Global Information Assurance Certification Paper

Charles Bolen
Security Essentials Certification (GSEC) v.1.4b option 1
Safe Sharing – A Desktop Users Primer
3 August 2003

Abstract
This paper is written to provide some insight for average Windows users who desire to utilize File and Printer Sharing for Microsoft Networks. The paper will show from a technical standpoint, what happens when File and Printer Sharing is enabled. This will be described both in terms of older versions of Windows (9x and NT) as well as the newer versions (2000 and XP), since there are significant differences between them. Secondly it will describe some of the vulnerabilities that are associated with File and Printer Sharing. Finally, some preventative measures will be presented.

Introduction
In October of 1992, with the release of Microsoft's Windows for Workgroups 3.1 (followed a little over a year later by version 3.11), networking capability was, for the first time, placed in the hands of personal computer users in an office or home setting. Windows for Workgroups was the first version of the Windows operating system to integrate network peer-to-peer file and printer sharing capabilities into the Windows environment.

At the time networked environments were more the exception than the rule and most PC users didn't fully realize how easy it would become to "hook 2 machines together". That such machines could now share files, printers or other peripheral devices, although not readily apparent, was to set off an explosion of interconnectivity. As with any good thing, it wasn't long before this connectivity was exploited by persons engaged in either simple malicious mischief or engaged in outright malevolent actions. In some cases hackers simply rose to the challenge of solving the puzzle of getting into another PC. Others, however, sought to gain access to restricted or confidential information or to turn the use of the PC to their own particular end, such as a Denial of Service attack. As the risks associated with this vulnerability became apparent, competent, forward-thinking system administrators began to regulate or block the use of some of the resource sharing features. Users began to cry out "WHY??". "What's the harm?" "What could happen?". The short answer is simply, PLENTY. The long answer is, of course, a bit more complicated.

First, what I will attempt to show in this paper is from a technical standpoint, how File and Printer Sharing is enabled. Along with this I will try to provide an explanation or a translation of some of the technical aspects into simpler terms. This may give the System Administrator a way to respond to the "What could happen?" sort of questions when sharing becomes an issue on the local network.

Second, I will show some of the vulnerabilities associated with sharing files and printers on a network of Windows-based machines. Many of these vulnerabilities

are found in older versions of Windows (9x and NT). However, some very recent vulnerabilities have arisen that are linked to services that nearly all versions of Windows use for a variety of functions, including file and printer sharing. The recently discovered RPC (remote procedure call) vulnerability is related to the service that allows one computer to request another to perform some action or service. This is precisely the kind of process that facilitates such things as sharing files and printers. This vulnerability affects nearly ALL versions of Windows and has just cropped up in the last two weeks.

Third, I will explain some preventative measures that need to be taken IF File and Printer Sharing is going to be enabled. Some of these measures will be of a general nature and others will be more specific to certain versions of Windows. Where possible, I will distinguish between these general measures and the version specific ones. None of these measures are fool proof and they will only tend to increase the margin of safety. As with any form of protection, there are measures and avenues available that can counter or circumvent it. Unfortunately, new countermeasures are emerging on an almost daily basis.

HOW DOES IT HAPPEN?

How do you share files and printers with other machines running Windows on the network? From a users perspective the procedure is simple. Select the item that is to be shared, either a printer or a folder (since sharing cannot be enabled for individual files), and bring up the alternate context menu with a right-click of the mouse. From the pop-up menu select either Sharing or Properties, as both will bring up the Properties window. Click on the Sharing tab, then click the radio button for Shared as:, assign the resource a name and click OK. If the system is using a FAT file system, there are some rudimentary password options that can be applied at this point. If the file system is NTFS, there are many more security options that can be configured. These password and security options will be discussed in more detail later.

Done. Simple as that. That is as far as most users go with their setup. But what have they done that they may not fully realize? What have they NOT done that perhaps they should have taken the time to do?

First of all, before they ever made the decision of which of their resources to share, the user has to have File and Printer Sharing for Microsoft Networks enabled in their network properties. In actuality, many PC dealers and manufacturers enable this when they install the Operating System (OS) for the user. With this option unchecked, the user would have a very sound defense against the kinds of attacks outlined later in this paper. Kind of like no one can break down the door if there is no door in the wall in the first place.

If File and Printer Sharing is not enabled, the user can still access files and printers on other servers and those shared by other computers on the network. It is not necessary to enable this service in order for such traffic to go out. When

this service is enabled, however, it also allows traffic to come in from outside of the users system. Inbound traffic could potentially come from anywhere on the Internet, if the user's system is connected to the Internet.

