Global Information Assurance Certification Paper

GIAC Certified Windows Security Administrator (GCWN)
Practical Assignment
Version 3.2, Option 2

Windows 2000 Vulnerability Analysis, Discovery and Patch Management In a Small or Resource-Limited Organization

By Richard A. Partridge

Submitted November 5, 2003

Table of contents

1.0 Abstract 1
1.1 Lab Environment 1
2.0 Vulnerability Analysis 1
2.1 Vulnerability Information Sources 2
3.0 Patch Application with QChain 4
3.1 Basic Procedure 5
3.2 Supported Patches 5
3.3 Unsupported Patches 8
3.4 CD Based QChain 8
4.0 Deployment And Integration Of Software Update Services 16
4.1 Server Setup 16
4.2 Client Setup 17
4.3 Patch Approval and Deployment 17
4.4 SUS Network Traffic Analysis 18
5.0 Vulnerability Discovery With Free & Open Source Tools 19
5.1 Vulnerability Discovery 19
5.2 Planning Vulnerability Scanning 20
5.3 Free Vulnerability Discovery Tools 24
6.0 Conclusion 26
Works Cited 27

1.0 Abstract

This paper will detail vulnerability discovery, analysis and mitigation as it applies to a small business or other organization with a limited budget. It begins with a technique to be used to identify existing and new vulnerabilities, to analyze them and determine when to apply the mitigating action. Information on the Microsoft QChain functionality and its use to prepare a host for connection to a Local Area Network (LAN) will follow, focusing on a CD-based method of hardening the host before plugging in the network cable. It will also show the workings of Microsoft technologies to maintain the patch status using QChain and Microsoft Software Update Server and Automatic Updates. Finally, this paper will review a number of vulnerability assessment tools with emphasis on those available free of charge and how to use them.

1.1 Lab Environment

The environment created for this paper is a fictitious small company with one full time system administrator and a limited budget dedicated to information technology. A small lab was used consisting of three personal computers running Microsoft Windows 2000 Advanced Server (2) and Windows 2000 Professional (1). One server was configured as a Domain Controller and DNS server for the domain ate-richard-partridge.eng, one was configured as a member server and the other configured as a workstation for employee use. All operating system baselines are Windows 2000 with Service Pack 4. The computers were networked through a four port switch and connected, when necessary, to the Internet through a Cable/DSL Router.

2.0 Vulnerability Analysis

What to patch and when to apply the patch are questions that must be answered. In spite of arguments of which operating system is better the truth is the operating system, regardless of vendor or source, is only as secure as the person administering it makes it. Determining the steps needed to secure a host before connecting it to the Internet and to maintain its security posture can be taxing for a seasoned information security professional and even worse for a system administrator in a resource-limited organization.

Code Red, SQLSlammer, and Lovsan/MSBlaster were devastating. The sad part is Microsoft had released patches for the vulnerabilities long before the exploits showed up. That each exploit was so successful was made possible by un-patched hosts. In one case the author is aware of a web server compromised by Code Red was attacking the organization. It didn't take long to contact the small business running the compromised web server. It took even less time for the business owner to state there was nothing wrong with his web server. A quick email to the appropriate Internet Service Provider resulted in the web server being blocked. The author later got a call from that same business owner. He ran the server himself and had bought it from a

vendor who said it was easy and proved it by standing up a web site before close of business. It was running Windows NT Server 4.0 with Service Pack 5 and Internet Information Server 4.0. As far as the owner knew there had never been any patches installed. When asked about anti-virus software the owner said the vendor told him it wasn't necessary and it would cause problems with the web site they developed for him.

Stories like this abound and the number grows from day to day but it doesn't need to be that way. With a little more time spent on properly administering a server that server can be maintained with a very reasonable level of security. One of the first things to do is find the information needed to make good decisions regarding what to patch and when.

2.1 Vulnerability Information Sources

The main source for information regarding operating system vulnerabilities is the operating system vendor, in this case, Microsoft. Every person who administers a Windows server should subscribe to Microsoft Security Bulletins and read and analyze each bulletin released.

Another source of timely, reliable information is the Computer Emergency Response Team Coordination Center (CERT-CC) at the Software Engineering Institute of Carnie-Mellon University. CERT-CC publishes advisories that detail vulnerabilities and mitigating actions. This is a vital source of information worth subscribing to.

The SANS Critical Vulnerability Analysis list should also be subscribed to along with SANS NewsBites. The Critical Vulnerability Analysis focuses on actions other organizations have taken to mitigate the risk associated with vulnerabilities.

Common Vulnerabilities and Exposures (CVE) is "A list of standardized names for vulnerabilities and other information security exposures - CVE aims to standardize the names for all publicly known vulnerabilities and security exposures" (The MITRE Corporation).

CVE is the source of tying various vulnerability descriptions and other sources of information together. An example is the vulnerability announced in *Microsoft Security Bulletin MS03-026, Buffer Overrun in RPC Interface Could Allow Code Execution (823980)*. A search at CERT-CC reveals this same vulnerability is titled *CERT® Advisory CA-2003-16 Buffer Overflow in Microsoft RPC*. Bugtraq ID 8205 covers the vulnerability as well and is titled *Microsoft Windows DCOM RPC Interface Buffer Overrun Vulnerability*. Though each organization is reporting on the same vulnerability the title assigned by each is at least slightly different. In some cases the different titles are so different as to be difficult to match to the vendor alert.

CVE ties it all together. Each message mentioned above refers to CAN-2003-0352, *Buffer overflow in a certain DCOM interface for RPC in Microsoft Windows NT 4.0, 2000, XP, and Server 2003 allows remote attackers to execute arbitrary code via a malformed message, as exploited by the Blaster/MSblast/LovSAN and Nachi/Welchia*

worms. The list of references lists each message mentioned above and many more, tying the different titles and descriptions together with the identifier CAN-2003-0352.

Finally the ICAT Metabase, which is a searchable vulnerability database, provides information about MS03-026 which is similar to CVE. The main identifier for each vulnerability in ICAT is the CVE (CVE/CAN) number. ICAT is well worth visiting when analyzing vulnerabilities.

When Microsoft prepares a patch for a vulnerability it is released by a Microsoft Security Bulletin. Security Bulletins generally contain an analysis of the vulnerability, a severity rating, Frequently Asked Questions and links to the patch and the accompanying Knowledge Base article. The text version of the security bulletin that arrives in email has links to the detailed bulletin and an end user version of the bulletin.

Before beginning review and analysis of vulnerability a method of determining when to apply the patch is needed. Microsoft makes appropriate recommendations in each security bulletin but a simple analysis grid may be helpful as well. The grid presented in Table 1 below compares the result of successful exploitation of the vulnerability to the ease of exploitation and provides a risk rating. Based on the risk rating a timeline for application of the patch can be set: High < seven days; Medium < 30 days; Low, administrator discretion.

Result - Ease	Public Script	Easy	Moderate	Difficult
Administrator, user, DoS, run code	High	High	Medium	Medium
Enumerate resources, other	High	Medium	Medium	Low

Table 1

The results are listed in two categories. The first is administrator level access, user level access or permissions of the currently logged on user, Denial of Service of any kind or the ability of the attacker to run any code in any context. These results also include the ability to take any action based on the level of access or to view, copy, or modify any data or system files. The second category, enumeration of resources and other, includes obtaining the results of a successful null session, listing the shares on the target and determining any other facts about the target. While these are broadly based risk ratings they do provide a baseline for an administrator to adjust for the organization.

