SMB 3.0

General Information found on the internet on SMB.

Introduction

Server Message Block (SMB) is a protocol that has long been used by Windows computers for sharing files, printers and other resources among computers on the network. The server message blocks are the requests that an SMB client sends to a server and the responses that the server sends back to the client.

There are also SMB clients for other operating systems. Samba is a well known SMB server implementation for UNIX and Linux that allows those operating systems to act as file and print servers for Windows and Linux clients. It's freeware and comes with most Linux distros. The current version is 3.6.5 and you can download the source code from the Samba.org web site.

Microsoft has improved the SMB protocol over the years. In 2006, they came out with a new version, SMB 2.0, in conjunction with Vista, and SMB 2.1 with Windows 7. Version 2 was a major revision with significant changes, including a completely different packet format. Windows 8 introduces another new version, SMB 3.0. If you want to delve deeply into the technicalities of the protocols, differences between them, and how they work at the protocol level, you can download this 394 page PDF document from Microsoft on the SMB Protocol Versions 2 and 3 Specification.

Each incarnation has included performance and security improvements, but there have been SMB vulnerabilities uncovered along the way that expose systems to potential attacks. For example, in 2010, Microsoft issued critical Security Bulletin MS10-020 to address a vulnerability in SMB that could allow remote code execution in various versions of Windows, ranging from Windows 2000 SP4 to Windows 7 and Server 2008 R2 (including the Server Core installation).

SMB signing

One security mechanism that has been in Windows SMB since Windows 98/NT is SMB signing. SMB signing is supported in all current versions of Windows; the best way to configure it is via Group Policy, although you can also do it by editing the registry. By default, SMB signing is required on domain controllers. It's enabled by default, but not required on SMB 1 clients, and disabled on SMB 1 servers.

The settings options were simplified in SMB 2. The "enabled" and "disabled" settings were done away with and you only specify whether SMB signing is required or not required. Again, the default setting for domain controllers is "required." For other SMB 2 servers and clients, the default is "not required."

Let's take a look at the Group Policy settings for configuring SMB signing in Windows 8 (similar to previous OS versions). First, open the Local Group Policy editor. You won't find it by typing **Group Policy Editor** or **gpedit.msc** on the Metro Start screen; you'll just get a message that "No apps match your search," as shown in Figure 1.

SMB 3.0

Figure 1

To open the Group Policy Editor console in Windows 8, first open the Run box (which you *can* find by typing its name on the Start screen) and then type **gpedit.msc** there. Now navigate in the left pane's tree to **Computer Configuration | Windows Settings | Security Settings | Local Policies | Security Options.**

In the right pane, scroll down to find the section of policies that begin with "Microsoft network client." Here you'll find the following policies:

- Digitally sign communications (always)
- Digitally sign communications (if server agrees)
- Send unencrypted password to third-party SMB servers

You'll also find similar policies that begin with "Microsoft network server," as shown in Figure 2.

