Global Information Assurance Certification Paper

SANS Practical
Track 5: Securing Windows
V3.1 – Option 2

From Batch Files to Security Settings; Designing a Custom IIS 5.0 Security Template

ABSTRACT

Over the past 5 years our company experienced tremendous growth. One negative side affect of this was loose standards on Windows Operating Systems configuration. This lack of standards affected support staff moral, interdepartmental relationships and system availability. Before we could migrate to Windows 2000 & IIS 5.0, we had to establish workable security & performance tuning standards. Through personal, extensive experience with IIS 4.0, plus additional resources, I proposed a standard Windows 2000/IIS 5.0 Web server build. It started with a common set of applications all Web servers would share, then VB Scripts & batch files I wrote were used by the department to install and configure the standard Web server build. This resulted in:

1. Overall performance improvement by 400%
2. All Web servers were identically built
3. A substantially hardened host, verified by an Open Source Vulnerability Assessment tool
4. Time required to build a server dropped by at least 50%

To better administer our Web servers I began porting our configuration standard over to a Security Template. This opened my eyes to a few security aspects that, up until now, were neglected. In addition, we can now audit a server against this template, so our standard can be maintained.

The following paper titled *"From Batch Files to Security Settings; Building a Custom IIS Security Template"*, details our current security settings. While they do not make up a complete IIS hardening tool, they position us as a company with highly available and robust Web servers.

Table of Contents

I. Description of System.....4

II. Security Checklist Used for Lockdown.....9

III. Security Settings.....10

IV. Apply, Test and Evaluate the Template..... 24

A. Apply the Template.....24

B. Test the Template's Security Settings..... 25

C. Test the System's Functionality.....31

D. Evaluate the Template.....33

V. Works Cited.....36

VI. Appendixes.....38

A. Tools Used to Test and Configure IIS 5.0 Web Server.....38

B. Additional References on CIFS/SMB, NT, NTLM & NTLMv2.....39

C. Security Template Links.....40

From Batch Files to Security Settings;
Designing A Custom IIS 5.0 Security Template

Note: The author assumes the reader has at least a basic understanding of Windows 2000 security (local & domain users plus the account database each of these draws upon) and network connectivity. If a detailed explanation of LAN Manager (LM), NTLM, NTLMv2, Server Message Block (SMB) or Common Internet File System (CIFS) protocol is required, then please consult one of the resources listed in Appendix B. Only the security parameters of these technologies in relation to our environment will be covered here. A brief description of the tools used is listed in the body of this paper and links to them can be found in Appendix A. Appendix C has list and links to some sample Windows 2000/IIS 5.0 security templates.

I. Description of System

A. Evolution to our current Web server environment

1. Our company experienced rapid growth, resulting in many different hardware and software configurations. In addition, some systems that were originally designed to be internal Web sites ended up being open to business partners, or duplicated and placed in a DMZ. At times, development teams were placed in the local administrator groups on Web servers (both development and production systems). Not only were members of a Web server farm configured differently (OS and Application settings), DLL versions varied as well. This resulted in inconsistent server performance and behavior. Troubleshooting took an increased amount of personnel and man-hours. It became increasingly difficult to rollout new features to a Web application; on some servers the required files (DLL versions) were present and on other servers they were not present. This resulted in Web applications functioning incorrectly or inconsistent behavior. This type of behavior was repeatedly experienced across Web server farms, where all farm members were supposed to be identical, with the many different groups/users with local administrator privileges; system build documents and configuration notes were quickly outdated, hence nearly useless. Our Web environment contained well over 200 development and production servers. With the rollout of Windows 2000 just beginning, this scenario called for more control and a better method of handling our Web environment.
2. It was at this point I recommended that all our Web servers share an evolutionary path. They all should have the same set of base applications, software updates, performance tuning modifications and security modifications applied to them during the build sequence. This would accomplish a number of objectives.
   a. **Reduce the number of Web server build variations.** By installing a base of applications on all Web servers, the build team could draw upon a standard set of files and avoid DLL version inconsistencies

(i.e. DLL Hell). In the event of having to install and configure multiple servers in a short period of time, this would reduce the amount of confusion faced (which files & applications were required for a particular Web application) by the build team and the number of errors that were made during this process. In addition, the installation and configuration process was optimized for speed and ease by the use of batch files & installation sequence. Where possible, applications were installed via a "custom setup" to reduce the number of components installed. This was conducted in a manner so as not to add to complexity to the build process or significantly increase the amount of time required building our Web servers.

b. **Present the support team with a consistent platform to troubleshoot.** With different administrators installing and configuring the Web servers, inevitably there were variations in the end product. This presented the support team with a very complicated server base to troubleshoot. With a base set of applications established for all Win2k Web servers I turned my attention on the configuration aspect. By utilizing batch files & scripts, all the servers were built and configured identically. Support time for the Web servers dropped markedly, thus easing the burden on the Support Team.

c. **Provide the development staff with a consistent platform for test, development and production.** The development staff required certain applications and settings in place for their Web application to function properly. Two prime application examples are Internet Explorer and MDAC. Instead of just relying on the default load of IE 5.0 or a download of IE 5.x to our servers, I used the Internet Explorer Administrator's Kit (IEAK) to create a "standard' IE 5.5 SP2 browser that had only the minimum components installed, which is installed in a "hands free" manner.

IEAK	**Purpose:** Windows-based application enables the most cost-effective and efficient way to deploy and manage Web-based solutions **Source:** See Appendix A

MDAC 2.7 RTM was installed via an "administrative" script. Standardized permissions, IIS log settings and IIS script mappings are a few of the software settings that were adopted for all servers. This allowed Web content to be more easily copied from a development server to a production server (farm), and placed into production status.

d. **Provide an evaluation platform that was representative of our environment, to internal clients for 3rd party application review.** Periodically it was requested of us to build a Web server to evaluate a 3rd party product, such as reporting software. The ease and speed in which a Web server could be installed and configured

aided our internal clients in their analysis of a vendor's product. If the product was found not to be compatible with our settings, we already had a checklist in place that guided troubleshooting. After any discrepancies were rectified, a qualitative report was then provided to management for their appraisal and decision.

e. **Provide a hardened Web server for placement in a DMZ environment.** With our national presence growing we needed a Web server that was resistant to Internet based attacks. This requirement was coupled with it remaining a member of the domain too, thus requiring more services than a standalone server. This presented a challenge in how far I could lockdown a server before applications no longer functioned properly. Work on this area started with reducing the number of components installed, not only with the operating system (including Indexing Services and Script Debugger) but also with the additional applications (IE 5.5 SP2). This area ranged from system services, IP Stack registry entries to IIS components (documentation, sample sites, etc.) and script mappings to additional components (IISLOCKD.EXE & URLSCAN 2.5).

IISLockd.exe	**Purpose:** Windows-Based application, IIS Lockdown Wizard works by turning off unnecessary features thereby reducing attack surface available to attackers **Source:** See Appendix A
URLScan.exe	**Purpose:** By blocking specific HTTP requests, the Urlscan security tool prevents potentially harmful requests from reaching the server and causing damage **Source:** See Appendix A

f. **Provide a Web server for intranet use.** This same platform used for Internet hosting would also have to allow internal users to authenticate via a Web browser to view company & HR Information, systems reports, access helpdesk applications and update backend systems.

g. **Provide a Web server for use with business partners.** In addition to an Internet Web server and an intranet Web server, it would also have to allow business partners to authenticate and access resources. Again, security was a prime concern, but network connectivity and functionality took precedence.

h. **Provide a scalable Web server for use as either in single server or server farm roles.** There existed the requirement for the server to host a Web site(s) by itself or as a member of a Web farm. Our company required a Web server with greater scalability than what Windows 2000 & IIS 5.0 offered out of the box. I have tuned our

Web servers to respond to our varying load and environments while still meeting these requirements. The bulk of these performance modifications were conducted via a batch file or VB Script.

B. The particular system the security template was piloted on is an intranet server. It utilizes our standard installation & configuration, and requires company employees to authenticate via browser and review/approve Purchasing Card (Pcard) expenditures. Pcard expenditures are uploaded to PARIS (Purchasing Accounting Reporting Information System), and stored in an MS SQL 2000 database. The PARIS application is made up of several components:
   - PARIS BackOffice Server
   - PARIS Web Server
   - PARIS COM+ Application

   In addition to the Pcard application, the only other application installed (that is not part of our base set up applications) was Network Associates Desktop Security 6.5.3 (PGP).

   Note: Due to corporate security policies, this evaluation server was not able to ftp daily data. The production system utilizes the native Windows 2000 ftp client, which was tested to ensure it functioned properly. Actual Pcard data was copied to the server, decrypted (via PGP), uncompressed (PowerArchiver) and imported into the production database (PARIS Back Office Server). The application owner reviewed the system for proper functionality.

   On a daily basis, updates to Pcard system are downloaded to the PARIS server via ftp. First, the file is unencrypted via Network Associates PGP Desktop Security and then uncompressed via Power Archiver 7.02. The PARIS BackOffice server imports the data via an ODBC connection to the MS SQL database. Pcard users can access the internal Web site and verify the information is correct and up to date. These transactions are also subject to a review that is conducted via an http connection.

   Our test, development and production Web servers reside in different geographical locations and it isn't realistic for an administrator to physically log on at the console. Most administration and troubleshooting activities are conducted remotely via a standard Web browser or a Terminal Services connection. For instance, an in-house written Active Server Page (ASP) is used to check if the Web server is functioning properly. It is assumed that if the page displays (in a browser) the current time and date that the IIS script engine is considered to be functioning properly. Domain administrators also require the ability to conduct normal, administrative functions such as reboot the server, view Event Logs, view/edit the registry, etc. We require SNMP for use by Compaq Insight Management (CIM). There is also anti-virus software (Trend Micro) and backup client (Veritas Net Backup) installed.

