How Samba was written

Autor:
Data de publicació: 22-12-2021

How Samba was written

Andrew Tridgell
August 2003

Method 1:

First off, there are a number of publicly available documents on the CIFS/SMB protocol. The documents are incomplete and in places rather inaccurate, but they are a very useful starting point. Perhaps the most useful document is "draft-leach-cifs-v1-spec-02.txt" from 1997 which is a protocol specification released by SNIA and authored primarily by Microsoft (with significant input from many other people, including myself). This document has expired as an IETF draft, and Microsoft has dropped their attempts to get CIFS accepted as an IETF standard, but the document is still available if you look hard enough with an internet search engine.

There are numerous other public specifications for various pieces of the protocol available. I maintain a collection of the ones I know about in http://samba.org/ftp/samba/specs/

Method 2:

I call this method the "French Cafe technique". Imagine you wanted to learn French, and there were no books, courses etc available to teach you. You might decide to learn by flying to France and sitting in a French Cafe and just listening to the conversations around you. You take copious notes on what the customers say to the waiter and what food arrives. That way you eventually learn the words for "bread", "coffee" etc.

We use the same technique to learn about protocol additions that Microsoft makes. We use a network sniffer to listen in on conversations between Microsoft clients and servers and over time we learn the "words" for "file size", "datestamp" as we observe what is sent for each query.

Now one problem with the "French Cafe" technique is that you can only learn words that the customers use. What if you want to learn other words? Say for example you want to learn to swear in French? You would try ordering something at the cafe, then stepping on the waiters toe or poking him in the eye when he gives you your order. As you are

being kicked out you take copious notes on the words he uses.

The equivalent of "swear words" in a network protocol are "error packets". When implementing Samba we need to know how to respond to error conditions. To work this out we write a program that deliberately accesses a file that doesn't exist, or uses a buffer that is too small or accesses a file we don't own. Then we watch what error code is returned for each condition, and take notes.

Method 3:

Method 3 is a greatly expanded variant of the "swear words" technique I have already mentioned. It involves writing something called a "protocol scanner". A protocol scanner is a program that tries all possible "words" in some section of a protocol and uses the response to automatically deduce new information about the protocol. It is like the French Cafe technique but with a very patient waiter.

For example, some section of the protocol might contain a 16 bit "command word" that tells the server what operation to perform. There are 64 thousand possible command words, so we try all of them and note which ones give an error code other than "not implemented". Then we need to work out how much supplementary data each command word needs, so the program tries 1 byte of blank data, then 2 bytes then 3 bytes etc until the server changes its response in some way. When the response changes then you know (with a fairly high level of confidence at least) that you are using the right quantity of data. You then try using non-blank data, putting in a filename or a directory name or a username until the server changes its response again. After a large number of tries the program eventually finds a combination of data that gives no error code at all - the server has accepted our request! We have just discovered a new phrase in "French".

Once the server has accepted the new request we need to work out what the request actually does. We know its a valid command, but what does it do? To determine that we send the new command then we follow it up with a series of already understood commands that ask the server for lots of detailed information about the files it has. Has a file size changed? Has a date changed? Has a file changed its name? Eventually we work out what the command does.

Method 4:

The final method that is worth describing here is the "differential" technique. This is used to discover interactions between different command words. Using the (now rather stretched) French Cafe analogy it is like trying to work out if you should use a different word for coffee if you are having it with a biscuit than if you are having it with cake. It goes like this.

You use your new knowledge of French to write a virtual waiter. A program that is supposed to behave like a real French waiter. Then you write another program that sends a random series of French phrases in turn to the real waiter and your virtual waiter. Your program then examines the replies carefully and notes any differences in how the two waiters respond. You keep careful notes.

When the two waiters respond differently then you look at your notes

and try the same sequence of phrases again, but this time leaving one of them out. Do the two waiters now behave in the same way? If they do then you know that phrase is critical to the difference between the two waiters, otherwise it isn't. In this way you can quickly determine the minimum set of phases that causes the two waiters to respond differently.

Once you have this minimal set then you stare at it hard and use the methods described earlier to see whats wrong with your virtual waiter. When you fix it you try again, and keep trying until your waiter behaves the same as the virtual waiter.

Now imagine using all of the above techniques (plus some other similar techniques I have not gone into here) over a period of 12 years. Thats how Samba was written.