This Sharing service is made possible by some basic networking services that Windows uses. By implementing these services within the operating system, there's no need for a dedicated file server, centralized security or any other advanced networking technologies. Server Message Block (SMB protocol) is what Windows uses to accomplish file and printer sharing. In earlier versions of Windows, SMB traffic relies upon NetBIOS or NetBIOS-over-TCP/IP (NBT) as the underlying method of achieving file and printer sharing. The SMB protocol performing the file and printer-sharing dirty work is invisible to end users. The NetBIOS protocol, on the other hand, is visible and provides a convenient method of access. [1] In Windows 2000 and later versions, Microsoft installed a new feature, that allowed SMB traffic to be directly hosted on TCP/IP, thereby providing a kind of "NetBIOS-less" SMB traffic.

NetBIOS was developed several years ago by IBM. It was intended to serve as a network protocol that would work on small networks of PCs. The keyword in that sentence is "small". The design goal was to build a small and fast protocol that would allow for human-assigned names of devices, such as "Joes Computer". The thinking was that it would be easier to use names rather than a complex numbering scheme such as TCP/IP's system. [1] Within this small network, one machine would broadcast a request to communicate with another by issuing a command to NetBIOS, asking for an enumeration of the names of the computers attached to the network. The other system would respond to the request, NetBIOS would send the names of the computers that have been registered and the connection could be established.

Another aspect of NetBIOS is that like all other services it is assigned to work off of certain ports on the computer. Each port is kind of like a pipe or window that opens out to the network beyond the host machine. Also like pipes and windows, ports can be opened or closed. NetBIOS works on ports 137, 138 and 139. The NetBIOS name service operates on port 137. This is how NetBIOS-based services find each other, since the names uniquely identify the machine and the services running on the machine. The NetBIOS datagram service operates on port 138. This provides information that is used by the SMB browser service to fill in the "Network Neighborhood" icon in Windows 9x. The actual file sharing service operates over port 139, the NetBIOS port.

The "NetBIOS-less" File and Printer Sharing used in Windows 2000 works on port 445. This still uses the SMB protocol, but here it is directly hosted on TCP, rather than using NBT to facilitate the communication and data transfer. This method has the advantage of providing a simpler system for carrying SMB traffic, plus it eliminates the need for NetBIOS broadcasts to resolve the names of other machines. It is possible to have both methods operating on a Windows 2000

machine. This is often necessary to provide backward compatibility with machines on the network running older versions of Windows. When both old and new versions are found on the network, the NetBIOS service must be in place to allow SMB traffic to travel over port 139, accommodating those machines that do not support direct hosting over port 445. When a request for SMB traffic is received, both methods are tried at the same time and the first method that responds is the one used.

Now that File and Printer Sharing has been enabled, and the decision has been made to share some of our resources with our co-workers, to what sort of vulnerabilities might we be exposing ourselves? What are some of the ways in which a snooper, intruder or attacker could access and/or compromise your files or your system?

WHAT CAN HAPPEN?

A NetBIOS attack has long been one of the favorite avenues of attack for hackers, particularly against machines running NT and particularly for those who may be new to hacking, or Newbies. Its popularity stems from the ease with which even an unskilled hacker can locate a computer on the Internet that is broadcasting its vulnerability to a NetBIOS attack. A check of the Internet Storm website on any day of the week will show that scans of port 137 number in the top three, if not at the very top, of the list of ports being scanned. One of the main reasons to scan for port 137 (as well as 138 and 139) is to search for computers that are vulnerable to a NetBIOS attack. It is estimated that at least 10% of the users on the Internet leave their hard disks open on port 139 and are vulnerable to a NetBIOS attack.[2] It is interesting to note that in recent weeks, the number of scans of port 445 have been increasing, signaling a shift in targeted OS's away from NT in favor of Windows 2000/XP.

Port scanning can be done using any number of tools that are readily available on the Internet. Some are free and some are commercial versions. A port scanner like Nmap will send out a flood of data packets to the network and depending on the response that comes back (or doesn't come back), can tell what services are listening and on which ports. When a computer is detected that is listening on a vulnerable port, the scanner can provide the IP address and much more information about the susceptible host.

Once a vulnerable host is found, the attacker can initiate a series of `net` commands. The first will probably be `nbtstat -A {IP address}`, which will show the machine name, the group that the PC belongs to, the name of active users on the machine, and the MAC (Media Access Control) address. Right away they have three important pieces of information. This is information that most users don't realize they are sharing so easily.