   Note: Our current Web server standard (both development and production) is a Compaq DL360: dual 1.4 GHz processors, 1 GB RAM & two SCSI-3 Hard Disks configured in a HW RAID-1 Array. This specific hardware was not available on loan while I ported our security checklist

over to a security template. With that in mind, I "made do" with a similar hardware configuration.
The following is a detailed list of the hardware and software used:

1. Hardware Configuration:
   - Compaq Proliant 2500
   - Dual 200 MHz Pentium Processors
   - 1.0 GB RAM
   - SMART 2-DH Array Controller
   - Two SCSI-2 Hard Disks configured in a HW RAID-1 Array
   - 100mb/Full Duplex NIC
2. Installed Software
   a. Windows 2000 Server, integrated Windows 2000 Service Pack 1
      i. Windows Components
         a.) Deselect Accessories and Utilities
         b.) Deselect Indexing Service
         c.) Deselect Internet Information Service
         d.) Management and Monitoring Tools
            - Select Network Monitor Tools Agent
            - Select SNMP
         e.) Deselect Script Debugger
         f.) Terminal Services
            - Select Enable Terminal Services
         g.) Terminal Services | Remote Administrative Mode
   b. Complete partitioning of array & start first batch file
      i. Second partition is created & formatted
      ii. First batch file runs and calls multiple VB Scripts for system hardening & tuning. The security modifications will be covered in Section III
3. IIS 5.0 installation is conducted by the guidelines set forth in Q259671 & Q281892
4. Internet Explorer 5.5 SP 2 installation is a "hands free" custom browser built from the IEAK
5. Compaq Management Agents, version 5.40
6. Veritas Net Backup Client, version 3.4.1
7. Trend Server Protect, version 5.35
8. VB Script 5.6 and MDAC 2.7 RTM installation
9. Windows 2000 post SP 3, IE 5.5 sp2 & IIS 5.0 patches
   i. Q318202_MSXML20_x86.EXE
      MS02-008 XMLHTTP Control Can Allow Access to Local Files (XML 2.6)
   ii. Q318203_MSXML30_x86.EXE
      MS02-008 XMLHTTP Control Can Allow Access to Local Files (XML 3.0)
   iii. MSXML4qfe.EXE
      MS02-008 XMLHTTP Control Can Allow Access to Local Files (XML 4.0)
   iv. Q323172_W2k_SP4_x86_EN.EXE
      MS02-048 Flaw in Certificate Enrollment Control Could Allow Deletion of Digital Certificate
   v. Q323255_W2k_SP4_x86_EN.EXE

MS02-055 Unchecked Buffer in Windows Help Facility Could Enable Code Execution

vi. Q324096_W2k_SP4_x86_EN.EXE
MS02-053 Buffer Overrun in SmartHTML Interpreter Could Allow Code Execution.

vii. Q324380_W2k_SP4_x86_EN.EXE
MS02-051 Cryptographic Flaw in RDP Protocol can Lead to Information Disclosure

viii. Q326830_W2k_SP4_x86_EN.EXE
MS02-045 Unchecked Buffer in Network Share Provider can lead to Denial of Service

ix. Q326886_W2k_SP4_x86_EN.EXE
MS02-042 Flaw In Network Connection Manager Could Enable Privilege Elevation

x. Q328310_W2k_SP4_x86_EN.EXE
MS02-071 Flaw in Windows WM_TIMER Message Handling Could Enable Privilege Escalation

xi. Q329115_W2k_SP4_x86_EN.EXE
MS02-050 Certificate Validation Flaw Could Enable Identity Spoofing

xii. Q329170_W2k_SP4_x86_EN.EXE
MS02-070 Flaw in SMB Signing Could Enable Group Policy to be Modified

xiii. Q329414_MDACALL_x86.EXE
MS02-065 Buffer Overrun in MDAC Could Lead to Code Execution

xiv. Q329834_W2k_SP4_x86_EN.EXE
MS02-063 Unchecked Buffer in PPTP Implementation Could Enable DOS Attacks

xv. Q324929.EXE
MS02-068 Cumulative Patch for IE 5.5 sp2 - December 2002

xvi. Q327696_W2k_SP4_x86_EN.EXE
MS02-062 Cumulative Patch for IIS 5 - October 2002

10. GE Capital Paris 5.0 Back Office Server
    i. A non-default location was chosen
11. GE Capital Paris 5.0 Web Server
    i. A non-default location was chosen
12. GE Capital Paris 5.0 COM+ Application
13. Network Associates Desktop Security 6.5.3
    i. A non-default location was chosen
    ii. Minimal components were selected
14. IIS Lock Down Utility w/URLScan 2.0 Filter
15. URL Scan 2.5 Baseline upgrade

II. Security Checklist used for lockdown

A. This security checklist was developed internally and specifically for our company environment. There was no single source of reference used. At times, best practices and lessons learned filled in the blanks where books and articles fell short of meetings our needs. I felt our checklist had reached the final stages of maturation in relation to batch files and scripts. With a stable checklist (for our environment), it was a logical choice to migrate these settings to a security template. Just as the checklist underwent revisions, the security template will undergo revisions and

mature. Here are the driving factors for our corporate security & tuning checklist:

1. Reduce the number of Web server build variations
2. Present the support team with a consistent platform to troubleshoot
3. Provide the development staff with a consistent platform for test, development and production
4. Provide an evaluation platform that was representative of our environment, to internal clients for 3rd party application review
5. Provide a hardened Web server for placement in a DMZ environment
6. Provide a Web server for intranet usage
7. Provide a Web server for use with business partners
8. Provide a scalable Web server for use either in single server or server farm roles

B. When presented with these requirements I chose a "one server fits all" approach. This meant all the servers received base sets of files that were installed via batch to provide for greater consistency. From IE 5.5 SP 2, through VBS 5.6 and up to MDAC 2.7 RTM, this was the base platform chosen. This particular server differs in the software in that GE Capital (Paris Web & Back Office server) & Network Associates (Desktop Security) was added on, but the base server shares 100% of the source files and 100% of the security modifications. The following section will cover the specific security modifications based on our listed requirements.

III. Security Settings

This section will follow the sequential steps from our build document. At this point all applications and updates, except the IIS Lock Down utility & the URLScan 2.5 upgrade, have been applied.

A. The TelnetUsers local group was created via the native NET.EXE command. No members were added to this group.

Basis: When this group is created, the only members that will be authorized to logon to the server via telnet are members of this group - TelnetUsers (Q250298). Even though the telnet service is not used on our Windows boxes this was a no-cost precaution against using an older, insecure protocol.

B. User Rights Assignment was configured with the NTRIGHTS.EXE utility.

NTRights.exe	**Purpose:** Command-line utility, grants/revokes NT-Rights to a user/group **Source:** See Windows NT 2000 Server Resource Kit in appendix A

We removed the Everyone group's right to:

Access this computer from the network
Logon locally

We also granted the Authenticated User (AU) group the right to:

Access this computer from the network

Logon locally

Basis: This configuration was originally started with IIS 4.0, AU being granted the Access This Computer From the Network and Log on Locally rights, so it can be linked to the IIS 4.0 Security Checklist. Q187506 has a table detailing the required rights (i.e. Log on Locally) in relation to Authentication Type (Basic – Clear Text). By default, the Everyone group had the "Access this computer from the network" right, which I removed. It's worth repeating this is a "One server fits all" build, so a compromise was made here resulting in the ability of AU possessing both "Access this computer from the network" and "Log on locally" rights. Since our servers reside in a controlled access data center, the ability of AU to physically log on at the console is minimal.

C. Audit settings were configured with the AUDITPOL.EXE utility.

Auditpol.exe

Purpose: Command-line utility enables the user to modify the audit policy of the local computer or of any remote computer
Source: See Windows NT 2000 Server Resource Kit in appendix A

System	Success	Failure
Logon	Success	Failure
Object Access		Failure
Privilege Use	Success	Failure
Process Tracking		
Policy Change	Success	Failure
Account Management	Success	Failure
Directory Service Access	Success	Failure
Account Logon	Success	Failure

Basis: Up until this point in our Web server builds, there was not a consistent set of source files drawn upon or a set of consistent configuration parameters for the enterprise. With that said, most of these modifications were reviewed by several teams (i.e. support team, several development teams, etc.) before being accepted. It would have been an easier sale if I could have taken our audit settings straight from a reference and implemented them, but it didn't happen like that. This is one area where both references and familiarity with the OS came together. Past experience has shown that there can be too much auditing. For instance, if both Success and Failure for Object Access (Scambray and Stuart 152) is chosen then there is considerable (negative) impact on system performance. On the other hand, some auditing may be necessary for monitoring and troubleshooting which some may deem excessive. "From Blueprint to Fortress" does not recommend enabling auditing under Process Tracking. I chose Failure to log unauthorized attempts to access files. This is useful for troubleshooting application problems (file system permissions were drastically reduced from previous

Web server builds) and stopping users with malicious intent. This is definitely an area that will have to be customized for each enterprise.

D. I knew I would be not be able to convince the build team or management to secure any services if it meant changing their (service) startup value via the MMC, so I looked for an alternative. The first article I came across (Securing IIS 5.0 Using Batch-Oriented Command Files) demonstrated just how easy this could be accomplished. Schultz (406-407) provides a variation on which services to place in a disabled state & manual startup state. The Windows NT WarDoc demonstrates the need to secure unnecessary services, specifically the Messenger service. Boswell (44) was able to demonstrate the importance of understanding the role services play in a Domain environment, "…Dynamic DNS requires Netlogon and the DHCP Client services". There were numerous reboots before a clean set of Event logs were achieved. Service startup values were quickly & easily set with REG.EXE (Reg Update) utility.

