

Since OPC XML-DA does not use DCOM it has a different set of security concerns. The specification does not provide any application-layer security mechanisms, and current implementations must rely on HTTP/HTTPS security mechanisms (i.e. basic authentication and/or transport layer security). OPC XML-DA does not utilize the Web Services Security standards such as WS-Security.

2.4 A Very Brief OPC Threat Analysis

Although a formal threat analysis was out of scope for this study, it makes sense to briefly consider the objectives and methods that an attacker could use to compromise an OPC server. The following section briefly describes possible attacker objectives, attacker tools and techniques.

2.4.1 Attacker Objectives

A threat analysis methodology we have successfully used in previous projects is to develop a set of high level attacker objectives. For OPC these are likely to be:

- *Reconnaissance* - Identify OPC server
- *Confidentiality* - Gain unauthorized read access to OPC server data
- *Availability* - Denial of OPC server service
- *Integrity* - Alter OPC server data

- *Integrity* – Create rogue OPC server

2.4.2 Attacker Tools and Techniques

To achieve the above objectives, an attacker has a variety of techniques at their disposal:

- *Use of Freely Available OS Reconnaissance and Attack Tools* – Tools like nmap, netcat and many others are freely available to exploit any underlying system vulnerabilities that may be present on poorly configured OPC hosts. These typically are designed for remote deployment.

- *Use of Freely Available OPC Browsers* - Typically these tools require the attacker to be part of the same domain or workgroup in order to easily browse and alter the OPC objects.

- *Custom OPC Attack Tools* - to conduct a more fully automated attack, a sophisticated attacker could develop custom security tools or exploit a poorly configured server or implementation flaw. These tools are also becoming increasingly available for download on the Internet.[12]

2.5 Four Possible OPC Risk Scenarios[13]

All the above vulnerability and threat analysis details may leave the reader wondering "*So what does all this really mean to the security of my control system?"* To answer this, we provide four possible risk scenarios below. This is only a small subset of possible scenarios since, as noted earlier in this report, a full analysis was beyond the scope of the project (Additional OPC risk scenarios can be found in papers by Langner[14] and Mora.[15]).

2.5.1 Risk #1: Collateral Damage by OPC-Unaware Malware

In this first scenario, general purpose malware inadvertently impacts an OPC client or server running in a control environment. For this to occur, the combined required vulnerabilities include:

1. The control system firewalls (or other EN/PCN perimeter security measures) are configured to allow a significant number of open ports

[12] For example, the *OPC Server Interface Vulnerability Assessment Tool* can be downloaded from http://www.neutralbit.com/

[13] Special thanks to Ralph Langner for providing the four example scenarios for this report.

[14] Ralph Langner; "*OPC Exposed Part II",* S4 Conference, *Digital Bond Press*, Miami, FL, January 2007

[15] Lluis Mora; "*OPC: Interface Implementation Vulnerabilities*", S4 Conference, *Digital Bond Press*, Miami, FL, January 2007

to and from the OPC server platform because data is needed by applications in the business network (This could also be the result of lack of attention to security);

2. The DCOM access rights that are very permissive due to the installation script provided by the OPC vendor;
3. An OPC server running on a host platform running either an outdated operating system (such as Windows-NT, SP4) OR a current operating system with critical patches not applied (such as Windows-XP SP2).

Based on the OPC End-user survey noted in White Paper #1, vulnerabilities #2 and #3 are extremely common, occurring in a majority of end user deployments 51% and 53% of the time, respectively. As we will discuss in Section 3 of this report, we have good reason to believe that vulnerably #1 is also common in many real world OPC deployments.

The threat in this scenario is an average worm that exploits well-known RPC/DCOM vulnerabilities such as the MSBlaster worm did in 2003. The possible impact starts with the worm entering the organization via a business user on the enterprise network, traversing through the control system firewall, infecting systems with OPC servers and then spreading onto other systems in the automation network.

According to statistics obtained from the Industrial Security Incident Database (ISID), the Blaster worm incident caused at least 6 separate process security incidents in 2003 by following this general scenario. Using ISID extrapolation techniques, this translates into a conservative estimate of between 60 and 120 incidents in the North American control market in 2003. Typically, incidents of this type have resulted in loss of production in over 40% of the cases.