When a Microsoft Security Bulletin is received the first step is to determine if any of the organization's systems are affected. A look at the list of affected systems in the bulletin should result in a quick decision. Next is the process of determining the ease of exploiting the vulnerability. Analysis of more than once source of information regarding vulnerability exploitation is important. Is there a script available publicly to exploit the problem? Can the vulnerability be exploited from the console? Does exploitation require extensive programming skills? Try to answer these questions and assign a value of Public Script, Easy, Moderate or Difficult to ease of exploitation. With this information the grid in Table 1 can be used to assign an organization risk and a timeline

for taking action. Here are two examples based on Microsoft Security Bulletins MS03-034 and MS03-039.

Microsoft Security Bulletin, MS03-034, Flaw in NetBIOS Could Lead to Information Disclosure (824105), details a vulnerability that will allow data exposure. The maximum severity rating is Low. Further review of the bulletin reveals exploitation of the vulnerability may be difficult and that only random data from RAM may be exposed (Microsoft Corporation). Therefore Ease of exploitation should be categorized as Difficult and successful exploitation result should fall under Enumeration, other. By applying this information to the grid in Table 1 it can be determined the risk rating is Low and the patch should be applied at the discretion of the administrator.

Microsoft Security Bulletin, MS03-039, Buffer Overrun in RPCSS Service Could Allow Code Execution (824146) details a vulnerability which would allow the attacker's choice of code to run or result in a Denial of Service. The first thing of importance is the severity rating of Critical. Further, Microsoft recommends the patch be applied immediately. A review of the details shows successful exploitation will result in the ability to run code and Denial of Service (Microsoft Corporation). Applying these details to the grid in Table 1 results in a risk rating of High and the need to apply the patch in less than seven days.

Once the analysis is complete the process of actually protecting the affected assets begins. In most cases this means the appropriate patch must be applied. As always, apply new patches to a test system first. If no problems are discovered develop a deployment plan and proceed to apply the patch to all affected production systems. Unfortunately, even many well funded organizations cannot justify the cost of dedicated lab systems which mirror their production systems for the purpose of testing patches. That being the case it is recommended the patch, or any mitigating action, be applied to the least critical system first then to progressively more critical systems until all are patched.

3.0 Patch Application with QChain

"The QChain tool allows you to install multiple security updates without having to reboot between installations. The QChain tool evaluates the drivers, DLLs, and executable files updated by each security update and ensures that only the most recent versions of the files are maintained after reboot" (Smith 493).

The functionality of the utility is built into Windows 2000 Service Pack 3, Windows XP and Windows Server 2003 (Microsoft Corporation 262841). QChain functionality in Windows 2000 and beyond can ease the patch application process, especially on new, out-of-the-box computers. Before going into detail on using QChain some background is in order.

QChain is available for Windows NT 4.0 and all NT based operating systems released since. The use of QChain on Windows NT and Windows 2000 prior to Service Pack 3 requires the actual executable, QChain.exe, whereas QChain functionality is an integral part of Windows 2000 with Service Pack 3 and later, Windows XP and Windows Server 2003. In the latter case the use of the executable, QChain.exe, is not needed (Microsoft Corporation 262841). This paper only discusses the use of QChain as it applies to Windows 2000 Service Pack 3 and later. The actual QChain.exe executable will not be used.

3.1 Basic Procedure

The basic steps to follow to use QChain functionality are detailed in *Microsoft Knowledge Base Article 296861, How to Install Multiple Windows Updates or Hotfixes with Only One Reboot*.

The procedure requires the Hotfixes to be downloaded to a local shared folder either on the target host or on another host on the LAN. The batch file written to complete the installations then lists the path to the Hotfixes and installs each. Once the chained installation is complete the host must be rebooted to complete patch installation.

An example batch file is provided as follows:

```
"@echo off
setlocal
set PATHTOFIXES=E:\hotfix

%PATHTOFIXES%\Q123456_w2k_sp4_x86.exe /Z /M
%PATHTOFIXES%\Q123321_w2k_sp4_x86.exe /Z /M
%PATHTOFIXES%\Q123789_w2k_sp4_x86.exe /Z /M"
```

(Microsoft Corporation 296861) (quotation marks not used in batch file instructions)

This example sets a temporary environment variable to the path. It then installs each patch with the /z and /m switches which prevent restart and run the patch in unattended mode respectively. It is important to remember to use the appropriate switches to not restart and run in unattended mode. Once the chained patches are installed the computer must be restarted for patch application to take affect.

3.2 Supported Patches

Unfortunately, not all patches released in Microsoft Security Bulletins are supported by QChain functionality. Supported patches are those that apply directly to the operating system. They are identified by a specific naming convention that has changed as operating system evolution has progressed. According to the *Microsoft Windows NT and Windows 2000 Hotfix Installation and Deployment Guide,*

> "Windows 2000 Hotfix.exe programs are named according to the following convention:
> Q######_XXX_YYY_ZZZ_LL.exe

where:
- Q###### = the Microsoft Knowledge Base article number (for example, Q123456)
- XXX = the operating system
- YYY = the service pack level
- ZZZ = the hardware platform
- LL = the language" (Microsoft Corporation)

This has changed to the current naming convention for Windows operating system patches as seen in the file name for the patch associated with MS03-039 for Windows 2000: Windows2000-KB824146-x86-ENU.exe.

The change took place when the hotfix installer engine for operating system patches changed from Hotfix.exe to Update.exe.

> "Microsoft packages hotfixes in an automatically installed format. You can install a hotfix by running the hotfix program file. The hotfix program file extracts the hotfix files and runs the Hotfix.exe or the Update.exe program.
>
> Hotfixes included in Windows 2000 Service Pack 3 or earlier use the Hotfix.exe program.
>
> Hotfixes included in Windows 2000 Service Pack 4 or later and hotfixes released after Service Pack 4 use a new engine to install hotfixes. Windows XP hotfixes also use this hotfix installation engine. The file name for the new installation engine is Update.exe" (Microsoft Corporation 262841).

Update.exe includes the QChain functionality.

The most positive identifier of supported patches is the set of switches supported by the patch. To see this, just run the patch from the command line with the /? switch. The output in Figure 1 below is from Q312895i.exe /?. Notice Windows 2000 Hotfix Setup in the Title Bar and HOTFIX followed by the options and an explanation of each. These are the options for patches that use Hotfix.exe as the package installer.

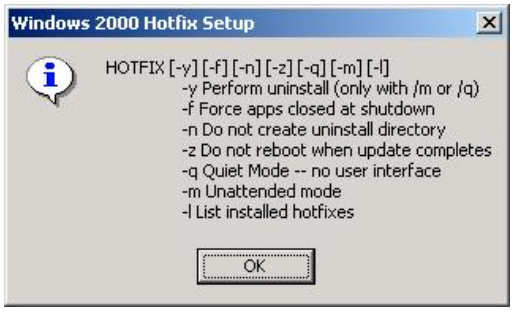

Figure 1

Now take a look at Figure 2. This results from running Windows2000-KB824146-x86-ENU.exe /? from a command line. This is the patch for the RPCSS vulnerability identified in *Microsoft Security Bulletin MS03-039, Buffer Overrun in RPCSS Could Allow Code Execution*.

In this case the Title Bar displays the Knowledge Base article number. Also notice how the switches have changed. While the –z switch has remained the same the switch required to run the patch in unattended mode has changed from –m to –u. UPDATE is also prominently displayed in the window.

