Active Directory Setup
IN719 Systems Administration

Introduction

Active Directory Domain Services (AD DS) is, at its heart, a database that stores information about users, groups, computers, and other entities in a network. It's a complex and powerful tool for managing networks and we will only get a small taste of its functionality in this paper. We're primarily interested in it because it will provide us with a centralised database of users. You already have some experience with this. Have you noticed that you can log into a variety of services on campus with the same user name and password? That's AD DS. On the other hand, have you been annoyed by the fact that you have to supply the same username and password multiple times? AD DS could probably solve that, but it's complicated.

1 The problem

Today our goal is to explore the idea of directory services using AD as an example, before configuring our AD DS Domains and add a few users to it. We'll also crack open PowerShell briefly to see a little bit about how it works with AD DS.

2 Explore AD DS

Most organisations, including OP, are managed using central directory services. Of course there are a wide range of other solutions out there, but Active Directory has become something of a de-facto standard, so our systems need to work with it, whether they run Linux, Windows or Mac OS X.

Now, to get a better understanding of what those services provide, let's have a look at how we can access and explore OPs Active Directory. On the I drive directory, you should find the tool AD Explorer, which we will use.

Download it to your machine and run it. It will ask you to decide which server to connect to, and which username and password to use. The credentials are unproblematic, but we need to figure out the names or IP addresses of the OP AD machines.

To do that, open your command line and start `nslookup` (which is default client of Windows). Next, run `set type=all`. This switches the DNS client to provide all DNS entries for a given name, not only specific ones like hosts (Type: A) or mail servers (Type: MX).

Now we need to query the OP DNS servers for directory services. Adapt the following command:

```
_ldap._tcp.dc._msdcs.DOMAIN
```

To make it work, you will need to replace `DOMAIN` with OP's fully-qualified domain.

What does that query do? Well, it looks for all servers that support LDAP (Lightweight Directory Access Protocol – look it up!), domain controller (dc) facilities via TCP, and specifically Microsoft domain controller servers for the OP domain.

Once it works you should see a list of machines and entries. Active Directory servers are generally set up redundantly (especially in organisations of OP's size) – you can guess why. Pick the first given server hostname from the results and use it in AD Explorer, along with your regular OP username and password.

Now you will see the directory services schema of OP. You should see a lot of `CN=` and `OU=` prefixes. Look up LDAP's directory structure specification to see what they mean (Wikipedia is a surprisingly good resource here).

Browse your way through the directory structure on this server (use the first branch `DC=op,DC=ac,DC=nz`) and try to appreciate the power of directory services, shown in the sheer number of types of objects that can be managed in AD. What you actually see here are not the resource instances, but their classes, somewhat comparable to an SQL schema, as opposed to the actual content. Click on selected classes to see their properties on the right (e.g. Domain Controllers, Users, etc.). You can explore instances by doubleclicking, but it is easier by searching.

Use the search dialog (check the tool bar) to search for actual instances. You will need to specify classes you want to look for, the attributes you want to search on, and the pattern (e.g. name) to match against. A good class to play with is 'user', which allows you to look for your own an other user's properties (e.g. e-mail address, etc.). Doubleclicking on the result will lead you to the complete entry. Feel free to try other queries.

Explore the AD a bit. Remember this tool, in case you have to explore an AD instance that you don't directly control, e.g. to perform authentication. But we will revisit that later in the lab (and in the course), but let's first turn to setting up Active Directory ourselves.

3 Configure AD DS

Carry out the following steps:

1. Log into your ad server and open the server manager.
2. Click on the top right menu "Manage" and select "Add Roles and Features".
3. Follow the wizard, and perform a "Role-based or Feature-based Installation" on the current server.
4. Once you reach the point "Server Roles", select "Active Directory Domain Services" and confirm the selection all the way through the wizard to start the installation. It may take a while.
5. Once the installation is complete, click the little yellow alert triangle (on the top right) and click "Promote this server to a domain controller". Another configuration wizard will start.
6. Choose to add a new forest. Use "directory.op-bit.nz" as your domain name. Click next.
7. Set your functional levels to Windows Server 2012 R2. We don't need to maintain compatibility with any older servers.
8. Select the DNS server option. (We may not even need it, but just in case.)
9. Choose and note your restore password. Click next.
10. Don't worry about the DNS delegation. It's not a problem for us. Click next.
11. The default NetBIOS name is fine. Click next.
12. The default paths are ok, but note that this is where your AD DS data will be stored. That's important when we plan backups. Click next.

13. Use the "View Script" option to save a script that can be used for an unattended install if you need to restore your domain. This script is the kind of thing that should go in your source code repo.
14. Run the prerequisites check, you'll see some messages but your setup should pass. The click install. Your server will require a reboot. That's the kind of event that you should document on your wiki. Any thoughts why?

After the reboot:

1. Use the Active Directory Users and Computers tool to add domain accounts for everyone on your team. Make those users members of the Domain Admins group. Note that these new accounts are distinct from the local system accounts you created earlier.
2. Create a new Organizational Unit in your domain called "Web Users".
3. Add a new domain account for a user named Joe Bloggs inside the organizational unit.

N.B.: The purpose of the organizational unit is to hold the user information for users of the web application we'll set up later in the semester. We want an easy way to identify the users of our web app without including all of the other kinds of users in our domain.

Use ADExplorer to verify that it is setup properly. To connect, use the machine's IP address.

4 View our users with PowerShell

Of course, you can also use Active Directory services from applications and in scripting environments. Let's try it out.

Open PowerShell on the server and run the following command:

```
Get-ADUser -Filter * -SearchBase "dc=directory,dc=micro-agents,dc=net"
```

This should produce a list of all the user accounts in your domain. Note especially the value of the `dc` (domain component) items in the *Distinguished name*. Now run the following:

```
Get-ADUser -Filter * -SearchBase "ou=Application Users,dc=directory,dc=op-bit,dc=nz"
```

(Note that this may not work at the current stage, but needs to be adapted to match your directory structure!)

What's the difference in the lists? Do you see why? Now experiment with other SearchBase strings. Can you write a query that returns only your user account? The purpose of this is to get a sense of the LDAP structure of AD DS. Ensure you understand what is going on here.