2.5.2 Risk #2: Accidental Shutdown of Control System by User

In this scenario, a benevolent but ill-informed corporate user is experimenting with OPC client applications or tools. By chance he/she connects to a SCADA system that has an integrated OPC server and a possible overload flaw. For this to result in an impact, the required combined vulnerabilities include:

1. The control system firewalls (or other EN/PCN perimeter security measures) are configured to allow a significant number of open ports to and from the OPC server platform because data is needed by applications in the business network;
2. The DCOM access rights that are very permissive due to the installation script provided by the OPC vendor;

3. A SCADA system with integrated OPC server that has a very large variable pool;
4. An OPC software vulnerability where a server browse results in crash of control system with large variable pools.

For the reasons noted previously, vulnerabilities #1 and #2 are common. As OPC grows in popularity and deployment scope, #3 is becoming increasingly common.

Vulnerability #4 may seem unlikely, but the fact is that an OPC item browse uses a lot of memory and has been shown to cause system crashes in the past. In one case an accidental OPC item browse on a system with approximately 40,000 variables resulted in the complete shut down of a major food processing plant. This particular system had a very large amount of processing power and memory - estimates by OPC expert Ralph Langner indicate that one can expect server breakdowns on average PC hardware with a variable pool in excess of 20,000 variables.

The impact from this scenario starts when the user browses the OPC server on the SCADA system from enterprise network and as a result crashes both the OPC server and the integrated SCADA system. Production is shut down and the root cause is unknown to the company. According to the survey results reported in White Paper #1, loss of OPC at this level will result in a loss of production in the majority of OPC deployments 54% of the time.[16]

2.5.3 Risk #3: Opportunistic OPC Denial of Service Attack

In this scenario, OPC-specific malware infects a company control system and causes shutdown of process operations. For this to result in an impact, the combined vulnerabilities required include:

1. The control system firewalls (or other EN/PCN perimeter security measures) are configured to allow a significant number of open ports to and from the OPC server platform because data is needed by applications in the business network;
2. The DCOM access rights that are very permissive due to the installation script provided by the OPC vendor;
3. An OPC software vulnerability that can be exploited to cause a denial of service in the OPC application and possibly in the computer hosting the OPC server;
4. A malware package that specifically exploits OPC is available.

[16] "OPC Security Whitepaper #1 - Understanding OPC and How it is Deployed", *Byres Research and Digital Bond*, April 2007, p. 12

Once again, vulnerabilities #1 and #2 are common place in the deployed OPC systems we surveyed. According to tests reported at the S4 conference in January 2007, of the 75 different OPC servers tested using freely available tools, 33% of these servers were vulnerable to basic DoS attacks[17]. This very limited research shows that many deployed OPC servers have vulnerability #3, and the number of vulnerable systems and variety of vulnerabilities is likely to grow as more security research is performed.

To date, vulnerability #4 has never been reported, but papers by both private experts and government security bodies indicate that narrow–focus, custom-built malware has become increasingly common in the past two years[18] [19]. The technology to create such a worm is readily available.

The steps to impact for this scenario are that the OPC malware enters the company via another vector (such as email) and begins to search for OPC targets. It detects any OPC servers on the control system from enterprise network and then attacks any vulnerable applications using the OPC vulnerabilities listed in CERT/CC vulnerability notes (for example see CERT/CC Vulnerability ID: VU#404833) or a zero-day vulnerability.

Once this scenario occurs, the OPC server will be unavailable and may require anything from a simple reboot to complete software re-installation and configuration to recover. The impact on production will be based on how the OPC server is used and if other critical control applications are on the same computer as the OPC server application.

2.5.4 Risk #4: Intelligent, Aggressive Attack against OPC Hosts

In this scenario, a skilled and aggressive attacker deliberately targets OPC software running on his target company's control system and causes a serious process upset using a man-in-the-middle (MITM) technique. For this to result in an impact, the combined vulnerabilities required include:

1. The control system firewalls (or other EN/PCN perimeter security measures) are configured to allow a significant number of open ports to and from the OPC server platform because data is needed by applications in the business network;
2. The DCOM access rights that are very permissive due to the installation script provided by the OPC vendor;

---

[17] Lluis Mora; "OPC: Interface Implementation Vulnerabilities", S4 Conference, *Digital Bond Press*, Miami, FL, January 2007

[18] Al Berg, "Seven trends to expect from virus and worm authors in 2006," Threat Monitor, January 4, 2006 (http://searchsecurity.techtarget.com/tip/1,289483,sid14_gci1155150,00.html)

[19] *"NISCC Briefing: Targeted Trojan Email Attacks,"* National Infrastructure Security Coordination Centre, London, UK, June 2005

3. Administrative rights allowing write access to ADMIN$, the registry and the Windows service control manager on the victim machine.