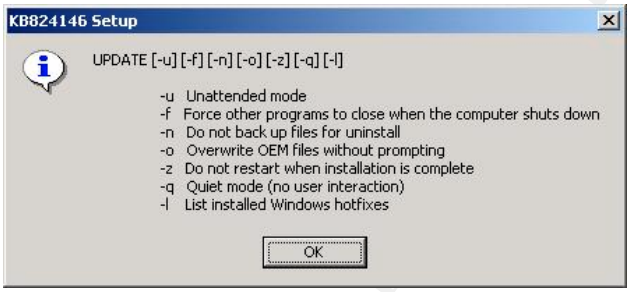

Figure 2

From looking at current operating system patches the following naming convention examples exist for patches using Update.exe with QChain functionality:

- Windows2000-KB824146-x86-ENU.exe
- WindowsServer2003-KB824146-x86-ENU.exe

Running each of these executables from the command line with the /? switch results in the same dialog box as shown in Figure 2 above.

As a side note the MS03-039, KB824146 patch for Windows NT 4.0 Server was downloaded and ran with the /? switch since the patch name, WindowsNT4Server-KB824146-x86-ENU.exe, fits the naming convention of the other operating systems. The result is in Figure 3 below.

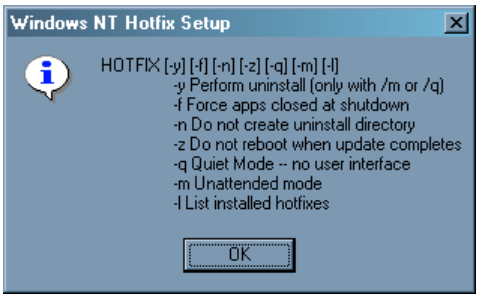

Figure 3

HOTFIX is still the executable that applies the patch so application of this patch to a Windows NT 4.0 system with QChain will require the actual QChain.exe utility be available and run once the chained patches have been run.

3.3 Unsupported Patches

Patches not supported by QChain functionality are any patches not installed using Hotfix.exe or Update.exe. Most patches that apply to non-OS components cannot be installed using QChain. This includes but is not limited to patches for Media Player, Internet Explorer (except in Windows Server 2003), Microsoft Data Access Components and Microsoft Desktop Engine 2000.

An example is the latest cumulative patch for Internet Explorer. By running q828750.exe /? from a command line the dialog box in Figure 4 is returned.

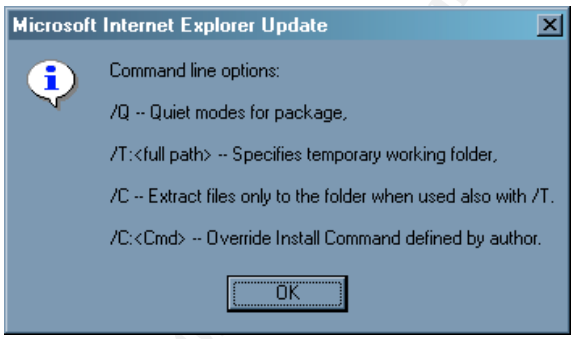

Figure 4

This patch is not installed by Hotfix.exe or Update.exe. The same goes for most patches that fall into the above categories and/or are not a direct component of the operating system. These patches must be installed manually or through some other means.

3.4 CD Based QChain

While assembling a batch file and directory structure to support QChain installations across the LAN may be an effective means of patching servers there are better ways, even on a limited budget. But what about that new server that just arrived? New servers generally arrive with the operating system installed. Corporate management may very well expect it in production that day. If that new server is running Windows 2000 Server there is an extremely high chance the administrator password will be blank, Internet Information Services 5.0 will be installed and not locked down, there is not one patch beyond the latest service pack (if the latest service pack is even there) and there will be no up to date anti-virus software. Plugging this server into the Internet is like holding up a lighted sign at a hacker convention saying HACK ME! Expect that server to be found and compromised before the close of business if connected to the Internet with no additional attention.

A server is at its most vulnerable as soon as the operating system is installed. At this moment most servers will have no anti-virus software, no patches and any passwords entered during installation will most likely be null or very easily guessed. It will only take a few moments of Internet exposure for such a server to be compromised. The importance of this situation cannot be stressed strongly enough. The author is aware of two such instances. One involved a server running Solaris 8. The administrator installed the operating system and had every intent of applying updates and patches and strong passwords right away. Then the administrator was called away from the new server for an emergency and left the server running and connected to the Internet. The administrator handled the emergency but forgot to return to the new server and left it running overnight with no patches and a null root password. The server was compromised with a root kit before the next morning. In another case a system was being configured with Windows 2000 Server. The vendor had connected the server to the LAN (and the Internet) without authorization and departed for the day, leaving the server running with a new installation that had been patched but was left with no anti-virus software. By morning the server was infected with NIMDA.

According to SANS list *The Ten Worst Mistakes Information Technology People Make* the top mistake is "Connecting systems to the Internet before hardening them" (SANS). This can be easily avoided by using QChain and a few other techniques to develop a CD that can be used to patch and reasonably secure a server before plugging in the network cable. This technique involves the use of a directory structure that includes all the software, service packs and patches needed to be applied before connecting the host to the Internet. The directory is saved to a CD-RW disk. The software, patches and configuration steps are accessed through HTML files for each operating system.

The root directory of the Build is named domainname-Build-CD-yyyymmdd. In this particular case it will be named ATE-R-P-Build-CD-20031018 to identify the company (fictitious). The Build CD directory structure is designed to ease application of patches to new and existing systems. The base directory structure contains all the patches needed for each operating system. In the event the data exceeds the space available on a CD it can be broken down into individual CDs for each operating system. If multiple CDs are used the root could be named domainname-W2K-Build-CD-yyyymmdd, etc.

For one base operating system such as Windows 2000 a single CD should suffice for Advanced Server, Server (with or without IIS), and Professional. If other operating systems are in use, such as Windows NT 4.0, Windows XP or Windows Server 2003, additional CDs may be needed to hold the patches and service packs and in fact may be easier to maintain. The same goes for server-based applications such as SQL Server.

In the example Build, Internet Explorer 6 with Service Pack 1 resides in its own directory, extracted from the Internet Explorer 6 SP1 CD. Likewise, Windows Media 9 resides in its own directory. Windows 2000 Service Pack 4 resides in the root directory.

Referring to Figure 5, below it can be seen patches identified by Microsoft Security Bulletins reside in their own directories. Each directory is named for the security bulletin followed by the operating system or component to which it applies. For example, MS03-039 applies to a number of Windows operating systems but in this case only the patch for Windows 2000 (and there is only one Windows 2000 patch) is needed so that directory is named MS03-039-2K. If this were to support NT 4.0 the base MS03-039 directory could be named MS03-039-NT40-2K with subdirectories for each individual patch named according to the operating systems to which it applies.

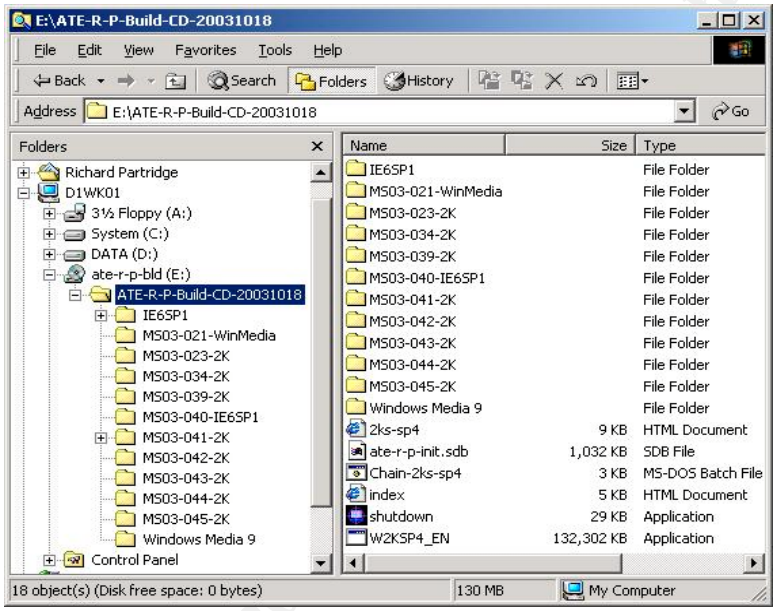

Figure 5

Figure 6 below shows an example of a template that can be used to store patches. The template can be customized as needed by the organization.

