

Securing Microsoft Services

Mark Burnett 2002-05-22

Securing Microsoft Services

by Mark Burnett

last updated May 22, 2002

Shut off unnecessary services. It is sound advice that is preached in just about every security book, checklist, or training class. But all too often the advice ends there, leaving systems administrators to wonder what exactly is an unnecessary service and how best to shut it off. Sure, it's easy enough to click on "Administrative tools" then "Services" to view the available services. And it's easy to double-click a service you do not use and set the "Startup Type" to disabled. But is there more to securing services than just that?

A service is an application that runs in the background, independent of any user session. Because services run unattended at startup, they are well suited for server-type applications such as a Web server. But this also has its drawbacks, because a user may not be aware that a service is running. Without any user interaction, one could be running a number of default services and never be aware of the potential security risks. This was made all-too-clear a while back as worms with name like "Code Red" and "Nimda" spread across the Internet, often exploiting users who were unknowingly running Web services on their workstations. In turn, these infected workstations spread the worm to thousands of other systems across the Internet.

To master Windows security, you must master Windows services. You must understand how services work, how they are exploited and how services are secured. This article will give you the how-tos of Windows services and point you in some further directions for learning more. This article will not tell you which services to start and which to shut off, but it will give you the information you need to determine what is best for your servers.

How are Services Exploited?

Windows services are exploited by manipulating the service to run a command or access the file system to read or write a protected file. Since most services are run in the security context of the SYSTEM account, they usually have privileged access to most system functions. This makes them particularly interesting to attackers. By manipulating a service, an attacker can escalate his own privileges to do just about anything he wants to do. For example, [Microsoft Security Bulletin MS02-006](#) addresses a buffer overflow in the SNMP service that would allow an attacker to remotely execute commands with the permissions of the SYSTEM account.

Other exploits are less serious but may still use flaws in a service to allow other unauthorized actions. For example, there have been flaws in the SMTP service could allow a spammer to disguise their identity by relaying e-mail through your server

The trick for the attacker is getting access to the service. For most Internet services, this is simply a matter of connecting to the assigned TCP port. For other services, one must have local console access to be able to do any serious exploiting. To protect a service, you must be aware of the exploits and minimize its exposure to those exploits.

How to Minimize Service Risks

The most obvious way to minimize service risks is to shut off a service. Of course, some services are required; in which case, it is critical to keep up-to-date with service packs and hotfixes. Once a serious risk is known, Microsoft generally releases a patch to address that vulnerability. The large majority of attacks are attempts to exploit known vulnerabilities. By keeping patched, you can greatly reduce your exposure.

Another way to minimize service exposure is to not run it from the local SYSTEM account. You can create a new, unprivileged account to run the service, thereby eliminating the all-powerful SYSTEM access. But doing this has its drawbacks. For example, you must do some research to determine the minimal privileges the new account should have. Obviously, with nearly 100 Windows services, creating a unique account for each one is not an easy task. Windows XP and .Net Server address this problem by introducing two new service accounts: LocalService and NetworkService. These accounts run with lesser privilege and, therefore, limit a service's power. Windows 2000 users may be better off sticking with the SYSTEM account unless security is a higher priority than the time required for research.

Some services, while not flawed, expose a server to undue risk by opening doors for other types of attacks. For example, in the wrong hands, the SNMP service could provide a wealth of information. If you must use the service, then control access to it by blocking the appropriate ports at your firewall. Once again, the best protection is to shut down unnecessary services.

How to Best Manage Services

Before I explain how to eliminate unused services, I want to introduce a tool named `sc.exe`, the Services Controller command-line tool. This tool is available as part of the Windows 2000 Resource Kit, the Platform SDK, and is installed by default with Windows XP. `Sc.exe` gives you much more control over services and has the added benefit of being scriptable.

The Services tool on the Administrative Tools menu has some significant limitations. For example, it can only show the service in one of three states: stopped, paused, or running. In fact, there are actually four additional states that can only be viewed using `sc.exe`: continue pending, pause pending, start pending, and stop pending. If a service hangs in a stop-pending state, the Services administrative tool will show the service as already stopped. To make matters more confusing, the NET START command will show the service as still running. Only the `sc.exe` command will give the service's proper state.

`Sc.exe` is an essential tool for managing services. For more documentation on using `sc.exe`, visit the [Microsoft Technet overview](#).

How to Start Services

Windows services have three startup modes: automatic, manual, and disabled. But it turns out those three descriptions are not as straightforward as they at first seem. Automatic startup means that the service will be loaded at boot time. Manual, however, would be better described as on-demand startup. When a service is set for

Manual startup, it remains dormant until a user starts it or an application requests it via the StartService API call. In other words, a service set for Manual startup will actually start automatically as it is needed. For example, if you run the Clipbook Viewer (Clipbk.exe) to view the local clipboard, the Clipbook service also starts. The Clipbook service also allows the clipboard to be viewed by remote computers, a service you probably do not want running in the background. The Clipbook service depends on the Network DDE service, which in turn depends on the Network DDE DSDM service, all of which are started, if set for automatic or manual startup.