Once again vulnerabilities #1 and #2 are common. The third vulnerability is more demanding, but is not unreasonable to assume that anybody capable of carrying out a MITM attack as described below will be able to get administrative rights using standard rootkit software or by other means. Finally some means of access into the enterprise network is required.

The steps to implement the attack are:

- The attacker would first gain access to the enterprise network and then gain the required access privileges into the control system by taking advantages of the two vulnerabilities noted above.
- The attacker would then remotely install a stealth OPC server on a victim machine using DCOM, rename the registry entry for victim OPC server to point to the stealth OPC server and remotely install and execute utility program for killing the original OPC server.
- Finally the attacker would use the stealth OPC server to deliberately send misleading information to the operators to induce them to take inappropriate actions (this is known as *operator spoofing*). If desired the attacker could also de-install and delete utility program to wipe out traces of this activity.

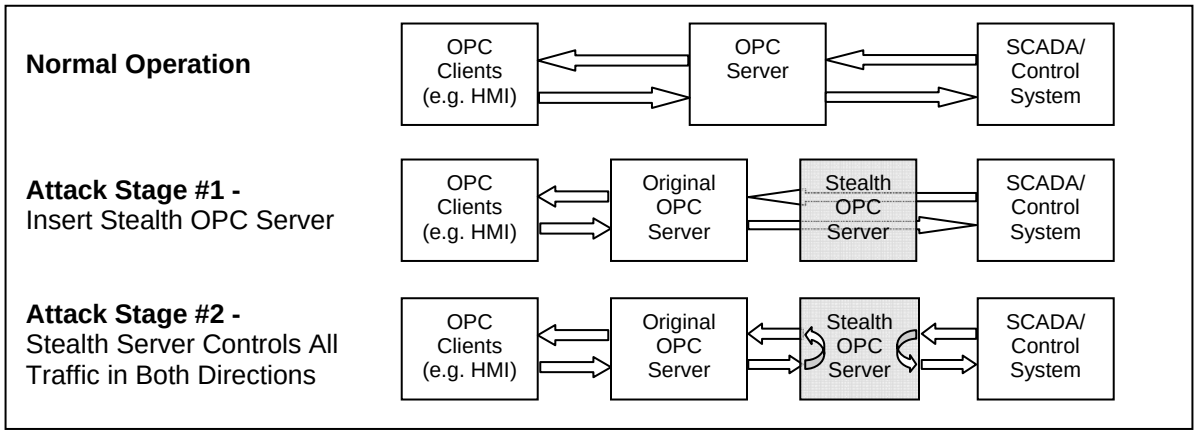

Figure 2-2: OPC Man-in-the-Middle Attack (courtesy of Ralph Langner)

The result would be malevolent process manipulation without being noticed by control room staff (if properly disguised). Attacks of this type against OPC systems have not been reported to date, but have been demonstrated by security researchers. As well, a similar, but non-OPC attack was been reported in 2002 against the ship loading facilities of a major oil company during a general strike. In this case the attackers used knowledge of the MODBUS protocol to remotely destroy all the PLC programs required for tanker loading, effectively shutting down the port for a day.

3 Analysis of Common OPC Configurations and Guidance

In this section we investigate system defaults and vendor guidance documents for OPC installations in terms of their impact on security.

For the most part, the security of any OPC installation largely rests upon how the end-user configures their OPC hosts. While software flaws and vulnerabilities can impact overall system security, incorrectly applied permissions or inappropriate accounts can do far more damage. Furthermore, the typical OPC host configuration is strongly influenced by the guidance provided by the software vendor - most end-users are likely to stick with the default vendor configuration unless there are very convincing reasons to do otherwise. Thus we believe that the OPC vendor has significant responsibility for helping users create secure OPC systems.