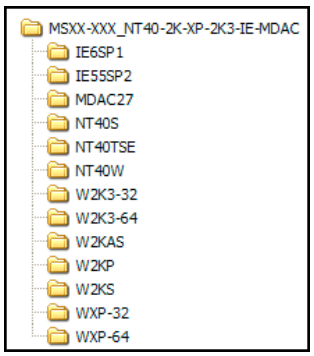

Figure 6

The HTML files serve as the portal to the patches and begin with an index (see Figure 7). The index lists, with links, each operating system supported and also includes any

notes or other information needed for reference. The index should be dated with the date of the latest update to the entire directory. If the CD only supports one operating system the index may not be necessary.

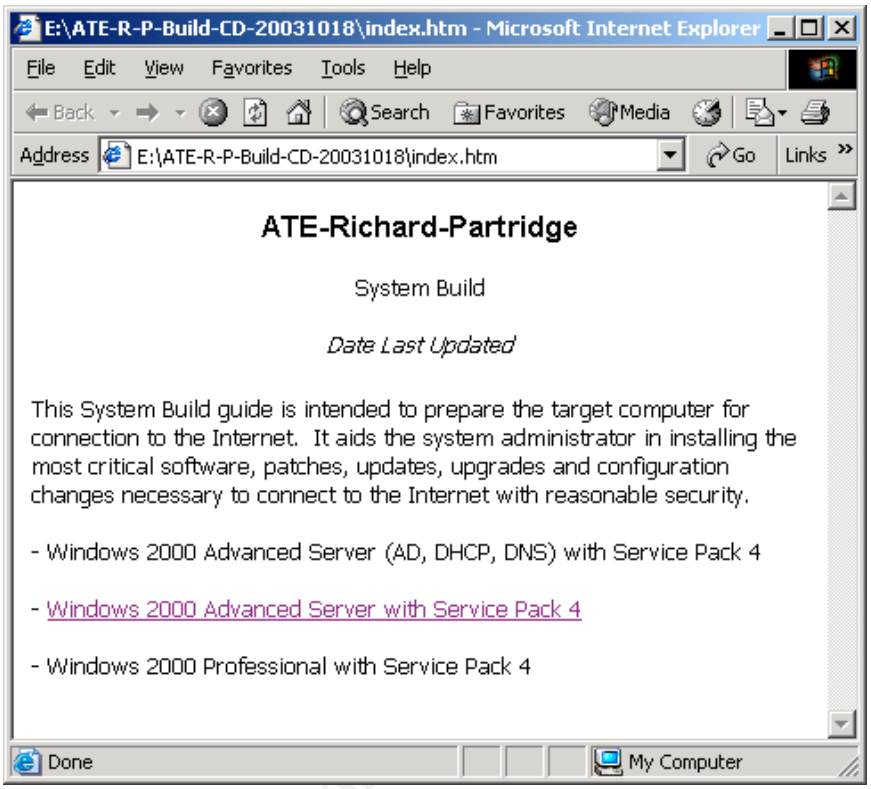

Figure 7

It is the build sheets that are the key to easing patch application and server deployment. Each build sheet is an HTML file. HTML was chosen because every Windows computer has a web browser installed and because of its functionality, specifically the ability to link to a patch or a source of information.

Resist the urge to design a build sheet that links to a vendor patch source such as Microsoft. The whole idea here is to configure and patch the computer before plugging it into the network. The computer must be secured to a reasonable level before it is networked.

Another excellent reason to download the patches right away and make them available from within the organization is the availability, or lack thereof, of vendor sources. During the Lovsan/Blaster outbreak Internet connectivity was hit-and-miss. At times Microsoft and Windows Update simply could not be reached. Many Internet Service Providers were shut down completely and this would have a very negative effect on a small business relying on a DSL or Cable connection. Similar situations occurred during the SQL Slammer and Code Red outbreaks. So while analysis of a vulnerability

announcement is going on download the patches and store them within the organization.

The availability of the patches is what makes the Build Sheets work. Each build sheet provides a sequenced link to each patch. More importantly, each build sheet can serve as a checklist for complete configuration should the administrator wish to develop it to that length.

The build sheet demonstrated here is intended to be for a Windows 2000 Advanced Server file and print server. As such the server does not need many of the components installed by default. It is intended to serve three scenarios. The first is the most likely where a new server is delivered with the operating system already installed. In that case the installation is most likely to be a default installation including Internet Information Services 5.0. It may or may not include the latest service pack. It is essential to remove unneeded components before any additional configuration is undertaken. The build sheet serves as a checklist to remove unneeded components. In the other case the build sheet serves to configure a new installation of the operating system. This situation allows unneeded components to be unchecked during the install. Finally, the build sheet can be used to bring a server of unknown patch and configuration status to a known secure state.

ATE-Richard-Partridge
System Build
Windows 2000 Advanced Server with Service Pack 4

1. Install Windows 2000 Advanced Server.
   During installation ensure:
   - Format all drives with NTFS
   - Administrator password - enter a strong password of at least 8 characters.
   - Windows 2000 Components
     -- Accessories and Utilities
     --- Accessability Wizard - uncheck
     --- Accessories - Uncheck all but Calculator, Paint and WordPad
     --- Communications - Uncheck all (Includes Chat, Hyperterminal, Phone Dialer)
     --- Games - uncheck
     --- MultiMedia - uncheck
   - Indexing Service - uncheck
   - Internet Information Services (IIS5.0) - uncheck
   - Script Debugger - uncheck
   - Terminal Services - check only if it will be used for remote administration
   - All others leave unchecked
   - Network Settings - Leave Typical Settings radio button checked
   - Workgroup or Domain - Leave “No, this computer is not on a network...” radio button selected
2. Install Windows 2000 Service Pack 4 by clicking here.
3. Install anti-virus software by clicking here.
4. Apply the latest anti-virus definition update by clicking here.
5. Upgrade to Internet Explorer 6 with Service Pack 1 by clicking here.
6. Apply the IE6SP1 cumulative patch (MS03-040) Q828750 by clicking here.
7. Upgrade Windows Media Player to Windows Media 9 by clicking here.
8. Apply the Windows Media 9 patch (MS03-021) KB819639 by clicking here.
9. Install Windows 2000 chained patches by clicking here.
   - When the script runs it will apply the initial Local Security Policy.
   - When chained patch installation is complete the computer will reboot automatically.

The Build sheet begins with a heading followed by a list of steps to take to configure the installation. The list may be customized to meet the needs of the administrator and organization. Following the configuration settings is a list of service packs, software and patches to be installed. Each item is a hyperlink that points to the relative path of the target executable. This is relatively easy to set up in Microsoft Word. Open Word then immediately save the document as a web page. Begin typing with heading and installation steps. Type in item 2. Install Windows 2000 Service Pack 4 by clicking here. Select the text then right-click and select Hyperlink. In the Insert Hyperlink dialog box navigate to the executable or directory you wish the link to point to then click OK. In this case you would navigate to the Windows 2000 SP4 file. In the case of Internet Explorer or other software you would navigate to the installation executable, usually setup.exe and select it. Continue with this till all patches/software are linked.