The only way to prevent this from happening is by setting the startup mode of each of these services to "Disabled". But disabled is not really disabled. A user with proper permissions can start a disabled service by simply changing its startup mode to Manual or Automatic.

Sc.exe offers even more control by allowing you to actually delete a service, which I will explain in more detail below.

How to Know Which Services You Need

Knowing which services to disable can be a complicated question. Some services should obviously be turned off, but other services are not so clear. The basic rule of thumb is that if you do not have a specific user for a service, then shut it off. If you think you may have a future use for the service, shut it off until the time comes to use it.

The services you need depend on the role of your server. If you have a Web server, you are obviously going to need IIS running, otherwise shut off IIS. If your server is not part of a domain (as is often recommended for IIS servers), you can safely disable all services related to Windows networking.

For Microsoft's advice on which services are needed for running IIS, see Microsoft support's [List of Services Needed to Run a Secure IIS Computer](#). Note that in this article, Microsoft has listed the License Logging Service as required. While they certainly may want you logging licenses, it is not required for IIS operations. (Also note that the appendix at the end of this article includes a list of services by category.)

How to Really Screw Things Up - Disabling Remote Procedure Call Service

Often, you can determine if you need a service by turning it off and see if anything breaks. But be careful when using this method. One service that should never be turned off is the Remote Procedure Call (RPC) service. This service is required by many other services: once you have disabled it, you will not be able to get back into the services administrative tool to start it up again. In some configurations, disabling Remote Procedure Call will prevent the system from booting. In short, do not disable this service.

Another service to watch out for is the Print Spooler service. Although you may not have a printer installed, you need this service running to be able to install service packs. The best advice is to set it for Manual startup.

How to Know What a Service Really Does

Some services have such vague descriptions that it is difficult to determine their actual function. Take the Single Instance Storage Groveler service. The description of this service says that it "Scans Single Instance Storage (SIS)

volumes for duplicate files, and points duplicates files to one data storage point, conserving disk space." What it does not mention, however, is that it is only used with the Remote Installation Service (RIS) and unless you use remote installation, you do not need this service.

To determine what a service really does, try one or more of the following:

- Check the more detailed descriptions at Microsoft TechNet's [Windows 2000 Services](#);
- Check the description in the file properties by right-clicking on a file and selecting the "Version" tab;
- Check the service's file dependencies as described below; and/or,
- Check what ports a service opens by running `netstat -an` at a command prompt before and after starting a service, comparing the results.

How to Know What Files a Service Uses

Knowing a service includes knowing what files a service uses. Often, you can determine if you need a service just by viewing which files it uses. To view file dependencies, I have written a short batch file that uses two Windows Resource Kit utilities, `reg.exe` and `depends.exe`:

```
@set imagepath=
@FOR /F "tokens=3" %%a in ('reg query HKLM\system\currentcontrolset\services\%1
    /v imagepath 2^> nul ^| find "imagepath" ') DO @set imagepath=%%a
@if defined imagepath (
@echo Dependencies for %imagepath%:
@call depends /a0f1c /oc:~svcdep.tmp "%imagepath%"
@FOR /F "tokens=1 delims=, skip=1" %%b in ('type ~svcdep.tmp ^| findstr /B /c:",'
^|
    findstr /V /c:"?" ^| sort') do @echo %%b
@del ~svcdep.tmp 2>nul
) else (
@Echo '%1' is not installed or is not a valid service
)
```

Save this file as `svcdep.bat` and start it with the short name of a service to get a list of files the service uses. For example, to view the dependencies for the File Replication service, type: `svcdep.bat ntfrs`.

You will see the following output:

```
Dependencies for %SystemRoot%\system32\ntfrs.exe:
"c:\winnt\system32\DBGHELP.DLL"
"c:\winnt\system32\DNSAPI.DLL"
"c:\winnt\system32\ESENT.DLL"
"c:\winnt\system32\GDI32.DLL"
"c:\winnt\system32\KERNEL32.DLL"
"c:\winnt\system32\MSVCRT.DLL"
"c:\winnt\system32\NETAPI32.DLL"
"c:\winnt\system32\NETRAP.DLL"
"c:\winnt\system32\NTDLL.DLL"
"c:\winnt\system32\NTDSAPI.DLL"
```

```
"c:\winnt\system32\NTFRS.EXE"
"c:\winnt\system32\RPCRT4.DLL"
"c:\winnt\system32\SAMLIB.DLL"
"c:\winnt\system32\SECUR32.DLL"
"c:\winnt\system32\USER32.DLL"
"c:\winnt\system32\WLDAP32.DLL"
"c:\winnt\system32\WS2_32.DLL"
"c:\winnt\system32\WS2HELP.DLL"
"c:\winnt\system32\WSOCK32.DLL"
```

If you scan this list, you will see that it uses DNS (DNSAPI.DLL), Winsock (WS*.DLL), and Remote Procedure Call (RPCRT4.DLL) libraries. This will not only give you a clue as to what types of actions a service performs, it also helps you see what other services it depends on.

