OPC Security Whitepaper #3
Hardening Guidelines for OPC Hosts

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 3 (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	June 28, 2007	E. Byres, D. Peterson	4th Draft for controlled public review
1.3a	August 31, 2007	E. Byres, D. Peterson	5th Draft for final DHS review. Includes comments from the DHS Recommended Practices Group
1.3b	September 9, 2007	E. Byres, D. Peterson	Correction of minor grammatical errors and added figures to Section 3.6
1.3c	November 13, 2007	E. Byres, D. Peterson	Correction of minor editorial errors

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 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.

Since OPC deployments can vary widely, it is essential that any of the recommendations in this report be tested on a non-critical test system before being deployed in a live control system.

byres research
intrinsically secure

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 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 last of the three, and outlines how a server or workstation running OPC can be secured in a simple and effective manner. Typically this “hardening” must be conducted in several stages. First the operating system (typically Windows) needs to be “locked down” in such a manner that will make it less susceptible to common O/S-based attacks. This involves five steps which are:

1. Ensuring up-to-date patching of the operating system and applications on the OPC host;
2. Limiting services to the required minimum for OPC;

3. Defining user accounts and privileges;
4. Limiting network access via the Windows Firewall;
5. Protecting the Windows Registry.

Next, the specific OPC components must be hardened using the OPC and DCOM configuration tools found in Windows. Unfortunately, completing this stage successfully is more complex; our testing indicated that there are a number of OPC applications that do not properly follow the DCOM specifications for Windows software. As a result, several of the steps suggested below may cause a malfunction of these OPC applications. Thus we suggest the OPC user consider the seven steps listed below as a menu to choose from rather than a list of unalterable requirements:

1. Controlling the authentication levels for various OPC actions;
2. Controlling the location of various OPC actions;
3. Managing the DCOM Permissions;
4. Limiting protocols used by DCOM/RPC and setting a Static TCP port;
5. Setting appropriate OPC servers accounts;
6. Restricting Transport Protocols for RPC;
7. Restricting TCP Port Ranges for RPC.

Of these seven, perhaps the most unusual is step 4, as it gives the end-user the opportunity to address one of the more vexing problems in OPC security, namely the problem of dynamic port allocation. Unfortunately it was also the solution most likely to cause issues with OPC software, since it was apparent that not all vendors of OPC products respect the static setting of port numbers. Thus we also provided step 7 as alternative method for port restriction, in case task 4 does not work correctly on your OPC software.

Next, the system needs to be tested to ensure these changes still allow all OPC applications to function correctly. Since we found a number of cases where OPC vendors were not respecting DCOM security settings and requirements, this testing is critical before any security settings are deployed on live production systems.

Lastly, verification of the fortifying effort is required to ensure no serious security holes have been left open. This includes the following steps:

1. Windows Service and Open Port Determination
2. Windows Event Log Analysis
3. Vulnerability Scanning

These stages are expanded upon in a detailed Action Plan for Hardening OPC Hosts within this report. Specific examples are also provided for each

task. In all, we believe by following these guidelines, the typical controls technician will be able to create a more secure and robust OPC deployment on their plant floor and OPC can continue to grow as a valuable solution in industrial data communications.

1 Introduction

This report is the third 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, its basic components and how it is 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 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 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 make this migration from the DCOM-based architecture to a .NET-based architecture, industry will have the opportunity for much 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 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 third 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 #2 we defined a set of vulnerabilities and possible threats to OPC hosts, based on OPC's current architecture (i.e. the use of DCOM). We also looked 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.

In White Paper #3, we use this information to give the OPC end-user a series of practical recommendations they can draw upon to secure their OPC host machines.

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.

It is also important to understand that this document details nearly every security measure that could be used to harden OPC installations. In order to

determine which of the mentioned countermeasures and strategies are feasible and advisable for a specific OPC deployment, a risk assessment should be conducted first. In addition, the industrial environment should be checked to ensure all design elements will function flawlessly with the proposed security countermeasures. Some suggested countermeasures will not work with -- or are not advisable for -- every OPC installation.

Finally, we cannot guarantee that following our recommendations will result in a completely secure configuration. Nor can we guarantee 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 Hardening Strategy for OPC Hosts

Building on the material from the previous white papers, this report attempts to detail all security measures and good practises that could be used to harden OPC hosts[4]. We suggest the OPC user consider the mitigations listed in this reports as a menu to choose from rather than a list of unalterable requirements.

Typically this "hardening" should be conducted in four stages. First, the Windows platform itself needs to be "locked down" to make it less susceptible to common Windows-based attacks, yet still allow OPC applications to function. Then the specific OPC components need to be hardened using the OPC configuration tools found in the Windows operating system. Next the system needs to be tested to ensure these changes still allow all OPC applications to function correctly. We found a number of cases where OPC vendors do not respect DCOM security settings and requirements, so the test stage is critical before any security settings are deployed on live production systems. Lastly, verification of the fortifying effort is required to confirm no serious security holes have been left open.

For the most part these configuration guidelines will apply to both clients and server hosts. The callback mechanism used by OPC essentially turns the OPC client into a DCOM server and the OPC server into a DCOM Client. In our examples we focus on OPC servers, but to take full advantage of these recommendations they should be followed on all nodes that contain either OPC servers or OPC clients. Several sections discuss clients specifically.

It is also important to note the examples shown below are primarily based on hosts running Windows XP/SP2 or Windows Server 2003/SP1 (or later). Earlier versions of Windows can still take advantage of many (but not all) of these suggestions, but will be considerably more difficult to configure. Thus if at all possible, a first step should be to upgrade any OPC host platforms to these newer operating system versions.

Finally, these examples were performed and lab tested in a workgroup setting; as a result, slight modifications may be required in domain-based environments. In real-life industrial settings domains may be beneficial as they provide the ability to apply these recommendations uniformly across a group of hosts via group policy. In workgroup environments all recommendations will have to be deployed individually on the host machines, increasing the administrative effort and the chance for error. In addition, we are aware of

[4] Please note that this report only focuses on OPC host security and does not attempt to detail good practices for securing the network components (such as firewalls) for OPC traffic. We hope to offer this information in a fourth white paper in 2008. In the mean time, interested readers should consider the Microsoft Technical Article "Using Distributed COM with Firewalls" by Michael Nelson at http://msdn2.microsoft.com/en-us/library/ms809327.aspx

some possible domain specific security features that can be added, but these were beyond the scope of this report and are not discussed in this document.

3 General Windows Hardening Recommendations

Since OPC is deployed on the Windows operating system in over 95% of the cases, this section discusses the general hardening of OPC hosts using standard Windows-based tools and techniques. Five security mechanisms are discussed:

1. Ensuring that operating system and application patches are at a currently version level;
2. Configuring the minimum services running on the host for a typical OPC deployment;
3. Limiting of user privileges through account management;
4. Limiting network access via the Windows IP Security Policies;
5. Protecting the Windows registry.

While none of these mechanisms are particularly revolutionary, the real trick is to secure the host in such a manner that makes it less susceptible to common Windows-based attacks, yet will still allow all OPC applications to function. This is often more difficult than it should be for two reasons. First, some requirements for OPC operation are at odds with good Windows security practices. Second, a number of OPC vendors appear to ignore a number of Windows DCOM specifications and requirements. That said, based on our lab testing of configurations listed in this section, we believe all will allow the correct operation of most OPC systems.

Since OPC deployments can vary widely, it is essential that any of these settings be tested on a non-critical test system before being deployed in a live control system.

All techniques discussed in this section are based on standard administrative tools available in the current "professional" versions of Windows[5]. Thus the specific examples illustrated below are intended for the Windows 2000/SP4, Windows Server 2003/SP1 and Windows XP/SP2 operating systems. These were chosen, since the survey results noted in White Paper #1 indicate these are the versions of Windows most likely to be used in OPC deployments.

3.1 Patch Management for OPC Hosts

As we noted in the introduction to this report, and expanded on in White Paper #2, poor patching of OPC hosts is a significant contributing factor for

[5] The Windows Vista operating system was not tested as it was unavailable at the time the lab testing was performed

OPC security issues. A number of the well-known worms (such as MSBlaster) released in the past few years have specifically targeted the underlying RPC and DCOM services for OPC. This has made users and vendors keenly aware of the need to patch operating systems and applications in industrial control systems. Unfortunately, the difficulty with patch management is one cannot automatically deploy new patches into the process control environment without risking disruption of operations. Thus careful policy and practice is required that balances the need for system reliability with the need for system security.

Based on our survey, it appears many users and vendors have developed effective patching procedures for PCs used in their control systems. 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*"[6], and the Edison Electric Institute's "*Patch management Strategies for the Electric Sector*".[7] Both provide excellent guidance for patch management in critical system.

3.2 Minimum Required Services

In order to make Windows hosts more secure, it is critical that all unnecessary services be disabled. Based on lab testing, the following are the minimum set of Windows 2000[8], Windows Server 2003 and Windows XP[9] services that are typically required on stand-alone OPC clients and servers. The name in brackets following the service name is the recommended Startup Type:

- COM+ Event System (Automatic)
- COM+ System Application (Automatic) (Required by XP)
- DNS Client (Automatic)
- Event Log (Automatic)
- IPSEC Services (Automatic)
- Net Logon (Manual)
- NTLM Security Support Provider (Automatic)
- Plug and Play (Automatic)

---

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

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

[8] http://labmice.techtarget.com/articles/win2000services.htm

[9] http://www.sysinternals.com/blog/2005/07/running-windows-with-no-services.html

- Protected Storage (Automatic)
- Remote Procedure Call (RPC) (Automatic)
- Security Accounts Manager (Automatic)
- Security Center (Automatic) (Required by XP)
- Server (Automatic)

As well, some OPC applications require additional services to be enabled to remain functional. For example, if the OPC application does not use the OPCEnum component (and thus needs to remotely browse the registry[10]) the following services are also required:

- Computer Browser (Automatic)
- Remote Registry (Automatic)

While not strictly a service, File and Printer Sharing should be disabled. This is done via the network connections panel.

Again, since OPC deployments can widely vary, it is essential that the effects of disabling any service be tested on a non-critical offline system before being deployed in a live control system.

3.3 Limiting User Privileges

In most control environments, the day-to-day operation of OPC-based applications does not require a highly privileged account. On the other hand, the configuration of OPC applications often does. Unfortunately, in many systems we see the highly privileged account settings being the norm, exposing the system to numerous security issues.

To address this, we recommend OPC administrators create two accounts, one for day-to-day operations and one for configuration.[11] Configure these accounts as follows:

- *Create an account (e.g. opcuser) and set it to be a low privilege account* - This will be used for the normal execution of OPC client and server applications. When the opcuser account is created it should be added as a member of the Users group.
- *Create an account (e.g. opcadmin) and set it to be a high privilege account* – This account will only be used for infrequent

[10] Remotely browsing the registry is no longer a recommended practice by the OPC Foundation. However some older applications may still require remote browsing to function correctly.
[11] http://www.opcconnect.com/dcomcnfg.php

configuration changes and for the initial installation of the OPC software. When the opcadmin user is created it should be added as a member of the Administrators group. It is often simplest to rename the existing administrator account to opcadmin.[12]

Finally the Guest account should be disabled and robust passwords (a mix of letters, numbers and special characters and not found in a dictionary) should be used for all accounts.

3.4 Limiting Network Access

In most control environments there is little reason to allow every device on the control network to communicate to OPC hosts. Typically there are only a small number of machines communicating using OPC. Because of this, it makes good security sense that network access should only be allowed between these few trusted machines. Windows 2000, Server 2003 and XP contain host-based firewall capabilities that can use IP filters and a security policy to restrict network traffic to OPC hosts.