The final entry, Install Windows 2000 chained patches by clicking here, points to the batch file used for the chained installation. Once again, use the relative path.

> **Note**: Patches applied by the build sheets should be sequenced. Application by patch release date is the best. In spite of Windows File Protection there are cases where newer files get overwritten by older files during updates and component additions. Windows 2000 is certainly far better at protecting system files than any operating system before it but it is still not perfect. So whenever a component gets removed or added or a new software package installed and whenever a patch is applied it is important to conduct a vulnerability scan of the server to ensure it maintains the desired security posture and reapply any patches that show not applied.

Service packs and other files may be left in the top level directory. In this instance there are a number of additional files.

The file named Chain-2k-sp4.bat is the main script for chaining patch installation with QChain. In addition to chaining the patches it also discovers which drive the CD is in and applies a security template database to Local Security Policy that enhances the security posture of the computer. Finally, the script uses the Shutdown.exe utility from the Windows 2000 Server Resource Kit to automatically restart the computer after the chained application of patches.

Here is the batch file followed by comments.

```
@echo off
rem  Written by Richard A. Partridge
rem  rick_ap@hotmail.com
rem  System security configuration and
rem  patch application.
rem  Try to find which drive the CD is in.
rem  Looks in drives C thru H.  If found
rem  execution continues.
rem  If not found script exits with the message
```

```
rem  Cannot find CD-ROM drive.  Additional tests
rem  can be added for more dirves.
rem  The letter A is initially used because
rem  this will not be on a floppy.
set x=A
if exist C:\ATE-R-P-Build-CD-20031018 set x=C
if exist D:\ATE-R-P-Build-CD-20031018 set x=D
if exist E:\ATE-R-P-Build-CD-20031018 set x=E
if exist F:\ATE-R-P-Build-CD-20031018 set x=F
if exist G:\ATE-R-P-Build-CD-20031018 set x=G
if exist H:\ATE-R-P-Build-CD-20031018 set x=H
if "%x%" == "A" goto nocd
echo CD is in Drive %x%
rem  Set local path to patches.
setlocal
set PATHTOFIXES=%x%:\ATE-R-P-Build-CD-20031018
rem  Configure initial security settings.
echo Configuring Local Security Policy.
copy %PATHTOFIXES%\ate-r-p-init.sdb %TEMP%\ate-r-p-init.sdb
attrib -r %TEMP%\ate-r-p-init.sdb
secedit /configure /db %TEMP%\ate-r-p-init.sdb
del %TEMP%\ate-r-p-init.sdb
rem  Start chained installation of patches
echo Installing patches.
%PATHTOFIXES%\MS03-023-2K\Windows2000-KB823559-x86-ENU.exe -z -u
%PATHTOFIXES%\MS03-034-2K\Windows2000-KB824105-x86-ENU.exe -z -u
%PATHTOFIXES%\MS03-039-2K\Windows2000-KB824146-x86-ENU.exe -z -u
%PATHTOFIXES%\MS03-041-2K\Windows2000-KB823182-x86-ENU.exe -z -u
%PATHTOFIXES%\MS03-042-2K\Windows2000-KB826232-x86-ENU.exe -z -u
%PATHTOFIXES%\MS03-043-2K\Windows2000-KB828035-x86-ENU.exe -z -u
%PATHTOFIXES%\MS03-044-2K\Windows2000-KB825119-x86-ENU.exe -z -u
%PATHTOFIXES%\MS03-045-2K\Windows2000-KB824141-x86-ENU.exe -z -u
goto reboot
:nocd
rem  This section runs when the CD
rem  cannot be found.
echo Cannot find CD-ROM drive.
goto end
:end
exit
:reboot
%PATHTOFIXES%\shutdown.exe \\%computername% /R
exit
```

For the batch file to work from a CD the first section determines which drive it is in. This is accomplished by using the if exist statement to search for the main directory on the CD. Once the condition is true that drive letter is saved in a variable. The batch file then sets a local path variable with the command: set PATHTOFIXES=%x%:\ATE-R-P-Build-CD-20031018. The full path can then be entered as %PATHTOFIXES% with the remainder of the unique path throughout the remainder of the batch file.

The next sequence of commands applies an enhanced security policy database to the target computer. It copies the database to the TEMP directory, removes the read only

attribute then applies the database using the secedit command before deleting the copied database from the TEMP directory.

> **Note**: During security database application testing it was discovered application would fail if the database was installed directly from the CD. A second test was set up with the database copied to the target host and the same failure occurred. At this point it was discovered the Read Only attribute was set. Finally, the database was copied and the Read Only attribute removed and application was successful.

Applying enhanced local security policy settings is a quick way to configure a computer. The policy should be set to meet the needs of the organization. This particular policy was based on the w2k_server policy available from The National Security Agency at http://www.nsa.gov/snac/win2k/index.html. A policy should be developed for application to new servers and basing the policy on a known good example, then modifying that only as necessary for the organization, is a good practice.

Once the security policy is applied the batch file applies the chained patches. Each patch is followed by the –z and –u switches to prevent reboot and apply the patch in unattended mode. Execution is then directed to the section that reboots the computer. The shutdown.exe utility from the Windows 2000 Server Resource Kit is used here and with the /R switch this command restarts the computer after a 30 second delay.

The remaining files in the build directory consist of the security template database file, ate-r-p-init.sdb, and the shutdown.exe utility.

This is a relatively simple implementation of QChain using a batch file. It is currently planned to be further developed to autorun the CD and eventually make use of the RUNONCE registry key to fully automate the entire process, including all reboots. There are advantages to using a batch file such as this for applying security settings and patches. If a standard security policy is used then the condition of the system once complete will be known and meet the security needs of the organization. In the case of one or two patches it may be just as efficient to apply them manually but in the case of eight as depicted in this paper the use of QChain functionality can save considerable time because it is not uncommon for a server to take from five to fifteen minutes to completely reboot. Considering a ten minute reboot this application method shaves over an hour from the time needed to get the system operational. The time saved when applying the large number of patches required just before the release of a new service pack could be multiple hours.

This same concept of a build CD can be applied to maintaining the security posture of operational systems. The CD structure is moved to a server-based share. The changes to support network based application are made to the batch file only since the relative links in the HTML files will remain usable. Alterations needed in the batch file are:

- Remove the section that looks for which drive the CD is in.

- Change the path to an absolute path to the share using a UNC path.
- Delete the :nocd section.

Once this is established the administrator can map a drive from the target server, open the index and appropriate page, then run the patches from the HTML files. As new patches are approved for installation they can be added to the bottom of the HTML file with a direct link to install the patch. After a number of patches have been added the batch file can be altered to apply them using QChain functionality.

4.0 Deployment And Integration Of Software Update Services

Patch application and maintenance can also be eased with the use of Microsoft Software Update Services (SUS). SUS allows the administrator to selectively approve patches for automatic or manual application to systems. It allows application of patches from within the organization which may be essential after a patch is released, followed by a scripted exploit appearing on the hacker IRC channels. The installation of SUS requires Internet Information Services be installed on the host and it also runs the IIS Lockdown Tool resulting is a more secure installation of IIS.

SUS is available for download from:
http://www.microsoft.com/downloads/details.aspx?FamilyId=A7AA96E4-6E41-4F54-972C-AE66A4E4BF6C&displaylang=en

4.1 Server Setup

Deployment, set up, configuration and synchronization of SUS are covered in detail in the *Microsoft Windows Security Resource Kit* and the *Microsoft Windows, Deploying Microsoft Software Update Services* whitepaper available for download at: http://www.microsoft.com/windowsserversystem/sus/susdeployment.mspx.

