INTRODUCING SAMBA 4 NOW, EVEN MORE AWESOMENESS

Samba 4 has been in development for a long time but an official first release is imminent, the developers say. Its biggest feature is Active Directory Server support, which removes the last hurdle in a pure-Linux server set-up, with only Windows PCs and Macs as clients. In this FOSS for Windows special issue of LFY, let's explore how to set up Active Directory Server, so that you can finally phase out those pesky Windows Servers while keeping Windows desktops and Macs intact.

If you have ever used Windows and Linux together on a network, I am betting my entire wallet (not that there's anything in it at the moment, but that's beside the point) that you have used Samba. If you haven't (or have, without knowing it), Samba is the software that lets you access Windows network shares from Linux. Not just that Samba's built-in first-class server lets you create Windows-accessible shares on your Linux box.

Samba 3 can also host a Windows NT 4-compatible Domain set-up, and can act as a PDC (primary domain controller) as well as a BDC (back-up domain controller). But who uses NT 4 Domains any more? Ever since Windows 2000 Server was released at the end of 1999, everyone's been using Active Directory (AD) to organise clients and hold user and computer data. And why not? AD is based on industry standards Kerberos 5 for authentication, LDAP for the resource database, and all of it glued together with DNS SRV records. So why did it take 13 years for Samba to finally support AD?

Proprietary extensions to LDAP and Kerberos 5, that's why. But those extensions

have now become open standards, and the AD LDAP schema has been reverse-engineered, so Samba can finally start acting as an Active Directory Server (ADS). You can now convert all your Windows Servers to Linux, without changing a line of code (Samba 4 can now do AD, and Mono can do ASP.NET, which are actually the only two reasons people use Windows Servers anyway).

Getting started

First of all, Samba 4 is still in development, but also stable enough for Canonical to include it in the official repositories for Ubuntu 12.04 LTS (Long-Term Support release). With the light testing I have been doing at home, it works pretty well, so if you are going to think about rolling this out across your corporate infrastructure, you could try adding a couple of Samba boxes as backup AD servers, and see how they handle the load.

You have probably realised by now that Samba 4 is currently available only on Ubuntu 12.04. At least, a sufficiently updated non-alpha version that works even though the packaging is broken (which we can fix, so there's no need to worry). I have a hunch this package will never be updated, but I also believe that once official releases are out, there will be a PPA providing packages for this.

Before we start, we need a box that has Ubuntu 12.04 LTS installed. Samba 3 should NOT be installed (the configuration isn't compatible, so if it has already been installed, remember your configuration and purge all Samba 3 packages).

Setting up Samba 4

I am assuming your DHCP server is set up and working fine. The DHCP server may be on a different machine, but you must be able to edit your DHCP configuration. More appropriately, you should be able to make your DHCP server point to custom DNS servers, so if you re doing this at home and thinking about using the DHCP server on your ADSL modem or Wi-Fi router, that won't work.

You will have to set up a DHCP box (you can install the DHCP server on this machine itself), and set it up as the gateway to the Internet.
As for a DNS server, you can have any DNS servers set up anywhere you want, but they must be able to forward DNS requests to this Samba box.
I am assuming that the Samba box has an IP address of 192.168.100.1. Replace this with your own IP address (and of course, it goes without saying: the Samba box must have a static IP).
Anyway, let's move on to…

Installation

Start by installing all required packages, as follows:

$ sudo apt-get install samba4 samba4-clients krb5-user bind9 attr ntp

This command will spew out a huge error message, so run it again to make sure all packages (other than samba4) are installed properly. Samba 4 is actually fully installed, but the default configuration creation script is broken. This is good, because Samba 4 configuration is pretty involved, and we want to be able to do it from scratch. However, as far as dpkg is concerned, the installation failed, and we need to fix that. So we will tell dpkg that samba4 is installed and fine, by opening up /var/lib/dpkg/status, searching for the string Package: samba4, and next to it, replacing the word half-configured with installed. Then we wil actually configure Samba 4 by running the following commands:

$ sudo rm /etc/samba/smb.conf
$ sudo /usr/share/samba/setup/provision –realm=network.local –domain=NETWORK –adminpass='aBcD1@3–server-role=dc

This sets up the AD, so there re a couple of things to note here. The realm is the Kerberos realm, and is generally the full domain name, including the suffix, of

your network. The domain is the WINS domain, equivalent to the Workgroup name, and is typically the first name of the domain in upper-case. The adminpass is the password for the administrator user (yes, in Samba, administrator starts with lower-case a, and that's what you need to use when logging in on Windows). The server role is dc, which sets up an AD(S) for you.
Watch the output of the provision command carefully. Near the beginning of the output, it will try and detect which IP address to use. If it detects the IP address of the wrong interface (one that you don't want to serve on), we will have to change this. Make a note of the wrong IP address it detected we will need that later.
Configuration is far from over, but we can now test if Samba 4 is actually working. Run:

$ sudo service samba4 start
$ sudo smbclient -L localhost -U%

The last command should spit out the following:

Sharename Type Comment
——— —- ———-
netlogon Disk
sysvol Disk
IPC$ IPC IPC Service

Congratulations, we ve installed
BIND.
Setting up DNS
We re in luck he
All we need to do is to tell BIND to include in its configuration a certain file, and

we ll be done.

