Global Information Assurance Certification Paper

Securing Windows Service Accounts

Gerald Rice
03/03/2005
GSEC Practical v 1.4c, Option 1

Table of Contents

Introduction

Quite often a Windows network is installed out of the box with little attention paid to security from a service account perspective. This is evident when you see a Windows network that has a single service account with a password that never changes for the multitude of services running on the servers. Another example is when Microsoft SQL Server 7.0 is installed and later upgraded to SQL Server 2000 without downgrading the account the SQL service runs under to a non-administrative account. This paper will look at some strategies for tightening up security around your Windows service accounts, as well as the special group of administrative accounts you use to work on your servers. There is a lot of confusion around the built-in accounts that exist for servers, so we will take a closer look at the accounts themselves. While the principles shown here apply to both workstations and servers, this discussion will primarily center on the server environment. We will look at both large production environments, and smaller ones. A large environment will typically have many administrators and many service accounts, which provide for a complex environment to manage from a security perspective. A smaller environment has its own set of issues to wrestle with, since resources are often not as plentiful.

Assumed Knowledge

This paper assumes that you understand how to implement and change accounts, passwords, and services. You will also need to have a basic understanding of domains, trusts, authorization, and authentication across Windows networks. It is also assumed that your servers have been hardened using best practices for Windows security, e.g. you are using NTFS, and you have secured the file system as suggested by Microsoft.[1]

The Definition of a Service Account

What is a service account anyway? In basic terms, a service account is an account that a service on your computer uses to run under and access resources. This should not be a user's personal account. While they may look the same, the separation of users from services is very important for both tracking and the ability to tighten down what an account can and cannot do. A service account could also be an account that is used for a scheduled task (sometimes referred to as a batch job account), or an account that is used in a script that is run outside of a specific user's context. A scheduled task account should not be a personal user's account for the same reasons that a service should not run under a personal user's account.

[1] "Windows 2000 Security Hardening Guide." Microsoft. 11 Apr. 2003. 20 Dec. 2004 <http://www.microsoft.com/technet/security/prodtech/win2000/win2khg/05sconfg.mspx#EGAA>

Hackers Love Service Accounts

Hackers will often target service accounts because they are often implemented in such a way that they have a higher level of access than a user account. They have also historically not changed passwords as often (if ever) as user accounts. Services are often installed under the built-in Local System account, which gives what are essentially local administrator privileges, so they are more predictable in how they will be able to be used if compromised. While local administrator privileges may seem somewhat innocuous since they are not usually useable on other computers on your network, the local administrator privileges can end up granting access to domain username/password combinations and or lead to account changes that allow for easier connections to other parts of your network. As a result, both locking down a service account and following good password change and audit procedures is an important part of keeping your systems secure.

Domains and Service Accounts

Most domains are set up in a flat structure, especially those in smaller production networks. This means you've got one domain and a whole bunch of users inside that domain. In larger enterprise type environments, you will often see more than one domain. This is a good way to help define what and where the service accounts go. The concept of a resource domain is outside our scope here, but splitting your flat domain into two domains, where your resources (computers, printers, service accounts, and maybe even administrator accounts) are in their own domain makes it much easier to block unwanted accounts from having any access into your production server environment. This will also make organizational and policy decisions more clear cut, since you have a clear delineation between where user accounts and service accounts reside. If you have a production domain that only has your servers in it, and the only accounts in it are either service accounts or administrator accounts, then you can ensure that all administration level access to your servers come from one set of accounts. At that point, you can also be sure that non-administrative automated access between servers is done through accounts that you are aware of, and have locked down specifically to run as service accounts.

All These Accounts!

There are a few different types of accounts in Windows. Windows 2000, 2003, and NT 4.0 all have a lot of the same types of accounts by default, but 2003 Server has since added a couple of new types of accounts. There is the built-in Local System account, the built-in Local Service Account, the built-in Network Service account, local and domain User Accounts (which would include service accounts), and more. Let's look at the mentioned accounts a bit closer.