Reg.exe	**Purpose:** Command-line utility, manipulates registry entries on local or remote computers **Source:** See Windows NT 2000 Server Resource Kit in appendix A

The following table lists the Service and startup value assigned to them:

Alerter		**Disabled**
Application Management	*Manual*	
Automatic Updates		**Disabled**
Certificate Services		**Disabled**
ClipBook		**Disabled**
COM+ Event System	Automatic	
Computer Browser		**Disabled**
DHCP Client	Automatic	
Distributed File System		**Disabled**
Distributed Link Tracking Client		**Disabled**
Distributed Link Tracking Server		**Disabled**
Distributed Transaction Coordinator		**Disabled**
DNS Client	Automatic	
Event Log	Automatic	
Fax Service		**Disabled**
File Replication		**Disabled**
FTP Publishing Service	Automatic	
IIS Admin Service	Automatic	
Indexing Service		**Disabled**
Internet Connection Sharing		**Disabled**
Intersite Messaging (ISM)		**Disabled**
IPSEC Policy Agent		**Disabled**

Kerberos Key Distribution Center		**Disabled**
License Logging Service		**Disabled**
Logical Disk Manager	Automatic	
Logical Disk Manager Administrative Service	*Manual*	
Messenger		**Disabled**
Net Logon	Automatic	
NetBackup Client Service	Automatic	
NetMeeting Remote Desktop Sharing		**Disabled**
Network Connections	*Manual*	
Network DDE		**Disabled**
Network DDE DSDM		**Disabled**
NT LM Security Support Provider	Automatic	
Online Presentation Broadcast		**Disabled**
Performance Logs and Alerts		**Disabled**
Plug and Play	Automatic	
Print Spooler	Automatic	
Protected Storage	Automatic	
QoS RSVP		**Disabled**
Remote Access Auto Connection Manager	*Manual*	
Remote Access Connection Manager	*Manual*	
Remote Procedure Call (RPC) Locator		**Disabled**
Remote Procedure Call (RPC) Service	Automatic	
Remote Registry Service	Automatic	
Remote Storage Engine		**Disabled**
Remote Storage File		**Disabled**
Remote Storage Media		**Disabled**
Removable Storage		**Disabled**
Routing and Remote Access	*Manual*	
RunAs Service		**Disabled**
Security Accounts Manager	Automatic	
Server	Automatic	
Smart Card		**Disabled**
Smart Card Helper		**Disabled**
SNMP Service	Automatic	
SNMP Trap Service		**Disabled**
System Event Notification	Automatic	
Task Scheduler	Automatic	
TCP/IP NetBIOS Helper Service	Automatic	
Telephony	*Manual*	
Telnet		**Disabled**
Terminal Services	Automatic	
Trend Server Protect	Automatic	
Uninterruptible Power Supply		**Disabled**
Utility Manager	*Manual*	

Windows Installer		*Manual*
Windows Management Instrumentation	Automatic	
Windows Management Instrumentation Driver Extensions		*Manual*
Windows Time		**Disabled**
Workstation	Automatic	
World Wide Web Publishing Service	Automatic	

E. File system permissions were set with Windows Explorer. This was both time-consuming and prone to an occasional mistake. Previous attempts to automate this with CACLS.EXE (native to Windows 2000) & XCACLS.EXE were not consistent.

Xcacls.exe	**Purpose:** Command-line utility displays or modifies the access control lists (ACLs) of files **Source:** See Windows NT 2000 Server Resource Kit in appendix A

This portion of building a server was undoubtedly the longest, resulting in the most complaints (from the build team). The goal of customizing the file system permissions was to limit access to legitimate accounts (and non-legitimate/compromised accounts) to only the folders/files required and to the minimum amount of access that was required. This was also heavily influenced by the design goals of reducing the number of Web server build variations (since past attempts at customizing the file system permissions were inconsistent), and providing a hardened Web server for placement in a DMZ environment (restricting the default NTFS permissions). To oversimplify the concept of (Windows NT) user account & code execution, when a user is successfully authenticated and executes code, that code runs under the context (i.e. privileges) of that user. "Thus, the actions performed by executing code is limited only by the privileges granted to the account that executes it. The goal of the malicious hacker is to run code with the highest possible privileges." (Scambray and Stuart 12). Q187506 was a general starting point for establishing our file system permissions, while Q271071 was drawn upon specifically for reference to the Temp directory and the requirement for Change access (via Authenticated Users). I replaced access to the Everyone Group (Network & Interactive groups) with the AU group (local, domain and trusted domain accounts). In addition, the AU group has access to fewer files & folders as well as fewer rights (List, Read, Read and Execute in place of Change). NTFilemon.exe was indispensable when it came to troubleshooting and filling in the holes for required permissions.

NTFilemon.exe	**Purpose:** Windows-based application that monitors and displays all file system activity on a system **Source:** See Appendix A

The IISLockDown utility further adds to securing the file system permissions by denying access to two local user accounts (IUSR_*MachineName* & IWAM_*MachineName*). Prior to implementing this utility, the use of the deny attribute was viewed as laborious (selecting the files via Windows Explorer) & potentially dangerous (I did not have an accurate list of files to use for Windows 2000 and it was felt system availability might have been limited if the wrong files were denied access to the IUSR & IWAM accounts.) A Microsoft recommended list (for Windows NT 4.0) for critical files is located on the MS IIS 4.0 Security Checklist. Past experience with moving these files (as recommended by the MS IIS 4.0 Security Checklist) has shown there would be increased administrative overhead, and it would not meet the design goal of providing the support team with a consistent platform to support. Britney's NT Hack Guide details the steps to uploading Netcat.exe to a Windows NT/2000 Web.

NetCat.exe	**Purpose:** Command-line utility, which reads and writes data across network connections, using TCP or UDP protocol **Source:** See Appendix A

Even though the vulnerability that lead to calling tftp.exe (by the IUSR_*MachineName)* has been patched, I felt that this might provide some future protection against similar exploits that involve calls from the IUSR_*MachineName* to "critical files". In NT 4.0 speak, the files are referred to as Critical Files and in Win2k speak they are referred to as System Files.

Permissions listed below follow the Windows NT 4 model:
Read = List, Read, Read & Execute
Change= List, Read, Read & Execute, Write

Set permissions on the C: drive & subfolders as follows:

Note: Set permissions on the current folder and propagate permissions on all subfolders.

Folder	User	Permission
C:**	Administrators	Full Control
	System	Full Control

Note: Starting at the root of the drive, the existing file system permissions were overwritten with Administrators and SYSTEM with Full Control. When required, subfolders had additional permissions applied, taking advantage of inheritance when possible.
Note: AU = Authenticated Users

C:\Documents and Settings\All Users**	AU	Read

C:\Documents and Settings\All Users\Application Data\Microsoft\Crypto\RSA**		
	AU	Change
	Creator Owner	Change

Note: This area was crucial to the dynamic content & COM+ applications functioning properly. A great deal of trial-and-error was conducted here in order to yield functional Web server.

C:\Documents and Settings\Default User**	AU	Read
C:\Program Files**	AU	Read

Note: The easiest way to set permissions on this group of folders is via Windows Explorer – selecting all listed folders (below) and setting permissions on them simultaneously.

C:\Program Files\Accessories**	AU	Remove
C:\Program Files\Compaq**	AU	Remove
C:\Program Files\Microsoft FrontPage**	AU	Remove
C:\Program Files\NetMeeting**	AU	Remove
C:\Program Files\Windows Media Player**	AU	Remove
C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions**		
	AU	Remove
C:\Winnt**	AU	Read

Note: This area (the entire C:\Winnt directory tree) was crucial to the dynamic content & COM+ applications functioning properly. A great deal of trial-and-error was conducted here in order to yield a functional Web server.

Note: The easiest way to set permissions on this group of folders is via Windows Explorer – selecting all listed folders (below) and setting permissions on them simultaneously.

C:\Winnt\addins**	AU	Remove
C:\Winnt\Application Compatibility Scripts**		
	AU	Remove
C:\Winnt\AppPatch**	AU	Remove
C:\Winnt\Debug**	AU	Remove
C:\Winnt\ime**	AU	Remove
C:\Winnt\Offline Web Pages**	AU	Remove
C:\Winnt\Registered Packages**	AU	Remove
C:\Winnt\Registration**	AU	Remove
C:\Winnt\Sch Cache**	AU	Remove
C:\Winnt\Security**	AU	Remove
C:\Winnt\Service Pack Files**	AU	Remove
C:\Winnt\Twain_32**	AU	Remove

Note: The easiest way to set permissions on this group of folders is via Windows Explorer – selecting all listed folders (below) and setting permissions on them simultaneously.

C:\Winnt\System32\certsrv**	AU	Remove
C:\Winnt\System32\clients**	AU	Remove
C:\Winnt\System32\config**	AU	Remove
C:\Winnt\System32\dhcp**	AU	Remove

C:\Winnt\System32\dtclog**	AU	Remove
C:\Winnt\System32\export**	AU	Remove
C:\Winnt\System32\LLS**	AU	Remove
C:\Winnt\System32\LogFiles**	AU	Remove
C:\Winnt\System32\NetMon**	AU	Remove
C:\Winnt\System32\NTMSData**	AU	Remove
C:\Winnt\System32\OS2**	AU	Remove
C:\Winnt\System32\reminst**	AU	Remove
C:\Winnt\System32\rocket**	AU	Remove
C:\Winnt\System32\rpcproxy**	AU	Remove
C:\Winnt\System32\shellext**	AU	Remove
C:\Winnt\Repair	AU	Remove
	System	Remove

Note: This reduces access to only the Administrators. In the event the SYSTEM account has been compromised, it (the SYSTEM account) will still not have access to the local SAM.)

Note: The easiest way to set permissions on this group of folders is via Windows Explorer – selecting all listed folders (below) and setting permissions on them simultaneously.

C:\Winnt\System32\Inetsrv\Data**	AU	Remove
C:\Winnt\System32\Inetsrv\iisadmin**	AU	Remove
C:\Winnt\System32\Inetsrv\metaback**	AU	Remove
D:**	Administrators	Full Control
	System	Full Control

Note: Starting at the root of the drive, the existing file system permissions were overwritten with Administrators and SYSTEM with Full Control. When required, subfolders had additional permissions applied, taking advantage of inheritance when possible.

D:\LogFiles\http**	***Sanitized Text***	Read
D:\Software\IIS Temporary Compressed Files**		
	AU	Read
D:\WebContent\mailroot\pickup	AU	Change

Basis: The initial permissions were based on Q260985. They were further refined after several rounds of testing and NTFileMon.exe.

D:\WebContent\wwwroot**	AU	Read

F. Registry Entries Relating to the Logon Dialog box
Note: In compiling our build document I came across many books, articles & Web sites. When possible I verified these recommendations against MS TechNet and the REGENTRY.CHM from the Windows 2000 Server Resource Kit. If the parameters were not covered in sufficient detail, then I posted a question on the relevant MS Newsgroup. I've always felt that MS TechNet & the Resource Kits have been the two best sources of

detailed information available for truly understanding and administrating MS products.