3.1 User Guidance Documentation

For this portion of the analysis we requested OPC/DCOM host hardening recommendations from a large number of vendors. Unfortunately we failed to receive any documents that explicitly focused on OPC security, although we understand that such documents may be under development in some organizations. In the end we were left with what we could obtain from vendor web sites or from organizations such as the OPC Foundation and CERN.

As we reviewed many of these OPC documents, we were struck by the impression that many authors believe security isn't really a major consideration for applications not directly on the Internet. For example, the OPC Foundations' XP SP2 Configuration guide[20] states that the Windows firewall could be disabled if the OPC server is safely behind a corporate firewall. Unfortunately we believe that this is a very antiquated and misguided viewpoint – numerous studies have shown that permeability of most network architectures with a single security perimeter control is inadequate and untenable[21]. Below we will summarize two documents that provided the most focused guidance, one from the OPC Foundation and the other from CERN.

3.1.1 OPC Foundation Guidelines

Due to the extensive security improvements in Windows XP SP2, the OPC Foundation released a document describing settings necessary to get OPC

[20] Karl-Heinz Deiretsbacher, Jim Luth & Rashesh Mody; "Using OPC via DCOM with Microsoft Windows XP Service Pack 2", *OPC Foundation*, 2004

[21] Avishai Wool, "A quantitative study of firewall configuration errors" IEEE Computer Magazine, *IEEE Computer Society*, June 2004, Pages 62-67

working if SP2 was installed on a host. Since the primary function of the Windows XP SP2 release was improved security, the recommendations in this document have a significant impact on OPC security. Table 4-1 shows a summary of these recommendations.

Host Firewall	Add DCOM (TCP port 135) and OPCEnum exceptions to firewall. Recommends for vendors to use the Microsoft API to automate firewall configuration during the install.
DCOM Settings	Add anonymous login to OPCEnum.exe. Although the screen shot shows adding Everyone and Anonymous login as the users, they recommend creating an opcuser's group.
Network Considerations	May be acceptable to permanently disable the firewall if behind a corporate firewall.

Table 3-1: Summary of OPC Foundation Security Configuration Guidelines

Unfortunately, while these guidelines offer several useful tips, the overall effect may be to decrease rather than increase the security of OPC systems by implying that very permissive settings are required on the Windows firewall for OPC to operate[22]. In particular, the suggestion that it might be acceptable to disable the host firewall if the system was safely behind a corporate firewall was troubling. While this may have been acceptable when the document was written in 2004, we believe this to be ill advised in most industrial settings today. Thus we encourage the OPC Foundation to have the document revised.

3.1.2 CERN Security Guidelines

Conseil Européen Recherche Nucleaire (CERN) released a well written document describing their practices for Windows XP SP2 deployments of OPC Servers and Clients[23]. This document provided good quality recommendations from a security perspective, and we have therefore built upon their suggestions for our hardening recommendations in White Paper #3. Table 3-2 shows a summary of these recommendations.

[22] In fairness to the OPC Foundation, the purpose of the whitepaper was to explain how to configure the Windows firewall to allow OPC traffic pass through it. This in turn would help discourage users from shutting off the firewall altogether.

[23] Jean-Pierre Puget, Renaud Barillere, Mark Beharrell; "IT-CO recommended DCOM settings for OPC", *European Lab oratory for Particle Physics*, 7 July 2005 http://itcofe.web.cern.ch/itcofe/Services/OPC/GettingStarted/DCOM/RelatedDocuments/ITCODCOMSettings.pdf

Host Firewall	Recommends disabling the built-in firewall during installation and re-enabling it later. Exceptions are added for DCOM (TCP port 135) and for all client and server applications.
Accounts	Recommends creating a local group for all users that should be able to access the OPC server.
DCOM Settings	For the OPC Server: (Application: Connect, Identity: CERN\opcadmin, Launch Permissions, Access Permissions) OPC Server Browser (Authentication Level: Connect)

Table 3-2: Summary of CERN Security Configuration Guidelines

3.2 Default Configuration Parameters for OPC Servers

One of the areas where Microsoft has made significant improvements in the past few years is in making applications, processes and the entire Windows operating system more secure "out of the box". Although a tough default security posture with restrictive access control and fewer enabled services may make an application more difficult to initially configure, in the long run this may be the best security solution. On the other hand, making the default too restrictive may cause the user to configure the application to the most insecure (permissive) status just to get it operational. Consequently, a balance of security and usability is required.

