About me

- I'm the Samba maintainer at Red Hat
- Samba Core Team member since 2010
About me

FOSS Hacker working on:

- Samba - The domain controller and file server
- libssh - The SSH Library
- cmocka - A unit testing framework for C
- cwrap - Client/Server testing made easy
- darktable - Image raw developer
- LineageOS - Android with Privacy Features
Abstract

Samba's command line user interface [...] and design principles [...] fade in and out of use according to some esoteric pattern.
The abstract was shamelessly stolen from Douglas's SambaXP talk in 2019:

What should we do with our user interface?
Flashback (2019)

Samba’s command line UI

- kind of haphazard
- patchy abstractions
- untested as a user interface
Flashback (2019)

Nobody can fix it

- experts are locked-in
- newbies are baffled
- old options can’t be dropped
I tried to fix it (as good as possible)
Agenda

- What are the current problems?
- How did we solve the issues?
- How do we prevent issues in future?
- What can still be improved?
- Has the documentation been updated?
What are the current problems?
A few examples ...
We have `-k` and `-k` yes. Same option with and without an argument.
We have -k and -k yes. Same option with and without an argument.
Editor or encryption?

1 $ ldbedit --help | grep '\-e'
2  \-e, \--editor=PROGRAM     external editor
3  \-e, \--encrypt           Encrypt connection for privacy

Will I enable encryption or will it open an editor, or both?
Editor or encryption?

```
1 $ ldbedit --help | grep '\-e'
2   -e, --editor=PROGRAM  external editor
3   -e, --encrypt          Encrypt connection for privacy
```

Will I enable encryption or will it open an editor, or both?
Editor or encryption?

Will I enable encryption or will it open an editor, or both?

```bash
$ ldbedit --help | grep '\-e'
2  -e, --editor=PROGRAM       external editor
3  -e, --encrypt              Encrypt connection for privacy
```
<table>
<thead>
<tr>
<th>No.</th>
<th>Command/Flag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>`$ ldbsearch --help</td>
<td>grep '-S'`</td>
</tr>
<tr>
<td>2</td>
<td><code>-S, --sorted</code></td>
<td>sort attributes</td>
</tr>
<tr>
<td>3</td>
<td><code>-S, --sign</code></td>
<td>Sign connection to prevent</td>
</tr>
<tr>
<td>4</td>
<td>`-S, --signing=on</td>
<td>off</td>
</tr>
</tbody>
</table>
Sorted or sign?

1 $ ldbsearch --help | grep '\-S'
2   -S, --sorted                 sort attributes
3   -S, --sign                   Sign connection to prevent
4   -S, --signing=on|off|required Set the client signing state
Sorted or sign?

1 $ ldbsearch --help | grep '\-S'
2 -S, --sorted                      sort attributes
3 -S, --sign                        Sign connection to prevent
4 -S, --signing=on|off|required     Set the client signing state
Sorted or sign?

1. $ ldbsearch --help | grep '\-S'
2. -S, --sorted sort attributes
3. -S, --sign Sign connection to prevent
4. -S, --signing=on|off|required Set the client signing state
## Scope or config file?

1. `$ ldbsearch --help | grep '\-s'`

2. `-s, --scope=SCOPE`  
   search scope

3. `-s, --configfile=CONFIGFILE`  
   Use alternative configuration

Will I set the scope or provide a config file with `-s`?
Scope or config file?

1 $ ldbsearch --help | grep '\-s'
2 -s, --scope=SCOPE               search scope
3 -s, --configfile=CONFIGFILE     Use alternative configuration

Will I set the scope or provide a config file with -s?
Scope or config file?

Will I set the scope or provide a config file with `-s`?

1. `$ ldbsearch --help | grep '\-s'`
2. `-s, --scope=SCOPE  search scope`
3. `-s, --configfile=CONFIGFILE  Use alternative configuration`
I want scope

$ ldbsearch --help | grep '-s'
-s, --scope=SCOPE search scope

• I want to set the scope, lets use the long option.
<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$ ldbsearch --help</td>
</tr>
<tr>
<td>2</td>
<td>-s, --scope=SCOPE</td>
</tr>
<tr>
<td>3</td>
<td>-i, --scope=SCOPE</td>
</tr>
</tbody>
</table>
Let's use `ldbsearch --scope`

1. `$ ldbsearch --help | grep '\-scope'`
2. `-s, --scope=SCOPE`  search scope
3. `-i, --scope=SCOPE`  Use this Netbios scope
Let's use `ldbsearch --scope`

1. `$ ldbsearch --help | grep '^--scope'`
2. `-s, --scope=SCOPE`  
   - search scope
3. `-i, --scope=SCOPE`  
   - Use this Netbios scope
Can we fix this?
Backwards compatibility dilemma

Fixing consistency across tools will create new problems!