The Built-in Local System Account

The built-in Local System account is essentially a local administrator on the computer. In fact, the built-in Local System account is actually in the administrators group (by having the BUILTIN\Administrators SID in its token), thereby gaining all of those privileges, and being given a few more![2] A service running under this account won't typically have administrative access on another server because it is not a domain account, but it will on the local server. When a service runs under the built-in Local System context, it takes on the Computer Accounts permissions when interacting over the network. This is an important fact to understand if you are in the habit of using Computer Accounts in security groups. It also allows for a limited amount of Active Directory access, and when implemented on a Domain Controller, full access to the Active Directory.[3] For this reason, it's particularly important to be sparing about the use of the built-in Local System account on your Domain Controllers. The built-in Local System account does not require an administrator to manage a password since Windows automatically changes the password every 7 days.[4]

The Built-in Local Service Account

The built-in Local Service account is new in Windows Server 2003. Its purpose is to be used to run services on your machine under an account that has much more limited privileges than the built-in Local System account. Unlike the built-in Local System account, this account accesses the network using null (or anonymous) privileges. This means that it will not likely have much access outside of its own computer, unless you have not locked down anonymous (null) session access on other machines. The password for this account is null and as such does not need to ever change. This account will operate with the same permissions and is essentially a part of the local Users group. You will find that brand new Windows 2003 Server installations (as opposed to upgraded installations) will have many services running under this account, rather than the built-in Local System account.[5] This account, like any other account, can be compromised, but its effectiveness for use by a hacker is more limited by its restrictions.

---

[2] "Platform SDK: DLLs, Processes, and Threads – LocalSystem Account." Microsoft. Dec. 2004. 05 Jan. 2005 <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dllproc/base/localsystem_account.asp>

[3] "Platform SDK: Active Directory – The LocalSystem Account." Microsoft. Oct. 2004. 05 Jan. 2005 <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ad/ad/the_localsystem_account.asp>

[4] Tony Redmond. "Running Exchange Services from LocalSystem." Windows IT Pro. Feb. 2001. 14 Nov. 2004 <http://www.windowsitpro.com/MicrosoftExchangeOutlook/Article/ArticleID/16371/16371.html>

[5] Sean Deuby, "Lessons for an AD to Windows 2003 Upgrade," Windows IT Pro Magazine January 2005: Page 44

The Built-in Network Service Account

The built-in Network Service account is also new to Windows Server 2003, and operates in much the same way as the built-in Local Service account, but its network access takes on the role of the Computer Account. This means that the service running under the built-in Network Service account will present the Computer Account credentials when trying to do things like access a share on another computer, or query the Active Directory. Like the built-in Local Service account, the password for this account is null and does not change. It is also essentially a member of the Users group, and as such has the same permissions.[6] This account will also be used for many services in a fresh installation of Windows 2003, just like the built-in Local Service account.

Created Service (User) Accounts

A created Local Service account can be defined as to what sort of access it can have on the local server, and if implemented in a specific way, can even automatically give a service access to another server (more on this later). A created Local Service account is really just a local user account. Domain Service Accounts are accounts that are similar to a created local service account. They can more easily access other resources on the domain since they are created at the domain level and are essentially just user accounts on the domain which are created specifically for the use of services. Created Domain Service Accounts do represent a significant threat since they can be used on any machine that needs them.

The Default Service Account

New services installed in Windows are often not secured by the installation application by default. Usually your new service will run under the built-in Local System service account or you will be given a choice of what account to run a service under (most installers take all of this into account.) Do not give a service the local administrator account to run under. It is understandable that it is easier, since there will not be any permissions issues if a service runs under an Administrator account, but if that service is compromised, it has administrative access. A regular user's account is also not appropriate as it makes it auditing much more difficult, as well as giving un-needed access to that users' files and whatever else that user may have been granted access.

