Securing a Samba File and Print Server

Samba Security Modes

There are two security levels available to the Common Internet Filesystem (CIFS) network protocol *user-level* and *share-level*. Samba's *security mode* implementation allows more flexibility, providing four ways of implementing user-level security and one way to implement share-level:

1. *security = user:* requires clients to supply a username and password to connect to shares. Samba user accounts are separate from system accounts, but the *libpam-smbpass* package will sync system users and passwords with the Samba user database.

2. *security = domain:* this mode allows the Samba server to appear to Windows clients as a Primary Domain Controller (PDC), Backup Domain Controller (BDC), or a Domain Member Server (DMS). See Samba as a Domain Controller for further information.

3. *security = ADS:* allows the Samba server to join an Active Directory domain as a native member. See Samba Active Directory Integration for details.

4. *security = server:* this mode is left over from before Samba could become a member server, and due to some security issues should not be used. See the Server Security section of the Samba guide for more details.

5. *security = share:* allows clients to connect to shares without supplying a username and password.

The security mode you choose will depend on your environment and what you need the Samba server to accomplish.

Security = User

This section will reconfigure the Samba file and print server, from Samba File Server and Samba Print Server, to require authentication.

First, install the *libpam-smbpass* package which will sync the system users to the Samba user database:

```
sudo apt-get install libpam-smbpass
```

If you chose the *Samba Server* task during installation *libpam-smbpass* is already installed.

Edit `/etc/samba/smb.conf`, and in the *[share]* section change:

```
    guest ok = no
```

Finally, restart Samba for the new settings to take effect:

```
sudo restart smbd
sudo restart nmbd
```

Now when connecting to the shared directories or printers you should be prompted for a username and password.

If you choose to map a network drive to the share you can check the “Reconnect at Logon” check box, which will require you to only enter the username and password once, at least until the password changes.

Share Security

There are several options available to increase the security for each individual shared directory. Using the *[share]* example, this section will cover some common options.

Groups

Groups define a collection of computers or users which have a common level of access to particular network resources and offer a level of granularity in controlling access to such resources. For example, if a group *qa* is defined and contains the users *freda*, *danika*, and *rob* and a second group *support* is defined and consists of users *danika*, *jeremy*, and *vincent* then certain network resources configured to allow access by the *qa* group will subsequently enable access by freda, danika, and rob, but not jeremy or vincent. Since the user *danika* belongs to both the *qa* and *support* groups, she will be able to access resources configured for access by both groups, whereas all other users will have only access to resources explicitly allowing the group they are part of.

By default Samba looks for the local system groups defined in `/etc/group` to determine which users belong to which groups. For more information on adding and removing users from groups see Adding and Deleting Users.

When defining groups in the Samba configuration file, `/etc/samba/smb.conf`, the recognized syntax is to preface the group name with an "@" symbol. For example, if you wished to define a group named *sysadmin* in a certain section of the `/etc/samba/smb.conf`, you would do so by entering the group name as *@sysadmin*.

File Permissions

File Permissions define the explicit rights a computer or user has to a particular directory, file, or set of files. Such permissions may be defined by editing the `/etc/samba/smb.conf` file and specifying the explicit permissions of a defined file share.

For example, if you have defined a Samba share called *share* and wish to give *read-only* permissions to the group of users known as *qa*, but wanted to allow writing to the share by the group called *sysadmin* and the user named *vincent*, then you could edit the `/etc/samba/smb.conf` file, and add the following entries under the *[share]* entry:

```
    read list = @qa
    write list = @sysadmin, vincent
```

Another possible Samba permission is to declare *administrative* permissions to a particular shared resource. Users having administrative permissions may read, write, or modify any information contained in the resource the user has been given explicit administrative permissions to.

For example, if you wanted to give the user *melissa* administrative permissions to the *share* example, you would edit the `/etc/samba/smb.conf` file, and add the following line under the *[share]* entry:

```
    admin users = melissa
```

After editing `/etc/samba/smb.conf`, restart Samba for the changes to take effect:

```
sudo restart smbd
sudo restart nmbd
```

For the *read list* and *write list* to work the Samba security mode must *not* be set to *security = share*

Now that Samba has been configured to limit which groups have access to the shared directory, the filesystem permissions need to be updated.

Traditional Linux file permissions do not map well to Windows NT Access Control Lists (ACLs). Fortunately POSIX ACLs are available on Ubuntu servers providing more fine grained control. For example, to enable ACLs on `/srv` an EXT3 filesystem, edit `/etc/fstab` adding the *acl* option:

```
UUID=66bcdd2e-8861-4fb0-b7e4-e61c569fe17d /srv  ext3    noatime,relatime,acl 0       1
```

Then remount the partition:

```
sudo mount -v -o remount /srv
```

The above example assumes `/srv` on a separate partition. If `/srv`, or wherever you have configured your share path, is part of the `/` partition a reboot may be required.

To match the Samba configuration above the *sysadmin* group will be given read, write, and execute permissions to `/srv/samba/share`, the *qa* group will be given read and execute permissions, and the files will be owned by the username *melissa*. Enter the following in a terminal:

```
sudo chown -R melissa /srv/samba/share/
sudo chgrp -R sysadmin /srv/samba/share/
sudo setfacl -R -m g:qa:rx /srv/samba/share/
```

The *setfacl* command above gives *execute* permissions to all files in the `/srv/samba/share` directory, which you may or may not want.

Now from a Windows client you should notice the new file permissions are implemented. See the *acl* and *setfacl* man pages for more information on POSIX ACLs.

Samba AppArmor Profile

Ubuntu comes with the *AppArmor* security module, which provides mandatory access controls. The default AppArmor profile for Samba will need to be adapted to your configuration. For more details on using AppArmor see AppArmor.

There are default AppArmor profiles for `/usr/sbin/smbd` and `/usr/sbin/nmbd`, the Samba daemon binaries, as part of the *apparmor-profiles* packages. To install the package, from a terminal prompt enter:

```
sudo apt-get install apparmor-profiles apparmor-utils
```

This package contains profiles for several other binaries.

By default the profiles for *smbd* and *nmbd* are in *complain* mode allowing Samba to work without modifying the profile, and only logging errors. To place the *smbd* profile into *enforce* mode, and have Samba work as expected, the profile will need to be modified to reflect any directories that are shared.

Edit `/etc/apparmor.d/usr.sbin.smbd` adding information for *[share]* from the file server example:

```
  /srv/samba/share/ r,
  /srv/samba/share/** rwkix,
```

Now place the profile into *enforce* and reload it:

```
sudo aa-enforce /usr/sbin/smbd
cat /etc/apparmor.d/usr.sbin.smbd | sudo apparmor_parser -r
```

You should now be able to read, write, and execute files in the shared directory as normal, and the *smbd* binary will have access to only the configured files and directories. Be sure to add entries for each directory you configure Samba to share. Also, any errors will be logged to `/var/log/syslog`.

Resources

1. For in depth Samba configurations see the Samba HOWTO Collection
2. The guide is also available in printed format.
3. O'Reilly's Using Samba is also a good reference.
4. Chapter 18 of the Samba HOWTO Collection is devoted to security.
5. For more information on Samba and ACLs see the Samba ACLs page .
6. The Ubuntu Wiki Samba page.