



## 2002 Buyers' Guide

Issue Archives

Subscribe/  
Change Address

Infosec Jobs

Editorial

Editorial Calendar  
Contact the Editors  
Staff Biographies

Vendor Links

Happenings

Advertising Info

Rate Card  
Editorial Calendar  
Testimonials  
Internet Opportunities  
Contact a Sales Rep  
BPA Statement  
List Rental Info  
Reprint Info

Security Wire Daily

Wireless SWD

Back/Missing Issues

About/Contact Us

Directions

Privacy Statement

Security Wire Digest

Read Current Issue  
SWD Archives  
Subscribe to SWD

Home

## FEATURES

August 2001

### IIS SECURITY

#### 10 STEPS TO BETTER IIS SECURITY

These quick and easy tips will help you harden your Microsoft Web server.

BY RUSS COOPER

You've done everything you can to install your Microsoft Internet Information Server (IIS) securely, making full use of the valuable c the Internet and all the resources at your disposal.

Now, how do you keep it secure? You want to maintain tight security, while one of your most precious resources--time.

A sound and practical security philosophy is to expend 20 percent of your ti percent of the problems. The measures below are quick and easy to implem thwart 80 percent of the potential attacks against your IIS server.

#### 1. Remove RDS Registry Keys

The most common attack against an IIS server exploits Remote Data Servic RDS is vulnerable because it allows a conduit to Open Database Connectiv functionality that would permit DOS commands to be run as System on the Through this, a malicious hacker can remotely gain access to an IIS server.

This attack persists as a threat three years after its discovery because it car corrected by a patch or service pack. Instead, you have to delete certain Re

There are two misconceptions about removing these Registry keys. One of t have stopped you from taking the appropriate action:

1. RDS is used to allow browser clients to construct queries directly agai data sources (e.g., a SQL database). With this functionality, a client b would establish a connection directly to the database, submit queries data. This isn't the usual method for executing client queries to the d RDS isn't needed in most cases. Most Web servers that permit client c against databases do so on behalf of the client (that is, using the clie and not System authority). The Web page accepts the client's query p and the Web server (not the client browser) submits the request to th This allows the database server to be protected from direct interactio client. If you're not allowing direct network connections between the c the database server, you're likely not using RDS functionality.
2. Allaire's ColdFusion ([www.allaire.com](http://www.allaire.com)) product also uses the term "R is a development system (Remote Development Service) that has not with Microsoft's RDS. So, removing the RDS Registry settings won't a ColdFusion.



To resolve the problems associated with RDS, remove the following Registry any sub-keys:

- HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\RDS\Server.DataFactory
- HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\Advanced
- HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\VbBusObj.VbBusObjCls

## 2. Don't Let Hackers Exploit DOS

The second most common method of attacking an IIS server is by causing the underlying OS to execute a command of the attacker's choice. To do this, the attacker must invoke a DOS shell, calling CMD.EXE on Windows NT/2000.

Many of the known IIS vulnerabilities allow a URL to be constructed that ultimately causes CMD.EXE to be invoked on the server. The attacker then appends DOS commands, such as ECHO, to the command-line parameter for CMD.EXE. This is equivalent to starting a DOS batch file.

Using this method, an attacker can invoke FTP, with instructions, in a command prompt on the server. The attacker opens an outbound FTP connection to a server of his choice and downloads a file or files that he wants to use (such as programs that give him a remote console on your server). The attacker then sends another URL to you to call the programs he just downloaded, and away he goes.

Many of us have grown up using the DOS command shell and like to have it on our servers for administration. Unfortunately, it can be exploited.

This risk can be mitigated, since most attacks against IIS servers aren't common means, typically, that a hacker discovers a weakness and crafts a way to exploit it. Usually, this information is published. Most attacks aren't by the creators of the software but by members of their hacker audience. The result is that if your IIS system has the default configuration, any published attack is likely to fail. Using someone else's weapon, an attacker is unlikely to do very much, if anything, to alter the system or get it to work.

So, if CMD.EXE isn't where it's expected to be or doesn't exist at all, the overwhelming majority of exploits that rely on it are going to fail. In such cases, an attacker will move on to another target.

On NT 4.0 systems, CMD.EXE can be deleted, renamed or moved to another location. Also, remove the COMSPEC environment variable, since it points directly to the location of CMD.EXE. If you renamed or moved CMD.EXE, you don't want to re-point COMSPEC, which would help an attacker. If you delete CMD.EXE, COMSPEC has nothing to do.