Another important issue is the little 'Allow service to interact with desktop' check box. If possible, it should not be enabled. It is a dangerous option, as it

[6] "Microsoft Windows Server 2003 Services Permissions." Microsoft. 12 Jan. 2005 <http://www.microsoft.com/resources/documentation/WindowsServ/2003/standard/proddocs/en-us/Default.asp?url=/resources/documentation/WindowsServ/2003/standard/proddocs/en-us/sys_srv_permissions.asp>

bypasses the built-in security boundary between the desktop and the services. An example of this is a service opening up a command window, which then is susceptible to messages from the desktop (including a simple CTRL-C keystroke) which would allow the user/malicious program to effectively have shell access and run commands under whatever context the service is running under.[7]

What Services Should I Worry About?

The basic idea of locking down a service account is to look at what the service really needs to do. Many services run under the built-in Local System account because they need to interact with the O/S at a low level (including interaction with the Kernel or restricted libraries). As a result, a lot of the services provided by Microsoft are designed to run this way by default. The situations to pay particular attention to are the services that are run in order to provide a service to someone; particularly services provided by third parties or in-house applications which may not have the resources to ensure that their services are secure at all times. You can be relatively assured that Microsoft's Plug and Play service is kept secure through patches and its own authentication mechanisms, but it may be difficult to know whether a web service created by a small 3 person software engineering group has been thoroughly tested for security vulnerabilities when the product was released, never mind a year after the fact. When you install or create a new service, try to figure out exactly what the service needs access to, and grant only what it needs. Restricting a service with an account that only has access to what it needs can also help to contain a vulnerability if one should occur.

Created Local Service Accounts

If you are installing a new service that runs on a single server and does not need to establish a connection out to another server or client somewhere using domain credentials, then the account can be a simple local account. Any accounts created on a local machine do not have to be part of a group, so you can get away with creating a service account that does not have access to almost anything (including interactive log-on). You can then check to make sure it has access to all the folder it needs to (assigning access where needed), and it won't have the ability to access other areas of the O/S it shouldn't be able to (unless some sort of privilege escalation hack is induced). When you assign a local account to a service account, Windows will automatically give it the Log on as a Service privilege. Be aware that local accounts with the exact same name and password on more than one box will be able to access each other's resources with correctly applied permissions due to Windows' passing of the

[7] Michael Howard. "Tackling Two Obscure Security Issues." Microsoft. 14 Aug. 2002. 11 Nov. 2004 <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dncode/html/secure08192002.asp>

username password pair when accessing another server. You can see this in action by creating an account with the same username and password on two different machines, then accessing a remote share from one of the machines using the created account. You should avoid having local accounts on different boxes with the same username and password.

Domain Service Accounts

If you are installing a new service that will need access to another server using windows file sharing, or something similar, this access can be allowed by creating a domain account for the service. This works well in a large production environment, since it is more difficult to maintain many accounts across many servers than it is to maintain them in a central location. An important fact to remember is that shared service accounts can be a major security vulnerability. Domain service accounts will allow one compromised account to be used across your entire domain where said account has permissions.[8] This should be weighed carefully against the administrative costs that will be incurred when using single created local service accounts. New accounts, by default, are placed in the Domain Users group. Microsoft domains require that domain accounts be placed into a group, so the best option is to create a new Domain group that does not have permission to interactively log-on to the domain and is otherwise locked down through your Active Directory Group Policy. You can then give permissions to the file shares or other resources that the service running under the new service account will need. You can also place the domain service accounts in the Domain Guests Group, but default behavior for domains is for the Domain Guests group to be placed in the Local Guests groups on member servers,[9] so placing your new Domain Service Accounts into their own group is usually the most secure way to go. One key issue with any user account being used as a service account is this fact:

> "Non-SYSTEM service account passwords are stored in cleartext in a portion of the Registry called the LSA Secrets, which is accessible only to LocalSystem."[10]

This means that anyone who has local administrator access on the server can access the registry and retrieve the service passwords. If that service account is a domain service account, then they have whatever level of access that service has on all the other servers that exists on in your domain (and trusting domains). In the end, running services under a domain account is not a desirable way to

---

[8] Jesper Johansson. "Hacking: Fight Back – How A Criminal Might Infiltrate Your Network." Microsoft. 06 Dec. 2004. 08 Jan. 2005 <http://www.microsoft.com/technet/technetmag/issues/2005/01/AnatomyofaHack/default.aspx>