1. DontDisplayLastUserName

   HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon
   Name **DontDisplayLastUserName** Note: add if necessary
   Type **REG_DWORD** Value **1** (binary)

   Basis: It's been repeatedly said that half of what's needed (to break into a system) is the username. Even with our servers in a controlled access data center, it makes sense to remove last username logged on.

2. LegalCaptionNotice

   HKLM\Software\Microsoft\WindowsNT\CurrentVersion\Winlogon
   Name **LegalNoticeCaption**
   Type **REG_SZ** Value **Security Notification**

   Basis: Dialog box title that is displayed after Ctrl+Alt+Del is pressed.

3. LegalCaptionText

   HKLM\System\Microsoft\WindowsNT\CurrentVersion\Winlogon
   Name **LegalNoticeText**
   Type **REG_SZ** Value

   Use of this system is explicitly limited to the employees of *Company Name* and individuals conducting business at the behest of *Company Name*. All data contained on *Company Name* computer systems is owned by *Company Name* and may be monitored, intercepted, read, copied or captured in any manner and disclosed by authorized personnel. Unauthorized use of this system is a criminal offense and will be prosecuted to the fullest extent of both state and federal law.

   For technical support, please call the *Company Name* Help Desk # ### ###-####.

   Basis: This is the Legal Notice Text that will be displayed in the "Security Notification" dialog box. The Chief Security Officer (CSO) was more than willing to provide this, since this is the same text used on devices capable of displaying a "message of the day". The story goes an unauthorized individual was able to logon via the console (on a Windows NT System), but was later caught. When charges were pressed against him, his defense was after pressing Ctrl+Alt+Del he was presented with a "Welcome" dialog box, thus an open door policy was deemed to exist. I've been unable to verify the validity of this story, but can see the legal plausibility of it.

4. ShutdownWithoutLogon

   HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon
   Name **ShutdownWithoutLogon**
   Type **REG_SZ** Value **0**

   Basis: It is unlikely an unauthorized user would gain physical access to the server (since it resides in a controlled access data center) and

shut it down via the Logon Dialog box. In the event an unauthorized user attempted to shutdown the server remotely (via the Logon Dialog box), this option would not be available.

G. Disable 8.3 name creation

Basis: This initially came from the MS IIS 4.0 Security Checklist (Step 3: Windows NT 4.0 Settings), "NTFS can auto-generate 8.3 names for backward compatibility with 16 bit applications. As 16-bit applications should not be used on a secure Web server 8.3 name generation can be safely turned off." If a deeper appreciation is desired, Schultz (95-96) provides an overview of how the 8.3 naming convention can be attacked on a Windows NT/2000 host, such as file name, file ownership & guessing a legitimate filename.

H. Enable the use of NTLM v.2

HKLM\System\CurrentControlSet\Control\Lsa

Name	**LMCompatibilityLevel**	Note: Add if necessary	
Type	**REG_DWORD**	Value **1**	(binary)

Basis: By default, all three versions of NTLM (challenge/response) authentication mechanisms are enabled.

- LAN Manager (LM) is the least secure and used to connect to share level security mode on computers running WFW 3.11 & Win9.x.
- NTLM version 1 is more secure than the LM challenge/response authentication and to connect to servers in a Windows NT domain that has at least one domain controller running Windows NT 4.0 SP3 or earlier.
- NTLM version 2 is the most secure form of challenge/response and is used to connect to servers in a Windows NT domain that have all the domain controllers upgraded to Windows NT 4.0 SP4 or later or when connecting to Windows NT hosts in Windows 2000 domain.

With the ease of which LM & NTLM hashes can be sniffed and cracked, I opted for using the more secure NTLMv2 when communicating with Windows NT 4.0 hosts.

I. Null Session Shares

HKLM\System\CurrentControlSet\Control\LSA

Name	**RestrictAnonymous**	Note: Add if necessary	
Type	**REG_DWORD**	Value **1**	(binary)

Basis: In an operating system rich with Authentication, Authorization & Auditing, there exists the means to authenticate an anonymous user (per se, a user who does not poses a user account) but may access a Windows NT (& Windows 2000) host by using the Access This Computer from the Network user right and null credentials logon (MS Windows NT 4.0 Security, Audit, and Control 103). The Everyone group contains the Interactive group ("Membership is granted to this group if the log-on process occurs on the local machine." (MS Windows NT 4.0 Security,

Audit, and Control 103)) and Network group ("…could potentially include any users who have been granted the Access This Computer from the Network user right and who have successfully logged on using network log-on process." (MS Windows NT 4.0 Security, Audit, and Control 103)) The anonymous user is also referred to the null user. Do not confuse the anonymous user with anonymous access (in IIS) & the anonymous (IIS) user. When IIS is installed, a local user is created which is named (by default) IUSR_*MachineName*. Schultz (138-139) describes the complexities of granting access to the Everyone group, "The Everyone group is critical to system functionality. The access that this group receives necessarily entails some degree of risk, but depriving this group of all access (especially Write access) causes system processes that expect, but cannot obtain, a certain level of access to particular files and programs to break (possibly even massively)." Windows 2000 creates an IPC$ share for inter process communication by default. A null connection is accomplished via IPC (Inter Process Communication). A constructive example of a null connection is a Domain Controller, from a different domain gathering a list of usernames of shares in your domain. A malicious example is the gathering this information (as well as Network information, registry key, etc.) during the enumeration phase of an attack. The initial enumeration of a Windows hosts will rely heavily on Null Session (Scambray and McClure 67). The following table (Scambray and McClure 75) lists the values that may be used.

Value	**Security Level**
0	None. Rely on default permissions.
1	Do not allow enumeration of SAM accounts and names.
2	No access without explicit anonymous permissions.

The default value (0) allows unauthenticated access to the IPC$ share (& enumeration of sensitive information: SAM accounts, registry key, share information, etc.) A value of 1 provides protection against the more common enumeration attempts, but is still insecure. A value of 2 provides the best protection, but also can break applications or functionality. Q246261 lists the conditions (for a Domain Controller) in which setting the RestrictAnonymous value to 2 is not suggested. We have not conducted sufficient testing to implement this level of security. A later section (IV. B.) demonstrates the techniques and tools used to evaluate this security setting.

J. Restrict Null Session Access

HKLM\System\CurrentControlSet\Services\LanManServer\Parameters

Name **RestrictNullSessAccess** Note: Add

Type **REG_DWORD** Value **1** (binary)

Basis: The default value (0) allows unauthenticated users to access all shared resources. When this value is set to 1, unauthenticated users can access only the server pipes listed in the value of the **NullSessionPipes**

entry and the shared directories listed in the value of the **NullSessionShares** entry (Regentry.chm).

K. Secure Print Drivers

HKLM\System\CurrentControlSet\Control\Print\Providers\LanMan Print Services\Servers

Name **AddPrinterDrivers**
Type **REG_DWORD** Value **1** (binary)

Basis: By restricting the ability to add printer drivers to Administrators and Server Operators, we are closing another potential security hole. "Print drivers often run with SYSTEM privileges. If a user can install one or more print drivers (the default Registry setting), that person could install a malicious program that runs as SYSTEM instead." (Schultz 200)

Note: The Event Logs sized are based on keeping two weeks worth of data. (MS Windows NT 4.0 Security, Audit, and Control 58). Guest access to the Application & System logs has been removed (Schultz 202). Both Application & System logs overwrite when they reach 10 MB, the System logs overwrite when they reach 100 MB.

L. Application Log Settings

1. Increase the size of the Application Event Log to 10 MB (10240 kb)
   HKLM\System\CurrentControlSet\Services\EventLog\Application
   Name **MaxSize** Note: Add if necessary
   Type **REG_DWORD** Value **10485760** (decimal)
2. Restrict Guest Access to the Application Event Log
   HKLM\System\CurrentControlSet\Services\EventLog\Application
   Name **RestrictGuestAccess** Note: Add if necessary
   Type **REG_DWORD** Value **1** (binary)
3. Application Event Log retention method: Overwrite Events as Needed
   HKLM\System\CurrentControlSet\Services\EventLog\Application
   Name **Retention** Note: Add if necessary
   Type **REG_DWORD** Value **0** (decimal)

M. Security Log Settings

1. Increase the size of the Security Event Log to 100 MB (102400 kb)
   HKLM\System\CurrentControlSet\Services\EventLog\Security
   Name **MaxSize** Note: Add if necessary
   Type **REG_DWORD** Value **104857600** (decimal)
2. Restrict Guest Access to the Security Event Log
   HKLM\System\CurrentControlSet\Services\EventLog\Security
   Name **RestrictGuestAccess** Note: Add if necessary
   Type **REG_DWORD** Value **1** (binary)
3. Security Event Log retention method: Overwrite Events as Needed
   HKLM\System\CurrentControlSet\Services\EventLog\Security
   Name **Retention** Note: Add if necessary
   Type **REG_DWORD** Value **0** (decimal)

N. System Log Settings

1. Increase the size of the System Event Log to 10 MB (10240 kb)

   HKLM\System\CurrentControlSet\Services\EventLog\System

Name	**MaxSize**	Note: Add if necessary	
Type	**REG_DWORD**	Value **10485760**	(decimal)

2. Restrict Guest Access to the System Event Log

   HKLM\System\CurrentControlSet\Services\EventLog\System

Name	**RestrictGuestAccess**	Note: Add if necessary	
Type	**REG_DWORD**	Value **1**	(binary)

3. System Event Log retention method: Overwrite Events as Needed

   HKLM\System\CurrentControlSet\Services\EventLog\System

Name	**Retention**	Note: Add if necessary	
Type	**REG_DWORD**	Value **0**	(decimal)

O. Enable SMB Signing

Basis: A registry entry can be created for when the host is acting in a server role, when the host is acting in a client role, or for both of these roles. In addition, the entry can be left as the default setting (not enabled – SMBs will not be signed), enabled but not required (SMBs will be signed if negotiated but will also connect with unsigned SMBs hosts), and required (will only accept connections from hosts that have signed SMBs). "The main value, therefore, is to lock out 'man-in-the-middle' clients (that is, clients inserted between the resources as the legitimate client)." "Using SMB packet signing can impose up to a 15 percent performance hit on file service transaction." (MS network client)

1. Secure Resource Sharing Through SMB Signing (client role)

   HKLM\System\CurrentControlSet\Services\Rdr\Parameters

Name	**EnableSecuritySignature**	Note: Add	
Type	**REG_DWORD**	Value **1**	(binary)

2. Secure Resource Sharing Through SMB Signing (server role)

   HKLM\System\CurrentControlSet\Services\LanManServer\Parameters

Name	**EnableSecuritySignature**	Note: Add	
Type	**REG_DWORD**	Value **1**	(binary)

P. Registry Entries to Harden the TCP/IP Stack

In meeting the goal of providing a hardened Web server for placement in a DMZ environment the registry received several entries to provide a more secure TCP/IP stack and specifically provide protection against a SYN-Attack, even though our exposed Web servers reside behind a firewall. These registry settings were taken from Security Considerations for Network Attacks; a brief description is provided after each registry entry. Prior to implementing these parameters in a production environment we put them through a series of progressive tests.