Our recommendation is to add a simple host-based firewall rule allowing traffic only to or from the IP addresses of other trusted OPC hosts. While this might seem to be simple, we discovered that in practice, setting up such a rule can be very cumbersome using the firewall configuration wizards available in Windows 2000, Server 2003 and XP. Thus these firewall wizards are not used and the following four-step process is recommended instead.

It is worth noting there are other technologies for controlling access between hosts that can be even more robust. For example, Microsoft's Domain Isolation model[13] is far more secure. However due to its complexity, detailed directions for configuring it are beyond the scope of this report - it may be covered in subsequent reports.

3.4.1 Creating the Filter Lists

Two filter lists are required to properly secure a host. The first list matches all traffic coming to and from trusted machines. The second list matches all

[12] NOTE: For simplicity in this report we refer to user accounts rather than account groups. However a better alternative is creating an opcadmin group rather than just adding an opcadmin user. Then within the opcadmin group an account can be made for everyone who should have administrative privileges to the OPC server. This will provide change management accountability for the OPC host. The same applies to creating opcuser group rather than a single opcuser account that multiple users access. For more information on account groups in domain environments see: http://www.microsoft.com/technet/security/guidance/networksecurity/sec_ad_admin_groups.mspx

[13] http://www.microsoft.com/technet/itsolutions/msit/security/ipsecdomisolwp.mspx

other traffic. In the examples below there is only one trusted machine, but this could easily be expanded.

First, launch the *Control Panel/Administrative Tools/Local Security Policy* application. Next, while making sure the "*IP Security Policies on Local Computer*" icon is selected, select "*Manage IP filter lists and filter actions*" under the *Actions* menu.

Now select the *Manage IP Filter Lists* tab and add the filter lists. Figure 3-1 shows what to expect while the filter list for traffic between trusted machines is being created. The filter list that matches all other traffic is the same except no destination IP address is specified.

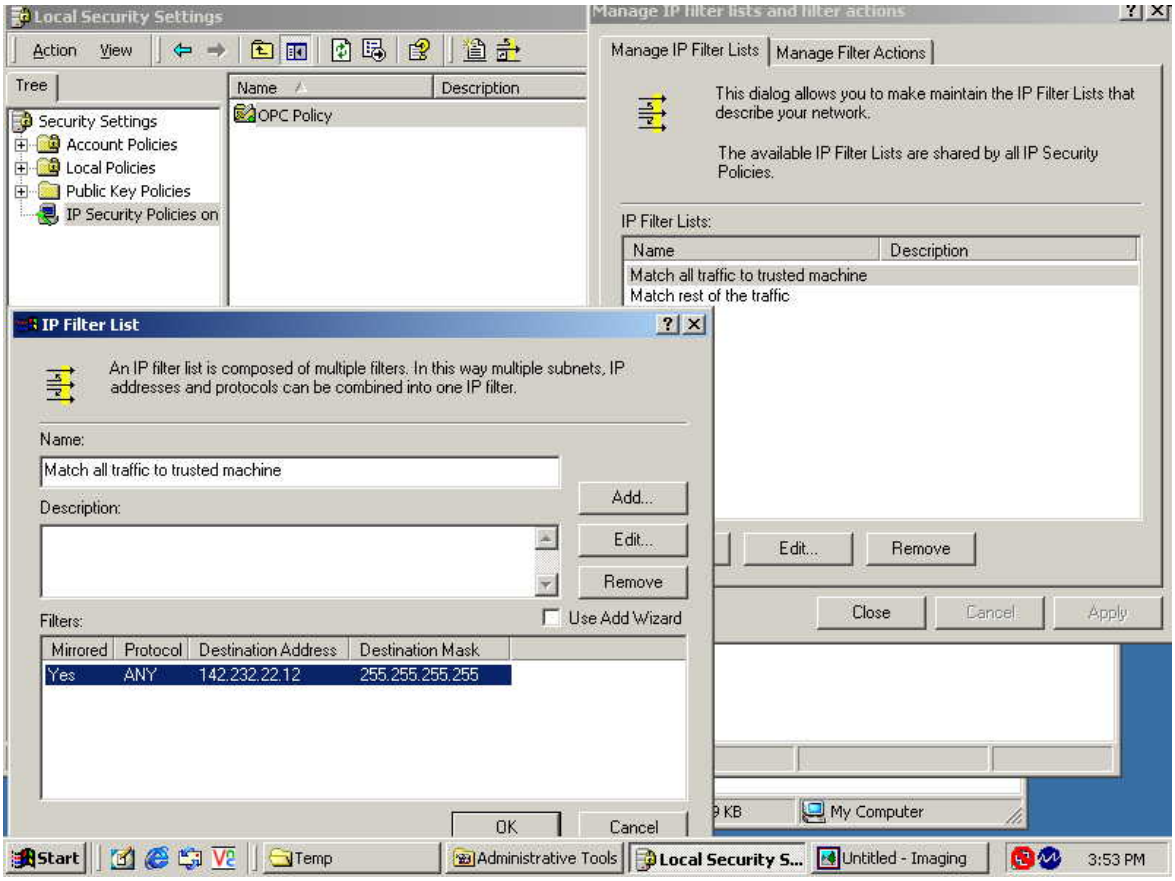

Figure 3-1: Creating the Filter Lists

Two configuration settings are rather subtle; "*Mirrored*" should be selected and Protocol should be *ANY*. Mirrored refers to matching traffic between trusted machines in both directions. ANY refers to allowing any protocol running on top of IP for trusted machines. It is possible the protocol could be narrowed down to only TCP, but care is needed to ensure that this doesn't impact other critical services you may require.

3.4.2 Creating the Block Action

Once the lists are created, actions for these lists are needed. In this case two actions are required. The first is Permit, and it exists by default. The other is Block and it needs to be created. If a filter list has an action of Block, then all traffic that matches the filter list gets dropped.

Using the Local Security Settings Tool, under the *Actions* menu item, select "*Manage IP filter lists and filter actions*". Now select the *Manage Filter Actions* tab to create the Block action. Figure 3-2 illustrates the action being created.

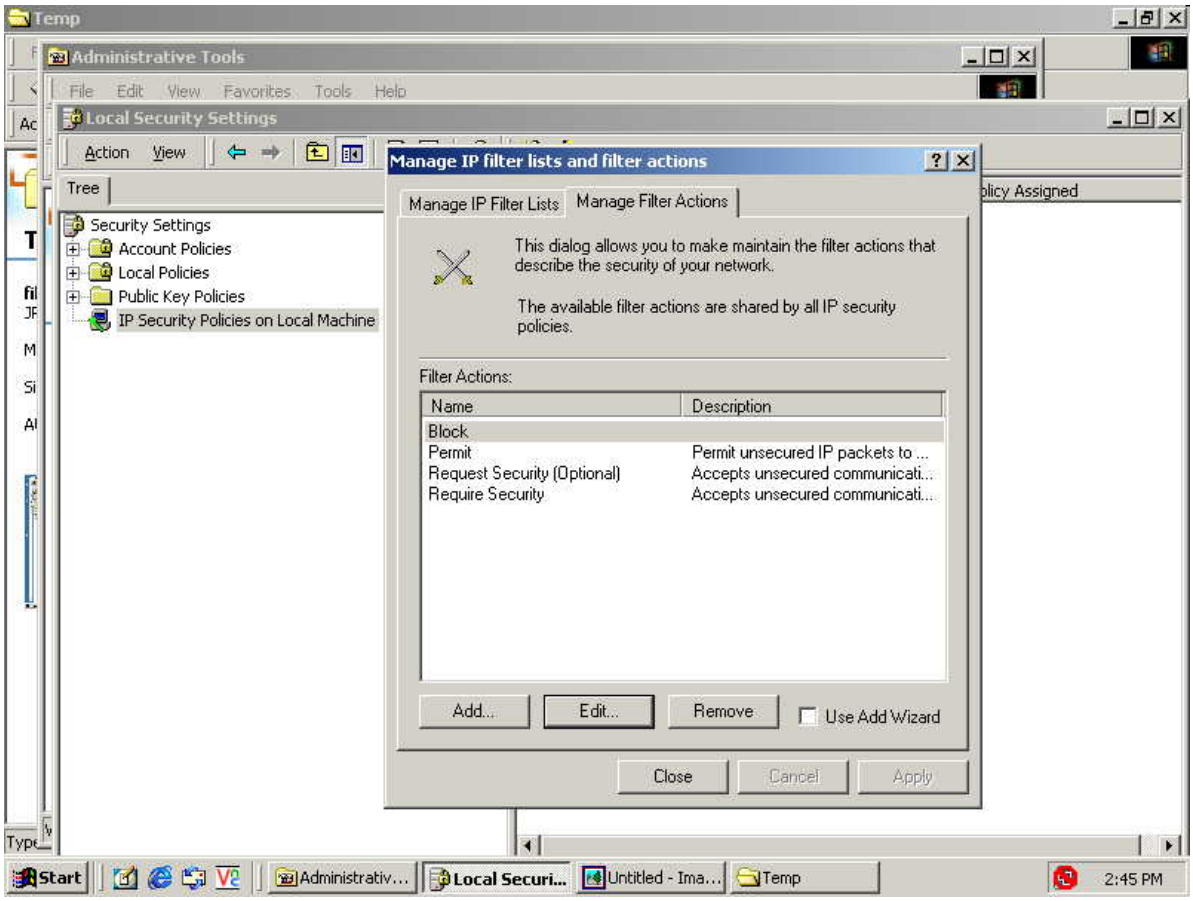

Figure 3-2: Creating the Block Action

3.4.3 Creating the Security Policy

After the Filter Lists and Block Action have been created, it is time to glue them into a security policy and apply them to all of the network interfaces.

Select *IP Security Policies* on *Local* and then under the *Actions* menu item of the Local Security Settings Tool, select "*Create IP Security Policy*". Give the policy a meaningful name (such as OPC Hosts Policy), deactivate the default response rule and add filter lists and actions. Set action to *Permit* for traffic between trusted machines and *Block* otherwise.

Unfortunately this step is not quite this easy as it could be because these policies have Internet Protocol Security (IPsec) features that need to be addressed. To use our lists and actions to simply filter IP traffic, do not select the default dynamic filter list, ignore the *Authentication* field, set *Tunnel Setting* to None and *Connection Type* to All. Figure 3-3 shows what to expect while the policy is being created.

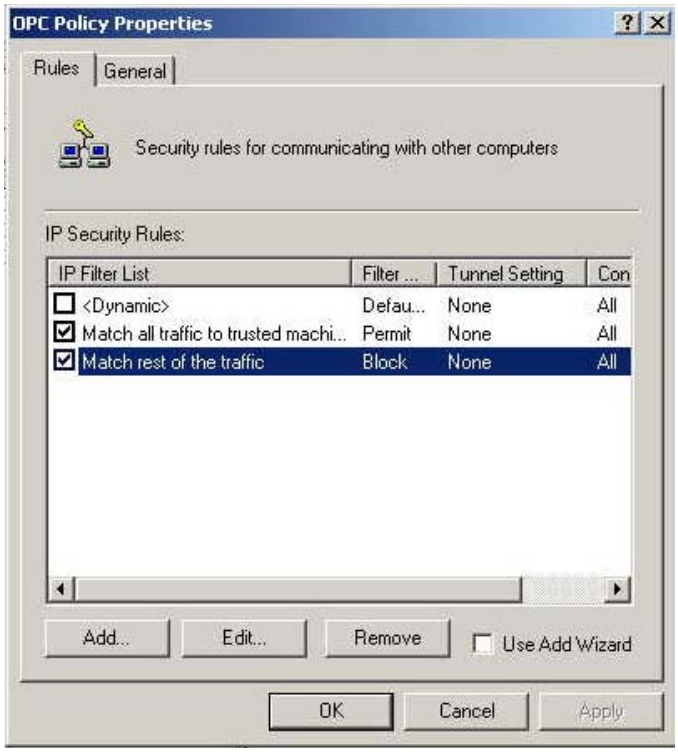

Figure 3-3: Creating the Security Policy

3.4.4 Assigning the Security Policy

The last step is to assign the policy. Simply right click on the policy and select assign. Figure 3-4 shows what to expect while the policy is being assigned.