To prepare the target host for SUS ensure it meets these minimum requirements listed in the whitepaper, *Microsoft Windows, Deploying Microsoft Software Update Services:*

> "- Pentium III 700 MHz or higher processor.
> - 512 megabytes of RAM.
> - 6 gigabytes (GB) of free hard disk space for setup and security packages" (Microsoft Corporation).

Once installation and configuration are complete the server should be synchronized with Windows Update. In the lab used for this paper the synchronization took approximately six hours over a 256k DSL connection and downloaded approximately 650 megabytes of data.

4.2 Client Setup

Windows 2000 with Service Pack 4 includes the Automatic Updates client which can be configured to access SUS. Enabling access to SUS requires registry changes or the application of an administrative template that can be deployed to individual hosts or applied to an entire Active Directory Domain through Group Policy.

The options for configuring Automatic Updates with SUS are the same as those for configuring for use with Windows Update:

- Notify me before downloading any updates and notify me again before installing them on my computer.
- Download the updates automatically and notify me when they are ready to be installed.
- Automatically download the updates, and install them on the schedule that I specify.

The difference is application of the administrative template supplied with the SUS download adds additional settings to point Automatic Updates to the local SUS server.

For servers it is recommended you select to be notified when updates are available and only download and install those that are needed. For desktop computers it is recommended the setting be to download and install on a supplied schedule at a specific time each day when employees will not be interrupted by the installation and subsequent restart.

4.3 Patch Approval and Deployment

Once the SUS server is configured and synchronized it's time to begin approval of the updates it is prepared to distribute. Apply the vulnerability analysis technique detailed in this paper to determine whether or not to approve a patch for deployment. Once the decision is made and the appropriate patches are approved they will be available to the clients depending on the client settings selected.

Exercise the same caution with patches applied from SUS as with patches manually applied. Wherever possible test the patch before approval and deployment. On servers, use the least critical server and progress till all have been patched. For desktop computers use a similar technique where the patch is applied to a non-critical computer first then to the rest. Keep in mind the test machines must not be clients of the SUS server because a computer is either a client or not. Once a patch is approved in SUS it deploys to all clients connected to that SUS server. For a formal testing environment an additional SUS server must be available. Lacking that it is advisable to update the test computers from Windows Update or manually by downloading the patch to a local share or directly to the test computer.

4.4 SUS Network Traffic Analysis

SUS uses a number of technologies and almost all of them run in the background. To get a better idea of what is going on a network capture of two types of SUS traffic was conducted.

SUS Server Synchronization with Windows Update is the first analysis. This capture was conducted shortly after configuration of the server with capture started just before a manual synchronization was initiated. The following is a brief review of the network capture.

The capture began with a DNS query to resolve www.msus.windowsupdate.com followed by the response of xxx.xxx.xxx.110. That was followed by establishment of an http connection to that host and an http GET of .msus/v1/aucatalog1.cab http/1.1. The reply to this GET continued for over 4000 frames and appeared to be in clear text.

The next request from the SUS server was another DNS query for crl.microsoft.com with the IP addresses of xxx.xxx.xxx.152 and xxx.xxx.xxx.247 returned. SUS then initiated an http connection to xxx.xxx.xxx.152 followed by http GET /pki/crl/products/CodeSighPVA/crl HTTP/1.0. This appears to be a request for a certificate revocation list for code signing digital certificates. The transmission was in clear text.

SUS then resolved download.windowsupdate.com, receiving a list of seven IP addresses. The first connection was to xxx.xxx.xxx.17 and issued http GET /msdownload/update/v3-19990518/cabpool/q328389.... This appears to be the download of the first update file and it continued from frame 4922 to frame 6303.

Additional downloads proceeded in the same fashion over the six hours of initial synchronization.

The conclusion from this review of the traffic capture is SUS synchronization uses standard Hypertext Transfer Protocol for communication with no encryption. It appears the SUS server initially requests a catalog of updates followed by a certification revocation list for code signing certificates to be able to validate the integrity of downloaded files.

SUS client synchronization and download is the next analysis. This capture was conducted just after the Group Policy enabling SUS was applied so the entire process of initial connection through patch download was captured. The initial communication involved a number of cabinet files and headers being transferred to the client. All transfers occurred within Hypertext Transfer Protocol with no encryption. File transfer then began, once again over HTTP with no obvious encryption. The complete synchronization of the client and download of applicable updates took approximately 38 minutes. During that time the client was not in use by a user.

SUS presents an inexpensive update deployment solution for a small or resource limited organization that wishes to have more control over the patches applied to its computers.

5.0 Vulnerability Discovery With Free & Open Source Tools

Exactly what needs to be patched on the server can be a very involved question to answer. What patches exist? Has a patch been overwritten? What about vulnerabilities that result from inappropriate configuration settings? There are a number of ways to begin answering these questions and each helps determine what vulnerabilities exist on the host.

5.1 Vulnerability Discovery

When all else fails hands-on inventory is always there. Begin by checking Add/Remove Programs for the list of applied Hotfixes. Compare this list against the list of Microsoft Security Bulletins for the operating system or component to determine what may be missing. This process can be time consuming and prone to errors. It also does not reveal overwritten patch files.

Ideally, a vulnerability scanner will be available to help in the discovery of unmitigated vulnerabilities. There are a number of excellent commercial vulnerability scanners available. These products are updated on a regular basis with new vulnerability signatures and exploits to help administrators keep their systems patched and secured. Internet Security Systems Internet Scanner and Harris Corporation STAT Scanner are two the author is directly familiar with. Both are very capable but each is suited to slightly different situations and each has its own unique way of working that demands attention to detail on the part of the operator.

Internet Scanner is a very versatile product. If a host can be plugged into a network Internet Scanner can scan it for vulnerabilities. The most important point about Internet Scanner is it generally does not require administrator level access to the target to accomplish its checks accurately. The main reason for this is Internet Scanner generally exploits the vulnerability rather than trying to determine if a specific patch has been applied by reading files or the registry. This means Internet Scanner can easily crash a target if a specific vulnerability exists. There are checks in Internet Scanner that only check for the presence of updates and patches. Most of these are for Microsoft systems and they do require the scan be conducted with the user logged on with an account that has administrator access to the target.

STAT Scanner is less intrusive and requires administrator access to any host scanned. STAT reads files and registry settings on Windows and other hosts to determine whether or not a specific vulnerability exists. The advantage this offers is reduced chance of causing problems on the host by an exploitative vulnerability check.

There are many other commercial vulnerability scanners available and though they are all effective they are also quite costly and most likely beyond the budget of a small business.

The final option for discovering vulnerabilities are open source and free vulnerability assessment tools such as Microsoft Baseline Security Analyzer (MBSA), Nessus, SuperScan and Nmap. MBSA and Nessus can both scan a Windows host for a number of vulnerabilities. MBSA relies on the catalog from Windows Update or the local SUS server to determine what patches are required. Nessus relies on plugins that the operator can select to scan for specific vulnerabilities and has the advantage of being able to scan for vulnerabilities on many networked hosts. Before beginning vulnerability scanning there are a few cautions and warnings of which to be aware.

5.2 Planning Vulnerability Scanning

Planning for a vulnerability scan is essential. It begins with determining what to scan for in order to assure No Security Through Self-Inflicted Denial of Service (NSTSIDoS). Know the hosts that are to be scanned and what vulnerabilities may exist on that host. Determine then what vulnerabilities to scan for and do not scan a host for a vulnerability that can not exist on it.

