# Database Servers on Windows XP and the Unintended Consequences of Simple File Sharing

David Litchfield [davidl@ngssoftware.com]
16th November 2005

An NGSSoftware Insight Security Research (NISR) Publication

http://www.ngssoftware.com

## Introduction

This paper presents some unexpected consequences of running database servers on Windows XP with Simple File Sharing enabled. In the real world, this kind of setup would typically be a developer's system and as it turns out, in some cases depending on the database software, you might not just be sharing your files but exposing both database services and data. In one case an attacker can easily gain DBA access to the database if Simple File Sharing is enabled. We'll examine the commercial databases, namely, Oracle, SQL Server, DB2, Sybase and Informix and see which are exposed, to what level and why.

## What is Simple File Sharing?

Before Windows XP, to gain access to a shared file on a Windows NT or 2000 box, you needed to have a valid user ID and password - that is assuming you weren't exploiting some other means to get access. This made it difficult for people out there that wanted to share files out to the general public, so with Windows XP, Microsoft introduced Simple File Sharing. With Simple File Sharing all access is granted via the guest account. In this way, if a user is sharing music or pictures from their XP system at home, they don't have to give out a user ID and password to everyone - people wanting access are simple given access through the guest account.

For those that want to share files but not with the world and their dog, these people could just use the classic way of sharing files. By far and above the most popular way of sharing files on Windows XP is with Simple File Sharing. As we'll see shortly, this has a significant impact on the security of a computer if a database server has been installed. But before exploring this let's look at the differences between simple and normal file sharing. The table below describes what happens when a remote user attempts to authenticate under certain scenarios:

| Simple File Sharing with Guest Account Active | | | |
|---|---|---|---|
| Valid Username | Wrong Password | Access Granted | Authenticated as Guest |
| Valid Username | Right Password | Access Granted | Authenticated as Guest |
| Invalid username | Any Password | Access Granted | Authenticated as Guest |

| Simple File Sharing with Guest Account Not Active | | | |
|---|---|---|---|
| Valid Username | Wrong Password | Access Denied | - - - |
| Valid Username | Right Password | Access Denied | - - - |
| Invalid username | Any Password | Access Denied | - - - |

| Classic File Sharing with Guest Account Active | | | |
|---|---|---|---|
| Valid Username | Wrong Password | Access Denied | - - - |
| Valid Username | Right Password | Access Granted | Authenticated as Username |
| Invalid username | Any Password | Access Granted | Authenticated as Guest |

| Classic File Sharing with Guest Account Not Active | | | |
|---|---|---|---|
| Valid Username | Wrong Password | Access Denied | - - - |
| Valid Username | Right Password | Access Granted | Authenticated as Username |
| Invalid username | Any Password | Access Denied | - - - |

It is clear that, for Simple File Sharing to work the Guest account must be enabled. Incidentally, this can lead to a point of confusion. The Windows XP Home Edition User Accounts control panel applet contains an entry for the Guest account. The UI tells us that the Guest account is "off" or "on", but what this actually refers to is whether the Guest account can be used to log on locally. What the UI says has no bearing on whether the Guest account is disabled or not. Running the "net user guest" command from a console window will tell you whether it is or not. The general rule is that, if Simple File Sharing is enabled, then the

guest account is enabled, too. Only if someone has gone out of their way to change the configuration would the guest account be disabled.

Simple File Sharing is discussed in full here:
http://support.microsoft.com/kb/304040

## The Impact of Simple File Sharing with regards to Database Servers

If a Windows XP box that has Simple File Sharing enabled has a database installed and that database supports OS based authentication in any way then it is possible for an attacker with no valid user ID and password to gain access to the database's services and data – sometimes with DBA privileges. We'll look at each database in turn and determine in what way it might be vulnerable.

**Oracle**
On Windows, if a user is a member of the ORA_DBA local group then they can connect to the database server as a SYSDBA without providing the password for the SYS user. When processing such a logon Oracle uses the NTLM SSPI AcceptSecurityContext() function. If the user has presented the correct username and password this function returns 0 and creates a token. The problem with this is that if Simple File Sharing is enabled all attempts to logon are successful – the user is authenticated as the guest user. However, as far as Oracle is concerned the authenticated principle is not "Guest" but whatever the remote user supplied as the username when they authenticated. If the username they presented is the name of a valid user in the ORA_DBA group then Oracle authenticates the user and gives them SYSDBA access having made the assumption, in good faith, that the remote user must have had the right password as AcceptSecurityContext() "said" they were successfully authenticated. All an attacker needs to do is discover the name of a member of the ORA_DBA group and create a user on their own system with the same name. As the password is irrelevant the attacker can then gain access to the Oracle server as a SYSDBA.

**Informix Dynamic Server**
With Informix, the client sends their user ID and password over the network in clear text. The Informix server then passes these to the LogonUser() function, specifying a logon from the network. With Simple File Sharing enabled, the LogonUser() function works slightly differently than one would expect given the matrix above; if the username passed by the authenticating client exists on the remote system then LogonUser() will not log the user on as guest but attempt to log on the actual user. Thus, if the password is wrong, the client will not be authenticated. However, as expected, if the remote client attempts to authenticate with a username that does not exist then they are logged on as the guest user. So, if a remote attacker wanted to gain access to an Informix database server running on XP with Simple File Sharing enabled they need only specify a username that doesn't exist on the system. They, of course, can use the guest account instead if has not been renamed.

**DB2**
Like Informix, DB2 uses the LogonUser() function. However, before calling the function it checks to see if the user exists or not. Thus, if a remote user presents a username that doesn't exist then they will not be able to gain access to the database resources. As such they could simply use the Guest account if the guest account has not been renamed to gain access – however, as the DB2 runtime client doesn't allow a blank password the attacker would need to write their own client.

**Microsoft SQL Server 2000**
By default, Microsoft SQL Server 2000 is not vulnerable. Like Oracle, SQL Server authenticates the client using the NTLM SSPI AcceptSecurityContext() function and the user is logged on as Guest, however, as SQL Server requires that a specific user be granted access, the remote user can log in – by default SQL Server doesn't allow Guest access to the database server. If, for whatever reason, someone has granted either the Guest account or the built-in Guests group access to the SQL Server then a remote user without valid credentials will gain access.

**Sybase**

Sybase is different from the other databases since it relies on external authentication by the operating system when authenticating OS users. As such, we need not explore Sybase further.

## Wrapping Up

We can see that enabling Simple File Sharing can have unintended consequences when it comes to database servers. If you are running a database server on Windows XP and you do want to share your files then I'd strongly recommend not using Simple File Sharing but rather plump for the more safe classic method. Further, you may also wish to consider using XP's Internet Connection Firewall to block access to the database services.