Once these four steps are complete, a rule that only allows traffic to or from the IP address of trusted OPC hosts should be in place.

Again, since OPC deployments can widely vary, it is essential that the effect of these rules be tested on a non-critical offline system before being deployed in a live control system.

3.5 Protecting the Registry

The registry is the central repository for configuration data in Windows. In order to protect the registry as much as possible, regular users should not be given "*Administrator*" rights, and "*Remote Registry Editing*" should be disabled from the "*Services*" panel of "*Administrative Tools*" on "*Control*

Panel". Note that restricting the ability to change values in the registry is not the same as restricting read access. Read access is needed only for systems that do not use OPCEnum for server browsing. If you have newer versions of OPC applications, there should be little need for registry browsing.

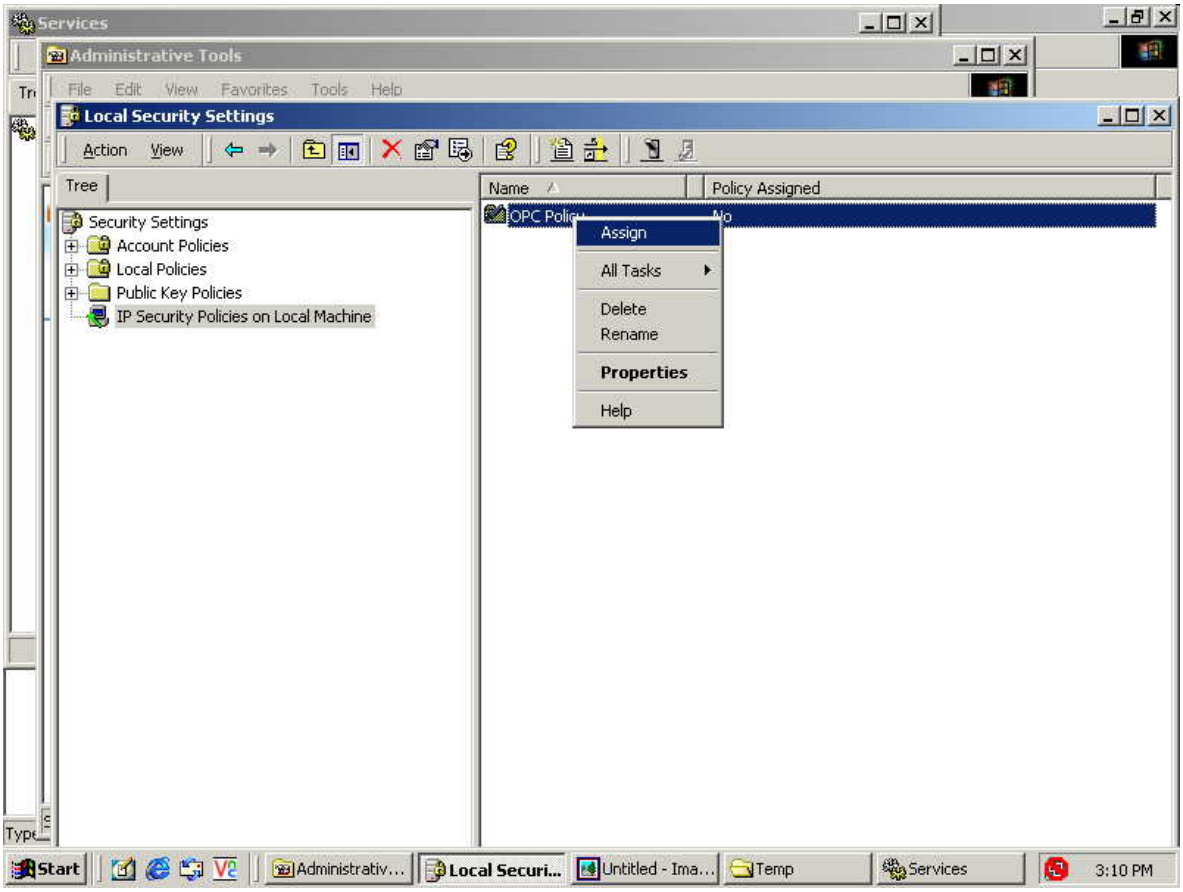

Figure 3-4: Assigning the Security Policy

When changing these settings there are several important tips that should be considered:

- Never change SYSTEM permissions from *Full Control* in the Registry. Any changes to this permission will cause your system to fail upon reboot.
- Consider removing permissions for the Power Users group if that group is not in use and replace all permissions for Users and Everyone group with Authenticated Users.

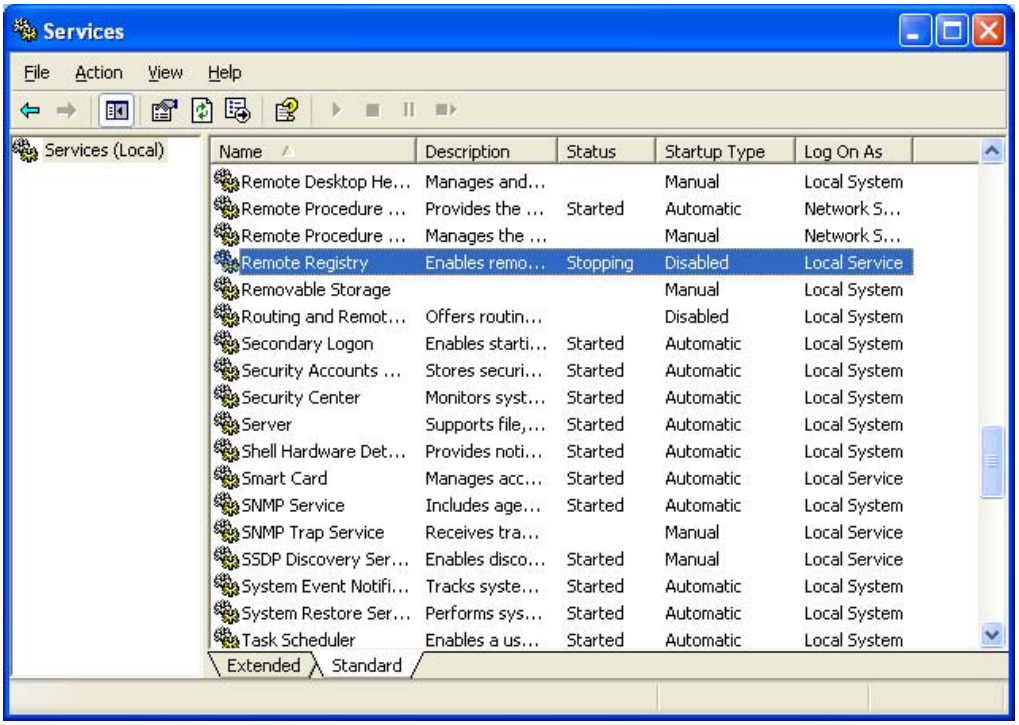

Figure 3-5: Remote Registry Service

3.6 Some Special Considerations for XP Systems

After all this setup, you may find that remote access using the opcuser and opcadmin does not work on your XP-based server. The reason is that for all out-of-the-box installations of XP in workgroup architectures, the system authenticates all remote users as "guest" regardless of the account name. The trick is to tell XP to use the "classic" authentication as shown in the screenshot below.

To access this setting launch the *Control Panel/Administrative Tools/Local Security Policy* application. Next, select *Local Policies/Security Option* as scroll down until you see the item *Network Access:Sharing and security model for local accounts* . Right click and you can access the *Properties* option.

If you configure this policy setting to Classic, network logons that use local account credentials authenticate with those credentials. This Classic model provides precise control over access to resources, and allows you to grant different types of access to different users for the same resource, which is exactly what is needed for OPC. Conversely, the Guest-only model treats all users equally as the Guest user account, and all receive the same level of access to a given resource, which can be either Read Only or Modify. This clearly doesn't work for the OPC security model we are proposing.

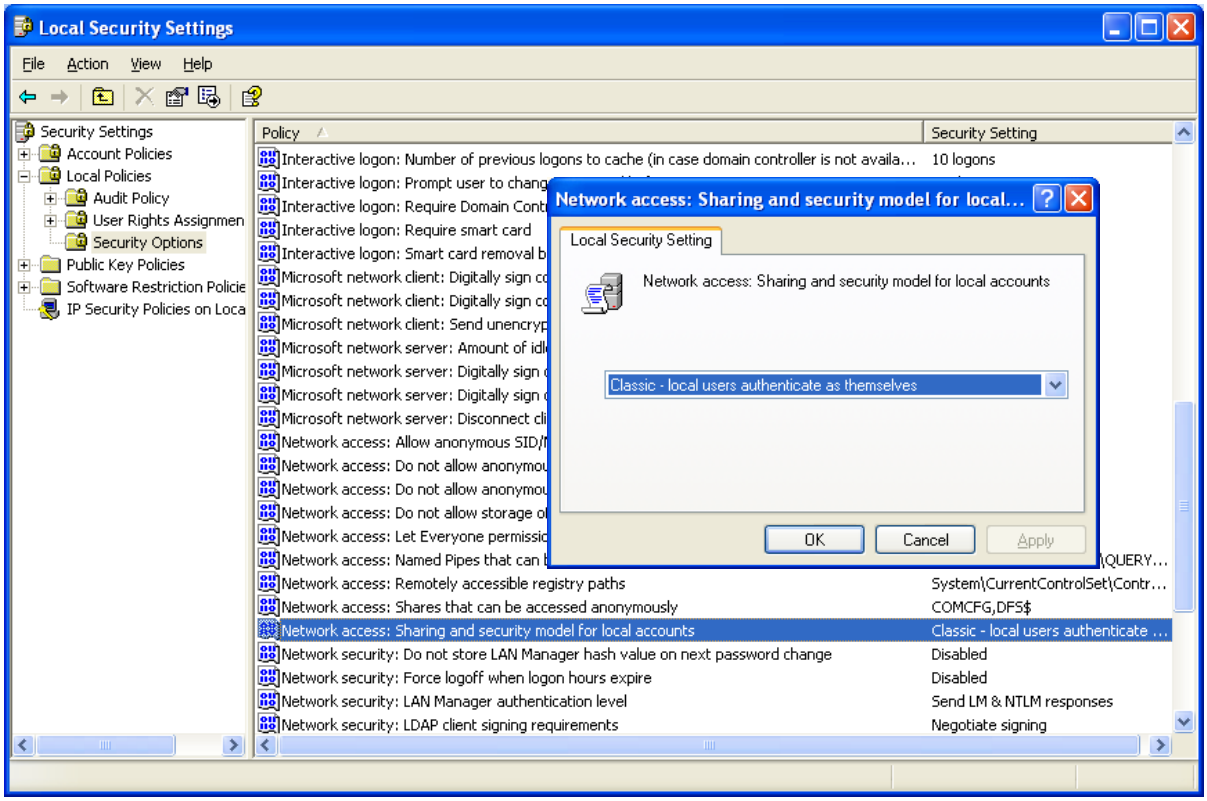

Figure 3-6: Setting the XP Remote Access to "Classic"

Note that this policy setting does not affect network logons that use domain accounts. The default for Windows XP computers that are joined to a domain and Windows Server 2003 computers is Classic. This setting also has no effect on Windows 2000 or Server 2003 computers.