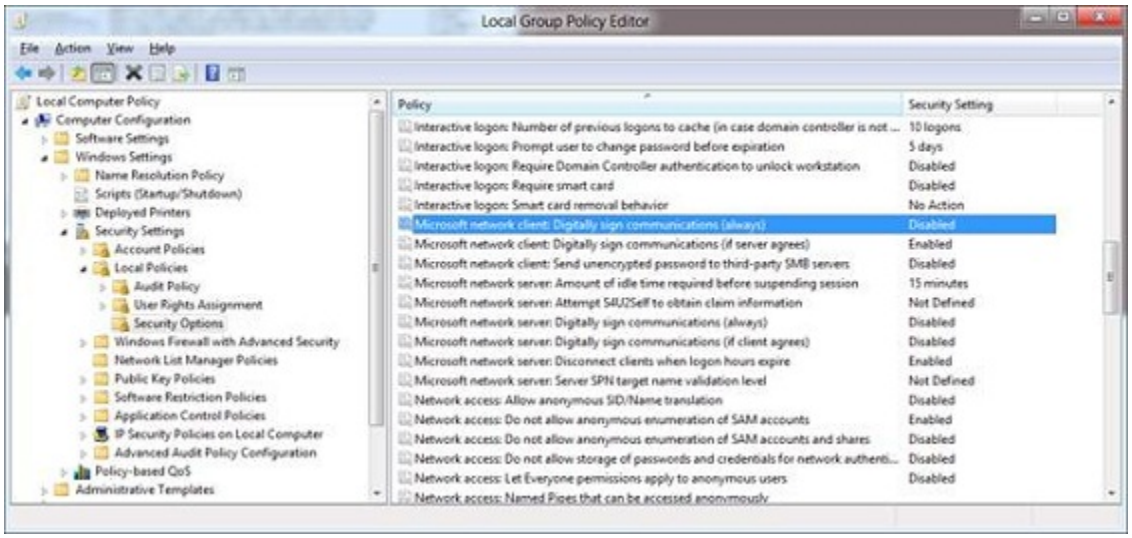

Figure 2

SMB 3.0

Windows operating systems include both a server SMB component and a client SMB component, and these are configured separately. Thus a computer can be configured to require packet signing when acting as an SMB server but not when acting as an SMB client, or vice versa.

You'll see that the client policy to digitally sign communications if the server agrees is enabled, whereas the other two are disabled. If you enable the policy to digitally sign communications (always), the client will require SMB packet signing and will refuse to communicate with a server that does not support signing.

What about that policy to send unencrypted passwords to third-party SMB servers? That's there so that if you have non-Microsoft SMB servers on your network and they don't support having passwords encrypted during the authentication process, you can use this policy to enable your Windows machine SMB client to communicate with them. You won't see a corresponding policy in the Windows network server section. Of course, sending passwords in an unencrypted state presents a big security risk, so it's best to keep this policy disabled unless you absolutely have to use it.

If you prefer to make changes via the registry, in your registry editor navigate to the following keys:

For the SMB client:
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\LanmanWorkStation\
Parameters

For the SMB server:
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\LanmanServer\
Parameters

Create a DWORD value called **RequireSecuritySignature,** as shown in Figure 3, and set its value data to 1.

