Advances in the Samba Testsuite

Andrew Tridgell
Samba Team

tridge@samba.org

In last years tutorial ...

- Last year I introduced the Samba4 test suite. At the time it provided the following:
  - Wide coverage of core file sharing operations
  - dual-server randomised testing
  - special purpose tests for mangling and locking
- Many of those tests have been improved or expanded. Update now!

The year of RPC testing

- This year the main focus has been on RPC tests
  - Good coverage of the most important RPC pipes
  - RPC scanners and diagnostic tools
  - tools for developing new IDL files
- The new tests build on the existing smbtorture test tool, but adds lots of RPC functionality
  - new RPC code now IDL based
  - new IDL compiler, with extensions to aid in building test code

smbtorture RPC syntax

- smbtorture for RPC tests is used like this:
  - smbtorture [binding_string] [options] [test_name]
  - For example, to test SAMR functions on the server MYSERVER using the RPC over SMB transport you would use:
    - smbtorture ncacn_np:MYSERVER -Uuser%pass RPC-SAMR
- Use the option -h for a list of tests

Binding strings

- RPC binding strings are used to specify a transport, a server and optionally a set of options
- Currently we support two transports:
  - ncacn_np means RPC over SMB
  - ncacn_ip_tcp means RPC over TCP
- For the server name you can use a hostname, a netbios name or an IP address

Binding string options

- At the end of a binding string you can provide a set of options. For example:
  - ncacn_ip_tcp:MYSERVER:[print,sign]
- Supported options include:
  - “print” means to print verbose decodings of all RPCs
  - “sign” means to use RPC signing
  - “seal” means to use RPC sealing
  - “bigendian” means to send big-endian RPCs
  - “validate” means to run additional tests on generated NDR

RPC-MGMT test

- The MGMT rpc pipe is a core part of DCE/RPC
  - provides statistics and enquiry functions on RPC pipes
- the RPC-MGMT test loops over all pipes that Samba4 knows about, and runs every MGMT RPC call on each pipe
- Very useful for finding what interfaces that are available on a server
  - also useful for finding what security options are available on a pipe

RPC-MGMT example

```
Testing pipe 'samr'
        server is listening
        server refused to stop listening - WERR_ACCESS_DENIED
        calls_in   9061  calls_out      0
        pkts_in   14624  pkts_out    9862
        principle name for proto 9 is 'DsRole'
        principle name for proto 10 is ''
        principle name for proto 16 is 'DsRole'
        principle name for proto 68 is ''
        uuid 12345778-1234-abcd-ef00-0123456789ab  version 0x00000000  'lsarpc'
        uuid c681d488-d850-11d0-8c52-00c04fd90f7e  version 0x00000001  'UNKNOWN'
        uuid 3919286a-b10c-11d0-9ba8-00c04fd92ef5  version 0x00000000  'lsads'
        uuid 12345778-1234-abcd-ef00-0123456789ac  version 0x00000001  'samr'
        uuid d335b8f6-cb31-11d0-b0f9-006097ba4e54  version 0x00050001  'UNKNOWN'
        uuid 98fe2c90-a542-11d0-a4ef-00a0c9062910  version 0x00000001  'UNKNOWN'
```

RPC-SCANNER

- The RPC-SCANNER test also uses the RPC MGMT interface to list interfaces, but then binds to each interface and scans to see how many RPC calls are available on the server
- This is useful to see if a server has more calls than you know about, which might mean new calls have been added

```
uuid 1ff70682-0a51-30e8-076d-740be8cee98b  version 0x00000001 'atsvc'
4 calls available
OK: matches num_calls in local IDL

uuid 12345778-1234-abcd-ef00-0123456789ac  version 0x00000001 'samr'
64 calls available
WARNING: local IDL defines 68 calls
```

RPC-EPMAPPER test

- The RPC-EPMAPPER test uses the RPC EPM calls to query the mappings available on a remote server
    - provides another method for finding out about interfaces offered by a remote server
- The EPM results also show what endpoints are available for each interface

RPC-SAMR test

- SAMR is one of the most important pipes in the CIFS world. It is the core of remote account management.
  - The RPC-SAMR test tries nearly every SAMR operation
  - test users, groups and aliases are created for testing write operations
  - read operations are tried on every user, group and alias in all domains reported by the server
- It is useful to run this test with the “print” option to see the details of every call

RPC-LSA test

- the RPC-LSA test is less complete than the RPC-SAMR test, but does cover the most important LSA functions
- Coverage includes:
  - mapping to/from SIDs and names
  - managing domains and domain information
  - managing privileges and secrets

RPC-NETLOGON

- The RPC-NETLOGON test is designed to test both netlogon functionality and BDC SAM synchronisation and delta calls
- The test will create a test machine name as a new BDC domain member to allow for BDC calls
- SAMR modification calls are used to trigger SAM database changes for individual deltas

RPC-SCHANNEL test

- The RPC-SCHANNEL test is designed to exercise the secure channel encryption used on some RPC pipes
- The test creates a new workstation and BDC domain member
- SCHANNEL varients tested include:
    - 64 and 128 bit schannel
    - signing and sealing
    - BDC and workstation

RPC-SPOOLSS test

- The RPC-SPOOLSS test uses the remote SPOOLSS RPC API to test printer enumeration and management
    - not as complete as some of the other tests
    - not all SPOOLSS functions have been encoded as IDL yet

Other RPC-* tests

- A number of tests have been written for other RPC pipes, but are not complete yet:
  - RPC-DFS - test distributed filesystem calls
  - RPC-WKSSVC - test workstation service calls
  - RPC-SRVSVC - test server service calls
  - RPC-ATSVC - test AT job scheduling calls
  - RPC-WINREG - test remote registry calls
  - RPC-EVENTLOG - remote event query
- While the tests are not complete, they are still likely to be useful if you are working on implementations of those pipes

The ECHO pipe

- For basic RPC tests, one of the most useful pipes is the “ECHO” pipe.
  - win32 source available at http://samba.org/ftp/unpacked/junkcode/rpcecho-win32/
  - also built into Samba4 server
- The echo pipe can test sending and receiving large amounts of data, with both signing and sealing
  - particularly useful for testing RPC fragmentation issues

big/little endian

- When working out the correct IDL for a new call, it is extremely useful to be able to see the call both in big and little endian formats
  - allows integer size and type to be easily spotted
  - reduces confusion over alignment
- I highly recommend getting hold of a Sun Sparc server, and installing the free “PC Netlink” server. That provides a big-endian server.
- Use the “bigendian” binding option to smbtorture to force sending of big-endian RPCs

pidl - an IDL compiler

- The basis of the new RPC code in Samba4 is the pidl IDL compiler
  - auto-generates client and server stubs
  - generates debugging functions for printing IDL structures
  - simple, portable design (written in perl)
- The compiler supports some new IDL syntax:
  - support for auto-setting of variables
  - support for relative pointers, and subcontexts
  - support for some types of non-NDR encodings

ndrdump

- When working on a new piece of IDL, it is useful to be able to test-parse some existing data with the new IDL.
- First save your NDR data to a file.
    - if you capture it with ethereal you can use the “Export Selected Packet Bytes” option
- Then run ndrdump, specifying the pipe, function and whether it is “in” or “out” data
- Fix any errors, recompile ndrdump and try again

How you can help

- If you or your company use the Samba test suite then please help!
  - send corrections to the IDL as you find them
  - send new IDL for missing functions or pipes
  - send new test code, or fixes to existing test code
- Thus far there have been almost no contributions to the Samba test suites from outside the Samba Team. I hope that will change.