

Windows Server 2003 Security Management

Volume 1, Issue 1

June 2004

In this issue...

Best practices:
Windows Server 20034

Ramp up DNS security
with these three steps8

Lock IT down: Secure Windows Server 2003 Active Directory

By *Brien M. Posey, MCSE*

If I were to tell you that Windows NT Server 4.0 was a lot more secure than Windows 2000 Server, you would probably think that I had lost my mind. Sometimes, though, truth is stranger than fiction. In some ways, Windows NT Server was more secure than Windows 2000 Server. However, Microsoft learned from their mistakes and implemented a Windows NT-like security structure into Windows Server 2003's Active Directory. Let's discuss these security issues and learn some tips you can use to build a secure Active Directory (AD) environment.

Physical security is job #1

When attempting to secure AD, it's critical that you implement physical security first. If anyone you wouldn't trust with the Administrative password has physical access to a domain controller or to your DNS servers, you don't have a secure AD. Many administrative and disaster recovery tools exist that can easily double as hacker tools.

Given physical access to the server, it is easily possible for someone with minimal computer knowledge to hack the server in a matter of minutes. So don't even bother trying to secure AD until you've made sure that all of your servers are placed in a secure location.

Windows NT vs. Windows 2000

Don't get me wrong. In many areas, Windows 2000's security is far superior to that offered by Windows NT. However, there is a basic law of computing that states that the more complex a piece of software is, the greater the chance that it will contain a security hole or a major bug that can be exploited. As we all know, Windows 2000 is a lot more complex than Windows NT.

Perhaps the best example of simplicity and security going hand-in-hand involves the domain model implemented by each server operating system. In Windows NT, the domain was pretty much the only organizational structure that existed. A domain often contained all of the users, groups, and computers for an entire organization. If an organization was really big, they could create multiple domains and have the domains trust each other; but each domain was an independent structure.

When Microsoft created Windows 2000, they realized that the Windows NT domain model just didn't scale well into larger organizations. So, they based the AD on a structure called a forest. A forest is basically a collection of domain trees. Within a forest, you can have many different domains and can even use parent and child domain trees. Just as was the case with Windows NT, each domain has its own Administrator. This is where the similarities end, though.

In Windows 2000, Microsoft decided they needed to make the domains more manageable. They created different levels of domain administration. For example, a member of the Domain Admins group could typically administer the current domain and any child domains beneath it. A member of the Enterprise Admins group had the ability to administer any domain within the entire forest. Herein lies the problem.

The fatal flaw in the Windows 2000 AD model is that every domain completely trusts every other domain within the forest. This causes a couple of problems. First, if security has not been applied properly, administrators can just add their accounts to the Enterprise Admins

Attention!

Subscribe to our FREE *Windows Server 2003 Security Management* newsletter, and you'll get an e-mail notification when we publish each quarterly edition. No more surfing to find it on the site. No more reminders on your calendar that it's time to look for it. You won't miss an issue. [Just instantly sign up now!](#)

If you would like to automatically sign up for weekly *general Windows 2000 Server tips*, [click here.](#)

group to gain control over the entire forest. If the domain is a bit more secure, rogue administrators need only to tamper with the SID history and launch an elevation of privileges attack against the forest. By manipulating the SID history, administrators could give themselves Enterprise Admins status.

There are other inherent weaknesses in the Windows 2000 AD security model as well. As you probably know, each domain requires at least one domain controller. Likewise, each domain controller contains information relating not only to the domain, but also to the forest. Such information includes AD's schema and some basic configuration.

Now, imagine you had an administrator who wasn't being intentionally malicious, but who installed a malicious application or incorrectly modified an AD. If the change that the administrator made was to a forest-level AD component, the change would eventually be propagated to every domain controller in the entire forest, thus corrupting every single copy of AD and potentially crashing the entire network.

Let's compare this situation to Windows NT. Even if one domain trusts another domain, both domains include a copy of the Security Accounts Manager pertaining to their own domain only. In this way, rogue administrators can't make a change to the SAM in their domain and then use that change to corrupt other domains. Likewise, there is no all-powerful group within Windows NT that a rogue administrator could use to gain control over every domain in the entire organization.