4 OPC/DCOM/RPC Hardening Recommendations

Once the underlying Windows system is secure, it is time to address the security of the OPC applications. This involves carefully setting up user accounts, putting in restrictions for DCOM objects and restricting RPC behavior. The configuration required is discussed below in three parts; OPC Hardening, DCOM Hardening and RPC Hardening.

It is important to note that this section is focused on guidance for the Windows Server 2003/SP1 and Windows XP/SP2 operating systems. Microsoft added a number of significant DCOM security enhancements to these versions[14] and the recommendations in this section are designed to take advantage of these improvements. Users of older operating system versions can still follow many of the guidelines below, but upgrading to the newer versions is highly recommended.

Since OPC deployments can vary widely, it is essential that any of these recommendations be tested on a non-critical test system before being deployed in a live control system.

The recommendations in this section require considerable care and off-line testing before they are deployed in critical systems. Our tests showed there are a number of OPC applications that do not properly follow the DCOM specifications for Windows software. For example, using the DCOM controls to set a static TCP port for an OPC application (as noted in Section 4.2.4) caused issues with the OPC software from a number of vendors. In response, we provided Section 4.3.2 *Restricting TCP Port Ranges for RPC*, as alternative method for port restriction. Thus the OPC user should consider the suggestions listed in this section as a menu of security options to choose from, rather than a list of unalterable requirements.

4.1 OPC Hardening Recommendations

By utilizing separate opcuser and opcadmin accounts or groups as suggested in Section 3.3, we can limit the security exposure by restricting what actions the OPC server and authenticated users can perform. We recommend the opcadmin account be used only when installing the OPC server or client software and making configuration changes, since this account can both launch and access OPC servers. Even then, the opcadmin account should be limited to a specific list of OPC servers or clients.

For the actual running of the server the opcuser account (or opcuser group account) should be used. As defined below, opcuser cannot launch an OPC server, but can access a running server.

[14] http://www.microsoft.com/technet/prodtechnol/winxppro/maintain/sp2netwk.mspx

Finally we suggest only running the OPCEnum service[15] when it is necessary to browse the OPC servers. When OPCEnum is run, limit its access to the opcuser and opcadmin accounts. Left in its wide open state, OPCEnum can present a considerable security risk and typically other users do not need to access it.

4.2 DCOM Hardening Recommendations

There are two main goals for successful DCOM hardening. First, we need to only give as much permission as is required for users per DCOM object. For example, if a computer is running three OPC servers, but only one needs to be accessed remotely, only allow remote access to that one server.[16] Similarly, if all OPC servers and clients are on a single host, then disable remote access and allow only local access.

Second, we need to use the different level user accounts created earlier for Launch and Access permissions. Again we suggest opcadmin be the only user account used to launch or configure OPC servers and should have the servers it can configure restricted. The opcuser account can be used by users who need only to connect and access running OPC servers.[17]

To achieve these two goals we use the DCOM Configuration Tool that is found under *Control Panel/Administrative Tools/Component Services*[18] shown in Figure 4-1. It can also be accessed by starting dcomcng.exe from the *Run...* option in the Start Menu.

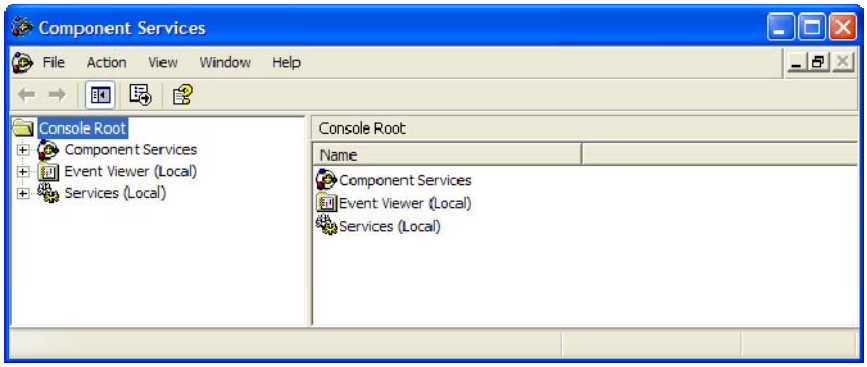

Figure 4-1: Component Services (DCOM) Configuration Tool

Once there, open up "*Component Services*". Within it, ignore COM+ Applications for now, and proceed to "Computers". Click on Computers to get the screen shown in Figure 4-2.

---

[15] http://www.sentech.co.nz/ScenicHelp/dcomsecurity.htm
[16] http://www.opcactivex.com/Support/DCOM_Config/dcom_config.html
[17] http://itcofe.web.cern.ch/itcofe/Services/OPC/GettingStarted/DCOM/RelatedDocuments/ITCODCOMSettings.pdf
[18] http://www.gefanucautomation.com/opchub/opcdcom.asp

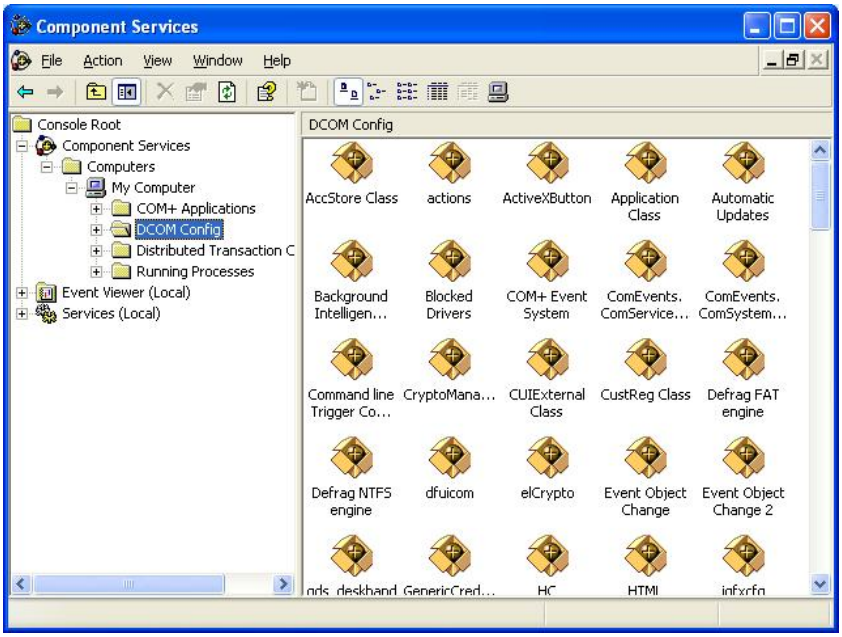

Figure 4-2: DCOM Configuration Screen

Open "*My Computer*", open the "*DCOM Config*", and see what DCOM objects can be configured. Figure 4-3 shows the DSxP Opc Server Simulator which is the server used for this example. On the plant floor you are likely to see the OPC servers you are using, but you may have to dig around for them.

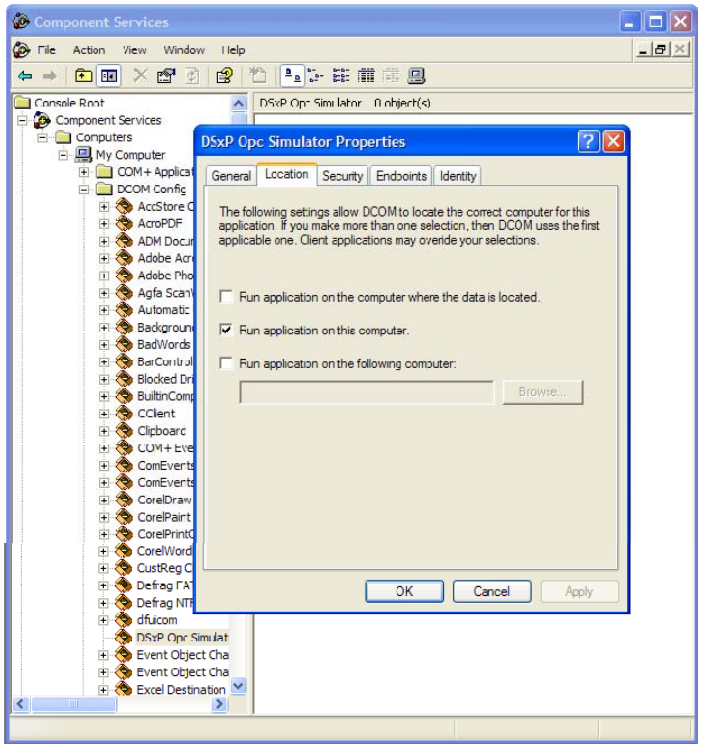

Figure 4-3: The Configuration Properties for an OPC Server

4.2.1 Controlling the Authentication Level

The first change to make is the Authentication Level of the OPC server as shown in Figure 4-4. These Authentication levels are defined as follows:

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

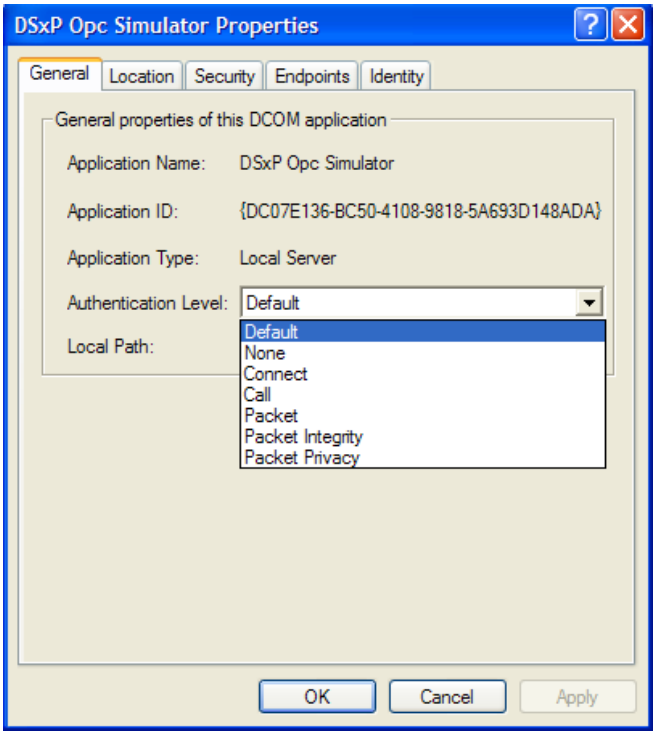

Figure 4-4: General Configuration Tab for an OPC Server

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

4.2.2 Controlling the Location

The "Location" tab lets you configure where the DCOM server can run. Here only the local computer is specified which is the typical situation in most environments. Figure 4-5 illustrates this.