[9] "Microsoft Windows Server 2003 Default Groups." Microsoft. 22 Oct. 2004 <http://www.microsoft.com/resources/documentation/WindowsServ/2003/standard/proddocs/en-us/Default.asp?url=/resources/documentation/windowsserv/2003/standard/proddocs/en-us/sag_ADgroups_9builtin_intro.asp>

[10] Joel Scambray and Stuart McClure, Hacking Windows Server 2003 Exposed (California: McGraw-Hill/Osborne, 2003) Page 18.

operate an enterprise environment. If you can move towards using the newer built-in Local Service and built-in Network Service accounts, you will go a long way towards further securing your services.

Examples of Services and Their Accounts

There are some specific services that continually come up when dealing with service accounts. Sometimes all that is needed to better secure these services is a quick trip to the vendor's website, and/or some testing. Microsoft SQL Server is a great example of this, because the service has had its share of security issues, so limiting its access is very desirable.

Microsoft SQL Server 7.0 and earlier required that you use a service account that was a local administrator on the server. When SQL Server 2000 came out, Microsoft also released some documentation on how to run a SQL Server on a non-administrative account. While it does not work in every situation, it does work in most, and can be applied (after testing) fairly easily.[11] Microsoft's IIS platform is also confusing because they have used a multitude of different accounts over time, which tends to 'provide' you with a lot of different accounts on your system. Accounts such as ASPNET, IUSR_Computername, and IWAM_Computername exist in order to help keep internet users isolated from the environment. In IIS 6.0, the IIS service normally needs to run under the built-in Local System service account, but you can run your worker processes under a different account, like the built-in Local Service account or the built-in Network Service account to further restrict those processes to only what they need. See http://www.informit.com/articles/article.asp?p=101750&seqNum=6 for some more information on IIS and its accounts. Backup services are also another problem area since each vendor seems to treat account access totally differently. Other services can be individually analyzed and looked at in the same way, thus allowing you to secure even the most commonplace applications' service accounts while still providing the needed services.

Securing Your New Service Account

If you choose to create a new service account, the service accounts themselves also need to be secured. Since Windows tends to think of every new account created as a user account, you get user-type privileges applied automatically. This includes placing your new service account in the Users group (so remove that immediately!). When you open the properties on your new service account, you will find a lot of default settings that you can remove. Change things such as:

- Making sure there is no profile
- Disallowing Terminal Services connections

[11] "10 Steps to Help Secure SQL Server 2000." Microsoft. 28 Jun. 2003. 16 Dec. 2004 <http://www.microsoft.com/sql/techinfo/administration/2000/security/securingsqlserver.asp>

- Disabling remote control
- Denying Dial-In access and etc.

Remember that in the case of service accounts, and all accounts for that matter, less is better. Don't give access where it is not necessary. You can even go as far as explicitly denying the permissions of service accounts that have more privileges than you would like. If your new service has to run with administrative privileges, deny that account access to all the directories besides the one or two that it needs.[12] The User Rights Assignment section of the Local Security Policy provides you with some simple ways to further lock down your accounts as well. You can use policies such as 'Deny logon locally' and 'Deny logon through Terminal Services' to specifically deny your new account the possibility of logging in to your system, in case the account does become compromised.

Account Lockout and Service Accounts

Remember that regular user accounts can have account lockouts policies without causing a lot of problems, but an account lockout for a service account can be disastrous, especially in a large environment with many services where a denial-of-service condition could be reached. A hacker could purposely lock out your services by just attempting log-in with your service account using a wrong password enough times. If you do want to implement account lockout, the concept of having a resource domain really pays off, since account lockout is a domain-wide policy. This way, you can implement account lockout on your user domains, and keep the resource domain without a lockout policy, while using auditing or an IDS or similar tool to keep an eye on bad logins for your resource domain. The chances of bad logins on a resource domain are significantly less. This is due to the fact that bad logins will only tend to occur when someone changes a password for a service, or when someone implements a new service and types a wrong password.

