Mac OS X Problems with SMB, OpenLDAP and 802.1x Implementations

Mac OS X Problems with SMB Implementation

One serious flaw I find most annoying about MAC OS X operating system is the access of Windows Shares via Samba. What is Samba?

"Samba is an Open Source/Free Software suite that provides seamless file and print services to SMB/CIFS clients. Samba is freely available, unlike other SMB/CIFS implementations, and allows for interoperability between Linux/Unix servers and Windows-based clients.” (Quoted from Samba Web Site)

Now saying that, Samba access to Windows 2000 and Windows 2003 works fine for Linux / Unix Distributions provided they are relative up to date without modification of the Windows Based Server.

Now Mac OS X is base of FreeBSD and of Mac OS X 10.4.8, also uses Samba to access Windows Share. But the Samba on Mac OS X is some what lacking. Samba on the Macs does not support signing requirements which are required by Windows Server machines which are on by default. You would think Apple would update their Samba on the MAC OS X considering Samba is Open Source and works perfectly fine on the other platforms.

You can verify this on the Mac system my running a terminal and run this command.

```
tail -f /var/log/system.log
```

Then run finder, and click connect to server

```
smb://windows.server.name/sharename
```

You will see in the log that this feature is not supported in Mac OSX Samba implementation.

To allow Mac OS X machines to access Windows Shares via Samba you have to disable Digitally sign communications (always) / Digitally encrypt or sign secure channel data (always) which can be found in

Administrative Tools

Security Settings -> Local Policies -> Security Options

Digitally sign communications (always) -> Disable
Digitally encrypt or sign secure channel data (always) -> Disable

Ashley Chew – Last Revision 22/03/2007
Make sure your policies are not overwritten by any other policies, and make sure the services are restart or simply reboot the machine.

Alternatively you can use regedit and edit the following keys.

For Digitally sign communications (always)

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanWorkstation\Parameters\RequireSecuritySignature

Which are 0 for false and 1 for true

Similarly for Digitally sign communications (always)

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanManServer\Parameters\RequireSecuritySignature

Which are 0 for false and 1 for true

That’s basically it, if you use finder and click connect to server. You should able to see that it no longer says the message in the log file saying its not supported. And it should be successfully authenticated and connected to the share.

But you shouldn’t have to do this since it supported in Samba and its functional implemented in Linux like Fedora so makes you wonder.

Mac OS X Problems with OpenLDAP implementation

Another gripe I have is with the Mac OS X is with their server version. For one thing their OpenLDAP server is based of the Open Source OpenLDAP server, again this is somewhat behind.

In a mixed based system ie in this case lets say Unix, Linux and Macs OSX. You wanted to share a common authentication mechanism which traditionally is now LDAP orientated. You would want a encryption method which is understood by all system

Typically the most common encryption method is crypt, md5 and sha. I would of chosen md5 encryption for password as its pretty much standard on all platforms you would think which is better then crypt, sha encryption is a good alternative but is used mostly in ldap system but is not understood on all platforms. But guess what md5 passwords are not supported by Mac OS X server, although its supported in the OpenLDAP which Mac OS X server is based on. So I had to revert back to crypt encryption method much to get a common login mechanism.

Ashley Chew – Last Revision 22/03/2007
Not to mention I think the Mac OS X OpenLDAP server is somewhat broken, occasionally the Mac OSX LDAP Server will refuse to start up although you try to restart it manually. I thought it was a once of thing but that is not the case, it even happened when I was at an Apple Training coarse with their default image and I asked why this happened. The reply I got was to reboot the machine (Isn’t that a Windows standard reply?). For some reason it again didn’t work then subsequent reboot it finally worked.

For the record I’ve decommissioned the Mac OS X Server, as there seems to be not as stable as I would like which include problems like the OpenLDAP Stabilty, NFS reshare over AFP (Performance Issue) etc and replaced it with a generic LDAP server. I’ve lost the ability to use Apple Management Utilities to manage individual accounts, but I’ve got better stability and performance by cutting out the middle Mac OS X Server.

### Mac OS X Problems with 802.1x Implementation

Now the other thing I have a serious problem is the 802.1.x or rather known as port locking via digital certificates authenticated by a radius server implementation on a Mac OS X system. So network access will be given to that port until the machine has authenticated itself.

Don’t get me wrong the 802.1x does work on a Mac OS X, buts its implemented on user level and not on a system level. When I say user level, it means the user has to log on first then authenticated themselves again the radius server. At a system level the machine will start a service to authenticate the machine to the radius server which will allow network access.

People may not see a distinction but it is a very important distinction. Let say you have a couple of hundred Mac OS X system which you want to secure your network with 802.1x so not just anyone can plug a machine into the network. Now because of a user level implementation of 802.1x, the Mac OS X system is basically isolated, the user has to log in then authenticate to the radius server to get network access for the machine. But the machine has no network, so how is a user going to authenticate to central authentication system such as LDAP if has no network. It’s the chicken and egg problem again. Unless you want to replicate all your accounts individually on the machines but defeats the purpose of a common authentication mechanism

Now this is solved by implementing 802.1x at a system level, where the system boots up and the machine though a service authenticate against the radius server which grants network access to the machine instead to a user. Talk about short sightedness on Apple’s behalf.

### Other OS X Problems

I’ve noticed other problems on the Mac OS X but I’ll not mention them as they do not really warrant any real mention as I don’t think they are that serious yet?