Next, the `net view \\{IP address}` command will show the name of any shared resources. When a share is discovered, the intruder can tap into it using

the `net use <drive letter>: \\{IP address}` command. Assuming that no password was required or that it was defeated, the intruder now has complete access to these files, just as if they were on his own machine. He is free to do whatever the permissions assigned to the share will allow. If the access is sufficient, he can also add files or programs to the victim's PC. This is the simplest and most direct attack. Once an intruder is inside your machine, they can also use that as a beachhead to probe deeper and gain further access. One drawback, from the intruder's standpoint, is that this could also leave some very prominent tracks, as when the user shuts down his machine and gets a warning that there are still 1 users connected to his computer and does he really want to shut down.

Let us suppose that the results of the attackers `net view` command produced a response like `There are no entries in the list`. Even though there are no shared resources visible, this does not mean that there is no way for an attacker to compromise your system. Tools such as NAT (NetBIOS Auditing Tool) will probe a system for a variety of vulnerabilities, even if no shares exist. The Tool will attempt to gain file system-level access by establishing a TCP connection on port 139 (NetBIOS). If the connection is established, protocol levels are negotiated to setup client/server capabilities and security levels. If further authentication is required (which it should be) the NAT makes various attempts at "guessing" usernames and passwords, likely without any awareness on the part of the owner of the target that such an intrusion is taking place. Once a session is established, NAT looks for file system shares and checks for write access. At this point the target is considered vulnerable and it is just a matter of to what extent. As can be seen from the description, NAT is almost a kind of self-propelled hacking tool.

Even though you may not have set up any shares on your system, or have set them up as hidden shares, you are still vulnerable to attacks described. This is because Windows creates certain default hidden shares. These default hidden shares are generally C$ (plus the root of any other partitions), ADMIN$ and IPC$. The most vulnerable of these is the IPC$ (Inter Process Communication) share, since this can provide a wealth of information about your system and it is the one that cannot be disabled, even through editing the registry.

The IPC$ share is usually exploited through the use of a null session. Null sessions are sessions established in which no credentials or authentication is required. In essence, a free pass. On the surface it may seem ludicrous that such sessions are allowed, but for reasons that are beyond the scope of this paper, there are numerous networking and operating system functions that both utilize and benefit from null sessions.

If the `net view` command did not allow the enumeration of shares because of permissions, a null session could be used to map the IPC$ share, `C:\tool>net use \\IPaddress\IPC$ "" /u:""`. With this completed successfully, the

`net view` command would now give us a list of shares on the target machine. Further, an attacker can make use of information acquired from a null session to mount a password attack against a system and then use it, for example, as a platform for distributing warez. A good example of this is described at http://www.giac.org/practical/Michael_Kriss_GCIH.doc

If the level of access gained by an intruder is sufficiently deep, they may even be able to install executable files, that can open up a vast array of security problems, too numerous to mention here. Programs that monitor keystrokes to catch passwords, programs that open up system and configuration information that can lead to full access for outsiders, programs that provide a backdoor for intruders to access your system without your knowledge, programs that launch attacks to other systems are all possibilities once your machine has been compromised.

Remember that one of the most easily obtainable pieces of information that can result from a NetBIOS hack is usernames on the system. With this information, an attacker now has half of what they need to fully connect to a system. This can lead to a brute force attack with a password dictionary program and potentially gain complete access to the vulnerable computer.

WHAT CAN BE DONE ABOUT IT?

Now that we have seen some of w h a t can happen and hopefully we have a better understanding of h o w it can happen, lets look at some of the methods and techniques that can be used to eliminate or to mitigate the vulnerabilities of a system. Not all of these will be applicable to all systems and I will try to specify particular Windows versions where I can.

First And foremost, the surest form of protection is DON'T ENABLE File and Printer Sharing. If you don't have it, they can't exploit it. Along with that, if you are not going to use File sharing, you can disable NetBIOS over TCP/IP. This would provide a significantly greater measure of protection against NetBIOS attacks by eliminating NetBIOS traffic through ports 137-139. It has been pointed out that disabling NetBIOS over TCP/IP in the Network properties may not truly stop NetBIOS from functioning. It may just be implementing a sort of filter that prevents NetBIOS information from going over the network or to the Internet. Also, port scanners would still be capable of enumerating shares and usernames on Windows 2000 machines by looking at port 445.

In order to fully disable NetBIOS over TCP/IP you can disable it in the device manager, where it shows up as a hidden device. Unfortunately, this could absolutely cripple a machine on a Windows based network, as it would no longer be able use UNC (Universal Naming Convention) names when trying to connect to other Windows 2000 machines.