Vulnerability scanning carries a certain amount of risk with it. Scanning for vulnerabilities that cannot exist on a host can cause the target to crash. There is no good reason to scan a Windows computer for a Solaris or Linux vulnerability. It makes the scan take longer at the least and may very well crash the target at the worst. This type of situation was witnessed when an information security audit team arrived, directed by corporate headquarters, with little advance warning. The audit team began scanning a large number of hosts with Internet Scanner and would not reveal the vulnerabilities for which they were scanning. On the last day the audit team decided to scan the LAN segments for the management ports of the organization's 30+ switches. In less than a minute after initiating the scan the switches started crashing. Before the scan could be stopped over 20 of the switches had crashed and over 2,000 networked hosts were now stand-alone hosts. The switches crashed so hard they had to be powered down and controllers removed and manually reset then reloaded before they would work properly. The entire process took over 30 hours with frequent problems popping up over the next three weeks. After the switches crashed the audit team released a copy of the Internet Scanner Policy they used to scan the switches. Of the 100+ checks in the policy not one vulnerability being scanned for could have existed on those switches. Had the proper planning taken place and the switches scanned for vulnerabilities that affected that make of switch the crashes would not have occurred.

This paper assumes the person conducting the vulnerability scan has full inside knowledge of the hosts being scanned. Planning then begins with deciding which hosts to scan and is best accomplished by grouping them into categories based on operating system and applications. Servers should be scanned separately from desktop

computers and domain controllers and any other highly critical servers should be scanned separately, preferably after assuring a complete, successful backup. By the same token, those highly critical targets should also be at minimum load. In other words, do not scan a busy Exchange server at the time everyone starts work and checks email. The same goes for a domain controller; don't scan it when it will be servicing logons as people come to work. Schedule scans of heavily used and critical hosts for off hours when the load on them is at its lowest and scan them individually. If necessary, terminate or disconnect all connections to the target. Remember, NSTSIDoS.

Once the hosts are categorized a scan policy or configuration containing the vulnerabilities to be scanned for must be assembled. Initially it may be best to scan for all vulnerabilities that can exist on that host to establish a baseline. Once patching and configuration changes are made based on scan results the follow-up scan could only include those things patched or changed.

Once the systems meet the organization's security requirements set up a schedule to scan all hosts at least once a quarter. Machines can be scanned for a list of critical vulnerabilities that include those known to have been patched. At the least it is recommended that periodic scans check for all critical vulnerabilities such as those that fall into the High or Medium category of Table 1 above. The SANS Institute's *Top Vulnerabilities to Windows Systems* (http://www.sans.org/top20/) provides another source of vulnerabilities to searched for, but categorize them based on the target host.

Plan to conduct a full vulnerability scan of any host prior to connection to the LAN. If a Build CD as above is used, once it has been applied the target host can be networked with the scan host via a small switch or crossover cable. A vulnerability scan can then be run to ensure all patches were successfully installed and all configuration changes made. Any problems turned up by the scan must be corrected and once they are the host is safe to connect to the LAN and Internet. In some cases a connection isolated from the LAN and Internet may not be possible so ensure the required patches and configuration settings have been applied, connect to the LAN and immediately conduct a vulnerability scan for verification.

Patching is not an exact science. There are often problems with applying a patch that won't show up till the host is scanned, or worse. Though Windows File Protection (WFP) prevents critical system files from being replaced there are other critical files that are not protected by WFP. Every time a component or software change is made to a host it must be scanned again for all critical vulnerabilities to ensure the change did not overwrite any critical patches previously applied.

Vulnerability scanning was an integral part of developing the lab systems used for this paper. The systems were initially set up with Windows 2000 with Service Pack 4. MBSA was installed on Windows 2000 Professional and another computer configured to dual boot Windows XP Professional and Red Hat Linux 9.0 was used running Red Hat to scan the lab computers with Nessus. The first scan was a baseline scan with MBSA

using the HFNETCHECK style scan. The results showed the following patches were needed:

- MS03-023	823559	HTML converter	Critical
- MS03-034	824105	Flaw in NetBIOS	Low
- MS03-039	824146	RPCSS	Critical
- MS03-040	828750	IE Cumulative	Critical
- MS02-032	320920	Windows Media Player	Critical

To enhance this initial assessment a search was conducted with Microsoft Security Bulletin Service at:
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/current.asp
By searching for Windows 2000 Advanced Server with Service Pack 4 the following Security Bulletins, and associated patches, were identified as applicable:

- MS03-022
- MS03-023
- MS03-026
- MS03-034
- MS03-039

A search for Windows 2000 Professional revealed the following Security Bulletins and their associated patches needed:

- MS03-023
- MS03-026
- MS03-034
- MS03-039

At this point it was decided to upgrade Internet Explorer on each host to Internet Explorer 6 with Service Pack 1. Searching for IE 6 SP1 in the Microsoft Security Bulletin Service showed the latest cumulative patch needed as referenced in MS03-040.

Once this scan analysis was complete a scan using Nessus was conducted. To demonstrate the potential for the problems a promiscuous scan can cause, the baseline scan with Nessus was conducted by enabling all vulnerability checks. During the scan all hosts remained operational but after the scan it was discovered there were network communication issues with each of the three hosts. Checking the System and Application Logs revealed no errors so the hosts were all rebooted and the communication difficulties did not return. Nessus also found more interesting problems with all three hosts. The Nessus scan showed the following patches needed:

- MS02-008, XHTML
- MS03-011, VM
- MS03-026, RPC DCOM

- MS02-009, IE
- MS02-040, MDAC
- MS03-033, MDAC
- MS02-025, Exchange

At this point it was clear a definitive answer to the exact patches to apply was not to be easily found. To avoid more confusion it was decided to analyze each of the vulnerabilities and develop a patch plan. The initial patch plan would require the installation of IE 6 with SP 1 and Windows Media 9 as part of the baseline install. Patches to be applied were then decided upon according to the risk analysis grid in Table 1 and the following patches were placed into the baseline build:

- MS03-021 for Windows Media
- MS03-023 for HTML Converter
- MS03-034 for Flaw in NetBIOS
- MS03-039 RPCSS
- MS03-040 IE Cumulative patch

Additional scans were conducted after the initial patching and the MBSA scan showed no critical vulnerabilities. The Nessus scan still showed three and one stood out as an example of one of the problems that can arise during vulnerability scanning. The Nessus scan showed all three systems as vulnerable to the RPC DCOM vulnerability detailed in *Microsoft Security Bulletin MS03-026 Buffer Overrun in RPS Interface Could Allow Code Execution (823980)*. Following the release of that bulletin additional vulnerabilities associated with RPC were discovered and Microsoft released *Microsoft Security Bulletin MS03-039 Buffer Overrun in RPCSS Service Could Allow Code Execution (824146)* with a patch that corrected the new vulnerabilities and included the corrections for the original vulnerabilities announced in MS03-026 (Microsoft Corporation). It was know all three lab systems had received the MS03-039 patch and that the scan showed that vulnerability did not exist. Analysis of this possible false positive began with checking the plugin code available at the Nessus site. Examination of the code revealed it checks for specific registry entries associated only with the patch for MS03-026. Additional searching of Nessus plugins revealed there are two plugins associated with the vulnerability identified in MS03-026. One checks for the presence of the patch by reading registry entries and the other actually exploits the vulnerability. The second plugin that exploits the vulnerability had been used in the scan but did not report the existence of the vulnerability. The final determination is the report is a false positive.