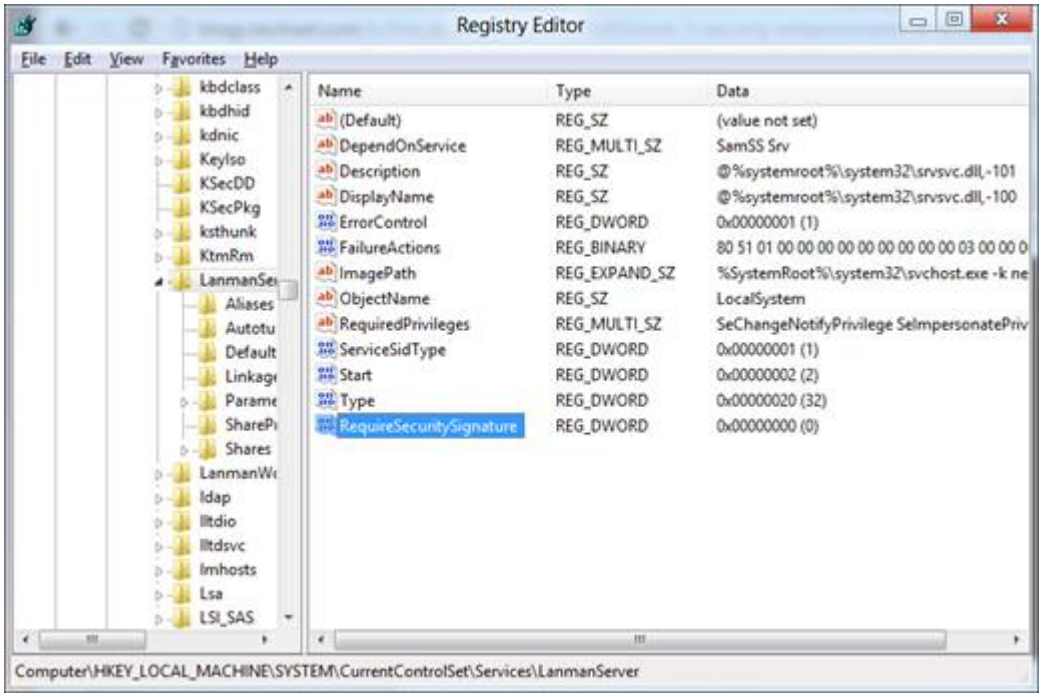

Figure 3

SMB encryption in Windows 8 and Server 2012

Microsoft has made a number of security improvements in SMB 3.0, which will be introduced in the Windows 8 client and Windows Server 2012. A new algorithm is used for SMB signing. SMB 2.x uses HMAC-SHA256. SMB 3.0 uses AES-CMAC. CMAC is based on a symmetric key block cipher (AES), whereas HMAC is based on a hash function (SHA). AES (Advanced Encryption Standard) is the specification adopted by the U.S. government in 2002 and was approved by the National Security Agency (NSA) for encryption of top secret information.

SMB 3.0 in Windows 8 and Server 2012 has the ability to encrypt the SMB data while it's in transit, at a much lower cost than deploying other in-transit encryption solutions such as IPsec. Encryption in transit protects the communications from eavesdropping if intercepted as it passes through the network.

You can enable SMB encryption for specific shares in Server 2012 via the File and Storage Services in Server Manager. You can do it when you create a new file share via the New Share Wizard, as shown in Figure 4.

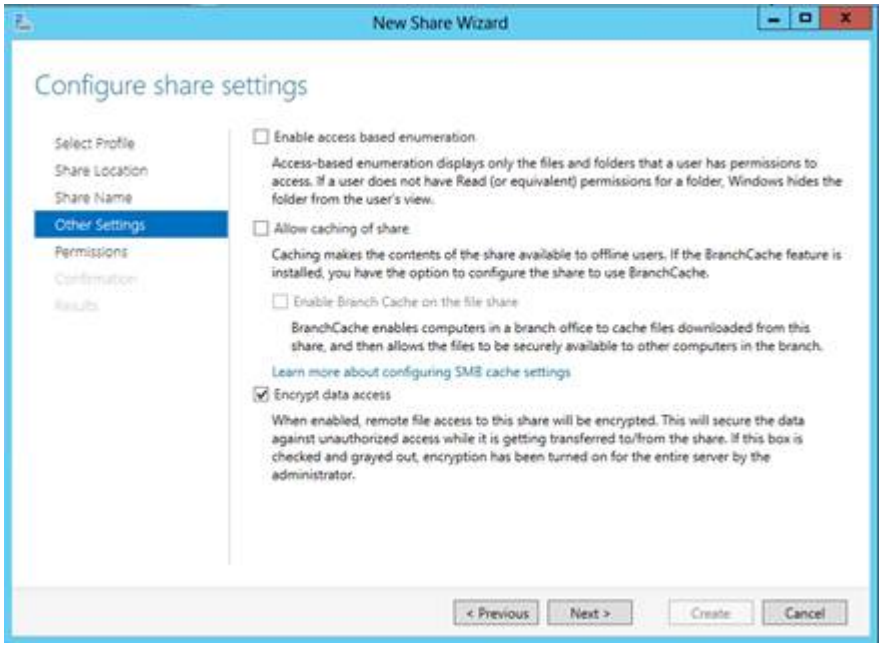

Figure 4

You can also enable encryption on an existing share. Here's how:

1. Open the Server Manager and in the left pane, click **File and Storage Services.**
2. Click **Shares** in the left pane.
3. In the middle pane, right click the share for which you want to turn on encryption.
4. Click **Properties** in the context menu, as shown in Figure 5.

