Unix Servers and MS AD Domains

“We don’t hurt Linux servers in Dorne”
- Oberyn Martell

“Everywhere in the world they hurt linux servers”
- Cersei Lannister

Network Prerequisites

The goal of this guide is to allow users of unix servers via SSH or TTY to authenticate using their Microsoft AD username and password. Additionally, the level of access that users get should be controlled by AD group.

Network DNS

For a successful multi-OS domain, DNS must be working perfectly. There are three main approaches to getting a working DNS setup. You will need to choose one.

The first is to simply use Windows server for DNS. This will allow the DNS records to be updated automatically, and will ensure the configuration is correct. A Unix-based DNS such as Bind or CoreDNS can be a backup in the scenario. DNS should be trivially enabled in the server manager on Windows.

The second is the allow Windows to control some portion of the DNS records.

The third is to create the DNS records that AD will expect to be there. There are a number of them. Here’s an incomplete list (good luck...):

Name	**Value**
_msdcs	NS <DNS server>
<ad server name>	A <ad server>
<ad server ip>	PTR <ad server name>
_gc._tcp.Default-First-Site-Name._sites	SRV 3268 <AD server>
_ldap._tcp.Default-First-Site-Name._sites	SRV 389 <AD server>
_kerberos._tcp.Default-First-Site-Name._sites	SRV 88 <AD server>
_gc._tcp	SRV 3268 <AD server>
_kerberos._tcp	SRV 88 <AD server>

_kpasswd._tcp	SRV 464 <AD server>
_ldap._tcp	SRV 389 <AD server>
_kerberos._udp	SRV 88 <AD server>
_kpasswd._udp	SRV 389 <AD server>
DomainDnsZones	A <DNS server>
_ldap._tcp.Default-First-Site-Name._sites.DomainDnsZones	SRV 389 <AD server>
_ldap._tcp.DomainDnsZones	SRV 389 <AD server>

And for the subdomain _msdcs.<domain> there's more:

_kerberos._tcp.Default-First-Site-Name._sites.dc	SRV 88 <AD server>
_ldap._tcp.Default-First-Site-Name._sites.dc	SRV 389 <AD server>
_ldap._tcp.Default-First-Site-Name._sites.gc	SRV 389 <AD server>
_kerberos._tcp.dc	SRV 88 <AD server>
_ldap._tcp.gc	SRV 389 <AD server>
_ldap._tcp.dc	SRV 389 <AD server>
_ldap._tcp.pdc	SRV 389 <AD server>
gc	A <domain controller ip>
<Domain controller invocation ID>*	CNAME <domain controller>
_ldap._tcp.<some long thing???>.domains	SRV 389 <AD server>

* This can change after restoring from backups. Find it from Powershell with >Get-AdDomainController

Other Requirements

- An AD account with domain admin privileges
- An AD group with at least 1 test user in it
- A username convention should be established to ensure that AD users do not overlap with existing Unix users
- If the system time drifts too much (~3 minutes) between the DC and the Unix server, all hell will break loose. To prevent this, you can use NTP.

Basic Process

There are a bunch of different tools that can be used to talk with a Windows AD server from Linux/BSD. Generally you will pick the set that is the easiest to install and configure on your distro.

The easiest method for joining a domain on Linux is realmd, a Red Hat project that can setup domain joining for you. Realmd uses either SSSD or Winbind under the hood to act as AD client. Sadly, RealmD sucks on Debian.

If realmd is not available, you must use either Winbind or SSSD manually. Generally, Winbind is better when using Ubuntu, otherwise try SSSD.

CentOS

Centos is easy. These steps should also work for Fedora.
Prerequisites

- The hostname of your machine must be the short version of its FQDN and should be unique
- Your hostname and FQDN should resolve to localhost on the server (edit /etc/hosts)

Install your packages

Verify

- The make sure that the domain can correctly be detected, run
  > realm discover <DOMAIN>

Connect

> realm join -v --user=<domain admin user> <DOMAIN>
will ask for password, configure

Test

- Be default, domain users will need to specify the domain in the username (e.g. joe@example.com)
- Try running > getent passwd <user>@<DOMAIN>
- If you do not want to specify the full domain, try editing sssd.conf

Ubuntu

Ubuntu doesn't have a nicely-packaged realmd like Centos does. For this reason, domain join must be done more manually with samba and winbind.

Prerequisites:

- The hostname of your machine must be the short version of its FQDN
- Time sync with ntp is a good idea. Things will break if the DC time is different than your server's

Install your packages

- `samba krb5-user winbind libpam-winbind libnss-winbind`
  - `You will need to enable Universe repo for this`
  - `sudo add-apt-repository universe`
- `Set your default realm (MY_DOMAIN_IN_CAPS.COM)`
- Enable winbind: `$ systemctl enable winbind`

Check `/etc/krb5.conf` for correctness

- Near the top of this file should be the line
  `default_realm = MY_DOMAIN_IN_CAPS.COM`

Configure Samba: /etc/samba/smb.conf

- The [global] section in the config file should be
  Workgroup = <DOMAIN - TLD (e.g. UMCSTLAB)>
  Realm = <DOMAIN (all caps)>
  Netbios name = <server's hostname>
  Security = ADS
  Dns forwarder = <dns server>

Add Winbind to the name service config /etc/nsswitch.conf

- At the end of the first three lines (passwd, group, shadow), add " winbind"

Obtain a kerberos ticket

- $ kinit <a domain admin ad user> # then type password
- $ klist # should show a ticket

Join the Domain

- $ net ads join -k

Test Domain join

- $ getent passwd <a domain user>

- This should show information about a domain user

Configure PAM

- Edit /etc/pam.d/common-session. Below the line `session required pam_unix.so`, add
  `session required pam_mkhomedir.so skel=/etc/skel/ umask=0022`

FreeBSD

Make sure the FQDN of the server resolves correctly

- Edit /etc/hosts
- Add the full FQDN of the server after the localhost names

Debian

Make sure the FQDN of the server resolves correctly

- Edit /etc/hosts
- Add the full FQDN of the server after the localhost names

Install required packages

- $ apt install realm samba-common

Test that the domain can be found

Join the domain

- $ realm join --user=<a domain admin> --install=/ example.com
- The --install option seems to be necessary on Debian, otherwise realm will complain that packages are missing even though they were installed as dependencies

Restart SSSD

- $ systemctl restart sssd
- $ systemctl enable sssd

Edit SSSD config file, /etc/sssd/sssd.conf, to set the proper username format and restrict logins to a group.

- Under the [domain/MYDOMAIN.COM] section, add the following
  ad_gpo_access_control = permissive # debian seems to need this
  enumerate = True # allows Linux to query for ad group membership

Troubleshooting

SSSD

Try changing the debug level for SSSD can checking the logs. The max level is 10.

- Edit /etc/sssd.conf
- Add to each section “debug_level = 10”
- Restart the service
- Check the logs

ADCLI / REALM

Joining domain failed: Couldn’t create computer account:
CN=<hostname>,CN=COMPUTERS,DC=...

- Try adding the computer to AD manually before you join with realm. From Powershell
  > New-ADComputer -Name <hostname> -path ‘CN=computers,dc=<xxxxx>,dc=<yyyyy>’

Winbind