Similar situations can arise with MBSA. An example situation is where patch A replaces somedll.dll, version 1, with somedll.dll, version 2. Later, patch B is released and it replaces the current somedll.dll with somedll.dll, version 3. When MBSA is then sued to scan for patch A it may report patch A is missing.

Regardless of the vulnerability scanner chosen perfection does not exist and analysis of scan results is necessary to confirm the security posture of the target system.

5.3 Free Vulnerability Discovery Tools

Microsoft Baseline Security Analyzer (MBSA) is probably the best free vulnerability scanning tool available to a resource limited organization running Windows Operating Systems. MBSA "is a tool that can determine which critical updates are installed on a target computer, as well as which security updates are required" (Smith 509).

MBSA can be downloaded from:
http://www.microsoft.com/downloads/details.aspx?displaylang=en&familyid=9a88e63b-92e3-4f97-80e7-8bc9ff836742

The white paper, *Baseline Security Analyzer*, describes the capabilities and use of MBSA is available at:
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/mbsawp.asp

Installation and configuration of MBSA is a simple process of running the downloaded executable. Before installation give some thought to the most suitable computer to use for scanning. It should be readily available to the administrator but not necessarily available to users. During the installation the option of limiting the user permitted to use MBSA is offered. It is recommended the option "Only for me (username)" be selected unless there are additional administrators who will use the computer and MBSA.

Using MBSA requires the user be logged on with an account that has administrator access to the target machines. In most cases this should be a domain administrator account. In environments where a domain is not used the person conducting the scan should map a drive to the admin$ share using UNC on the target using an account with administrator access to the target. Do this with the NetBIOS name of the computer and not the IP Address. This may prove tedious if more than ten or so hosts must be scanned and are not members of a domain but it is the only way MBSA works.

If a single computer is the target in a stand alone environment it is best to map the drive as above and select the computer to scan by using the NetBIOS name of the target. Using the NetBIOS name ensures authentication that using the IP Address will not.

Scan results are output to the user interface and for a quick scan these results are good. While the results are saved in XML format it is not easily imported into a database or spreadsheet for analysis or presentation to management. One option is to use the Copy feature in the user interface to copy the security report view to the clipboard. It can then be pasted into a text document and imported into a spreadsheet but this still leaves out the specific details of each item in report. Each specific item can be copied and pasted as above as well. This is more time consuming but when dealing with a small number of systems it may be worthwhile.

The next tool for vulnerability discovery is Nessus. Nessus is an open source security scanner that runs on Linux and UNIX systems. It is provided in two components. The daemon, or server, must run on a Linux/UNIX system. This is the component that actually accomplishes the scan. The client provides the user interface and there are clients available for Linux/UNIX and Windows.

Nessus provides a wealth of vulnerability checks, called plugins that can scan for many vulnerabilities on many systems. Unlike MBSA, Nessus can run some plugins against a target without administrator access but many of those plugins are in the category of dangerous because they actually exploit the vulnerability. If the vulnerability exists and successful exploitation results in a crash of the target then that could be the result from a Nessus scan. As always, use caution when conducting vulnerability scans with Nessus or any other tool. NSTSIDoS.

The final category of vulnerability assessment tools is the port scanners. While not strictly vulnerability assessment tools, port scanners can find hosts on a network and determine what services are running on the host (what ports are listening).

SuperScan version 4 is a good Windows port scanner and is available from Foundstone and can be downloaded from the Scanning Tools page at http://www.foundstone.com/index.htm?subnav=resources/navigation.htm&subcontent=/resources/freetools.htm. Once at this page scroll down and select Scanning Tools then, Under SuperScan select Download this tool now.

The downloaded archive unpacks into a directory that can be placed in the Program Files folder. A shortcut can be created and placed on the Programs menu of the desired user profile.
Running SuperScan produces its user interface where the target IP Addresses can be entered along with many other options. The output is available in the user interface and can also be output to an HTML file that can be saved for later analysis.

By far the best known port scanner is Nmap. Nmap is available from Insecure.org and may be downloaded at http://www.insecure.org/nmap/nmap_download.html. Nmap is ported to Linux/UNIX systems and has a command line and GUI interface. It is also included with most Linux distributions. The user interface, Nmap Front End, is much like that for SuperScan and allows set up of target IP Addresses and scan options.

NmapWin is the Windows port of Nmap and is available for download from the same source as Nmap.

Each of these port scanners can determine which ports are open on a host which may lead to the discovery of previously unknown services running.

6.0 Conclusion

Deploying and maintaining secure information systems is not an easy task. It requires attention to detail and knowledge of the vulnerabilities that can exist on the systems used. By subscribing to vendor alerts such as Microsoft Security Bulletins the information needed when new vulnerabilities are discovered becomes more readily available. Subscribing to the CERT Advisory mailing list will enhance that information as will periodically checking web sites such as CVE and ICAT Metabase.

Once the details of a vulnerability are known it can be analyzed and assigned a risk rating that meets the needs of the organization and applied within the timeline set by the organization.

Applying patches to mitigate vulnerabilities as they arise is as essential as applying patches for known vulnerabilities to new systems before they connect to the LAN or Internet. An unprotected system is at very high risk of compromise and to reduce that risk it must be patched. Having a plan to patch and a systems that utilizes QChain functionality can significantly reduce the risk of connecting a new system to the Internet.

Once connected maintaining the security posture of the organizations' systems is key and that can be accomplished from within the organization with Microsoft Software Update Services. Using SUS gives the organization more detailed control over which patches get applied and which do not get applied.

Finally, analyzing information systems for vulnerabilities with MBSA, Nessus or any of the available port scanners will reveal problems before they become serious and in the case of a new systems it will ensure the host is secured before being connected to the Internet.

Works Cited

Microsoft Corporation. *Microsoft Knowledge Base Article – 296861 How to Install Multiple Windows Updates or Hotfixes with Only One Reboot*. 28 Oct. 2003. 4 Nov. 2003, <http://support.microsoft.com/default.aspx?scid=kb;en-us;296861>

---. *Microsoft Knowledge Base Article – 262841 Hotfix.exe Program Description and Command Line Switches*. 23 Sept. 2003. 4 Nov. 2003, <http://support.microsoft.com/default.aspx?scid=kb;en-us;262841>

---. *Microsoft Security Bulletin MS03-034 Flaw in NetBIOS Could Lead to Information Disclosure*. v1.1, 3 Sept. 2003. 4 Nov. 2003. <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/Bulletin/MS03-034.asp>

---. *Microsoft Security Bulletin MS03-039 Buffer Overrun In RPCSS Service Could Allow Code Execution (824146)*. v1.0, 10 Sept. 2003. 4 Nov. 2003. <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/Bulletin/MS03-039.asp>

---. *Microsoft Windows, Deploying Microsoft Software Update Services*. Jan. 2003. 4 Nov. 2003, <http://www.microsoft.com/windowsserversystem/sus/susdeployment.mspx>

---. *Microsoft Windows NT and Windows 2000 Hotfix Installation and Deployment Guide*. 2001. 4 Nov 2003, <http://www.microsoft.com/technet/archive/default.asp?url=/technet/archive/security/tools/tools/hfdeploy.asp>

SANS. *Mistakes People Make that Lead to Security Breaches, The Ten Worst Mistakes Information Technology People Make*. 23 Oct. 2001. 4 Nov. 2003, <http://www.sans.org/resources/mistakes.php>

Smith, Ben, et al. *Microsoft Windows® Security Resource Kit*. Redmond: Microsoft Press, 2003.

The MITRE Corporation. Common Vulnerabilities and Exposures. *About CVE*. 18 Jun. 2002. 4 Nov. 2003, <http://www.cve.mitre.org/about/>