Service Account Naming and Passwords

Another problem area with service accounts is naming. Naming a service account something obvious, like SQLService is useful to the administrators, but it's even more useful to the hacker. While obfuscation is not usually the best way to secure a computer, in this case it may slow a hacker down enough to not want to try every account available.[13] This is obviously less effective on a domain where there are no accounts other than service accounts, but will help in those domains where everything is centralized.

Passwords for these accounts should be lengthy and complex. Administrators

[12] Brien M. Posey. "Secure Apps to Stop Network Attacks." ZDNet. 27 Feb. 2003. 16 Dec. 2004 <http://insight.zdnet.co.uk/software/applications/0,39020466,2131151,00.htm>
[13] "Security Settings – User Rights." Computer Performance. 07 Jan. 2005 <http://www.computerperformance.co.uk/w2k3/gp/group_policy_security_user_rights.htm>

do not normally need to memorize them, and they may not be able to be changed as often as a typical user's account does. Changing passwords for service accounts is a necessary evil. The fact that users do not log in with the service accounts makes it more difficult to make password changes because there is not an individual who will normally touch their account on a daily basis and get the warnings about an upcoming password expiration. If you do not change the password on your service accounts, you have given a hacker a great opportunity for a login that will be good for as long as that service is around![14]

When you have a small number of services to change passwords for, setting up change schedules and timelines is not terribly difficult, but in the large production domain scenario, there could be hundreds of different accounts and the necessity for changing them becomes even more important since it is much more difficult to keep tabs on how the accounts are being used. Setting a schedule and having a direct, step-by-step implementation plan can do wonders in enabling your large production domain service account password changes to go more smoothly and quickly. Creating scripts or using tools that will enable you to change password and restart services can also be very helpful in large environments. There are even some tools starting to surface to automatically make these sorts of changes for you, although automating these changes needs to be carefully considered, since you could accidentally bring a lot of services down at one time. Scripting changes and testing can actually be a very good way to ensure the service is working correctly. If you have a script that makes the change and bounces the service, you could also have a script that tests the service to ensure it is up and operational.

A common issue with large environments is the need to use a specific domain service account for many services (i.e. for a server farm full of web servers). If you have a hundred services running that use the same service account, changing them all at once is impossible, and changing them slowly causes problems with the resources they need to access. One solution for this is to create a service account group and place the service account in that group. Next, give that group access to everything that the service account needs access to on the network or on the local servers. When it's time to change all the passwords, you can simply create a new domain service account and place it in the same group. Now the new service account has the same access that the old one did, and you can slowly move services over to the new account (with the new password).

Testing!

Any time that you are changing a service to run under a different account, or you are changing the password that service has been using for any length of time,

[14] Douglas Ford. "8 Simple Rules for Securing Your Internal Network." GIAC. 12 Sep. 2003. 14 Dec. 2004 <http://www.giac.org/practical/GSEC/Douglas_Ford_GSEC.pdf>

you run the risk of breaking that service. It is very important to spend a little extra time before-hand testing your changes. It is much simpler to set up a test environment these days due to lower server costs and virtual server technology, so setting up a test environment for these changes should be a quick and easy way to head off problems you may experience later. This also will allow you to spend time trying to lock your service accounts (and entire servers) down even further. Every minute spent in testing means possible hours of time saved when a problem arises while implementing the change in production. In large environments, it's a good idea to do your changes first in test, and after the implementation plan is worked out, rolling out the changes in a very controlled and methodical fashion, so if there are going to be any problems, you can catch them early and eliminate or at least reduce any outages or inconveniences.

Auditing Service Accounts

Auditing of your service accounts is also very important in order to keep your systems secure. In the case of service accounts in a dedicated resource domain, you are not going to have many accounts being added, dropped, or otherwise changed. As a result, any change that occurs should raise a flag, and should be checked out. Using an event log aggregator and looking for specific events can be helpful in discovering security problems and/or services that are not working correctly. There are many different events possible, but here is a quick list of some of the more useful event log entries:

- Event ID 528 – Successful logon (type 5 is a Service Account logon)
- Event ID 529 – Failed logon due to bad password or user name
- Event ID 529 – Failed logon due to account lockout
- Event ID 534 – Failed logon due to inadequate rights (account trying to interactively logon that does not have permission, or other rights problem)

There are many different event IDs for logons, and Windows 2000/2003 give you more specific information than Windows NT ever did. Check out the article at http://www.windowsnetworking.com/kbase/WindowsTips/WindowsNT/AdminTips/EventLogs/W2KandNTSecurityEventLogDescriptions.html for some more guidance.