As we investigated the range of DCOM configuration guidance, our goal was not to be exhaustive (or even present best practices) but to understand the range of likely configurations. In most cases, our assessment was based on freely available documentation and OPC servers.

One of the first things we noticed was that the majority of vendors did not tighten down the access permissions for DCOM objects after the installation process was complete. In fact, as OPC security expert Ralph Langner points out in his paper "*OPC Exposed Part II*"[24], a number of major vendors made recommendations that left the end users' OPC security configuration wide open. For example, Langer offered the following sample from a major PLC manufacturer's OPC documentation[25]:

> *"As of Service Pack 2 for Windows XP, communications of OPC also requires the following permissions to be set up:*
>
> - *Local and remote launch for the Anonymous Logon in Launch Permission*

[24] Ralph Langner; "OPC Exposed Part II, S4 Conference", *Digital Bond Press*, Miami, FL, January 2007

[25] After careful consideration the identity of this vendor has been removed to avoid subjecting the users of this product to unnecessary risk.

- *Local and remote activation for the Anonymous Logon in Launch Permission*
- *Local and remote access for the Anonymous Logon in Access Permission*

These settings are made automatically when you install the XXXX NET CD."

These settings basically allow any individual with network access to both launch and access arbitrary OPC services across the network. Even worse, Mr. Langner notes that some vendors even suggest that the user reconfigure standard DCOM access rights for the entire host and not just their application. Since many users do not fully understand the impact of these configuration suggestions, we feel that this type of vendor guidance is irresponsible and will result in settings that violate most corporations' security policies.

In comparison, many IT applications have executable hardening scripts for installation. These scripts may initially reduce the security of the system to expedite the actual installation of an application, but they then automatically correct this at the end of the installation process. We suggest that control system vendors using OPC start adopting this technique to reduce the post installation configuration burden upon the end-user.

Honeywell and Matrikon provide two good vendor practices that can make securing OPC easier. Honeywell not only extended OPC to provide increased security, but provided very specific configuration recommendations, especially on user accounts and permissions. Although we did not examine as many live OPC Servers as we would have liked, Matrikon was the only vendor of those we examined that changed DCOM permissions at the completion of the installation process, showing what can and should be done by vendors.

In summary, the focus of most user documentation is on initial configuration and the steps necessary to ensure proper functioning of the applications. Unfortunately, this is typically done by strongly downgrading the security posture of the applications and offering no guidance on how to upgrade it again at the completion of installation. The security of most OPC systems would be greatly improved if vendors improved the quality of configuration guidance to include improving security settings and provided easy to use hardening scripts to automatically enable more reasonable security settings after installation.

4 Conclusions

In this report we looked at the possible vulnerabilities present in OPC systems as they are deployed today. We then used this data to create four scenarios that represent some of the possibilities for OPC-based attacks. These four point us to several interesting conclusions. First, they highlight the fact that attacking OPC deployments does not require special skills or esoteric controls knowledge. All the tools and information needed to carry them out can be downloaded from the Internet.

The second conclusion we can draw is that two core vulnerabilities, namely excessively open firewalls and overly permissive DCOM access rights, lay at the heart of many scenarios. If either vulnerability is addressed, then the chance of these scenarios occurring is significantly reduced. What is especially interesting is the fact that these vulnerabilities could be considered the responsibility of and within the control of the knowledgeable OPC end user.

In other words, while there are a significant number of high profile vulnerabilities that are due to the design of both DCOM and the OPC standard, the misconfiguration of either the underlying operating system or DCOM configuration settings appear to offer a much more attractive attack surface for the average attacker. OPC security (particularly authentication and authorization) is tightly bound to operating system security, so a poorly configured OPC application can adversely affect the security of the underlying operating system and vice-versa.

Given Microsoft's large market share and early design decisions, there are significant vulnerabilities in OPC systems simply as a result of the underlying operating system. There is a rich library of attack techniques and tools that novice (or automated) attackers can use to easily identify OPC servers on a network and then extract useful information. Due to the weak configuration of many of the OPC deployments active today, these attacks can be extremely effective. The OPC end user can have a direct influence on reducing this attack surface.