1. NoNameReleaseonDemand

   HKLM\System\CurrentControlSet\Services\Netbt\Parameters

Name: **NoNameReleaseOnDemand** Note: add
Type: **REG_DWORD** Value: **1** (Boolean)

Basis: Determines if the computer releases its NetBIOS name when it receives a name-release request from the network (protect the machine against malicious name-release attacks.)

2. EnableDeadGWDetect
   HKLM\System\CurrentControlSet\Services\Tcpip\Parameters
   Name: **EnableDeadGWDetect** Note: add
   Type: **REG_DWORD** Value: **0** (Boolean)

   Basis: TCP is allowed to perform dead-gateway detection. TCP may ask IP to change to a backup gateway if a number of connections are experiencing difficulty.
3. EnableICMPRedirect
   HKLM\System\CurrentControlSet\Services\Tcpip\Parameters
   Name: **EnableICMPRedirect** Note: add
   Type: **REG_DWORD** Value: **0** (Boolean)

   Basis: When this parameter is set to 0 (False) Windows 2000 will not alter it's route table in response to ICMP redirect messages that are sent to it by network devices such as a router.
4. EnableICMPRedirects
   HKLM\System\CurrentControlSet\Services\Tcpip\Parameters
   Name: **EnableICMPRedirects** Note: add
   Type: **REG_DWORD** Value: **0** (Boolean)

   Basis: Q293626 states a flow in how Windows 2000 looks for EnableICMPRedirect entry verse EnableICMPRedirects (plural). It was decided to include both values, based on an MS News Group post.
5. EnablePMTUDiscovery
   HKLM\System\CurrentControlSet\Services\Tcpip\Parameters
   Name: **EnablePMTUDiscovery** Note: add
   Type: **REG_DWORD** Value: **0** (Boolean)

   Basis: When this parameter is set to 1 (True) TCP attempts to discover the Maximum Transmission Unit (MTU or largest packet size) over the path to a remote host. By discovering the Path MTU and limiting TCP segments to this size, TCP can eliminate fragmentation at routers along the path that connect networks with different MTUs.
6. KeepAliveTime
   HKLM\System\CurrentControlSet\Services\Tcpip\Parameters
   Name: **KeepAliveTime** Note: add
   Type: **REG_DWORD** Value: **300,000** (decimal)

   Basis: Controls how often TCP attempts to verify that an idle connection is still intact by sending a keep-alive packet.
7. PerformRouterDiscovery
   HKLM\System\CurrentControlSet\Services\Tcpip\Parameters
   Name: **PerformRouterDiscovery** Note: add
   Type: **REG_DWORD** Value: **0** (decimal)

Basis: Controls whether Windows NT attempts to perform router discovery. This will prevent bogus advertisements.

8. SynAttackProtect
   HKLM\System\CurrentControlSet\Services\Tcpip\Parameters
   Name **SynAttatckProtect** Note: Add
   Type **REG_DWORD** Value: **2** (decimal)
   Basis: Causes TCP to adjust the retransmission of SYN-ACKS to cause connection responses to time out more quickly if it appears that there is a SYN-ATTACK in progress. NeonSurge/Rhino9 Publications has a nice description on SYN Floods starting at the three-way handshake, normal communication, SYN Flooders and recommendations to mitigate the vulnerability.
9. TcpMaxHalfOpen
   HKLM\System\CurrentControlSet\Services\Tcpip\Parameters
   Name: **TcpMaxHalfOpen** Note: add
   Type: **REG_DWORD** Value **100** (decimal)
   Basis: Controls the number of connections in the SYN-RCVD state allowed before SYN-ATTACK protection begins to operate.
10. TcpMaxHalfOpenRetried
    HKLM\System\CurrentControlSet\Services\Tcpip\Parameters
    Name: **TcpMaxHalfOpenRetried** Note: add
    Type: **REG_DWORD** Value: **80** (decimal)
    Basis: Controls the number of connections in the SYN-RCVD state for which there has been at least one retransmission of the SYN sent. Before SYN_ATTACK attack protection begins to operate.

IV. Apply, test and evaluate the template

A. Apply the Template

In its current form, the template is incomplete and requires several modifications (detailed in IV.D.). Once these have been added there will be several comparisons between our current build process (batch files & scripts) and using the template. This will eventually lead to the operational testing phase of adopting the template. When that aspect finally completed the template will be applied during the build process after all the applications are installed, including the IIS Lockdown tool. Currently, the IIS Lockdown tool further restricts our file system permissions (i.e. critical files), and this aspect will be incorporated into the security template. The IIS Lockdown tool installation will become an unattended installation and will include the URLScan 2.5 update. At this point, the security template will be applied via a batch file, with the results being saved for review and to build a history of the particular server. I will be conducting the initial reviews. Once these have successfully passed a break-in period, the build team will perform a cursory review for discrepancies.

I'm recommending a quarterly audit cycle for compliance with our build form. The obvious reason for this is to uncover any security settings that are not in compliance with our standard. For an intranet server, the

ramifications are less serious than for an Internet facing Web server. From here we can work with the application owner on how to best correct the problem and preserve application functionality. On the proactive side, the template can be used to verify that both development and production servers are configured exactly the same. I've spent too many long days troubleshooting a production server trying to determine why a Web application works in development, but not in production. Our current method of troubleshooting this problem amounts to brute force and ignorance. It is my hope that the security template can help reduce or eliminate this problem.

Once there has been at least one quarterly cycle conducted by hand then I will have the momentum to automate this. Since we have yet to implement one audit, this area hasn't received any attention (whether by GPO or a third party application).

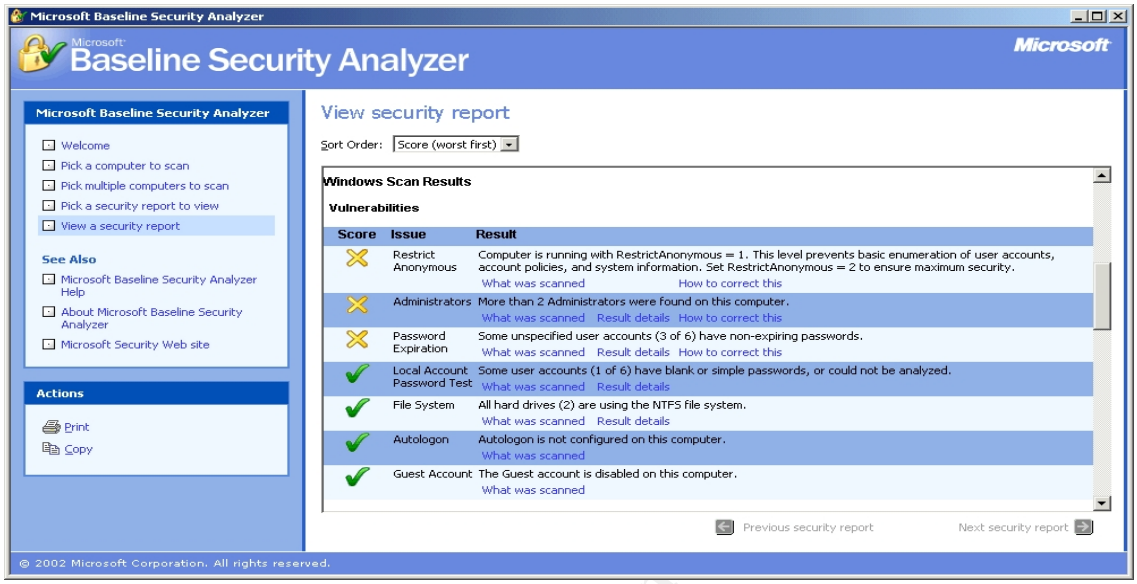
B. Test the Template's Security Settings

There were a couple of routes taken in testing these security settings. These are by no means the only ones available, nor are they the most strenuous tests that could be taken. These tests represent the methodology I've adopted for testing the security of a networked system. The first approach is to conduct an audit using the Microsoft Security Baseline Analyzer v. 1.1 (MBSA 1.1).

MBSA 1.1	**Purpose:** Windows and command-line application, can perform local or remote scans of Windows systems **Source:** See Appendix A

The second approach is to use some of the techniques & tools employed by the black hat community.

The current version of the MSBA (1.1) was run from a remote workstation. I prefer to have only the necessary components installed on servers. Since the MSBA 1.1 can be run from a remote workstation, this is the auditing method I use. It was run under a Domain Administrator's credentials to ensure it had sufficient permissions & access. As is always the case, the "Check for an update" box is selected to ensure the patches are up to date. The shortcomings addressed here will be discussed in the next section, D. Evaluate the Template. The following screen print describes what the MSBA considers common vulnerabilities.



A few years ago I came across the Windows NT WarDoc that described the methodology of enumerating and penetrating a Windows NT host. The mindset is "How to Improve the Security of Your Site by Breaking Into It." This time, I logged into a workstation (Win2k Professional) locally with NO domain privileges. This ensured I had the same level access as the bad guys. A few tools were used to test the security of the Web server:

Nbtstat.exe	**Purpose:** Command-line utility, displays protocol statistics and current TCP/IP connections using NBT (NetBIOS over TCP/IP) **Source:** Native to Microsoft Windows Operating Systems
NetWatch.exe	**Purpose:** Windows based tool, shows who is connected to shared directories **Source:** See Windows NT 4 Server Resource Kit in appendix A
RMTShare.exe	**Purpose:** Remote Share – Command-line utility, set of or deletes shares remotely and can grant and remove ACLs on those shares **Source:** See Windows NT 4 Server Resource Kit in appendix A
Enum.exe	**Purpose:** Tool to enumerate, using null and user sessions **Source:** See Appendix A

The first test came using Nbtstat to view the NetBIOS name table from the remote host.