Do that by running the commands below:

```
$ samba-tool dns update 192.168.100.1 network.local server A 192.168.24.129
192.168.100.1
$ samba-tool dns update 192.168.100.1 network.local @ A 192.168.24.129
192.168.100.1
```

Here, 192.168.100.1 is the IP address of the Samba box, server is
of the box, and 192.168.24.129 is the wrong IP that was detected so adjust tho
accordingly. You will need the administrator password for these commands, but you
don t need to run them as the root user.

Then open up /etc/bind/named.conf, and add the following line:

```
include “/var/lib/samba/private/named.conf”;
```

Open /etc/bind/named.conf.options and add the line given below inside the braces:

```
tkey-gssapi-keytab “/var/lib/samba/private/dns.keytab”;
```

If you really have to use AppArmor, you need to add the following rules to
/etc/apparmor.d/usr.sbin.named inside the braces, anywhere towards the end (if
you are using a 64-bit build of Ubuntu, replace i386 with x86_64):

```
/var/lib/samba/private/** rkw,
/var/lib/samba/private/dns/** rkw,
/usr/lib/i386-linux-gnu/samba/bind9/** rm,
/usr/lib/i386-linux-gnu/samba/gensec/** rm,
/usr/lib/i386-linux-gnu/ldb/modules/ldb/** rm,
/usr/lib/i386-linux-gnu/samba/ldb/** rm,
```

Now do the following:

$ sudo service apparmor restart
$ sudo service bind9 restart

Our DNS infrastructure is now running. You can run the following lines:

$ dig SRV _ldap._tcp.network.local
$ dig SRV _kerberos._tcp.network.local

And both should return an answer section. If they do, congratulations this DNS set-up works.

Final touches

Start by testing Kerberos functionality. Type in:

$ kinit administrator@NETWORK.LOCAL

The realm needs to be typed in upper-case. If it works, it should tell you something about the ticket expiring after a particular number of days (I was informed that mine would expire in 41 days). Also, if you type in klist -e, it will list the Kerberos ticket you have received.

We also installed an NTP server, because Kerberos is very fussy about time, and you can use the Samba box as a time server to let your clients sync their clocks with the AD server. It should be configured to use the NTP Pool Servers, so it doesn't need any configuration.

We are now ready to join our first computer to the AD.

Joining the domain

When you are joining the domain from a Windows box, type the domain name in upper-case (like we did with the kinit command). When asked for a username and

password, type in administrator as the username, along with the administrator password. Reboot the computer and log in to the domain by typing NETWORK\administrator (replace NETWORK with your own WINS domain and make sure it is all caps) as the username, and the administrator password.

Administering the server

Samba 4 doesn't come with any administration tools, so how do you add, remove or modify users, computers, OUs, forests, etc? Well, you simply use Windows Servers official tools. Load up a client with the remote admin tools for Windows Server. If you are using Windows XP, install the Windows Server 2003 Service Pack 2 Administration Tools Pack, available at http://bit.ly/PjvjSG.

If you are using Windows Vista (you must run at least Service Pack 1), install the RSAT (Remote Server Administration Tools) for Windows Vista, available at http://bit.ly/LtEdgk.

If you are using Windows 7, install the RSAT for Windows 7 SP1 (you don't need SP1 to actually install the RSAT) from http://bit.ly/NcnwmR.

Be warned that while the 2003 and Vista tools weigh in at around 15-30 MB, the Windows 7 RSAT is a whopping 240 MB download per architecture.

With Windows 7, just installing the RSAT won't do you have to activate it by heading to Control Panel, searching for Turn On Windows Features, opening the applet and activating the entire Remote Server Administration Tools For Windows 7 group, and then rebooting the PC. It's almost as if using Windows 7 is a crime.

Once the remote admin tools are installed, you can use the classic DSA.MSC to administer the Samba 4 server just as if it were a Windows Server. You can create forests, add backup servers to the forests (which may be more Samba servers, or Windows Servers our set-up works with Windows Server 2008 R2 and should work just dandy with Windows Server 2012, when it comes out), and just generally

do anything you wish without limitations\ it’s fully featured.

Samba 4 also supports Group Policy out-of-the-box, so you can launch GPEDIT.MSC and edit group policy to your heart’s content.

Roaming profiles with Windows 7

If you want to support Windows 7 Roaming Profiles, you need to create the □profiles□ share and do some tinkering with DSA.MSC. First, open up /etc/samba/smb.conf (on the Samba box) and include the following lines:

[profiles]

path = /var/samba/profiles

read only = no

Create the profile directory, as follows:

$ sudo mkdir /var/samba/profiles

Then, start up DSA.MSC, select all the users you want to add roaming profile support to, right-click the selection and click Properties. In the Profile tab, locate the Profile Path input field, and type:

\\server.network.local\profiles\%USERNAME%

Replace server.network.local with your Samba box □s DNS name. A

So what’s next?

Well, play around with it. Remember, you can also add Macs to an AD set-up. Mac OS X has built-in support for AD. Linux computers can also join AD you can use either Centrify or Likewise Open to do the job.

Source : http://www.opensourceforu.com/2013/03/introducing-samba-4-now-even-more-awesomeness/