• We need to introduce new options
• Complexity might increase
• User scripts will break
Another problem: Logging

Do Samba tools log to

- *Standard Output (stdout)* or
- *Standard Error (stderr)*

by default?
Logging

1 $ smbclient --help | grep 'std'
2   -E, --stderr
3       Write messages to stderr instead of stdout
4
5 $ smbd --help | grep 'std'
6   -S, --log-stdout
7       Log to stdout
Logging

1 $ smbclient --help | grep 'std'
2   -E, --stderr          Write messages to stderr instead of stdout
3
4
5 $ smbd --help | grep 'std'
6   -S, --log-stdout      Log to stdout
Problems discovered so far

- There is no consistency in logging to `stdout` or `stderr`
- We have several command line parsing implementations
- We have a lot of duplicate options
How did we solve the issues?
For tools written in C

- Rewrite of command line parser
- Only **one** cmdline parser implementation in C
- Uses the client credentials API for all tools
## New important common options (1)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`--use-kerberos=desired</td>
<td>required</td>
</tr>
<tr>
<td><code>--use-krb5-ccache=CCACHE</code></td>
<td>Credentials cache location for Kerberos</td>
</tr>
</tbody>
</table>
New important common options (1)

1. `--use-kerberos=desired|required|off`  Use Kerberos authentication
2. `--use-krb5-ccache=CCACHE`  Credentials cache location for Kerberos authentication
Corresponding smb.conf option

```
client use kerberos = desired|required|off
```
FIPS mode

- `client use kerberos = required` will be forced
- Beside Kerberos, SMB doesn't offer any other secure authentication methods in FIPS mode
What is FIPS?

- Standard for Security Requirements for Cryptographic Modules by the US government
- Issued by the National Institute of Standards and Technology (NIST)
What is FIPS 140-2 mean?

- Allows only a subset of "strong" crypto algorithms for security relevant functionality
What does FIPS 140-2 mean for Samba?

RC4 and MD5 are not available
1. NTLM doesn't work, only KRB5
2. SMB1 doesn't work
3. SMB guest shares don't work
Legacy support

- `-k` and `-k yes|no` are kept as legacy options for now
- They will be removed in future!
New important common options (2)

--client-protection=sign|encrypt|off
Configure used protection for client connections
Corresponding smb.conf option

```plaintext
client protection = sign|encrypt|off|default
```

- default will use the default options of client smb encrypt and client signing.
Client protection

• This option will do the right thing for you.
• Works for SMB and DCEPRC connections!
Removed common options

1. -e | --encrypt
2. -S, --signing=on|off|required  Set the client signing state

Use --client-protection now!
Removed common options

- e|--encrypt

- S, --signing=on|off|required  Set the client signing state

Use --client-protection now!
Logging

- All tools and daemons log to stderr by default now!
- Can be changed using `--debug-stdout` option.
1 $ smbd --log-stdout # has been removed
2 $ smbd --debug-stdout
smbd/winbindd/nmbd

1 $ smbd --log-stdout # has been removed
2 $ smbd --debug-stdout
Debugging memory (talloc)

All tools are providing:

1. `--leak-report` enable talloc leak reporting on exit
2. `--leak-report-full` enable full talloc leak reporting on exit
Debugging memory (talloc)

All tools are providing:

1. `--leak-report` enable talloc leak reporting on exit
2. `--leak-report-full` enable full talloc leak reporting on exit
Tool options changed

Example ldbmodify:

1. `-s` is not used for `--configfile` anymore
2. `-e` is not available for `--editor` anymore

ldb tools use: `-s | --scope`
Tool options changed

Example ldbmodify:

1. `-s` is not used for `--configfile` anymore
2. `-e` is not available for `--editor` anymore

`ldb tools use: -s|--scope`
Tool options changed

Check Samba 4.15 release notes for details!
Other issues solved

- New cmdline parser fixes Kerberos support for winexe!
- BUG: #14616
Big code cleanup

• This work is a big code cleanup
• Gets rid of a lot of global variables
3

How do we prevent issues in future?
Improved cmdline parser

- New cmdline parser has a sanity checker!
- Walks over all short and long options looking for duplicates.
Improved cmdline parser

$ net help

cmdline_sanity_checker: Duplicate option --long|-l detected!
Unit tests

- The new command cmdline parser has unit tests
- Sanity check is run in unit tests
What can still be improved?
Add tests to execute each binary

./bin/tool --help
Compare with manpage

- Compare --help output with manpages
Cross tool consistency

- Make sure certain options behave the same and are documented consistently.

-`v|--verbose`
-`q|--quiet`
Example for --verbose

- **v**
  
  2 × -v, --verbose  Be verbose
  1 × -v, --verbose  Print more details of checking
  1 × -v, --verbose  Print all DN pairs that have been compared
  1 × -v, --verbose  Show default options too
Example for --quiet

- `q
  2 × -q, --quiet` Be quiet
  1 × -q, --quiet` Do not print anything but relay on just exit code
  1 × -q, --quiet` don't print details of checking
Has the documentation been updated?
Yes, it has!
Doc update

• All existing manpages have been updated for the common commandline parser options!
Will we get shell-completion one day?
Maybe!
Questions?

Twitter: @cryptomilk

Blog: blog.cryptomilk.org