Nbtstat –A *Remote IP Address*

The goal of the bad guy is to obtain a username. In a default configuration, the username of the currently logged on user will be listed in

the NetBIOS Name Table. Since our servers reside in a controlled access data center the most likely user who would be logged on (locally) would be an administrator. By comparison, since this would require an administrator to visit the server, it is also unlikely an administrator would be logged on locally, where the ability to remotely administrator the server via Terminal Services is present. Terminal Services has been configured to limit the amount of time a session can remain idle or disconnect before it is terminated. Under this configuration, an administrator will not be inadvertently logged in (& possibly allowing his/her username to be gleaned in this manner. It was gratifying to NOT see a username listed in the (NetBIOS) Name Table, since I was also logged on locally to the Web server undergoing this audit.

The next test was attempting a Null Session connect. Two utilities were used for this, Net.exe (Net Use & Net View) and NetWatch.exe. Here is the sequence of steps I followed:

Connect via the null user

(Net Use *Remote IP Address*\ipc$ "" /user:"")

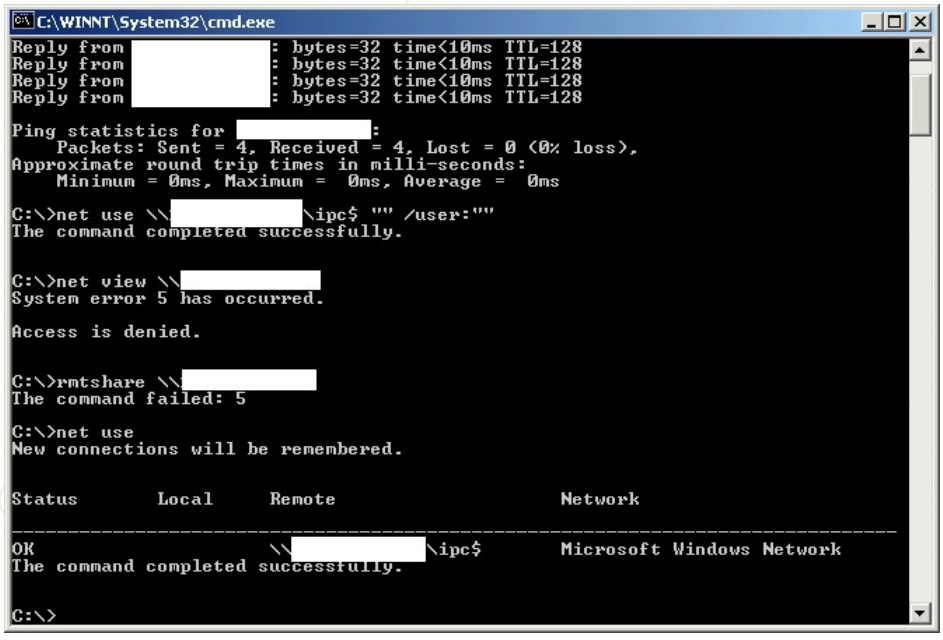
Attempt to enumerate the remote host via Net View

Net View *Remote IP Address*

View & manage remote shares (including hidden)

Rmtshare *Remote IP Address*

The following screen-prints reveals this simple attack approach has been blocked.



If a detailed explanation of the "The command failed: 5" is needed, then the Net command can be used. At the command prompt, type:

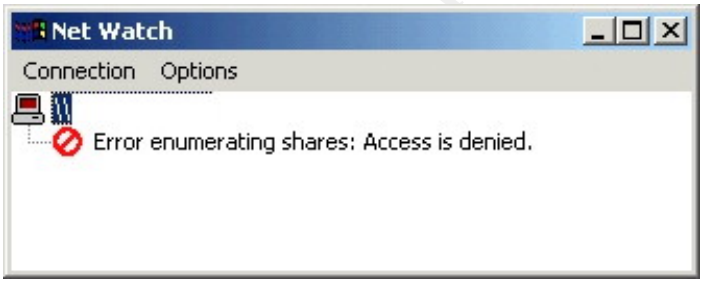
Net Helpmsg 5

Displays

Access is denied

To round out the file system attack, the NetWatch utility was used. Normally, this GUI utility will start, displaying a connection to the local host. From the Connection pull-down menu the Add Computer command is used to connect (with a Null Connection still in effect) to the remote Web server. The first indication this connection is not behaving normally is:

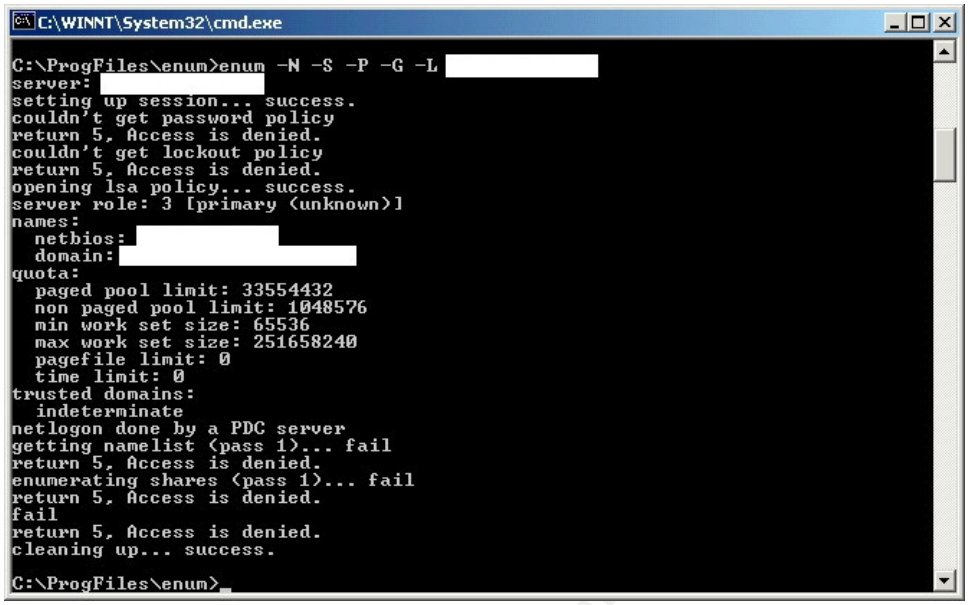
Acknowledging this by clicking Yes shows how the RestrictAnonymous=1 setting has provided some protection.



Continuing with enumeration, the Enum.exe is used in an attempt to glean various "interesting information". The following table lists the switches & explanation used:

-N	Get namelist
-S	Get sharelist
-P	Get password policy information
-G	Get group and member list
-L	Get LSA policy information

This revealed the servername & domain (**sanitized on this screen print**), which we were already able to obtain from the Nbtstat command. The information we were really after was not available to use (return 5, Access is denied.)



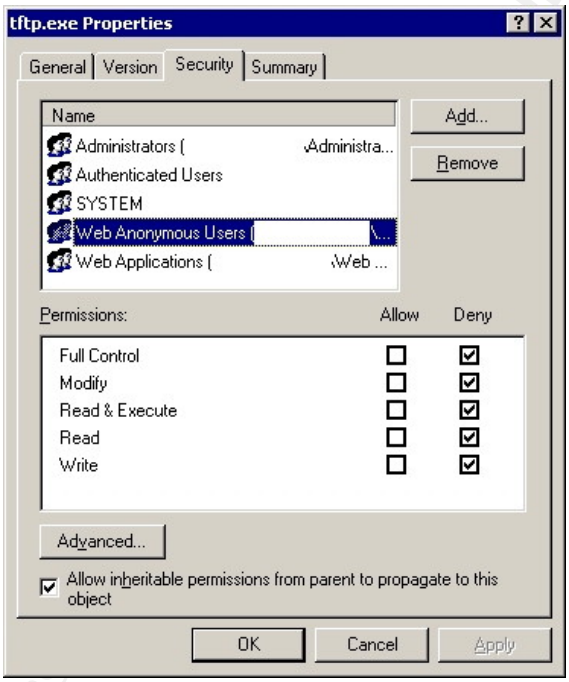
It is desirable to set RestrictAnonymous=2, but some networked applications would no longer function correctly. Under these conditions, the RestrictAnonymous=1 will have to do. At the very least, it does offer some protection against a file system attack.

The Web attack portion of the Windows NT WarDoc was not followed. I felt the material described specific vulnerabilities in IIS 4.0 and these either were not present (code improvements or that option not installed) or they had been addressed by a Service Pack or hot fix. At this point I switched to the style of Britney's NT Hack Guide. Again, I felt that either these vulnerabilities were not present or MS had released a patch (which was installed on this system). What I did follow was the methodology. I was not able to reproduce a successful attack on a Windows IIS 4 server so it was unrealistic to reference a Web based attack on this server (conducted by myself). Many modifications have been performed to secure the Web server:

- IIS was installed in non-default location
- IIS was installed with the minimal components
- IIS logging was enabled, in a non-default location
- Unnecessary script mappings were removed
- IIS Lockdown utility was installed, resulting is stricter file system permissions (see the following screen print)
- URLScan 2.5 upgrade was installed, with logging enabled in a non-default location

The item of interest, in relation to the Britney's NT Hack Guide the file system permissions was that even if I did have a list of "sensitive files" for Windows 2000, manually setting the deny ACL (for the IUSR_*MachineName*& IWAM_*MachineName*) would not be implemented due to administrative overhead. Of further note, in Exploit #1 of Britney's NT Hack Guide covers an exploit that I consider not to be present (either

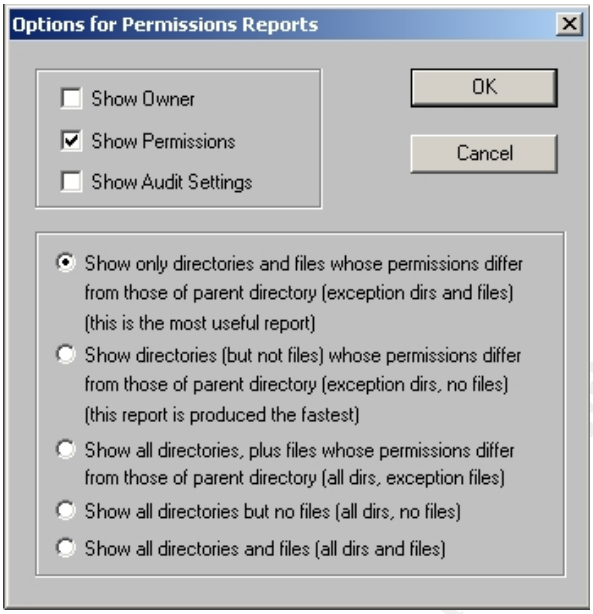
due to code improvements in IIS 5 or patch by a Service Pack/hot fix). It is the style of the attack, calling a sensitive file from the IUSR_*MachineName* account that I was trying to provide protection against. By exploiting a directory traversal in IIS the attacker is able to call the cmd.exe (as the IUSR_MachineName account). The next step is to upload, via the TFTP protocol, NetCat. Both of these files (cmd.exe & tftp.exe) have the deny attribute set on the ACL (for the IUSR & IWAM accounts). It is through a defense-in-depth perspective this is accomplished (minimal components, current patch status, reduced script mappings, URLScan utility and finally the deny ACL on particular files). The following screen print demonstrates how the ACLs on System Utilities are set via the IIS Lockdown utility.