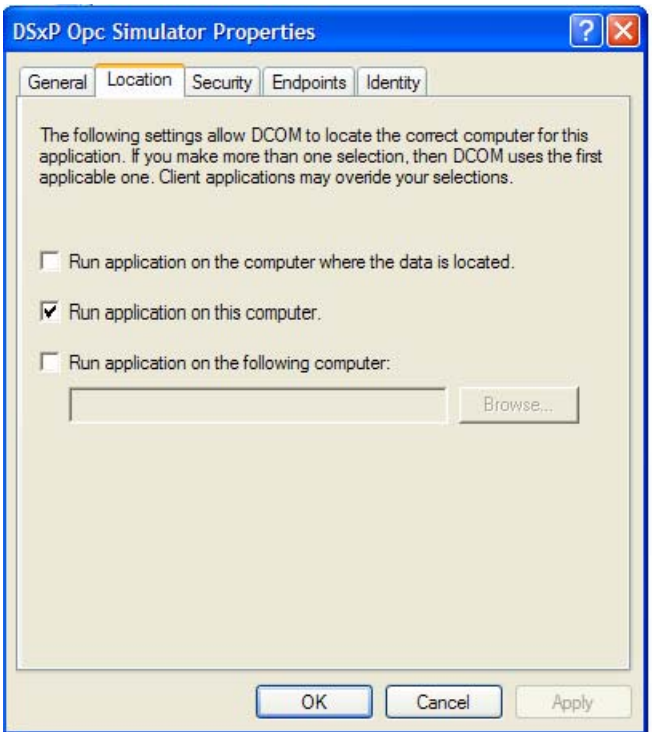

Figure 4-5: Location Configuration Tab for an OPC Server

4.2.3 Managing DCOM Permissions

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

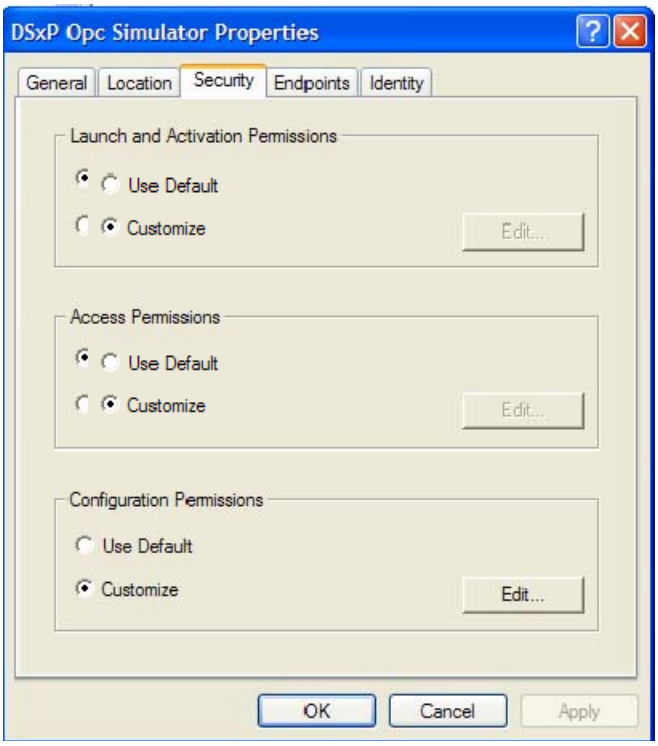

Figure 4-6: Security Configuration Tab for an OPC Server

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

- *Launch Permissions* - Remove all existing entries and add the opcadmin account created earlier. If a particular OPC server is meant only to be used locally, then remote access to that server can also be disabled.

- *Access Permissions* - Remove all existing entries and add the opcadmin and opcuser accounts. Again, if a particular OPC server is meant only to be used locally, then remote access to that server can also be disabled.

- *Configuration Permissions* **-** Remove all existing entries other than the Everyone account. Modify everyone to be read-only, and add opcadmin with full control.

These settings are shown in Figure 4-7. As noted above, if the server or client is only to be used locally (i.e. the clients and servers are all on the same machine) then *Remote* should be turned off.

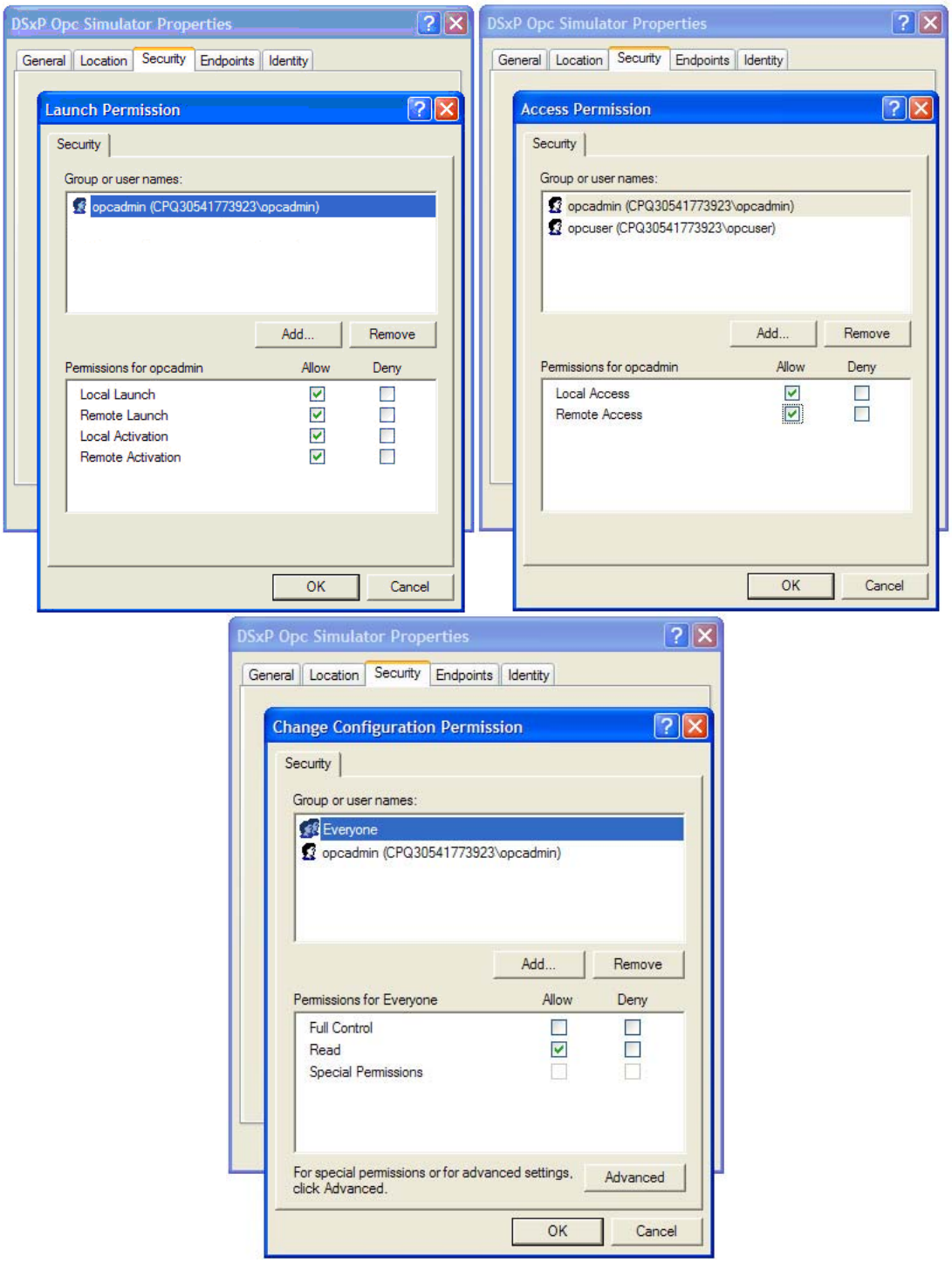

Figure 4-7: Launch, Access, and Configuration Permission Tabs for an OPC Server

4.2.4 Limiting RPC Ports and Protocols

The "*Endpoints*" tab allows you to select what protocols and ports can be used by this server and is shown in Figure 4-8. This tab gives us the possibility to

address one of the more vexing problems in OPC security, namely the problem of dynamic port allocation.

Most other TCP server applications use fixed port numbers to identify all incoming packets. For example, MODBUS/TCP uses port 502 and HTTP uses port 80. This consistency makes firewall rule creation relatively simple – if you want to block all MODBUS traffic through the firewall, simply define a rule that blocks all packets containing 502 in the destination port field.

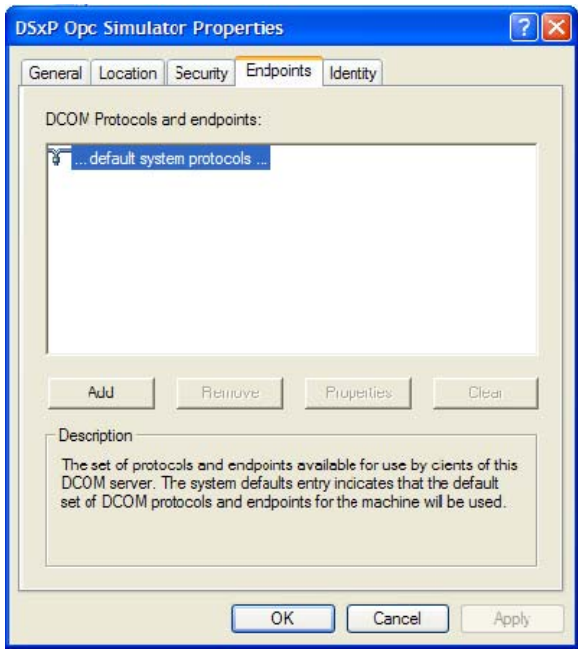

Figure 4-8: Endpoints Configuration Tab for an OPC Server

The default setup for DCOM (and RPC) complicates the situation by allowing the OPC server to dynamically pick its own port numbers. The reason is that while only one web server will typically exist on a given host, there can be multiple DCOM servers on the same device and each needs its own port number. It is certainly possible to have an administrator manually set these port numbers for each server, but early design decisions dictated this might not be an ideal solution, so dynamic allocation became the default.

Today, with security becoming a priority over administrative simplicity, it is worth considering the option of statically setting these ports for each OPC server. Of course it is critical to make sure two OPC servers on the same host do not get set up using the same port number.

Unfortunately not all vendors of OPC products respect the static setting of port numbers, so this technique must be tested carefully. Matrikon and NETxEIB OPC software products worked well with static ports, but several other products did not. Undocumented registry changes did get static setting of port numbers working on a few other vendors' products, but this was very

complex. **Thus it is important is to check with your OPC vendor before trying this technique on a live system.** If they do not support setting of static endpoints, we offer an alternative mitigation in Section *4.3.2 - Restricting TCP Port Ranges*.

If you want to use static port numbers for OPC traffic and your vendor supports them, select "*Add*" on the "*Endpoints*" tab and the screen in Figure 4-9 should appear. Then set the Protocol Sequence to "*Connection-Oriented TCP/IP*" and enter a port value for the static endpoint. Be certain this port number is not used by any other application in the host. In this example we have configured the host so the OPC server application will use TCP port 7000.

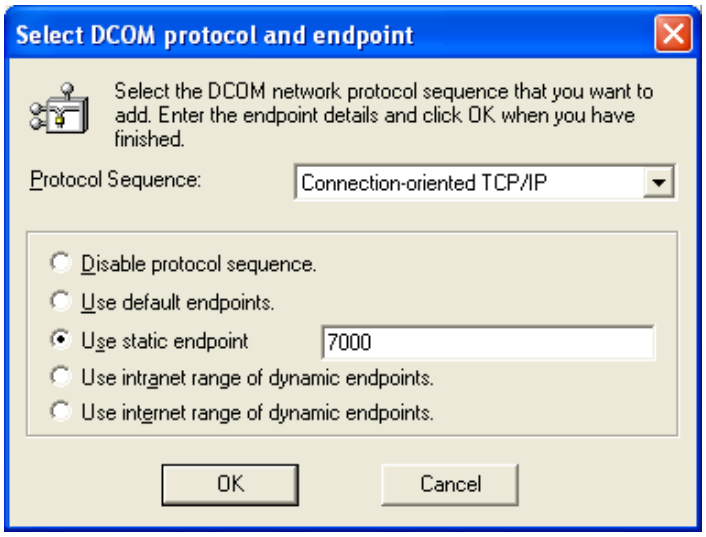

Figure 4-9: Security Configuration Tab for an OPC Server

4.2.5 Setting the OPC Application's Account

Finally, the "*Identity*" tab lets you configure what user account the DCOM application will run under. As shown in Figure 4-10, the OPC software should set to run as the opcuser account.