In order to get events in your security log related to accounts, you will need to implement auditing and logging. You will want to make sure you have enabled the following entries in the Audit Policy.

- Audit Account Logon Events – Domain account logons for any domain computer authorizing against your domain controller.
- Audit Account Management – Changes to any accounts are recorded with this setting (passwords, name changes, deletions and additions, etc.)
- Audit Logon Events – Logon Events auditing will enable you to see when local accounts are logged on (as opposed to Domain Accounts).

Auditing should also include a simple check of all of the accounts in your

domain and on the servers. It is possible that after a system is compromised, the hacker will create an account for themselves that will look like a service account.[15]

The Pecking Order

If you can get away with using a local account in which you have removed most privileges, then that is the best account to use, assuming you have a good password change policy and implementation plan in place. Permissions issues and manageability may force you use a more standard account for your services, and as long as they really do not need network access, the built-in Local Service account would be the next step up. If your service will need some network-level access, then the built-in Network Service account is the account to try. The next account you can look to is the built-in Local System account, as many services actually require this account to work at all. A Domain account can be used as a service account, if properly locked down, and you need permissions in a lot of places for many services. So a quick list from most desirable to least desirable looks like:

- Created Local Account (permissions locked down)
- Built-in Local Service Account
- Built-in Network Service Account
- Built-in Local System Account
- Domain Account

Wrapping Up Your Service Accounts

Service accounts are an important part of production Windows domains, and are often a source of great confusion and misinformation. Locking down your service accounts should be a basic component of your hardening guide for all computers. The fact that Microsoft does not differentiate between a service account and a user account is both good and bad, in that it is easier to work with one system for all account types, but it is also less secure when your service accounts could easily be used as a regular user. While it requires a bit of time to lock down a new service account to allow access only to what it needs, it is well worth the time spent, since it will help you in cases where a service might be compromised but the access granted to that service is restricting a hacker's ability to wreak widespread havoc. Defense-in-depth requires that you look at more than the perimeter, and service accounts are one major place where the in-depth strategy can serve you well. While there is a lot to remember during the creation and operation of service accounts, the hope is that this paper has sufficiently prepared you to apply accounts to your services in a more secure manner. Remember, secure services start with hard limits!

[15] Chad Todd. "You Got Hacked! Now What?" Microsoft Certified Professional Magazine Online. Sep. 2002. 16 Dec. 2004 <http://mcpmag.com/features/article.asp?EditorialsID=295>

References

"10 Steps to Help Secure SQL Server 2000." Microsoft. 28 Jun. 2003. 16 Dec. 2004 <http://www.microsoft.com/sql/techinfo/administration/2000/security/securingsqlserver.asp>

Balter, Dan. Exam Cram 2 – Managing and Maintaining a Windows Server 2003 Environment (Indiana: Que, 2004)

Ben Smith, Brian Komar, and Microsoft Security Team, Microsoft Windows Security Resource Kit (Washington: Microsoft Press, 2003)

Bragg, Roberta. "Fourteen Privileges That Can Be Abused in Windows 2000, Part 1." Informit. 14 Sep. 2001. 19 Nov. 2004 <http://www.informit.com/articles/article.asp?p=23332>

"Default Permissions and User Rights for IIS 6.0." Microsoft. 23 Aug. 2004. 14 Dec. 2004 <http://support.microsoft.com/default.aspx?scid=kb;en-us;812614>

Deuby, Sean. "Lessons for an AD to Windows 2003 Upgrade," Windows IT Pro Magazine January 2005.