Another nice thing about the way that Windows NT's trust relationships worked was that trust relationships could either be one-way or two-way,

and they were never transitive in nature. This meant that if you had a Users domain and an Admin domain, you could either allow both domains to trust each other or you could configure the network so that the Users domain trusted the Admin domain, but not vice versa. It also meant that if Domain A trusted domain B and domain B trusted domain C, then domain A didn't trust domain C unless you told it to.

Windows Server 2003 security

You're probably wondering what all of this has to do with Windows Server 2003. I went into the long comparison between Windows NT and Windows 2000 because in Windows Server 2003, Microsoft incorporated the best of both worlds. And so, to properly secure your Windows Server 2003 network, you need to understand the strengths and weaknesses of both security models.

The biggest AD security weakness in Windows 2000 is that all domains within a forest are linked together by a common administrative structure, the forest itself. In Windows Server 2003, the forest structure still exists and works almost identically to the way it did in Windows 2000.


What is different between the forest structure in Windows Server 2003 and that of Windows 2000 Server is that Windows Server 2003 makes it relatively easy to establish trust relationships between forests. Interforest trusts were possible in Windows 2000; but, in Windows Server 2003, interforest trusts are actually useful. When a

trust relationship exists between forests, administrators can grant access to a resource in a user from a foreign forest in the same manner that they would if the user existed within the local forest.

Single forest vs. multiple forests

A single forest environment is ideal for most small to medium-sized companies. Single forest environments are easy to manage. But larger companies often need each office or each department to be able to have full administrative capabilities over its own users and computers. In such environments, there is often a high degree of distrust between these various groups. In a situation like this, interconnected forests are ideal because they give each group total autonomy.

At the same time, even though the administrative burden is distributed, such a model usually has a much higher administrative burden than a single forest environment, which results in higher administrative costs to the company as a whole. My point is that, in a Windows Server 2003 AD environment, there is a trade-off between cost and security.



IT Security Survival Guide
Second Edition

- Safeguard against the threats associated with today's collaborative workspace
- Evaluate your system's security risks
- Identify e-info vulnerabilities

plus Bonus CD

Buy Now

Interforest trusts

Let's discuss the specifics behind using multiple forests as a mechanism for securing your organization's AD. First, each forest has its own AD; there is no common thread of any kind tying the forests together. So, it's possible to configure each forest to use a common DNS server. Assuming that the DNS server and backup DNS server are managed by someone trustworthy, DNS server consolidation is a great way to reduce cost and lessen the administrative burden. On the flip side, sharing a common DNS server can also be a single point of failure for the network if no backup DNS server is used.

There are some prerequisites you must meet before you can establish a trust relationship between forests in Windows Server 2003. Perhaps the most difficult of these is that any forest involved in the trust must be running at Windows Server 2003 forest functional level. Windows 2000 allowed you to run AD in either mixed or native mode. The functional level in Windows Server 2003 is very similar to this. Setting a forest to Windows Server 2003 forest functional level requires that every domain controller within the forest run Windows Server 2003.

Also, to create an interforest trust, you must be a member of the Enterprise Admins group. You must also have your DNS server configured so that it can resolve the names of domains and servers within the forest with which you're establishing the trust relationship.

Finally, as you may recall from Windows 2000, every forest has a root domain and all other domains fall beneath the root. Windows Server 2003 can create an interforest trust only from the root domain, because interforest trusts are transitive at the

domain level. This means that if you were to establish a trust between Forest A and Forest B, then every domain in Forest A will trust every domain in Forest B, and vice versa. Forest trusts are not transitive at the forest level, though.

For example, if Forest A trusts Forest B, and Forest B trusts Forest C, Forest A will not trust Forest C unless you tell it to do so. As you can see, the transitive nature of interforest trusts makes them fairly powerful. If your forest has multiple domains, you don't want an administrator of some lower-level domain creating an interforest trust without your knowledge or consent. That would cause huge security problems. This is why you can create an interforest trust only at the forest root level.