On Windows 2000 systems, removing CMD.EXE is a little more difficult because of Windows File Protection (WFP). CMD.EXE will automatically be replaced by a protected version if you delete, rename or move it. However, you can assign explicit access permissions to members of the Administrators group. You should explicitly deny all access to SYSTEM and IUSR/IWAM accounts (see [#8: "Limit Permissions, Lock Out At Least One Account"](#)) as well as any other accounts that you use in your Web site.

## 3. Secure ODBC Operation

ODBC is the most-used method for accessing databases on a Windows system. Unfortunately, ODBC inherently allows for DOS commands to be embedded in SQL calls. Several vulnerabilities have arisen that demonstrate how DOS commands can be chained together in ODBC queries, resulting in the invocation of CMD.EXE. Windows provides a mechanism to prevent ODBC calls from invoking DOS commands.

Look in your Registry for the following key:

- HKEY\_LOCAL\_MACHINE\Software\Microsoft\Jet

Under that key, you'll usually find a sub-key, 3.5 or 4.0, and under that key another sub-key, Engines.

Check for a DWORD value called SandboxMode and make sure its value is 3.

- HKEY\_LOCAL\_MACHINE\Software\Microsoft\Jet\3.5\engines\SandboxMode
- HKEY\_LOCAL\_MACHINE\Software\Microsoft\Jet\4.0\engines\SandboxMode

You only need the latest key, so if you have 4.0, you don't need 3.5.

In general, this setting should be fine for 95 percent of you. There are other options for this value, and some of you may need to allow for a less-restrictive setting depending on what your ODBC applications need to do. For more information, see "Expression Can Execute Unsafe Visual Basic for Applications Functions" in [Microsoft Knowledge Base](#).

#### 4. Remove Dangerous Extension Mappings

Numerous vulnerabilities have been found in how IIS handles specific Web page extensions. For example, .HTR was intended to provide a helper application for Web visitors to change their passwords. This added functionality introduces a vulnerability that an attacker might exploit. Other extensions had purposes that have since passed the times in IIS's evolution, but now are no longer (or rarely) used.

Out of the box, IIS supports a wide variety of page requests. Extension mappings map specific types of URLs to be handled by different programs, such as ISAPI. In the vast majority of IIS installations, all that's needed is support for .HTM, .HTT, .ASP, and .ASA, which doesn't require any extension mapping, and .ASP/.ASA, which requires mapping to ASP.DLL.

Extension mapping is part of Internet Services Manager, found by right-clicking on the virtual Web root. Choose Properties, then Home Directory, then Configuration. Note all of the mappings and their verbs (or exclusions on IIS 4.0) for future reference.

The next step is to determine what mappings you actually need on your site. A Web developer should be able to tell you exactly which mappings he uses. You can check through your Web directories to see what file extensions are present. Make sure you only check your production directories and not any sample directories. Then remove the mappings that aren't in use. There's one caveat--a vulnerability, announced in May, regarding the .printer extension on IIS 5.0, that could allow execution of remote commands. The .printer extension allows for Web-based printing, something most sites don't need (but is needed by Outlook Web Access sites). Removing the mapping for .printer mappings because of Windows 2000 Group Policies, which, by default, indicate that Web printing is available. If you remove the mapping, then reboot, the mapping is restored. To prevent this, do the following:

- Launch the Microsoft Management Console and load the snap-in for Group Policy.
- Select Computer Configuration, then Administrative Templates, then Internet Options.
- Check the setting for Web-based printing, and make sure that it's disabled.
- If the server is part of a domain, make sure that Web-based printing is disabled in the domain Group Policy.

## 5. Clean Up Your Server Before Going Into Production

One of the most commonly overlooked steps when putting an IIS server into production is removing all of the stuff that might have been added to aid in development. For example, FrontPage Server Extensions might have been used by Visual InterDev during the development of the Web application, or sample files from IIS's default installation might have proved useful to understanding some new feature.

A production IIS server needs to be as clean as possible, free of unnecessary files. Vulnerabilities have been uncovered in samples that IIS ships and installs by default, including one of the three Registry keys that cause the RDS problem. Two are used in normal operation of RDS, but the third was only present when samples were installed. Initially, many people removed the two required keys, but overlooked the sample key.

Samples aren't intended for a production environment and haven't been fully tested. They can give an attacker access into a production server, even when the production directories have been secured. To eliminate this vulnerability, remove all unnecessary Web directories, applications and files from your production machines before putting them into service.