4.3 RPC Hardening Recommendations

4.3.1 Restricting Transport Protocols to TCP

To make the Remote Procedure Call (RPC) mechanism more secure, it makes sense to restrict the available transport level protocols and to limit the range of potential transport protocol ports. Forcing OPC clients and servers to use only TCP (rather than UDP) will allow intervening firewalls to statefully police TCP streams that carry DCOM traffic. Hence, it is recommended to only list TCP in the list of available DCOM protocols. To do this, edit the "*HKEY_LOCAL_MACHINE\Software\Microsoft\Rpc\DCOM Protocols*" registry entry so that it only contains the item "*ncacn_ip_tcp*".

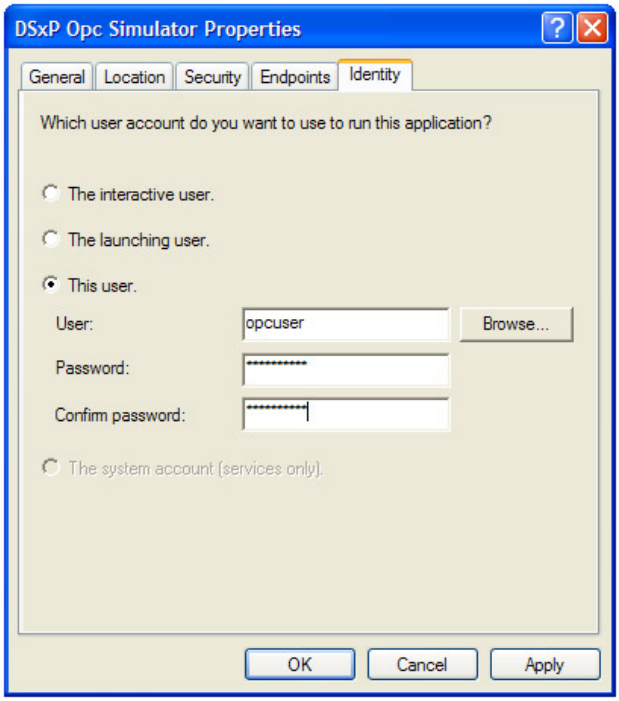

Figure 4-10: Identity Configuration Tab for an OPC Server

4.3.2 Restricting TCP Port Ranges

As an alternative to defining a static port for the OPC servers, one can make changes to the Windows registry that will limit the range of potential RPC ports used by an OPC server and allow simpler firewall rules. For example, administrators can define a small range of ports for RPC to use on the OPC host. This involves making registry changes and rebooting. To change the registry, create an *Internet* key under the following location:

HKEY_LOCAL_MACHINE\Software\Microsoft\Rpc

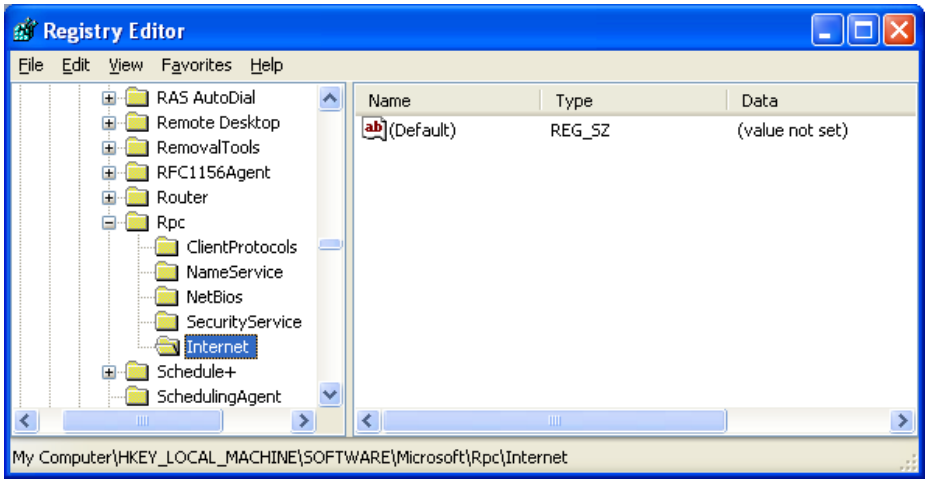

Figure 4-11: Creating a New Registry Key

Next create the following entries in this location:

- Ports (type REG_MULTI_SZ)
- PortsInternetAvailable (type REG_SZ)
- UseInternetPorts (type REG_SZ)

The value for *Ports* should be the desired port range you want to use for OPC servers. For example, you could allocate 100 ports by entering "7000-7100" in Ports. We recommend you use a range of ports above port 5000 since port numbers below 5000 may already be in use by other applications. Furthermore, previous experience shows a minimum of 100 ports should be opened, because several system services rely on these RPC ports to communicate with each other.

The value of *PortsInternetAvailable* should be set to "Y" for the Ports range to be noted. The value of *UseInternetPorts* should also be set to "Y for the Ports range to be noted. It is important to remember this will affect all RPC services and not just OPC applications so check with your vendor before trying this.

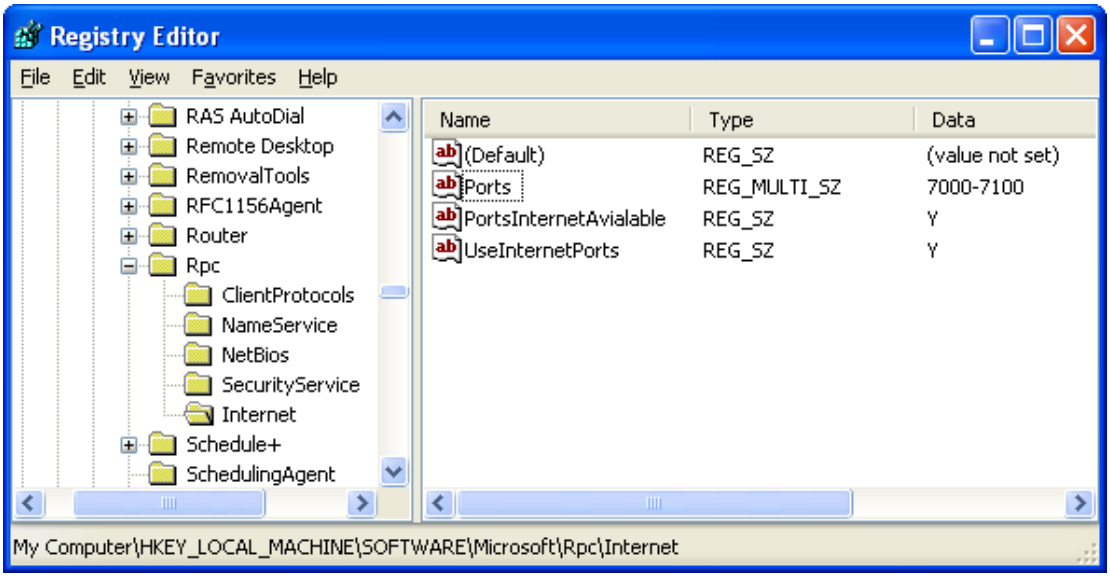

Figure 4-12: Adding the Registry Values

Also note that since OPC uses callbacks, you must use TCP for communications through a firewall if you want this mitigation to work. The reason for this is when the server makes a call to the client, the source port will not be within the range specified about and thus when the client sends a reply to the server's source port, it will not be able to penetrate the firewall. This is not a problem with TCP because most firewalls keep track of TCP connections and permit bidirectional traffic on connections, regardless of the source port, as long as they are opened from a machine on the inside. For

guidance on forcing OPC to use TCP, see Section 4.3.1 *Restricting Transport Protocols to TCP*.

4.4 More Special Considerations for XP Systems

One might assume these configurations are for OPC servers only. Unfortunately this is not the case; starting with Windows XP/SP2, the DCOM configuration must deal with what Microsoft calls "Limits". This means the accounts opcadmin and opcuser have to be added under "Limits" in the global COM security settings for all clients and servers.

To do this we again use the DCOM Configuration Tool found under *Control Panel/Administrative Tools/Component Services*[19] shown in Figure 4-13 . It can also be accessed by starting *dcomcng.exe* from the *Run...* option in the Start Menu.

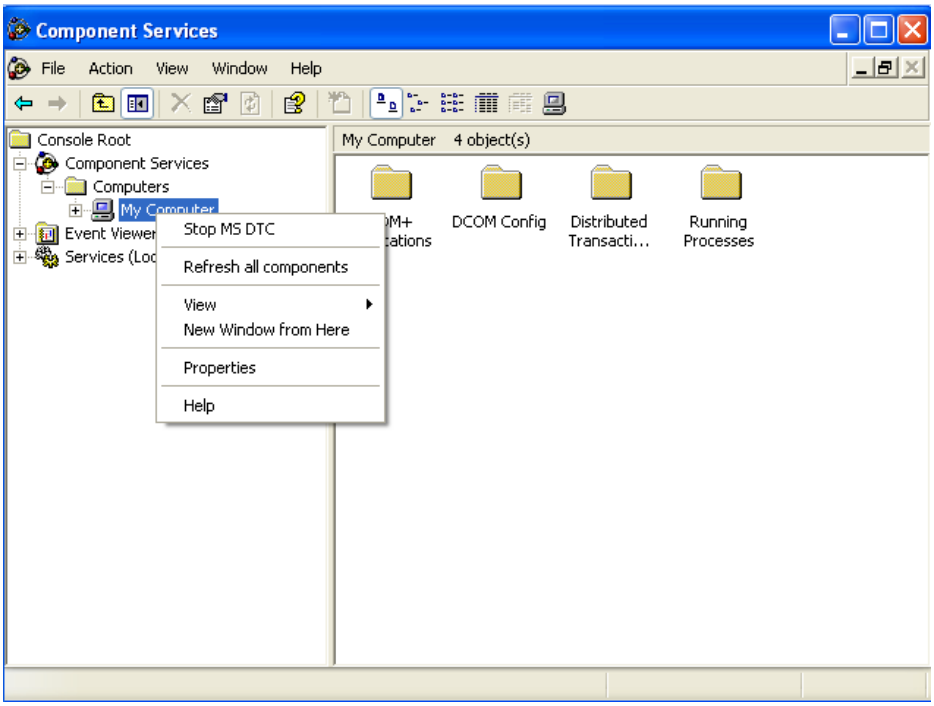

Figure 4-13: Component Services (DCOM) Configuration Tool

Now select the *COM Security* tab and an option to edit the Access Permissions and Launch Permissions will appear (see Figure 4-14). Each of these needs to be edited to add the accounts opcadmin and opcuser. This editing is identical to that described in Section 4.2.3.