Another interesting thing about creating trusts with Windows Server 2003 is that you don't necessarily have to create a full interforest trust. Suppose your business needs to establish a trust relationship with a supplier. You probably need to establish a trust relationship with only one of the supplier's domains. You probably aren't interested in the supplier's human resources or marketing domains. In such a case, you can create what's called an external trust.

An external trust is a trust relationship between domains, similar to the trust relationships that existed in Windows NT. An external trust can be established from any domain within your forest and links to a domain in a foreign forest. Aside from being able to establish the external trust at any domain level, there are other critical differences between an external trust and an interforest trust.

Unlike an interforest trust, an external trust is completely nontransitive, which means the trust applies only to the domains that the trust is assigned to. Other domains within the two

forests don't acknowledge the trust relationship.

Whether you are forming an interforest trust or an external trust, you have the option of creating a two-way trust, a one-way incoming trust, or a one-way outgoing trust. A two-way trust simply means that both domains trust each other. A one-way incoming trust means that users in the current domain or forest can be authenticated by the foreign domain or forest. Likewise, a one-way outgoing trust means that users in the foreign forest or domain can be authenticated by the local domain or forest.

Cross-forest authentication

Windows Server 2003 interforest trusts support cross-forest authentications. Suppose a user who normally logged into Forest A made a business trip to the company hosting Forest B. With forest authentication, users from Forest A could log into Forest B just as though they were logging into Forest A.

This might seem strange at first, since neither the domain controllers nor the global catalog in Forest B would have any knowledge of a user from Forest A. When the user tries to log in, the computer checks the domain controller and then the global catalog for the user's account. Because the account is not found, the system implements a cross-forest, name-matching function. This function compares the user's credentials with those found within all recognized namespaces (forests). The comparison is made via Kerberos and NTLM, so the process is secure.

Cross-forest authorization

Another feature that's great about Windows Server 2003 is cross-forest authorization. This allows you to

assign permissions to users within both the local forest and trusted forests directly through an Access Control List (ACL). This comes in handy for both granting and denying permissions.

Suppose you were an administrator for your company's research and development department, and that your job was to keep all of the files on your server confidential. The forest-level administrator for your company didn't know what he was doing, and he created an interforest trust with a competitor. If you wanted to keep users at the competitor's firm from being able to access your data, you

could give those users an explicit deny at the root level of each of the servers in your domain.

As nice as this capability sounds, though, there is a catch. You must completely type in the names of users or groups from trusted forests. Enumeration and wildcards aren't supported. This means that you can't just implement a blanket policy that says "don't let anyone from that other forest access any of my data." You could, however, get the names of each of the domains belonging to the other forest and deny access to the Everyone group belonging to each of those domains.

The best of both worlds

Even though Windows 2000 is newer than Windows NT, some of the improvements actually decreased security in your organization. Windows Server 2003 gives you added flexibility to restore that security. One way to achieve effective security within an organization is to implement multiple forests and create trust relationships between them. However, this isn't a process to be taken lightly, because there are many prerequisites and the process tends to increase costs and the administrative burden. ❖

From the TechProGuild subscription product. [Subscribe today.](#)

Best practices: Windows Server 2003

By Brien M. Posey, MCSE

If you've ever deployed Windows NT Server or Windows 2000 Server, you probably know that Microsoft designed those products to be nonsecure by default. Although Microsoft has provided many security mechanisms, it's been up to you to implement them. But when Microsoft released Windows Server 2003, the company switched philosophies. The new philosophy is that the server should be secure by default.

This is generally a good idea, but Microsoft didn't take it quite far enough. While a default Windows 2003 installation is certainly more secure than a default Windows NT or Windows 2000 installation, it is still anything but totally secure. Let's discuss some relatively easy measures that you can take to make Windows Server 2003 even more secure.

Know your role

Understanding the server's role (i.e., intended purpose) is absolutely critical to the security process. There are many roles for which a Windows Server can be configured. For example, a Windows 2003 server can act as a domain controller, a member server, an infrastructure server, a file server, a print server, an IIS server, an IAS server, a terminal server, and the list goes on. A server can even be configured to fill a combination of roles.