Development servers with direct access to the Internet can also be compromised. In the past couple of months, many IIS servers have been defaced by the sadm worm. After infecting a Sun Solaris box, sadmind/IIS scans for vulnerable IIS servers. The Solaris box then attempts to exploit the IIS vulnerability to deface the server's home page. Most of the compromised servers were not the primary Web server for their domain. Many of them were simply boxes under development that few people had direct Internet access.

The point is that not only production servers need to be secured. If you're developing a Web server, ideally you don't want it exposed to the Internet until it's been secured.

## 6. Install Only What You Need

A default installation is likely to be insecure. So, rather than doing a default installation and then looking for a checklist of what to remove, install only those components that are required for the IIS server you're working on. Items can always be added later.

An example of why this helps is the recent SMTP problem with Windows 2000 IIS 5.0. By default, all installations of IIS 5.0 include an SMTP server, which most people don't need. If the server is part of a domain, the SMTP server can be made to act as an SMTP relaying. SMTP relaying is the reason most spam exists today.

## 7. Re-Partition Key System Files

Most exploit scripts rely on the fact that directories are going to reside in their default partition. Obscurity isn't security. However, if you move or install key

files to partitions other than the default root drive, published scripts won't v modifying them to look in the correct partition. This can help you thwart ma

For example, numerous vulnerabilities have been associated with the traver directories, using the ..\ command. These vulnerabilities allow attackers to . restricted directories such as \WINNT or \WINNT\SYSTEM32.

Normally, these directories and the sensitive applications they contain are f root partition. By default, IIS installs itself into the same partition. However instead tell IIS to install itself into another partition, you reduce this vulner:

Directory-traversal attacks cannot work across partitions, so the attacker w to access \WINNT or \WINNT\SYSTEM32 in this environment. If \INETPUB is same partition as \WINNT, those sensitive applications won't be available to attacker, and the exploit will fail.

Of course, you can rename \INETPUB, but it's a pretty simple step to create for your Web directories and place everything in it.

## **8. Limit Permissions, Lock Out Attackers**

An often overlooked IIS feature is its ability to enforce permissions on Web creates anonymous accounts: IUSR\_machinename and IWAM\_machinename the account under which all non-authenticated users access pages. All Web (those that use a global.asa) are started under the IWAM account. These tw are critical on public Web sites.

To prevent these accounts from being exploited, make sure that they have i permissions to files that should not be directly accessed.

Of most importance is the %SYSTEMROOT% directory and its sub-directorie (\WINNT). You should explicitly deny access by the IUSR account--add it to i permissions of that directory, and give it no rights. The IWAM account is sin most sites, although it may require access to some sub-directories, dependi applications you have and where you installed them. Testing will help you d your Web site is working properly after you deny IWAM access.

## **9. Read Your IIS Logs**

You might be amazed at what you can find in your IIS logs. Most attacks ag servers are executed by sending the server a specially crafted URL, which is recorded in your logs.

Assuming you log to a flat text file--the IIS default--do a search for files th CMD.EXE or ECHO. If you've been under attack, either by an automated too individual attacker, chances are you'll find one of those text strings in your

Being aware of the attacks against you is useful in helping determine what : do to prevent exploits. For example, you could ban the attacking IP address site (through IIS Manager) or filter them at your upstream router. You migh [Google](#) or other search engines for your domain name to see if you're expos other page from your site. If you find pages that you didn't want in the sear you can use the ROBOTS.TXT file to control the way search engines crawl y

## **10. Keep Fixes Current**

You must try to stay current with Microsoft hotfixes and service packs that Microsoft releases. In August, Microsoft released a new tool called HFNETCHK, which you can use to call Microsoft's Security site and retrieve an up-to-the-minute XML file that contains information about all of the hotfixes that your system might need. It works on Windows 4.0 and Windows 2000 and covers the OS, IIS, Internet Explorer and SQL Server.

The tool is an executable that runs on your server. It checks to see what you have installed, what hotfixes should be applied and whether they're installed. You can find the Microsoft Security Bulletins referenced in the output, apply the hotfix or service pack as needed, and run the tool again.

That's the easiest way to keep up to date. You should also subscribe to Microsoft's [Security Bulletin](#), which will keep you posted on the latest vulnerabilities, patches and workarounds. [NTBugtraq](#) will also keep you informed of the most important security issues.

### **A Little Effort, a Lot of Security**

Simply installing the IIS server securely is important, but it's not enough. Protecting your server against outside attacks is an ongoing job, but it doesn't have to be a full-time job. The steps and principles discussed in this article will enhance your security enormously and give you precious time for all of your other mission-critical tasks. Install only what you need, keep abreast of current issues and fixes, and respond quickly to sensitive areas on your server. It's truly an ounce of prevention that's worth a pound of cure.

