Samba4: War Stories

Andrew Bartlett
Samba Team / Red Hat

Samba Domains: Who would have thought?

- First Samba Domain work back in 1996
- Samba ntdom, TNG, 2.0
- Samba 2.2 makes domain logons production
- Samba 3.0 takes it seriously
  - But massive changes still take place
- Samba4: The Active Directory challenge

A new start

- Why not just Samba 3.0?
  - 'Active Directory'
  - Kerberos logins
  - Chance to ditch NTLM
  - Hoping for group policy
  - Provide a way 'out' when NT4 compatibility vanishes
  - New infrastructure for multiple protocols

Samba4: Where are we at?

- Past the 'rigged demo'
  - But real networks are more complex
- Domain logons work
  - Clients can join
  - KDC issues ticket with PACs
  - Clients login to a Kerberos environment
  - Similar 'user experience' to Samba3
  - No trusted domains, forests, poor group manipulations, etc...

War stories

- Tour of some of the fun and pain gluing Samba4 together
- LDB LDAP backend
- Kerberos KDC and Heimdal

Creating an LDAP Backend

LDB – Our internal database

- LDAP-like internal database
- Multiple Backends
  - TDB: a local shared memory file
  - LDAP: back onto a remote LDAP server
  - Testsuite tests between these for consistency
- Modules interface
  - Allows LDB requests to be modified before they are stored, or on searches
- Samba4 is tightly built around this database

LDB – LDAP

- LDB has always supported an LDAP backend
- But in Samba4, it assumes a very particular LDAP server: Samba4
- The challenge:
  - Use a standard LDAP server
  - But using it for the main database is a different matter

Why an LDAP backend?

- Reuse corporate LDAP servers
  - Avoid political minefield of 'competing with OpenLDAP' (etc)
  - Leverage these in the same way Samba3 has

It should be pretty easy

- It has always been an aim of LDB to allow switching backends
  - Trivial for 'standard' operations
  - Harder for AD-like operation
- Tested Samba4 against Samba4
  - It works
  - It gave me hope: this should not be too hard
  - It showed our basic LDAP client code worked

Almost LDAP

- AD is best described as 'almost, but not entirely unlike LDAP'
- Close enough to be tempting, but far enough away to be a lot of work
- Also the reason for this challenge: people want more than AD
  - They want a standards compliant server too.

1st challenge: Schema

- The AD schema is standards-like:
  - Some standard elements
  - Many extensions
  - Also conflicts, even in OIDs!

AD v Standard Schema

- Examples:
  - The 'top' schema has many extra permitted attributes
  - The 'person' schema doesn't require a 'sn' (surname) attribute
    - In AD, machines a people too!
  - The 'cn' attribute is single valued
  - Some attributes are declared as OIDs, but are actually objectClasses

2nd Challenge: Behaviours

- AD has data dependent syntax
  - Effectively aliases on values
  - SIDs can be either S-1-2-3 or the binary blobs
  - ObjectCategory can be either 'short' 'long' forms.
  - person
  - dc=Person,dc=schema,dc=configuration....

More interesting behaviours

- AD has 'Extended DNs'.
- These include the GUID and SID in the returned DN
- USNCreated and USNModified
- Per replica attributes, but visible to clients
- Indicates the global sequence number for creation and last change

3rd Challenge: OpenLDAP

- OpenLDAP 2.3 always requires schema checking
  - This required producing a 'better' schema
- Builtin schema
  - OL 2.3 also builds in certain schema elements (you can't replace them)
  - One of the OIDs is duplicated: Microsoft 'stole' the OID for MiddleName.
- Also enforced operational attributes
  - Pushed me to create the entryUUID module

Success

- First success with OpenLDAP 2.1
  - With schema checking off
- Now functional with OpenLDAP 2.3
  - We have a Samba schema
  - We also add 'extensibleObject' to every record
- Next move is to Fedora DS
  - Having trouble determining minimum schema
- Total Time: about 2 months

Future

- AD Migrations
  - Samba4 and this work could allow migration from AD
  - Replace the Fedora DS 'winSync' module?

Kerberos Challenges

Almost Kerberos

- Microsoft added the PAC
- New GSSAPI flags
  - DCE_STYLE GSSAPI
  - Extra GSSAPI/Krb5 leg

Creating the PAC

- Embedded User and Group Information
  - Included in each Kerberos ticket
  - Required for AD Domain logon
- Format documented by Microsoft
  - After much fuss
  - Many fields 'reserved'
- Signed by the KDC
  - To prevent spoofing
  - Signing also documented, but less clearly

How we tackled the PAC

- Client first
  - In this case, the member server
- Capture Samples
  - Test network, and 'public' passwords
  - Grab keys using SamSync
- Bytewise matching
  - Ensure we can encode/decode byte-for-byte
  - Sign with test network keys
  - Validate with test network keys

Stealing the Keys

- Using SamSync to get secrets
- Obtain the krbtgt and host keys
  - Uses the fact that arcfour-hmac-md5 is the 'best' cryptotype

PAC Formats

- Steps to fully determine the format
  - Read the MS Doc carefully
  - Grab an example PAC from Windows 2003
  - Refine parser
    - Part format matches part of netlogon
    - Pretend the rest is IDL, when it really isn't
      - And a bit of hand-marshaling
  - Ignore Signature

PAC Signatures

- A signature must be validated to be any use..
- Signature over the PAC, with zero'ed out signatures:
  - But the key type is not zeroed
  - Key usage 'other'
- How do you find the signatures to zero them?
  - We assume fixed offsets from the rear of the packet
  - Probably should parse buffer level separate to NDR

Pointers, padding matters

- In cryptographic challenges, don't let things vary:
  - Padding
  - Pointers
- Reworked IDL and PIDL to make these match exactly
- Wrote **standalone** smbtorture test:
  - parse sample PAC
  - generate PAC
  - sign PAC (using known keys for the PAC parsed)
  - compare buffer: should be byte exact

The remaining challenge

- We create, sign and validate the PAC
- But:
  - Win2003 and WinXP still didn't accept the PAC
  - Looked into MITM and mimic attacks to further attack the problem.
  - Finally one more timestamp had to match.

Kerberos Implementation

- We use a fork of Heimdal Kerberos
  - Compiled as part of Samba4
  - Synchronized regularly
    - Currently behind, due to an upstream restructure

Kerberos Hopes

- PK-INIT is one of my big hopes
- Killing NTLM is the other
- Ideal password and token system
  - One password/token
  - No matter the platform
  - Consistent groups, policy etc
  - Not just syncronised systems.