The problem with this is that each server role has its own security needs. For example, if your server is going to function as an IIS server, you need to enable the IIS services. However, if the server is going to function solely as a file and print server, enabling IIS would be a huge security risk.

The reason I'm telling you this is to point out that there is no way that I can just give you a set of steps to follow and expect those steps to work in every situation. A server's security needs vary tremendously by the server's role and by the server's environment.

Because there are many ways to harden a server, I'll discuss the steps necessary for configuring a server to act as a simple, but secure, file server. I'll try to point out some things that you might do differently if the server is filling an alternate role. Just please understand that this isn't intended as a comprehensive guide to securing every type of server.

Physical security

To achieve true security, your server must be in a secure location. Normally, this means placing the server behind a

Help us help you

We'd like to know what topics you'd like to read about in the next issue of the *Windows Server 2003 Security Management* newsletter. Send us some mail at itmanager@techrepublic.com and tell us which of the following topics you'd like to read about. Also please tell us if there's a topic we haven't listed that you're interested in.

- How to work with Windows Server 2003's IP Security Monitor
- How to configure wireless security in Windows Server 2003
- Solutions to Windows Server 2003 group policy problems
- How to control access to shared resources in Windows Server 2003
- How to administer Windows Server 2003 PKI Services
- A primer of Windows Server 2003 security features

Help from your peers

If you don't find the answer to your Windows Server 2003 question in these pages, you may find it in our Discussion and Technical Q&A areas in the words of your peers. Here are a couple of places to get you started:

<http://techrepublic.com.com/5208-6230-0.html?forumID=3&threadID=137214&start=0>

<http://techrepublic.com.com/5208-6286-0.html?forumID=11&threadID=138528&start=0>

<http://techrepublic.com.com/5208-6286-0.html?forumID=11&threadID=133777&start=0>

locked door. Physical security is extremely important because many administrative and disaster recovery tools exist that can double as hacker tools. Anyone with such tools and a minimal skill level can hack a server in a matter of minutes once they have physical access to the machine. Your only hope against preventing such attacks is to place the server in a secure area. This is true of any Windows 2003 server, regardless of its role.

Creating a baseline

Aside from establishing good physical security, the best advice that I can give you when deploying a series of Windows 2003 servers is to decide on your security requirements prior to deployment and to enforce those policies immediately after deployment.

The best way to do this is to create a security baseline. A security baseline is a list of documented and accepted security settings. In most cases, your baseline settings will differ considerably depending on the server's role.

So the best thing to do is to create several different baselines that you can apply to various types of servers. For example, you might have one baseline for file servers, another for domain controllers, and still another for IAS servers.

Windows 2003 contains a tool called the Security Configuration And Analysis Tool. This tool allows you to compare a server's current security policy against a baseline security policy contained within a template file. You can either create these templates yourself or use one of the included template files.

The security templates are a series of text-based INF files stored in the %SYSTEMROOT%\SECURITY\TEMPLATES folder. The easiest way to examine or modify the individual templates is through the Microsoft Management Console (MMC).

To open the console, enter the MMC command at the Run prompt. When the empty console loads, select the Add/Remove Snap-in command from the File menu. This will cause

Windows to display the Add/Remove Snap-in properties sheet. Click the Add button found on the properties sheet's Standalone tab, and you will see a list of all of the available console snap-ins. Select the Security Templates snap-in from the list, and then click the Add, Close, and OK buttons.

Once the Security Templates snap-in is loaded, you can view each of the templates. As you navigate through the console tree, you will see that each template mimics the group policy structure. The template names reflect each template's purpose. For example, the HISECDC template is a high-security domain controller template.

If you're trying to secure a file server, I recommend starting with the SECUREWS template. As you look through all of the template's settings, you will find that the template can be used to make the server more secure than it currently is, but it may not meet your needs. Certain security settings may be too strict or too relaxed.

I would recommend either modifying the existing settings to meet your needs or creating a brand new policy. You can easily create a new template by right-clicking on the C:\WINDOWS\Security\Templates folder within the console and selecting the New Template command from the resulting menu.