Assuming that you have enabled File and Printer Sharing and that NetBIOS over TCP/IP is also enabled, the best and most secure protection is simply don't let anybody else from outside your network take part in it. Put up a firewall. Block all incoming and outgoing traffic on the NetBIOS ports and include port 445 if you have Windows 2000 or XP. Firewalls can be set up using either hardware or software. Some software products are even available at no cost, such as Zone Alarm, and can provide a fair amount of protection. Other more sophisticated versions would generally be priced accordingly. A firewall can also be configured on a piece of hardware. A setup consisting of just two network cards in an old PC running Linux can configure ports as either open or closed and provide excellent regulation of network traffic.

The next most effective direct measure you can take to protect your files and your system is to make it as difficult as possible for the intruder to access your shared resources. SET A PASSWORD ON YOUR SHARES. A STRONG PASSWORD. It would seem that such basic security measure should not have to be mentioned, but it cannot be emphasized enough. Granted there are ways to crack passwords, but some of these can be very time consuming. A strong password can be like a locked window. The lock (or strong password) may cause some intruders to simply move on, looking for easier targets. Setting a password for access to any shared resource is undoubtedly the most important measure the user can implement to protect shares.

Any password is better than no password, but some passwords are better than others. The stronger the password is, the better the protection it will provide. Strong passwords are simply more complex passwords. Passwords that contain letters AND numbers are better than those that are made up of just letters. Better still are those that contain special characters, such as * or # or @. The best password would be one that contains all of these characters. At a minimum, they should contain two of these three possibilities.

At the same time that passwords would be set on the shares, there are other security options that can be put in place. In Windows 9x and other FAT file systems, those options were limited to making files Read Only or granting Full Access. With either option they could set a password to control who had access. A third option allowed for setting a different password for each level of access, where users with one password would have just Read Only access while users with another password could have Full Access. That was the extent of access control with FAT file systems. When sharing printers there is simply a password option.

In the NTFS file system versions of Windows, such as NT, 2000, and XP, there is a much greater range of options that the user can configure to control access and set permissions. Under the Sharing tab, there is a Permissions option that allows the user to specify precisely who can access the files and what level of access they will have, Read, Write, or Execute. One thing to stress at this point

is NEVER allow Full Access to the Everyone group, since the Everyone group is just what it says. Its Anyone. Often this group will appear as the default and should, at a minimum, be replaced by the Authenticated Users group. Whenever possible, restrict access only to specific users or groups. At the same time, don't grant more permission than is necessary. If Full Control is not needed and Read will suffice, then just allow Read and explicitly deny Change or Full Control.

Under the Security tab, there is a wider range of Permission options that the user can allow or deny. The same rules as above would apply here: don't allow any more than is necessary. Also, there is an Advanced button that that allows the user to configure a much more extensive list of features and controls.

One more feature to recognize at this point is the Check box at the bottom of the Security tab, just below the Advanced button, that would "Allow inheritable permissions from parent to propagate to this object." If the user is configuring the Security/Permission settings on a new folder, a simple checkbox allows or prevents the permissions of the parent folder from being passed on to (inherited by) the new folder. This is a very powerful little feature that keeps the less secure settings of the parent folder from being automatically passed to the new folder.

The wide array of permissions and security options available in the newer versions of Windows may seem daunting or bothersome to some users. In reality, they are a wonderful tool and as with any tool, with power comes responsibility. If you want to tighten up the Security settings on your shared folders, you have to take the time to go through these options and decide how to structure the settings.

Along with this is another common sense step and that is to not share any more of your resources than necessary. To sum it up quite simply, don't share your entire C: drive just to let someone into your "tmp" or "shared" folder. In fact, it is a very good idea to designate just one folder as an area for shared files. Some new or inexperienced hackers will be anxious to get to your system files or to your root directory. Confining them to a very small part of your system may give you the protection you need. At the same time, bear in mind that at this point you have already been "had" by an intruder and your problems may extend well beyond one small part of your system.

Another simple step that can be taken to protect your shares is to hide them. When naming your shared resources, simply adding a "$" at the end of the name will cause to disappear from a list generated by a basic `nbtstat` command. This can provide an additional, albeit small, measure of defense.

The default hidden system shares are, of course, known to all but these can be disabled by either disabling the Server service or through editing the Registry at

`HKeyLocalMachine\SYSTEM\CurrentControlSet\Services\LanManServer\Parameters`.
This Registry edit will not, however, disable the IPC$ share. Also, bear in mind that disabling these shares may cause problems if you are operating in a Windows-networked environment.