With the use of DumpACL, a list of these System Utilities can be generated and incorporated into a security template.

DumpACL.exe — **Purpose:** Windows application that dumps the permissions (DACLs) and audit settings (SACLs) for the file system, registry, printers and shares in a concise, readable format
Source: See Appendix A

The following is a screen shot of the options available from DumpACL.



C. Test the System's Functionality

1. Administrative Functions

The administrative functions for this system are no different than any other Web server we have in place; a normal backup schedule must be maintained, anti-virus software must be updated, Compaq Insight Agents must be able to communicate with the Insight Management station and administrators must have connectivity for remote monitoring and a Terminal Services client connection.

The Veritas backup client was tested after the security modifications were applied by batch file, and after the modifications were applied via the security template. In both cases a successful backup was performed. In order to test the full functionality of the backup software, the D:\WebRoot folder was deleted, then restored from tape. The system functioned properly.

Antivirus software updates were installed without incident and Event Viewer & Virus Definitions indicated this system functioned properly.

This server was listed in CIM. We were able to perform all functions we require of CIM with no errors.

Just as the previous three applications were exercised, Network Associates PGP was observed for abnormal behavior. After the key rings were imported into PGP; fresh, encrypted files were downloaded and decrypted. To fully test the integrity of the data it was also unzipped (via Power Archiver) and imported into the database via the Paris5 administration utility. Please see the section IV. C. 2. covering the details from the application owner's testing.

The ability to remotely administrator this system after the template had been applied was conducted by multiple tools. Name resolution (both NetBIOS & IP address) was tested as well as the tool's functionality. For instance; Event Viewer, System Monitor, registry editor and some

command line utilities from the ResKit (ShutDown.exe, SCList.exe & SrvInfo.exe) & SysInternals (PsInfo.exe & PsShutdown.exe).

Shutdown.exe	**Purpose:** Command-line utility, allows you to remotely shut down or reboot a computer running Windows 2000 or Windows NT 4 **Source:** See Windows NT 2000 Server Resource Kit in Appendix A
Sclist.exe	**Purpose:** Command-line utility can show currently running services, stopped services, or all services on a local or remote computer **Source:** See Windows NT 2000 Server Resource Kit in Appendix A
SrvInfo.exe	**Purpose:** Command-line utility, displays information about a remote server, including available disk space, partition types, and status of services **Source:** See Windows NT 2000 Server Resource Kit in Appendix A
PsInfo.exe	**Purpose:** Command-line utility, list information about a system (local or remote) **Source:** See PsTools in Appendix A
PsShutdown.exe	**Purpose:** Command-line utility shuts down and optionally reboots a computer (local or remote) **Source:** See PsTools in Appendix A

Secondly, the IIS snap-in was used since it utilizes DCOM (communicates on random ports) unlike the previous set of tools. Past experience has shown that IIS may become unstable and continue to serve up static Web content. While in this condition, the SCM (Service Control Manager) will still report the Web services are "Running". In order to check the status of the IIS script engine (stable/corrupt) a simple ASP (status.asp) was written displaying "Hello World *date time*). This page reduces the time when isolating either a Web server problem or a Web application. This page, along with the default page for the Pcard application, was requested via IE 6 sp1 (with an empty cache). The system functioned as expected, with the status.asp updating the displayed time with each refresh.

From an administrative point, this system functions properly and is typical of what we expect of a new installation: no abnormal behavior or Event Viewer entries. Considering the conditions it was developed under, this was not surprise. We took a security checklist that was developed specifically for our environment and migrated over to a security template

2. Web Application Web User

Pcard users are required to review the credit charges and approve them. This involves connecting to the Pcard Web server via browser,

logging into the application (via a COM+ application & application user database) and reviewing their Purchasing Card statement.

The Pcard administrator utilizes functionality of the application and a second Web site (located on the same server) associated with the Pcard application. In addition to Pcard reviews, she also conducts approvals for Pcard users. She was able to login to both Web sites and conduct reviews and approvals without incident. From her perspective, this was an additional Pcard server that looked and acted just like our current (production) Pcard server.

D. Evaluate the Template

Since this checklist has been developed in house specifically for our environment, it is 100% compatible. It does not interfere with the development environment or the production environment. The settings reduce our exposure to malicious (internal) users and well as harden the system sufficiently to be placed in a DMZ. In no way did this template interfere with OS or application functionality. By no means is this template too secure. The opposite is true in that there are holes in our system hardening. I briefly flirted with a couple security templates from Microsoft and from the National Security Agency (NSA). These attempts demonstrated the need to develop a custom security policy. Since I was more familiar with batch files & scripts, it seemed this was the appropriate choice (based on my time constraints).

In order to appreciate what a well-designed security template can do, the review of this template will follow the security template's layout.

- Account Policies | Password Policy – Missed the mark with the batch file approach. This aspect was not even addressed in the batch file approach. Even though I've used the native NET.EXE account on several occasions, it was never considered for setting the Password Policy. At the very least password age (minimum & maximum) and password length could have been set.
- Account Policies | Account Lockout Policy – Missed the mark with the batch file approach. This aspect was also not even addressed in the batch file approach. I'm not aware of native utility (other than MMC) or ResKit utility that allows scripting this. Our utilization of VB Script is limited, so this avenue wasn't heavily considered.
- Account Policies | Kerberos Policy – Not addressed here.
- Local Policies | Audit Policy – Satisfied with the batch file approach. The options selected in the batch will be duplicated in the security template.
- Local Policies | User Rights and Assignments – Satisfied with the batch approach but there is potential to accomplish more here. The only two rights presently addressed are "Access this computer from the network" and "Logon locally". The underlying motive on both of these was to replace the Everyone group with the Authenticated Users group. All the assignable rights will be audited for the presence of the Everyone group,

and then possibly replaced with the Authenticated Users group. Some of the additional resources already on hand for guidance are the ResKit and sample templates (list is Appendix C).

- Local Policies | Security Options – Core to maturing the security template. The default list of options provided (in a new template) is a solid starting point, but will have to be built on. For example, there are ten registry additions for hardening the TCP/IP stack alone, with more under review. Q214752 explains in step-by-step detail how to add custom entries to a security template. In addition to being straightforward & accurate, it is easy to accomplish (adding additional template entries.) For a more complex entry (i.e. LM Authentication Level) Eric Shultz has a template available for review (listed in Appendix A). Once the custom entries have been created they will undergo testing to ensure the desired results are obtained.
- Event Logs | Settings for Event Logs – Good, but not complete coverage from the batch file approach. Log retention was not addressed in the batch file approach. The security template will round out and polish the Event Log settings.
- Restricted Group – Good coverage, but only partial use from the batch file approach. By creating a TelnetUsers group (and leaving blank) I blocked users from connecting vial telnet (in addition to disabling the service). The security snap-in allows for greater flexibility and control with this by creating a Restricted Group policy (MS Win2k Security TechRef 315-6). When the template is applied, the group will be created, and left empty. This can also be used to audit the members of the local Administrators group. A good thing just got better.
- System Services – Satisfied with the batch approach but there is a lot of potential here. Many services had their startup value changed to manual or they were disabled. This may be revisited and a few more services placed in a Manual startup state (from an Automatic startup state). The real bonus of using the template is setting the permissions on the service itself. Once again, the prospect of replacing the Everyone Group with the AU group will be closely looked at.
- Registry – Lightly used, good potential, but untested in our environment. The "Winreg" key is only the portion of the registry that has had the ACL changed. By using the template to automate this, a small amount of time will be saved as well as removing the chance of the keyboard being fat-fingered (since this was done manually). Once again, the prospect of replacing the Everyone Group with the AU group is very tempting. The setting of permissions on particular keys (Run, RunOnce) so the SYSTEM & Administrators only have Read access is also tempting for Trojan prevention but it is not very realistic on production servers at this time.
- File System – Core to maturing our template. The fact that the template can automate the setting of files system permissions alone is more than enough justification to migrate (to a template approach). Not only will it require less time to set permissions, but the chance of a mistake being

made is markedly lower (than by having an operator manually set permissions.) MS Win2k Security TechRef (318) provides step-by-step instructions for customizing files/folders ACLs. I look forward to the day file system permissions are audited. In addition to ensuring the development and production environments are in alignment, the newly formed testing environment will also benefit.

Just as the drive to automate and standardize our Web server environment spawned multiple batch files and scripts, the need for a more encompassing security policy with the ability to audit (for compliance) brought the security template to the forefront. While the batch files saved time, they didn't cover all aspects that were required (such as account lockout policy and file system permissions). The time required to build, tune and secure a Web server dropped drastically with batch files, but the ability to audit this configuration didn't exist (in an automated sense). By comparison, the process of examining our batch file approach so it can be migrated to a template provided a "critique" of our current method. In addition to the glaring holes and subtle absences that were discovered, a better method for implementing file system permissions will be used (via the template).