At the same time, the OPC vendor community also needs to bear some responsibility for improving OPC security. The poor quality (from a security point of view) of most OPC user documentation is resulting in deployments with an unnecessarily weak security. We strongly believe that the security of most OPC systems would be greatly enhanced if vendors improved the quality of configuration guidance to include recommended security settings and provided easy to use hardening scripts to automatically enable more reasonable security setting after installation. A few vendors have moved in this direction, but the vast majority have not.

The good news is that the hardening practices for the Windows operating system are well known by the IT security community and can be adopted by the controls community to significantly reduce these risks. In other words, OPC's reliance upon the Microsoft platform is both a curse and a blessing - while Windows has flaws, there are a wealth of practices for hardening Windows servers that can be easily applied to OPC clients and servers. In addition there are a number of DCOM specific security settings that can also be applied by the knowledgeable end-user. We will discuss these solutions in detail in our final report in this series, *OPC Security White Paper #3 – Hardening Guidelines for OPC Hosts*.

Glossary

ACL - Access Control List: List of rules in a router or firewall specifying access privileges to network resources.

API - Application Programming Interface: The specification of the interface an application must invoke to use certain system features.

CATID - Category Identifier: Specifies the active OPC specifications.

CCM - Component Category Manager: A utility that creates categories, places components in specified categories, and retrieves information about categories.

CERN - Conseil Europeen Recherche Nucleaire: European Laboratory for Particle Physics.

CIFS - Common Internet File System: Updated version of Server Message Block application-level protocol used for file management between nodes on a LAN.

CIP - Common Industrial Protocol: CIP is an open standard for industrial network technologies. It is supported by an organization called Open DeviceNet Vendor Association (ODVA).

COM – Component Object Model: Microsoft's architecture for software components. It is used for interprocess and interapplication communications. It lets components built by different vendors be combined in an application.

CLSID - Class Identifier: An identifier for COM objects.

CORBA - Common Object Request Broker Architecture: Architecture that enables objects, to communicate with one another regardless of the programming language and operating system being used.

CSP - Client Server Protocol: An Allen-Bradley protocol used to communicate to PLCs over TCP/IP.

DDE – **Dynamic Data Exchange**: A mechanism to exchange data on a Microsoft Windows system.

DCOM – **Distributed Component Object Model**: This is an extension to the Component Object Model to support communication among objects located on different computers across a network.

DCS – Distributed Control System: A Distributed Control System allows for remote human monitoring and control of field devices from one or more operation centers.

DDE - Dynamic Data Exchange: An interprocess communication system built into Windows systems. DDE enables two running applications to share the common data.

DLL - Dynamic Link Libraries: A file containing executable code and data bound to a program at the application's load or run time, rather than linking during the compilation of the application's code.

DMZ - Demilitarized Zone: A small network inserted as a "neutral zone" between a trusted private network and the outside untrusted network.

DNP3 - Distributed Network Protocol 3: A protocol used between components in SCADA systems (primarily in the power and water industries).

DNS – Domain Name System: A distributed database system for resolving human readable names to Internet Protocol addresses.

EN - Enterprise Network: The corporation-wide business communication network of a firm.

ERP - Enterprise Resource Planning: Set of activities a business uses to manage its key resources.

GUI - Graphical User Interface: Graphical, as opposed to textual, interface to a computer.

GUID - Globally Unique Identifier: A unique 128-bit number that is produced by the Windows operating system and applications to identify a particular component, application, file, database entry or user.

HMI - Human Machine Interface: A software or hardware system that enables the interaction of man and machine.

HTML - Hypertext Markup Language: The authoring software language used on the Internet's World Wide Web.

HTTP - HyperText Transfer Protocol: The protocol used to transfer Web documents from a server to a browser.

HTTPS - HyperText Transfer Protocol over SSL: A secure protocol used to transfer Web documents from a server to a browser.

IIS - Internet Information Server: Microsoft's web server application.

IDL - Interface Definition Language: Language for describing the interface of a software component.

IDS - Intrusion Detection System: A system to detect suspicious patterns of network traffic.

IPX - Internetwork Packet Exchange: A networking protocol used by the Novell Incorporated.

IPSEC – Internet Protocol SECurity: An Internet standard providing security at the network layer.