Another less often mentioned tactic that will work with some operating systems is to add a Scope ID to the NetBIOS name. In versions of Windows prior to Windows 2000, Scope ID could totally block all NetBIOS traffic from any machine that did not have an identical Scope ID. As mentioned earlier, however, Windows 2000 will also allow such traffic over what is essentially a "NetBIOS-less" route, through port 445. This could allow a Windows 2000 machine to open communications with a machine with a different Scope ID. So depending upon the platform from which an attack may be launched, implementing Scope ID may or may not provide some additional protection.

To set the Scope ID in versions prior to Windows 2000, you do so under the WINS tab in Network Properties. Since Windows 2000 is not constrained by Scope ID, it needs to be implemented by directly editing the Registry. This is not a task for the timorous or the uninitiated and is best left out of most systems. In addition, some problems may be created as a result of implementing Scope ID, particularly on Windows 2000 domain controllers. A "delayed" boot may result, as documented in this almost humorous quote from Microsoft Knowledge Base Article – 255195.

> "When a Windows 2000 domain controller has a NetBIOS scope ID defined, it may appear to stop responding (hang) during boot with a "Preparing Network Connections" message. If the computer is allowed to sit for **two hours** (emphasis added) or longer, the boot process may finish. [3]

As mentioned earlier, when an attacker attempts to connect to you system, they may be able to do so as an anonymous user with no password, i.e. a null session. There is a Registry edit that can restrict or eliminate null sessions, however the same caveat as above applies to Registry editing. In earlier versions of Windows, sessions were either allowed or prohibited. Windows 2000 has added a third option, which restricts sessions only to users that are explicitly authorized. To regulate this feature through the Registry, go to `HKLM\SYSTEM\CurrentControlSet\Control\LSA\`, select `RestrictAnonymous`, click on Edit → Modify and set the value to 1 in Windows 9x or NT. In Windows 2000 a value of 2 will stop all null sessions and a value of 1 will restrict them to authorized users only.

Restricting null sessions may also cause problems with broken Trust in both NT and Windows 2000 and may also generate problems with print queues, so proceed cautiously. It is of limited utility to protect one part of your system by breaking another.

After all of that: Enjoy Safe Sharing

[1] A NetBIOS-Over-TCP/IP Name Resolution Services
http://www.ehsco.com/reading/19960915ncw1.html

File and Printer Sharing (NetBIOS) Fact and Fiction
http://cable-dsl.home.att.net/netbios.htm

CIFS: Common Insecurities Fail Scrutiny
http://www.ussrback.com/docs/cifs.txt

SANS Institute. "SANS Security Essentials V: Windows Security"
p. 3-13 Windows 2000, Course Material (February 2003)

Accuracy in the Networking Media
http://www.networkmagazine.com/article/NMG20000510S0032

NetBIOS Based Hacking Tutorial By Gaurav Kumar
http://www.mycgiserver.com/~ethicalhackers/netbios.html

[2] Port Microsoft
http://www.iss.net/security_center/advice/Exploits/Ports/groups/Microsoft/default.htm

Ports 135 - 139
http://users.pandora.be/lechat/Ports 135 - 139.htm

The Unofficial NT Hack FAQ - Section 05
http://secinf.net/windows_security/The_Unofficial_NT_Hack_FAQ/The_Unofficial_NT_Hack_FAQ__Section_05.html

NULL Sessions In NT/2000
http://www.sans.org/rr/papers/67/286.pdf

NetBIOS Null Sessions: The Good, The Bad, and The Ugly
(UPDATED January 2, 2003)
http://www.brown.edu/Research/SysAdmins/articles/netbios_null_sessions.html

Weak Passwords + Null Session = Windows 2000 Exploit
http://www.giac.org/practical/Michael_Kriss_GCIH.doc

NTBugtraq Archive. Re: More NetBIOS over TCP/IP in Win2K:
http://archives.indenial.com/hypermail/ntbugtraq/2000/May2000/0095.html

[3] NetBIOS Scope ID Causes Windows 2000 Domain Controller to Stop Responding on Boot
http://support.microsoft.com/?kbid=255195

Using and Troubleshooting the TCP/IP Scope ID
http://support.microsoft.com/default.aspx?scid=http://support.microsoft.com:80/support/kb/articles/Q138/4/49.asp&NoWebContent=1

Null Sessions
http://netsecurity.rutgers.edu/null_sessions.htm

Microsoft Knowledge Base Article – 204279: Direct Hosting of SMB Over TCP/IP
http://support.microsoft.com/default.aspx?scid=http://support.microsoft.com:80/support/kb/articles/Q204/2/79.ASP&NoWebContent=1