It is one thing to sell the template on its ability to provide quicker, more consistent and more encompassing security during the build process, but quite another to receive the go ahead to implement auditing our environment for compliance with the build (tune & secure) process. The ideal argument (to convince management) is a painful email stating the new (Web application) code works on the development server but not on the production server, and we had lost consistency across our development and production servers all the while we possessed the tools to prevent this.

This is a prime example of The Immutable Security Administration Law #7 states, "The most secure network is a well-administered on". Appendix A has a link to the MS "Security Screen Savers". Once this security template is complete, it will assist us in providing better administration (more complete and in a timely manner) to a key component of our infrastructure.

Works Cited

"Microsoft Internet Information Server 4.0 Security Checklist" Microsoft TechNet Web Site. 24 July 2001. Microsoft Corporation. 31 December 2002 <http://microsoft.com/technet/security/tools/chklist/iischk.asp?frame=true>

"Microsoft network client: digitally sign communications (always)." Microsoft Tech Net Web Site, Microsoft Corporation 10 Feb 2003 < http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/windowsnetserver/proddocs/server/568.asp>

"Q142641 Internet Server Unavailable Because of Malicious SYN Attacks." Microsoft TechNet Web Site. 24 July 2001. Microsoft Corporation. 31 December 2002 <http://support.microsoft.com/default.aspx?scid=kb;en;Q142641&sd=tech >

"Q164882 Practical Recommendations for Securing Internet-Connected Windows NT Systems." Microsoft TechNet Web Site. 14 Nov. 2002. Microsoft Corporation. 27 December 2002 <http://support.microsoft.com/default.aspx?scid=kb;en-us;Q164882&sd=tech>

"Q187506 List of NTFS Permissions Required for IIS Site to Work." Microsoft TechNet Web Site. 10 June 2002. Microsoft Corporation. 27 Dec. 2002 <http://support.microsoft.com/default.aspx?=kb;EN-US;187506>

"Q214752 How to Add Custom Registry Settings to Security Configuration Editor." Microsoft TechNet Web Site. 10 Oct 2002. Microsoft Corporation. 27 Dec 2002 http://support.microsoft.com/default.aspx?scid=kb;en-us;Q214752&sd=tech

"Q246261 How to Use the RestrictAnonymous Registry Value in Windows 2000." Microsoft TechNet Web Site. 13 Aug. 2002. Microsoft Corporation. 31 Dec. 2002 <http://support.microsoft.com/default.aspx?scid=KB;en-us;q246261>

"Q250908 Creating a Local Group Can Restrict Other Users from Gaining Access to a Windows 2000-Based Computer Through Telnet." Microsoft TechNet Web Site. 11 Oct. 2002. Microsoft Corporation. 5 Feb. 2003 <http://support.microsoft.com/default.aspx?scid=kb%3ben-us%3b250908>

"Q259671 How to Change the Default Installation Paths for FTP and the Web." Microsoft TechNet Web Site. 15 Aug. 2002. Microsoft Corporation. 5 Feb. 2003 <http://support.microsoft.com/default.aspx?=kb;en-us;q259671>

"Q271071 Minimum NTFS Permissions Required for IIS 5.0 to Work." Microsoft TechNet Web Site. 4 Dec. 2002. Microsoft Corporation. 27 Dec. 2002 <http://support.microsoft.com/default.aspx?scid=kb;EN-US;271071>

"Q281892 Windows 2000 Unattend.Doc File Provides Incorrect FrontPage Server Extensions Parameter." Microsoft TechNet Web Site. 11 Oct. 2002. Microsoft Corporation. 5 February 2002 < http://support.microsoft.com/default.aspx?scid=kb;en-us;q259671>

”Q293626 Cannot Disable ICMP Redirects By Changing ‘EnableICMPRedirect’ Registry Value.” Microsoft TechNet Web Site. 11 Oct. 2002. Microsoft Corporation. 4 Feb. 2003 <http://support.microsoft.com/default.aspx?=kb%Ben-us%B293626>

“Securing IIS 5.0 Using Batch-Oriented Command Files.” Microsoft TechNet Web Site. Sept. 2000. Microsoft Corporation. 4 Feb. 2003 <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/iis/deploy/config/seciis50.asp>

“Security Considerations for Network Attacks.” Microsoft TechNet Web Site. May 2002. Microsoft Corporation. 04 Feb. 2003 <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/prodtech/network/secdeny.asp>

Boswell, Bill. “Mementos of Windows 2000.” Microsoft Certified Professional Magazine October 2002: 44-50.

Davis, John “From Blueprint to Fortress: A Guide to Securing IIS 5.0.” Microsoft TechNet Web Site. July 2001. Microsoft Corporation. 27 Dec. 2002 <http://microsoft.com/TechNet/prodtechnol/iis/deploy/depovg/secreiis.asp?frame=true>

Howard, Michael “Secure Internet Information Services 5 Checklist.” Microsoft TechNet Web Site. 29 June 2000. Microsoft Corporation. 27 December 2002 <http://www.microsoft.com/TechNet/prodtechnol/iis/tips/iis5chk.asp?frame=true>

Microsoft Technical Reference. Microsoft Windows 2000 Security Technical Reference. United States of America; Microsoft Press, A Division of Microsoft Corporation, 2000

Microsoft Technical Reference. Microsoft Windows NT 4.0 Security, Audit, and Control. United States of America, Microsoft Press, A Division of Microsoft Corporation,1999.

NeonSurge and Rhino9. SYN Floods and SYN Cookies. 31 December 2002 <http://www.aliceinwonderland.com/library/website_archives/rhino9/synflood.html>

Scambray, Joel and Stuart McClure. Hacking Exposed Windows 2000: Network Security Secrets & Solutions. Berkeley: Osborne/McGraw-Hill 2001.

Schultz, E. Eugene. Windows NT/2000 Network Security. United States of America: Macmillan Technical Publishing, 2000

Sideb0ard, Thorsten. Britney’s Guide to Hacking NT in 5 Easy Steps. January 2001. 2 January 2003 <http://www.interphaze.org/bits/britneysnthackguide.html>

Appendix A
Tools Used to Configure & Test IIS 5.0 Web Server

"DumpACL.exe" SystemTools Software Inc. 20 Mar 2003
http://www.somarsoft.com/
"Enum.exe" by Jordan Ritter, BindView. 20 Mar 2003
http://razor.bindview.com/tools/desc/enum_readme.html
"Internet Explorer Administration Kit 5.5" Microsoft Corporation. 20 Mar 2003
http://www.microsoft.com/windows/ieak/previous/techinfo/ie55/default.asp
"IIS Lockdown Tool" Microsoft Corporation. 20 Mar 2003
http://microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/tools/locktool.asp
"Microsoft Baseline Security Analyzer" Microsoft Corporation. 20 Mar 2003
http://microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/Tools/MBSAhome.asp
"Microsoft Windows 2000 Server Resource Kit" Microsoft Corporation. 20 Mar 2003 < http://www.microsoft.com/mspress/books/1394.asp>
"Microsoft Windows NT Server 4.0 Resource Kit Supplement 4" Microsoft Corporation. 20 Mar 2003 http://www.amazon.com/exec/obidos/tg/detail/-/0735608377/qid=1048367326/sr=1-3/ref=sr_1_3/002-0568791-1256864?v=glance&s=books
"NetCatNT.exe v1.1" by Hobbit, @Stake. 23 Mar 2003
<http://atstake.com/research/tools/network_utilities/>
"NTFilemon.exe" SysInternals. 20 Mar 2003
http://www.sysinternals.com/ntw2k/source/filemon.shtml
"PsTools" SysInternals. 20 Mar 2003
< http://www.sysinternals.com/ntw2k/freeware/pstools.shtml>
"Security Screen Savers" Microsoft Corporation. 20 Mar 2003
http://microsoft.com/Downloads/details.aspx?displaylang=en&FamilyID=6015F85B-9A3A-4AEB-8E50-28005312398A
"Urlscan Security Tool" Microsoft Corporation. 20 Mar 2003
http://microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/tools/URLScan.asp

Appendix B
Additional References for CIFS/SMB & LM, NTLM & NTLMv2

Hobbit. CIFS: Common Insecurities Fail Scrutiny. Jan 1997. 28 Jan 2003 http://web.textfiles.com/computers/cifs.txt

Leach, Paul and Dan Perry. CIFS: A Common Internet File System. Microsoft Corporation, MSDN Web Site. 10 Feb 2003 http://www.microsoft.com/mind/1196/cifs.asp

Scambray, Joel and Stuart McClure. Hacking Exposed Windows 2000: Network Security Secrets & Solutions, Ch. 5: Hacking CIFS/SMB. Berkeley: Osborne/McGraw-Hill 2001.

Sharpe, Richard. Just What is SMB, v1.2? 8 Oct 2002. 10 Feb 2003 http://samba.anu.edu.au/cifs/docs/what-is-smb.html

Smith, Randy Franklin. Inside SP4 NTLMv2 Security Enhancements. Windows & .NET Magazine – Security Administrator, September 1999. 24 Mar 2004 < http://www.ntsecurity.net/Articles/Index.cfm?ArticleID=7072>

Appendix C
Security Template Sources

"Guide to the Secure Configuration and Administration of Microsoft Internet Information Services 5.0." National Security Agency Web Site, Systems and Network Attack Center. 4 Mar 2002 National Security Agency. 24 Mar 2003 < http://www.nsa.gov/snac/win2k/guides/w2k-14.pdf>

"Q316347 IIS 5: HiSecWeb Potential Risks and the IIS Lockdown Tool." Microsoft TechNet Web Site. 15 Jan 2002. Microsoft Corporation. 24 Mar 2003 < http://support.microsoft.com/default.aspx?scid=kb;en-us;316347>

Reid, Gavin "IIS 5.0 and Windows 2000 Hardening Guide." Aug 2002. 24 Mar 2003 < http://www.shebeen.com/w2k/>

Schultze, Eric "Web_Secure.inf." SystemExperts Web Site. 24 Jan 2000. SystemExperts. 24 Mar 2003 http://www.systemexperts.com/win2k/HardenWin2K.html