Active Directory membership

We have implemented ADS domain member support in Samba 3.0.
In this talk I will describe some of the things we have learnt about implementing ADS support.

Please ask questions during the talk!

Joining an Active Directory domain

Joining an active directory domain consists of the following steps

□ find a ADS DC
□ made LDAP connection
□ kinit to get krb5 TGT for administrator
□ authenticate LDAP with SASL/GSSAPI/krb5
□ create machine account
□ set security descriptor on machine account
□ set password for machine principal

Finding a ADS DC

Samba implements two main methods for finding a ADS DC.

The first is to lookup the _ldap._tcp.REALM SRV records in DNS. These give the DNS names of any DCs for the domain. We found that it is important to sort the DNS replies to try to connect to a close DC if possible.

The second method is to use netbios lookups for 0x1b or 0x1c names. This should not be needed if DNS is correctly configured, but it is not uncommon that DNS is badly broken.

SASL/GSSAPI/Krb5 auth

LDAP connections to the ADS server need to be authenticated with SASL/GSSAPI/Krb5.

I found the default GSSAPI/SASL code in Cyrus-SASL quite troublesome. In order to make Samba ADS reliable and robust to poor DNS configurations I ended up re-implementing our own SASL code inside Samba. The main thing this gained was the ability to directly specify the kerberos principal to use in GSSAPI.

Creating the machine account

The machine account is an LDAP record of objectclass ’Computer’.
The core fields are:

- □userPrincipalName HOST/hostname@REALM
- □servicePrincipalName HOST/hostname
- □userAccountControl 0x1000 (UF_WORKSTATION_TRUST_ACCOUNT)

The userAccountControl is a bit tricky.

Setting the machine account SD

Machine accounts have a default security descriptor that doesn’t allow modification or deletion by the machine principal. This needs to be modified to allow the domain member to change its own password and to remove itself from the domain.

To change the security descriptor you fetch the hidden field ntSecurityDescriptor, decode it (it is in NDR format), modify it to allow modification and deletion by the owner, then put it back on the machine account.

Setting the machine password

The basic kerberos5 protocol doesn’t contain a mechanism for an adminsistrator to remotely set a users password. To solve this Microsoft extended to kpasswd protocol to allow the setting of a password for one principal using an authentication context of another principal. This is documented in draft-ietf-cat-kerb-chg-password-02.txt.

It is a fairly minor modification to the basic kpasswd protocol, and is implemented quite easily in terms of UDP packets and krb5_mk_priv().

SPNEGO

The major addition to the core SMB protocol is SPNEGO authentication. SPNEGO is an additional layer of security negotiation at the startup of a SMB session. The way it works is:

□the client sets the enhanced security flag

□the server provides a kerberos principal name in the negprot reply

□the client sends a session setup containing a kerberos ticket

See negTokenInit.dat and negTokenTarg.dat

ASN.1 and BER/DER

Most of the new protocols in ADS are based on ASN.1 and BER/DER. While ASN.1 has a reputation as a overly complex system it is really a reverse engineering dream. The on-the-wire encoding formats are self describing! This allows new parts of the protocol to be very quickly decoded. This saved me weeks of effort when implementing ADS in Samba. In particular, grab yourself a copy of dumpasn1.c (look in google).

Kerberos encoding types

Kerberos supports a number of "encoding types", which are encryption algorithms used to protect tickets and other data on the wire. The two most common encoding types are des-cbc-crc and des-cbc-md5 and these are supported by all Kerberos implementations. For ADS Microsoft added a new "type 23" encoding type that is based on the MD4 hashes used in SMB authentication.

If you are using MIT kerberos then you will need the latest patches or CVS version to support encoding type 23.

Winbind ADS

An important component of the NT4 domain support in Samba is the winbind daemon, which provide user/group services via NSS. For Samba 3.0 we now have an ADS backend for winbind that uses LDAP to provide user/group services from ADS via NSS.

The backends can also be mixed, allowing Samba to source user/group data from a mixture of trusted domains some of which are NT4 based and some ADS based.

Kerberos time synchronization

The kerberos protocol has a feature that clients must have their clocks tightly synchronized to the KDC or fetching the initial ticket will fail. While this is cryptographically a good idea it can be a great pain for users.

The trick to solve this problem involves noticing that a SMB negprot reply contains both the current time on the server and the servers timezone. This is much more useful than the NT ’net time’ command because ’net time’ needs authentication, but the authentication will fail if the time is not synchronised!

The Kerberos PAC

A lot of fuss has been made about the Kerberos PAC, mostly due to the early licensing on the Microsoft documentation for this data structure.

The PAC isn’t all that mysterious. It’s just a system to make ADS scale a little better, so that the DC doesn’t need to be contacted whenever a user logs into a server. The PAC is a data blob added to the ticket by the KDC that contains the information a server will need when it logs in a user, thus avoiding some network traffic. It is basically just a NDR encoded varient of the USER_INFO_3 structure embedded in the ticket supplied during SPNEGO/Krb5 authentication.

The domain flatname

The ’flatname’ of a domain is the old netbios name. Even for netbiosless ADS domains this is still needed in some parts of MS-RPC.

In Samba we determine the flatname via either fetching the ’flatname’ field of the trustedDomain LDAP record when contacting a trusted domain, or by using the servicePrincipalName fields of the domain controllers machine account record.

I hope to eventually completely remove the need to know the flatname in Samba, probably by using some newer varients of current MS-RPC requests.

Auto-configuration

To support auto-configuration Samba uses a query on the ldapServiceName field using a LDAP_SCOPE_BASE ldap query on the ADS DC. This can be performed before authentication allowing the code to auto-determine the Kerberos REALM directly from the DC.

A typical value for ldapServiceName would be vnet3.home.samba.org\

:win2000-vnet3$@VNET3.HOME.SAMBA.ORG

LDAP server name: win2000-vnet3
Realm: VNET3.HOME.SAMBA.ORG
Bind Path: dc=VNET3,dc=HOME,dc=SAMBA,dc=ORG

OpenLDAP problems

We struct two basic problems with using OpenLDAP for our LDAP client code. The first was lack of support for paged requests. Paged requests are essential for large ADS domains, as Microsoft LDAP servers by default are limited to a page size of 1000. Luckily it is relatively straightforward to construct LDAP paged requests from basic LDAP elements, so we were able to solve the problem quite quickly.

The second problem was with referrals. It seems that LDAP referrals generated by a Microsoft ADS server can confuse the OpenLDAP client code. To avoid this we disable referrals in the client code.

Dynamic DNS

Microsoft uses a new dynamic DNS authentication method called ’GSS-TSIG’. This uses GSSAPI/Krb5 to authenticate dynamic DNS requests to allow domain members to register themselves in the DNS domain. This extension is documented in draft-ietf-dnsext-gss-tsig-02.txt.

The client side of this extension doesn’t look too difficult, but is not yet implemented in any freely available code that I know of. I plan on implementing it shortly.