Once you have created a security template that meets your needs, go back to the Add/Remove Snap-in properties sheet and add a snap-in called Security Configuration And Analysis. When the snap-in loads, right-click on the Security Configuration And Analysis container, then select the Open Database command from the resulting menu. Since no database currently exists, make up a name for the security database. Click Open, and the necessary database will be created using the name that you provided.

Next, right-click on the Security Configuration And Analysis container and select the Import Template command from the shortcut menu. You'll see a list of all of the available templates. Select the template that contains your security policy settings and click Open. After the template has been imported, right-click on the Security Configuration And Analysis container once again and select the Analyze Computer Now command from the shortcut menu. Windows will prompt you for a location to write the error log. Enter a file path and click OK.

At this point, Windows will compare your server's existing security settings against those in the template file. You can see the results of the comparison by navigating through the Security Configuration And Analysis console. Each group policy setting displays both the current setting and the template setting.

Once you've had a chance to look through the list of discrepancies, it's time to enforce the security policy based on the template. To do so, right-click on the Security Configuration And Analysis container one last time and select the Configure Computer Now command from the shortcut menu. The tool will then modify your computer's security policy to match the template policy.

Group policies are hierarchical in nature. A group policy may be applied at the local computer level, the site level, the domain level, or the OU level. When you implement security based on a template, you're modifying the computer-level group policy. Other group policies aren't directly affected, although the final policy may reflect a change due to a setting in the computer policy being inherited by higher-level policies.

Modifying built-in accounts

For years, Microsoft has been preaching that you need to rename the Administrator account and disable the Guest account to achieve good security. In Windows Server 2003, the Guest account is disabled by default, but renaming the Administrator account is still a good idea because it's common for attackers to try to compromise the Administrator account.

There are a number of hacker tools that reveal the Administrator account's real name by examining the account's SID. Unfortunately, you can't change this account's SID, and there is really no way of preventing such a tool from determining the Administrator account's real name. Even so, I encourage everyone to rename the Administrator account and to change the account's description for two reasons.

First, less sophisticated hackers may not know of the existence of

such tools or have access to them. Second, renaming the Administrator account to a unique name makes it easy for you to monitor attacks against the account.

Another tip pertains to member servers. Member servers have their own built-in local administrative account that is completely separate from the domain Administrator account. You can configure every member server to use a different administrator account name and password. The idea is that if someone were to figure out the local administrator account name and password on one member server, you wouldn't want them to be able to use those credentials to hack your other servers too. Of course, if you have good physical security in place, no one should be able to gain access to a server to be able to use a local account.

Service accounts

Windows Server 2003 is designed in a way that minimizes the need for service accounts. Even so, some third-party applications absolutely insist on a traditional service account. If possible, always use a local account as the service account instead of using a domain account, because if someone were to gain physical access to the server, they could dump the server's LSA secrets, and compromise the password. If you use a domain password, the password can be used from any computer within the forest to gain access to the domain. If a local account is used, though, the password is useless from anywhere other than the compromised machine and doesn't provide any access to the domain.

System services

There is a fundamental law of computing that states that the more code

running on a system, the greater the chance that the code will contain a security vulnerability. One of the primary security strategies that you should focus on is to reduce the amount of code running on your server. Doing so reduces security risks and will also improve the server's performance.

In Windows 2000, there were a lot of services that were running by default, but were totally unnecessary in most environments. In fact, a default installation of Windows 2000 even included a fully operational IIS server. In Windows Server 2003, Microsoft turned off most of the services that aren't absolutely necessary. Even so, there are some services that are running by default, but are open to debate.

One such service is the Distributed File System (DFS) service. The DFS service was primarily designed to make a user's life easier. DFS allows an administrator to create a logical namespace containing resources from multiple servers or partitions. To a user, all of these distributed resources appear to exist within a single folder.

I personally like DFS, especially because of its fault tolerance and scalability features. However, if you did not use DFS, you would force

users to know the actual path to a specific resource instead of being able to access all resources through a single path. In some environments, this may translate to better security. In my opinion, though, the rewards of DFS far outweigh the risks.

Another such service is the File Replication Service (FRS). The FRS is used to replicate data between servers. This is a mandatory service on domain controllers because it's responsible for keeping the SYSVOL folder synchronized. For member servers, however, this service isn't mandatory unless you are running DFS.