Eric Cole, Jason Fossen, Stephen Northcutt, and Hal Pomeranz. SANS Institute Track 1 – SANS Security Essentials - Windows Security (2004)

Ford, Douglas. "8 Simple Rules for Securing Your Internal Network." GIAC. 12 Sep. 2003. 14 Dec. 2004 <http://www.giac.org/practical/GSEC/Douglas_Ford_GSEC.pdf>

Guhanick, Barbara. "Service Account Vulnerabilities." SANS. Aug. 15 2001. 03 Nov. 2004 <http://www.sans.org/rr/whitepapers/application/5.php>

Howard, Michael. "Tackling Two Obscure Security Issues." Microsoft. 14 Aug. 2002. 11 Nov. 2004 <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dncode/html/secure08192002.asp>

Joel Scambray and Stuart McClure, Hacking Windows Server 2003 Exposed (California: McGraw-Hill/Osborne, 2003)

Joel Scambray and Stuart McClure, Hacking Windows Server 2000 Exposed (California: McGraw-Hill/Osborne, 2001)

Johansson, Jesper. "Hacking: Fight Back – How A Criminal Might Infiltrate Your Network." Microsoft. 06 Dec. 2004. 08 Jan. 2005 <http://www.microsoft.com/technet/technetmag/issues/2005/01/AnatomyofaHack/default.aspx>

Martin C. Brown and Don Jones. "Security in Microsoft IIS." Informit. 13 Nov. 2003. 28 Dec. 2004 <http://www.informit.com/articles/article.asp?p=101750&seqNum=6>

"Microsoft Windows Server 2003 Default Groups." Microsoft. 22 Oct. 2004 <http://www.microsoft.com/resources/documentation/WindowsServ/2003/standard/proddocs/en-us/Default.asp?url=/resources/documentation/windowsserv/2003/standard/proddocs/en-us/sag_ADgroups_9builtin_intro.asp>

"Microsoft Windows Server 2003 Services Permissions." Microsoft. 12 Jan. 2005 <http://www.microsoft.com/resources/documentation/WindowsServ/2003/standard/proddocs/en-us/Default.asp?url=/resources/documentation/WindowsServ/2003/standard/proddocs/en-us/sys_srv_permissions.asp>

"Platform SDK: Active Directory – The LocalSystem Account." Microsoft. Oct. 2004. 05 Jan. 2005 <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ad/ad/the_localsystem_account.asp>

"Platform SDK: DLLs, Processes, and Threads – LocalSystem Account." Microsoft. Dec. 2004. 05 Jan. 2005 <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dllproc/base/localsystem_account.asp>

Posey, Brien M. "Secure Apps to Stop Network Attacks." ZDNet. 27 Feb. 2003. 16 Dec. 2004 <http://insight.zdnet.co.uk/software/applications/0,39020466,2131151,00.htm>

Redmond, Tony. "Running Exchange Services from LocalSystem." Windows IT Pro. Feb. 2001. 14 Nov. 2004

Reilly, Michael D. "Windows NT Users and Groups." Windows IT Pro. Jul. 1998. 05 Nov. 2004 <http://www.windowsitpro.com/Windows/Article/ArticleID/3597/3597.html>

"Security Settings – User Rights." Computer Performance. 07 Jan. 2005 <http://www.computerperformance.co.uk/w2k3/gp/group_policy_security_user_rights.htm>

Todd, Chad. "You Got Hacked! Now What?" Microsoft Certified Professional Magazine Online. Sep. 2002. 16 Dec. 2004 <http://mcpmag.com/features/article.asp?EditorialsID=295>

Thomas W. Shinder, Debra L. Shinder, and D. Lynn White, Configuring Windows 2000 Server Security (Massachusetts: Syngress, 2000)

"W2K and NT Security Event Log Descriptions." WindowsNetworking. 16 Mar. 2004. 06 Dec. 2004 <http://www.windowsnetworking.com/kbase/WindowsTips/WindowsNT/AdminTips/EventLogs/W2KandNTSecurityEventLogDescriptions.html>

"Windows 2000 Security Hardening Guide." Microsoft. 11 Apr. 2003. 20 Dec. 2004 <http://www.microsoft.com/technet/security/prodtech/win2000/win2khg/05sconfg.mspx#EGAA>