IP - Internet Protocol: The standard protocol used on the Internet that defines the datagram format and a best effort packet delivery service.

I/O - Input/Output: An interface for the input and output of information.

ISA - Instrumentation, Automation and Systems Society: ISA is a nonprofit organization that helps automation and control professionals to solve technical instrumentation problems.

IT - Information Technology: The development, installation and implementation of applications on computer systems.

LAN - Local Area Network: A computer network that covers a small area.

LM - LAN Manager: A now obsolete Microsoft Windows networking system and authentication protocol.

LDAP - Lightweight Directory Access Protocol: A protocol for accessing directory services.

MBSA - Microsoft Baseline Security Analyzer: A tool from Microsoft used to test a system to see if Microsoft best practices are being used.

MIB - **Management Information Base:** The database that a system running an SNMP agent maintains.

MODBUS - A communications protocol designed by Modicon Incorporated for use with its PLCs.

NETBEUI - NetBIOS Extended User Interface: An enhanced version of the NetBIOS protocol.

NetBIOS - Network Basic Input Output System: A de facto IBM standard for applications to use to communicate over a LAN.

NTLM - New Technology LAN Manager: A challenge - response authentication protocol that was the default for network authentication for Microsoft Windows New Technology (NT) operating systems.

OLE - Object Linking and Embedding: A precursor to COM, allowing applications to share data and manipulate shared data.

OPC - OLE for Process Control: A industrial API standard based on OLE, COM and DCOM for accessing process control information on Microsoft Windows systems.

OPC-A&E - OPC Alarms & Events: Standards created by the OPC Foundation for alarm monitoring and acknowledgement.

OPC-DA - OPC Data Access OPC-DA: Standards created by the OPC Foundation for accessing real time data from data acquisition devices such as PLCs.

OPC-DX - OPC Data Exchange: Standards created by the OPC Foundation to allow OPC-DA servers to exchange data without using an OPC client.

OPC-HDA - OPC Historical Data Access: Standards created by the OPC Foundation for communicating data from devices and applications that provide historical data.

OPC-UA - OPC Unified Architecture: Standards created by the OPC Foundation for integrating the existing OPC standards.

OPC XML-DA - OPC XML Data Access: Standards created by the OPC Foundation for accessing real time data, carried in XML messages, from data acquisition devices such as PLCs.

OPCENUM – OPC ENUMerator: A service for discovering and listing OPC servers.

OPC Unified Architecture - OPC UA: Standard to tie together all existing OPC technology and replace the underlying DCOM protocols in OPC with SOAP based protocols.

PLC – **Programmable Logic Controller**: A PLC is a small dedicated computer used for controlling industrial machinery and processes.

PCN - Process Control Network: A communications network used to transmit instructions and data to control devices and other industrial equipment.

PROGID - Program Identifier: A string that identifies the manufacturer of an OPC server and the name of the server.

RPC – Remote Procedure Call: A communications protocol for invoking code residing on another computer across a network.

SAP - Systems, Applications and Products: A German company that produces client/server business software.

SCADA – Supervisory Control And Data Acquisition: A system for industrial control consisting of multiple Remote Terminal Units (RTUs), a communications infrastructure, and one or more central host computers.

SID – Security Identifier: A unique name that is used to identify a Microsoft Windows object.

SP - Service pack: A bundle of software updates.

SPX - Sequenced Packet Exchange: A transport Layer protocol used by Novell Incorporated.

SMB - Server Message Block: A Microsoft network application-level protocol used between nodes on a LAN.

SNMP - Simple Network Management Protocol: A protocol used to manage devices such as routers, switches and hosts.

SOAP - Simple Object Access Protocol: A protocol for exchanging XML-based messages using HTTP.

SSL - Secure Socket Layer: A de facto standard for secure communications created by Netscape Incorporated.

TCP - Transmission Control Protocol: The standard transport level protocol that provides a reliable stream service.

UDP - User Datagram Protocol: Connectionless network transport protocol.

URL - Uniform Resource Locator: The address of a resource on the Internet.

WS-Security - Web Services Security: A communications protocol providing a means for applying security to Web Services.

XML - eXtensible Markup Language: A general-purpose markup language for creating special purpose markup languages that are capable of describing many different kinds of data.