[19] http://www.gefanucautomation.com/opchub/opcdcom.asp

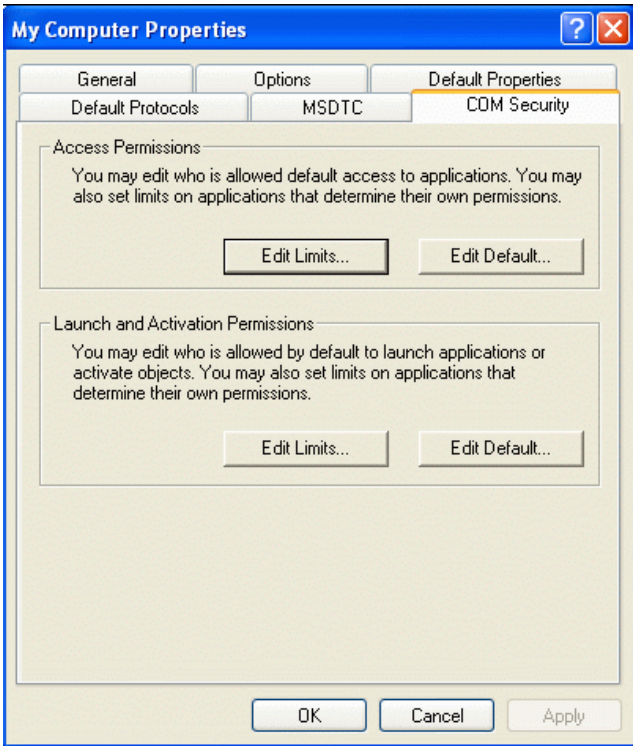

Figure 4-14: COM Security Tab

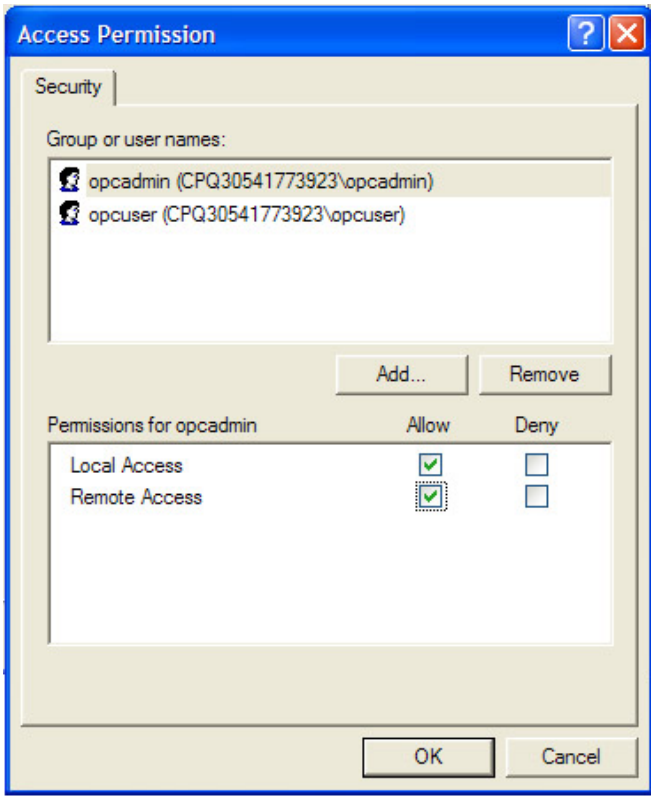

Figure 4-15: Adding opcuser to the Access Permission

5 OPC Host Hardening Verification

Even after applying the techniques for hardening Windows, OPC, DCOM and RPC described in the previous chapter, we are still left with a number of unanswered questions with regard to our OPC server:

- Have the hardening techniques been properly applied?
- What other specific exposures should be addressed?
- When is the system under attack and what kinds of attacks are being used?

To help answer these questions, some active and passive verification techniques can be used. These involve vulnerability scanning using freely available tools and the enabling and monitoring of Windows auditing features. Note, it is difficult to completely automate this verification process so a manual process is used in the following examples.

5.1 Windows Service and Open Port Determination

The first task is to determine if the configuration of the OPC servers has resulted in the correct servers starting, and if using static ports, if the ports are set correctly. There are many tools to do this, but one of the simplest is the built-in Windows utility "NETSTAT".

Netstat displays all active TCP connections, the ports on which the computer is listening and a number of useful Ethernet, IP and TCP statistics. To use Netstat, simply open command line window and type "netstat –o". The "-o" parameter displays all active TCP connections and includes the process ID (PID) for each connection. You can find the application based on the PID on the Processes tab in Windows Task Manager. Other similar tools include "fport" from www.foundstone.com.

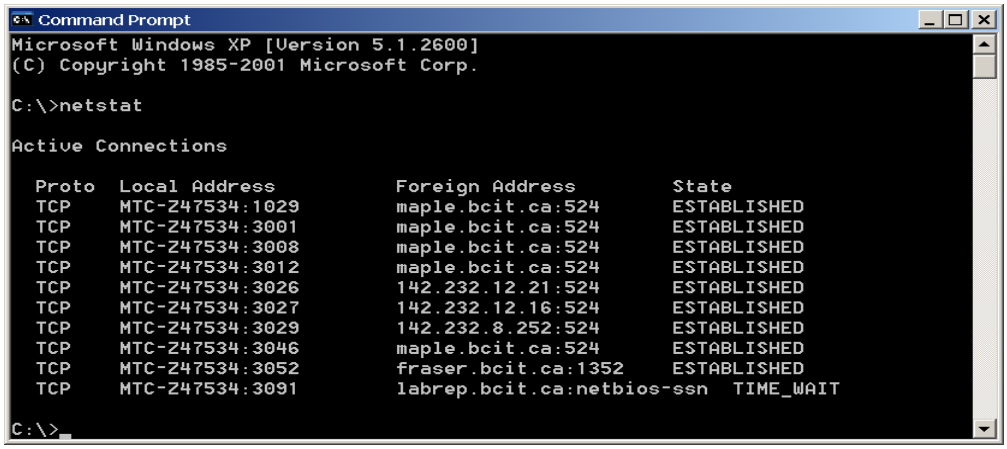

Figure 5-1: Typical NETSTAT Output

5.2 Windows Event Log Analysis

Windows 2000, Server 2003 and XP provide a rich set of features for identifying malicious activity and policy violations. Unfortunately, many are not enabled by default. Furthermore, typically the challenge is not in getting the data, but in deciding which information is most valuable when monitoring OPC based applications.

The first step is to enable Auditing to identify and log malicious activity against OPC Servers. On standalone systems, auditing is configured using the *Local Security Policy*. Although we identify a minimal set of Audit Policy recommendations, changes are often required. However in general the settings in the table below will work well.

Policy	Recommended Security Setting	Discussion
Audit Account Logon Events	Success and Failure	Since we are differentiating between the user account necessary to remotely access the OPC/DCOM components (opcuser) and the application administrator (opcadmin), it makes sense to log both successful and failed events. Note that interactive logins on the OPC server should be a relatively uncommon.
Audit Logon Events	Success and Failure	
Audit Object Access	Failure	Enabling object access auditing generates a significant amount of activity; so only failed attempts to access OPC objects should be enabled.
Audit Policy Change	Success	

Table 5-1: General Auditing Settings

Since login events are limited to interactive console logons, we must enable per object auditing on core OPC components. In *Security Options*, enable "*Audit: Audit the access of global system objects."* The object audit settings should be as listed in the table below.

Object	Settings
OPC Server Browser (OPCEnum.exe)	Traverse Folding / Execute File: Failed
Opc_aeps.dll, opcbc_ps.dll, opccomn_ps.dll, OPCDAAuto.dll	Traverse Folding / Execute File: Failed
OPC Server Application	Traverse Folding / Execute File: Failed

Table 5-2: Object Auditing DCOM/OPC files

It is important to remember that in order to get the most accurate picture of hostile activity across the network and on multiple clients and servers, we must be able to integrate data from a variety of sources, including routers, firewalls, intrusion detection/prevention systems, Windows event logs, and application specific logs generated by OPC servers. This can be a challenge given the different terminology, different message formats and different types of data (such as IP addresses, port numbers, GUIDs, application names, etc) generated by all these systems. This is a non-trivial task where more research and product development is needed.

5.3 Vulnerability Scanning

Apart from enabling and analyzing security logs on OPC client and server systems, we recommend that active methods be used to assess hosts for security deficiencies. The tools and techniques described in this section can identify a number of security gaps.

The focus of this section is only scanning for misconfiguration vulnerabilities in DCOM and OPC Servers and not identifying other vulnerable services or components that need to be upgraded. When evaluating existing techniques, we discovered that existing tools fall short when it comes to providing information about the state of DCOM and OPC security and at times they provide conflicting information. Two popular tools we used to check the security of OPC hosts are Microsoft's Security Baseline Analyzer and Tenable Network Security's Nessus Scanner. Other scanners can be used as well.

5.3.1 Microsoft Security Baseline Analyzer 2.0

The Microsoft Baseline Security Analyzer (MBSA) is a free tool useful for checking systems to ensure they are set up in accordance with Microsoft best practices and to ensure the basic Windows hardening techniques described above are followed. It also helps to identify gaps in Microsoft system and application updates. July 2005, Microsoft released version 2.0 of this tool, which, according to the Microsoft web site, is now used in many commercial security products.

We recommend using MBSA to scan the OPC server locally since it provides the most useful information and is the least intrusive. Scans can also be conducted remotely if proper domain/local user credentials are available, remote registry browsing is enabled and access to the well known Microsoft TCP and UDP ports is available. Unfortunately this would involve practices that we specifically advise against for OPC hosts, thus we can not recommend remote MBSA scans.

MBSA provides an easy-to-read report using simple pass/fail criteria and can be sorted according to severity. Although MBSA is by no means

comprehensive, we were disappointed to see it contains no analysis of DCOM configuration weaknesses. However it is still a useful tool.

As a test, we scanned our OPC server running a completely patched Windows 2000/SP4 in the default state without any of our hardening recommendations applied. It provided us with a report that included the following vulnerabilities:

1. Administrative Vulnerabilities
    - *Local Account Password Test* – MSBA determined that we were using weak passwords for our opcadmin, and opcuser accounts.
    - *Restrict Anonymous* – MSBA detected that we had RestrictAnonymous set to 0, which allowed null sessions to be established.
    - *Password Expiration* **–** MSBA determined that password expiration was not enabled. However password expiration may not be appropriate for control system environments.
    - *Windows Firewall* – MSBA identified that the built-in Windows 2000/XP firewall was not in use.
    - *Update Compliance* - MSBA provided an exhaustive list of security updates and hotfixes.
2. Additional System Information
    - *Services* **–** identified a number of unnecessary services running on the server.
    - *Shares* **–** identified old share names and permissions that were not required.

Although MBSA checked for common operating system level hardening issues, MBSA provided no DCOM-specific information and only provided information on Microsoft security updates. It did not list any 3rd party software in the reports. Still it is a very useful tool.

5.3.2 Nessus Vulnerability Scanner

Nessus is one of the most popular vulnerability scanning tools on the market. Although Nessus is a general-purpose scanner, it includes checks for multiple network layers and different types of devices. It features a large number of vulnerability checks for Windows and Windows-based applications. This is especially true if Administrator level credentials are provided.

One word of caution - Nessus has a track record of crashing embedded devices such as PLCs and RTUs and even some poorly implemented Windows applications. Sometimes the operating system can become unresponsive and unreliable during Nessus scans. **Thus we recommend these scans only be run on offline systems.**

Our scans of a default OPC Server configuration on a partially-patched Windows 2000 SP4 Workstation produced a large amount of information (after we provided Administrator level credentials to Nessus).

1. *Port Scans* – Given the use of multiple non-standard ports, port-scans against OPC are not very useful, but do help identify unnecessary system services (IIS etc) that may be running on an OPC host. They also help confirm if the TCP port number restrictions in suggested in Section 4.2 and 4.3.2 are effective.