How to Really Disable a Service

As I mentioned earlier, setting a service to disabled only prevents a service from running; it does not remove it from the system. However, `sc.exe` allows you to take that one step further by actually deleting a service from the registry. By typing

```
sc.exe delete <service name>
```

you can remove all registry entries related to the service so that it no longer shows up in the Services administrative tool. This is a good idea when you are sure that you will not use a service, but be careful doing this because `sc.exe` has no option to put a service back once it is deleted.

Some servers are sensitive enough that you may want to take this one step further by actually removing the service files from your system. For this, you must first list the service file dependencies using the batch file listed above. Next, you need to determine which service files are unique for the service and which files are shared Windows libraries. Some are obvious, but others are not so clear. You may want to make backups before taking this step just in case you delete a critical file.

If you look at the file dependencies for the FAX service, you will get the following list of files:

```
"c:\winnt\system32\COMCTL32.DLL"
"c:\winnt\system32\FAXEVENT.DLL"
"c:\winnt\system32\FAXSVC.EXE"
"c:\winnt\system32\FAXTIFF.DLL"
"c:\winnt\system32\GDI32.DLL"
"c:\winnt\system32\KERNEL32.DLL"
"c:\winnt\system32\MSVCRT.DLL"
"c:\winnt\system32\NTDLL.DLL"
"c:\winnt\system32\RPCRT4.DLL"
"c:\winnt\system32\SHELL32.DLL"
"c:\winnt\system32\SHLWAPI.DLL"
"c:\winnt\system32\USER32.DLL"
```

If you look at the list, you will see the three files: `faxevent.dll`, `faxsvc.exe` and `faxtiff.dll`. Since those three files are

obviously dedicated to fax functions, you should be pretty safe deleting them. You do not even have to delete all files for this step to be effective, just removing a few key files render a service unusable. Note that this step is only recommended for servers for which security is critical and for those who are brave enough to do some experimentation. You certainly do not want to try this on a production server.

How to Enforce Service Startup Modes

Once you have determined which services you want running and which you want disabled, it is important to keep them that way. But setting a service with a startup mode of "Disabled" does not mean it will always stay disabled. But services do not have any way of locking in their startup mode. Furthermore, the Services administrative tool offers no way to set a service's permissions to determine who can and who cannot start a service.

Domain security policies do allow you to enforce service startup modes and set service permissions; however, this is not available for stand-alone servers, such as Web servers. In these cases, you must rely on security templates. Use the Security Templates MMC snap-in to configure the startup modes and security settings for each service. Next, use the Security Configuration Editor (or secedit.exe) to apply the template.

Conclusion

Mastery of Windows services is a skill that not enough administrators possess. Your server security can be greatly enhanced if you understand services and how to control them. Take the time to be familiar with Windows services and develop a services policy for your Windows-based servers and workstations. The time would be well spent and the payoff is stronger security across your network.

Appendix: Services by Category

Clustering and Load Balancing	Remote Access
Distributed Transaction Coordinator Intersite Messaging	Internet Authentication Service Remote Access Auto Connection Manager Remote Access Connection Manager Routing and Remote Access
Collaboration	Remote Administration
ClipBook NetMeeting Remote Desktop Sharing	Remote Registry ServiceTe Telnet Terminal Services Terminal Services Licensing
Communications	Remote Installation
Fax Service Telephony	Boot Information Negotiation Layer Single Instance Storage Groveler Trivial FTP Daemon
Disk and File Management	Removable and Remote Storage

Distributed File System Distributed Link Tracking Client Distributed Link Tracking Server File Replication Indexing Service Logical Disk Manager Logical Disk Manager Administrative Service	Remote Storage Engine Remote Storage File Remote Storage Media Remote Storage Notification Removable Storage
Event Monitoring, Logging, and Alerting	System Administration
Alerter COM+ Event System Event Log Performance Logs and Alerts SNMP Trap Service System Event Notification	Application Management License Logging Service RunAs Service Task Scheduler Windows Installer Windows Management Instrumentation Windows Management Instrumentation Driver Extensions Windows Time
Hardware	System Services
Plug and Play Smart Card Smart Card Helper Uninterruptible Power Supply	Protected Storage Remote Procedure Call (RPC) Security Accounts Manager
Internet Clients	TCP/IP Networking
DHCP Client DNS Client	Internet Connection Sharing QoS Admission Control (RSVP) TCP/IP NetBIOS Helper Service
Internet Server Services	Windows Networking
DNS Server FTP Publishing Service IIS Admin Service Network News Transport Protocol (NNTP) Simple Mail Transport Protocol (SMTP) Simple TCP/IP Services Site Server ILS Service SNMP Service TCP/IP Print Server World Wide Web Publishing Service	Computer Browser IPSEC Policy Agent Kerberos Key Distribution Center Messenger Net Logon Network Connections Network DDE Network DDE DSDM NT LM Security Support Provider Remote Procedure Call (RPC) Locator Server Windows Internet Name Service (WINS) Workstation
Media Services	Other
On-line Presentation Broadcast Windows Media Monitor Service Windows Media Program Service Windows Media Station Service Windows Media Unicast Service	Print Spooler Utility Manager
Other OS Support	

File Server for Macintosh
Print Server for Macintosh

[Privacy Statement](#)

Copyright 2006, SecurityFocus