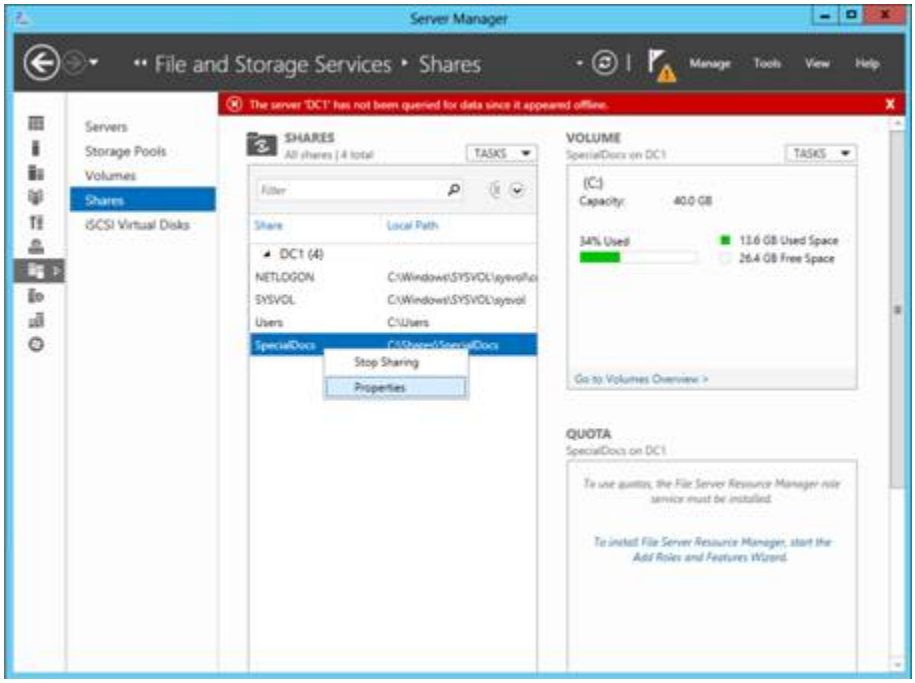

Figure 5

5. In the Share Properties dialog, select **Settings** and check **Encrypt data access.**
6. Click **OK**.

Another way to enable SMB encryption on a share is to use PowerShell. Type the following:

Set-SmbShare –Name <sharename> -EncryptData $true

You can also use PowerShell to enable SMB encryption for all shares on the server:

Set-SmbServerConfiguration –EncryptData $true

If you configure encryption for the entire server as described above, the **Encrypt data access** checkbox on the individual shares will be shown as checked and also grayed out so that they can't be changed on an individual basis.

Once you've enabled SMB encryption for a share, client operating systems that don't support SMB 3.0 will not be able to access that share unless you explicitly allow unencrypted access for those clients. To do that, you'll need to use the PowerShell command below:

Set-SmbServerConfiguration –RejectUnencryptedAccess $false

Dialect negotiation

There are a number of different "dialects" spoken by the SMB protocols, based on SMB version. When an SMB connection is made, the client and server negotiate to identify the highest level dialect both machines can support. This is done by way of the client sending a negprot message to the server with a list of the dialects it supports. The server responds to let the client know which of the dialects it will use.

This is important when making an SMB connection to another computer that uses a different protocol packet format. A security improvement in Windows 8/Server 2012 detects attempts to manipulate this negotiation to cause the systems to use a lower level dialect. If such an attempt is detected, Windows disconnects the SMB connection and logs the event.

Disable SMB 1.0

For better security, you can disable SMB 1.0. Only do this if you don't have any Windows XP (or below) computers on your network that need to connect to the SMB server and you don't have third-party devices that need to communicate via SMB 1.0. Use PowerShell to disable SMB 1.0 as follows:

Set-SmbServerConfiguration –EnableSMB1Protocol $false

Summary

SMB 3.0

The Server Message Block protocol was first developed for the original IBM PC and Microsoft adopted it for Windows and has continued to improve on it over the years. SMB 3.0, which will be a part of Windows 8 and Windows Server 2012, has several security enhancements that help you make SMB connections on your network more secure and guard against man-in-the-middle and eavesdropping attacks related to SMB communications. In this article, we discussed the basics of SMB security and showed you how to enable and configure some of these new SMB security features.

Reference:

http://www.windowsecurity.com/articles/Secure-SMB-Connections.html

http://support.microsoft.com/kb/2686098

31 OCT 2012