2. *SMB Share Enumeration* – If anonymous browsing is enabled (or login credentials are provided) Nessus identifies remotely accessible shares.

3. *RPC Enumeration* – The RPC scanning module provides output gathered from probes to RPC/DCE. No useful information about OPC applications could be gained from the RPC scans during our tests.

4. *Password Policy & History* – For this module, passwords that have changed and other enforcement mechanisms such minimum length, strength, force logoff time, and number of logins until lockout are reported. Some of these may not be appropriate for control system environments.

5. *Remote Registry Access* – Nessus determined whether or not remote registry browsing is possible.

6. *User Enumeration* – Nessus remotely determined the Security Identifiers (SIDs) and names of identified privileged and unprivileged user accounts.

7. *Known Vulnerabilities in Windows and 3rd Party Components* – Using "local" and remote checks, Nessus identified potentially vulnerable software versions.

8. *Remote Service Enumeration* – In addition to standard services (Computer Browser, DHCP Client, etc.) Nessus identified the OPC Server Browser and OPC Server when run as a service.

9. *Installed Software* – Nessus provided the name and version information on installed OPC client and server applications, in addition to other 3rd party software.

5.3.3 Audit Files for Nessus Vulnerability Scanner

Tenable Network Security has developed Nessus plugins that will audit the configuration of a device under test to an established configuration. Digital Bond has created an audit file based on the security recommendations in white paper. The audit file, available as Digitial Bond subscriber content, will allow an OPC user to determine if their OPC implementation meets the good practice security recommendations in Part 3 of the OPC white paper series.

The audit capability is available in Nessus 3 to Tenable Direct Feed subscribers and Security Center users. The "Policy Compliance" plugins (ID's 21156 and 21157) must be enabled the credentials for an account with Windows Administrator privileges must be entered into Nessus. The audit file for OPC servers is added via the compliance tab.

Some of the settings require customization per OPC server. For example, auditing the DCOM permissions requires the CLSID of the OPC server be entered into the audit file. This varies by vendor and product, but it is easily determined on the OPC server and Digital Bond has a large list of CLSID's. Additional instructions on the use and results from the OPC security audit file are available at Digital Bond's website.

6 A Summary of OPC Host Hardening Practises

6.1 An Action Plan for Hardening OPC Hosts

In earlier sections of this white paper we pointed out the best way to harden an OPC host is to do it in stages. One begins by locking down the operating system that the OPC server or client resides on, which in most cases is some version of Windows. Next, one should tackle the OPC applications by restricting the OPC accounts, limiting DCOM object access and constraining RPC protocol options. Lastly, to verify the hardening has been successful, it is important to check for remaining security vulnerabilities using security analyzer tools.

While it seems like a lot of effort, it is important to remember that effective security does not start or stop with these three steps. Security is an ongoing process and thus we recommend the following overall process for users of OPC technology:

1. **Determine whether OPC or DCOM is in use in your facility:** This may seem like a trivial task, but some applications may not adequately document what lower level API is used. We located at least one company that was unaware that DCOM was in use on its control system because it was bundled into a control product with a different name.

2. **Document how OPC or DCOM is deployed in your facility:** This includes determining what systems and devices communicate using OPC and how critical this communications is for your operation. List all OPC servers and client applications on each host in your facility.

3. **Evaluate possible operating system hardening practices:** Sections 3 and Section 6.2 (below) highlight common areas of concern and good practices for operating system hardening. Also investigate guidelines from your IT department and other bodies such as NIST and US-DoD[20].

4. **Select the appropriate operating system hardening practices for your environment:** Chose the hardening practices effective for your facility from the results of step 3.

5. **Evaluate possible OPC/DCOM hardening practices**: Review the guideline listed in Sections 4 and 6.2 of this report. Also review the recommendations of your OPC vendor and other bodies such as the OPC Foundation, for security settings.

[20] For example see http://csrc.nist.gov/itsec/SP800-68-20051102.pdf and http://iase.disa.mil/stigs/checklist/W2K3_Checklist_V5-1-10_20070525.zip

6. **Select appropriate OPC/DCOM hardening practices for your environment:** Chose the OPC/DCOM hardening practices effective for your facility from the results of step 5.

7. **Test hardening practises on offline test systems:** Make sure that you have tested any hardening techniques on non-critical systems and conduct functional testing to ensure OPC servers are operating properly. Only after you are sure that they will not impact your process should you deploy them on critical systems.

8. **Consult with your vendor/system integrator to address possible security incompatibility issues:** Unfortunately some applications may not function properly when either OS or OPC/DCOM hardening practices are applied. Work with your vendor/integrator to determine and resolve these issues.

9. **Implement hardening practises on operational systems:** Once all hardening techniques have been confirmed on offline test systems, deploy them on online system. Then conduct functional testing to ensure all OPC servers are operating properly.

10. **Verify the deployed OPC/DCOM and OS hardening practices:** After implementing hardening practices, make sure they are operating as expected using techniques described in Section 5.

11. **Implement other security countermeasures:** The host hardening guidelines described in this document are not sufficient on their own - it is prudent to have a defense-in-depth approach to security. This will include other solutions such as patch management, firewalls, antivirus deployment and so on.

12. **Monitor OPC hosts for intrusions or unusual activities:** This can be done using host and network based monitoring tools as well as Windows Auditing and Logging tools as discussed in Section 5.

6.2 Summary of High Risk Vulnerabilities and Mitigating Good Practices

Using the results from White Paper #2, we have summarized the key findings relating to common operating system vulnerabilities that are most critical for OPC deployments. We have then added the recommended practices for mitigating them based on the guidelines in this report. Please remember this is only a summary and is by no means a complete list of vulnerabilities or mitigations.

Vulnerability	Good Practice
Inadequate Patching of Host	Follow guidance from OPC vendor and existing organizational guidelines. (Section 3.1)
Unnecessary Services	Disable unnecessary services and ensure OPC hosts are single purpose platforms. (Section 3.2)
Unnecessary Access to Host from Other Devices	Use Windows IP Filtering (Section 3.4)
System Enumeration & Profiling	Disable Unnecessary Services (Section 3.2) and Confirm with Vulnerability Scanning (Section 5.3)
Weak Passwords	Beyond the scope of this document. Follow established industry or organizational best practices.
Remote Registry Access	Harden registry and disable remote editing (Section 3.5). If possible disable remote browsing.
Inadequate Security Logging	Enable system auditing for OPC and DCOM objects to identify unauthorized access attempts. (Section 5.3)

Table 6-1: High Risk O/S Vulnerabilities and Possible Mitigating Practices

Vulnerability	Good Practice
Lack of Authentication for OPC Server Browser	Disable OPC Server Browser and Anonymous Login after initial configuration (Section 4.1)
OPC Server Executes with Excessive Permissions	Configure OPC Server components to run with restricted permissions (Section 4.2)
Overly Permissive Settings for OPC Server Browser	Remove Everyone access to OPCEnum and require authenticated users and/or follow vendor recommended practices. (Section 4.2)
Unnecessary Protocol Support for OPC Server	Force RPC to only use TCP for transport and either use static ports or restrict port ranges (Section 4.3.1)
Excessive Open TCP ports on OPC Server	Force RPC to either use static ports (Section 4.2) or restrict port ranges (Section 4.3.2)
Lack of Confidentiality in OPC Communications	Enable "Packet Privacy" if possible (Section 4.2)
Lack of Integrity in OPC Communications	Enable "Packet Integrity" if possible. (Section 4.2)
Use of Historically Insecure Transport	Ensure patching and upgrade to OPC-UA when available.
OPC Security Configuration Lacks Fine Grained Access Control	Can not be addressed at this time

Table 6-2: High Risk DCOM/OPC Vulnerabilities and Possible Mitigating Practices

6.3 Some Final Thoughts

Based on our research, the challenges of securing OPC deployments are clear. The inherent architectural issues with the current versions of OPC, the default security posture and poor compliance to DCOM security settings of many OPC products, and the lack of unambiguous guidance with regard to security, all contribute to the difficulties of securing OPC deployments in most companies.

This does not mean OPC users should throw up their hands in despair. OPC's reliance upon the Microsoft platform is both a blessing and a curse - while Windows has flaws, we were able to uncover a wealth of practices for hardening Windows servers that can be applied to OPC clients and servers. Furthermore, the fact that a few OPC vendors are providing good security guidance and a degree of hardening during the installation process shows that it is possible to reduce the pain of security that many users are feeling.

What is needed from the vendor community is an immediate and focused effort towards improving OPC/DCOM installation processes and security guidance. Waiting for the day when there is widespread availability and deployment of the more secure OPC-UA is not a solution – that is simply too far in the future to help today's OPC end-users.

End-users can also do much to improve their security posture with regards to OPC. First, many of the vulnerabilities in OPC hosts that we discussed in White Paper #2 are well within the control of the knowledgeable end-user. Using a well-defined security plan, such as the one supplied in this document, the end-user can significantly reduce their OPC security risk. Second, the end-user community can start demanding better OPC guidance from their vendors – as we noted in White Paper #2, a few vendors already do an excellent job, so the challenge is to move the remaining vendors in this direction. Only end-users wielding the power of the purchase order can make this happen in a timely fashion.

Finally, it is critical the OPC end-user keep both operating systems and OPC applications as current as possible. The security of most software products have improved significantly in the past five years. This is especially true for Microsoft Windows and various OPC products. The eventual release of OPC-UA based software is likely to significantly help reduce the security effort and risk currently faced by industry today. This can only happen if the community embraces the new UA technologies over the next few years.

7 Areas for More Research in OPC Security

Since the focus in this project was on the hardening of OPC hosts, a number of other interesting security possibilities were not pursued during our research. We feel that these are worth investigating in future studies and have listed them below.

7.1 Firewall and Network Related Solutions for OPC Security

Readers may have noted that there is no discussion in this white paper on best practises for firewall configuration for OPC systems. This was considered out of scope for this project focusing on OPC hosts, but is an area urgently needing further research.

7.2 OPC Tunnelling Solutions for Security Robustness

Given the difficulty in developing firewall rule sets for DCOM-based applications (and the challenges of OPC use across multiple Windows domains), there are a number of 3rd party products or built-in techniques to tunnel OPC/DCOM traffic over a single port. Although these techniques may make the life of the systems administrator simpler, it is not clear if they improve security. Detailed analysis of these tunnelling solutions is urgently required.

7.3 Network Intrusion Detection/Intrusion Prevention Signatures

In the past few years intrusion detection signatures for SCADA protocols such as DNP3 and MODBUS have been developed based on likely misuse of valid protocol patterns. We believe that a similar approach could be conducted for OPC to alert on unauthorized attempts to access OPC Server GUIDs, Program IDs, or other client or server messages.

7.4 Enhancements to Network Vulnerability Scanners

Although scanning tools such as Nessus and MBSA proved useful for identifying Windows OS vulnerabilities, very little DCOM/OPC specific information was provided by these tools.

7.5 Research Implementation Vulnerabilities in OPC Components

Over the past several years, a number of tools have been released that attempt to find implementation flaws in ActiveX and COM components. Although Internet Security Systems Incorporated's Scanner/Intrusion Detection System (IDS) has a signature for an OPC Buffer overflow[21], to our

[21] http://xforce.iss.net/xforce/xfdb/13393

knowledge no implementation flaws have been disclosed in the OPC Foundation Components such as Proxy/Stub DLL's or OPC Applications.

7.6 Use of Domain Isolation in Control Environments

Domain Isolation is technique based on IPSec and Group Policy to prevent access from untrusted devices to trusted devices on a corporate network. While very promising on the surface, just how effectively this technology can be used in the industrial controls environment requires additional research.

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 Européen 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: An 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.