If you have a file server that isn't a domain controller and isn't using DFS, I recommend disabling the FRS. Disabling the FRS decreases an attacker's ability to replicate a malicious file across multiple servers. The FRS is enabled by default.

Another service worth taking a look at is the Print Spooler service. The Print Spooler manages all local and network print queues and controls all of the print jobs within these queues. The Print Spooler is required for all printing operations, and is enabled by default.

The flip side to this is that not every server requires printing capabilities. Unless a server is acting as a print

server, you should disable the print spooler. After all, why should a dedicated file server run the print spooler? Normally, no one should be sitting at the server console working, so there should be no need to print locally or from across the network.

I realize that often during disaster recovery operations, it might become necessary to print an error message or an event log. However, I recommend simply turning on the Print Spooler service when it is needed rather than leaving it on all the time for non-print servers.

Believe it or not, the Print Spooler is one of the most heavily exploited Windows components. There are countless Trojans that work by replacing the Print Spooler's executable file. The reason for such an attack is that the Print Spooler operates as a system-level service and, therefore, has a high level of privileges. So any Trojan posing as the Print Spooler can also gain these high-level privileges. To protect your server from such an attack, just prevent the Print Spooler service from running. ❖

Ramp up DNS security with these three steps

Michael Mullins, CCNA, MCP

A few months ago, I explained how to improve DNS security in "[Strengthen vulnerable spots to improve DNS security](#)." This solution highlighted the most common problems of current DNS implementations.

Afterward, I heard from several readers, who asked for more in-depth information to help secure these valuable network assets. Let's look at three common problems and solutions. I'll tell you how to make the recommended changes in both Windows and UNIX.

Stop cache poisoning

Cache poisoning occurs when a name server makes a recursive query and caches bogus data for a domain name. This can result in denial of service (DoS) or man-in-the-middle attacks. However, you can eliminate this vulnerability.

In Windows 2000 or Windows Server 2003, follow these steps:

1. Go to Start | Control Panel.
2. Click Performance And Maintenance, and click Administrative Tools.
3. Double-click DNS.
4. In the console tree, select the applicable DNS server.
5. Go to Action | Properties.
6. On the Advanced tab, select the Secure Cache Against Pollution check box in the Server Options section, and click OK.

In UNIX flavors of BIND, edit the named.conf file, and make the following changes:

```
acl internal {
xxx.xxx.xxx.0/xx; }; ! Your
network block
options {
recursion no;
allow-query { internal; };
...};
```

Disable recursive queries

External name servers should run in a passive mode. They should never send queries on behalf of other name servers or resolvers.

By default, your Windows DNS server performs recursive queries. Recursion is a DoS attack tool used by crackers to shut down a name server and make a site inaccessible to outside users.

You should definitely disable recursion. In Windows, issue the following command at a command prompt:

```
dnscmd <ServerName> /Config
/NoRecursion 1
```

In UNIX flavors of BIND, implementing security against cache poisoning (as demonstrated in the previous section) also turns off recursion.

Use a single interface

By default, DNS listens and responds on the appropriate ports on all configured interfaces. If your server is multihomed, then you have a potential security breach on multiple IP addresses.

In addition, this increases the complexity of your access control lists on your routers and switches. However, you can configure your DNS server to listen on only one IP address.

In Windows 2000 or Windows Server 2003, follow these steps:

1. Open DNS.
2. In the console tree, select the applicable DNS server.
3. Go to Action | Properties.
4. On the Interfaces tab, select Only The Following IP Addresses.
5. In the IP Address text box, enter an IP address for the DNS server you want to enable for use, and click Add.

In UNIX flavors of BIND, you can't natively control which ports are open on a multihomed interface. If the "named" service is running, all IP addresses will listen for traffic.

Final thoughts

After implementing these changes to your name server configuration, verify that you allow only TCP/UDP port 53 traffic to and from your server. This step completes the basic lockdown of your DNS servers.

As I've mentioned before, these servers are vital to the healthy functioning of your network. You must actively monitor them and keep them patched and up to date. ❖