**RUSS COOPER** ([russ.cooper@rc.on.ca](mailto:russ.cooper@rc.on.ca)) is "Surgeon General" of TruSecure (formerly TruSecure) and editor of NTBugtraq.

### **SIDEBAR**

SubOS: Armor for tomorrow's secure browser?

BY PETE LOSHIN

Illustration by Christine Hajar

"If a bad guy can persuade you to run his program on your computer, it's not your computer anymore." That's Law #1 of "The 10 Immutable Laws of Security," a Microsoft Security Response Center [white paper](#).

Hackers love to "own" your servers, but browsers offer tasty targets, too. Researchers building fundamentally new approaches to securing the browser warn that current browsers continue to be vulnerable...for now.

Every time you download a Web page with active content--such as program code like JavaScript or VBScript--you risk inviting a bad guy to own your computer.

Can you differentiate between "bad guys" and "good guys" when it comes to their code?

No, you can't, says Scott Schnoll, security expert and maintainer of the (unofficial) Internet Explorer Security Center (IESC) [Web site](#). The safe solution: "Don't allow content to execute in your browser," he says.



Schnoll recommends turning off browser options that allow any code--script controls, Java and so on--to run, except for code coming from trusted sites. HTML and simple images can't infect a computer, steal your credit card info wipe out your hard drive," he says.

In the face of the inability of commercial browsers, such as Netscape and Internet Explorer, to stop malicious applications, two researchers have begun work on a secure browser."

The heart of the Web browser security problem is really the lack of flexibility in operating systems, according to Sotiris Ioannidis, a Ph.D. candidate working in the Distributed Systems Lab at the University of Pennsylvania, and Steven M. Bellovin, an AT&T Fellow in the Communications Information Systems Research Department at AT&T Labs Research.

Ioannidis and Bellovin addressed this weakness by developing a protection mechanism they call SubOS, described in a paper, "[Building a Secure Browser](#)", presented at a recent USENIX conference.

SubOS prevents helper applications (such as Adobe Acrobat and RealAudio) from gaining more system privileges than they should. According to the paper, "Under SubOS, any application (e.g., ghostscript, Perl) that might operate on possibly malicious objects (e.g., postscript files, Perl scripts) behaves like an operating system, restricting their accesses to system resources."

Each new object that comes into a system is assigned its own sub-user ID and limited privileges. Applications are treated like users, with access restricted to necessary system resources. In effect, SubOS operates like a buffer OS between the potentially dangerous objects and the OS. Bellovin and Ioannidis worked with OpenBSI and say it would be possible to implement on other systems, such as Windows.

SubOS security mechanisms need not be limited to browsers. Mail clients could also be made secure using this model, too.

"As for browser security--today is an even day, so I think that the mailer is more dangerous than the browser," Bellovin says. "I'll change my mind tomorrow if I see something very risky, because they accept complex payloads from arbitrary places."

The secure browser is a work in progress.

"For now, it's a research effort, and I don't foresee an immediate change," Ioannidis says. "It requires new operating system code to do things right, and there are a few organizations that can effectively do that. It also requires a different mix of application writers; that's what we were trying to stimulate."

### **What Do We Do Now?**

SubOS--secure applications may be the wave of the future, but what can be done in the meantime?

The two primary strategies for browser security are (1) limiting when code is executed and (2) making sure that all security fixes are installed.

Schnoll says users should protect themselves against malicious content by checking Netscape Communicator's Preferences to disable Java and JavaScript; Microsoft Internet Explorer provides greater flexibility by allowing users to set the appropriate security level for each of four Security Zones. Unfortunately, "Most users don't know how to c

security-related browser settings or update their Web browser with security and patches."

Security professionals should increase security awareness among users through policies and enforcement.

**Some tips:**

Disable Macros in Microsoft Office applications through document security settings.

Use system and network management tools for updating enterprise-wide software to help install security updates and patches as well as to set security defaults.

Consider alternative applications (and operating systems) that are less vulnerable to common attacks.

Educate users to avoid opening e-mail attachments from unfamiliar sources and encourage use of data formats that don't incorporate executable content, such as Microsoft Office Macros.

**PETE LOSHIN** ([pete@loshin.com](mailto:pete@loshin.com)) is a senior editor-at-large for *InformationWeek*. He produces the Internet-Standard.com Web site and has authored more than 100 articles on Internet protocols and security.

[HOME](#)