smbcmp

a handy network capture diff tool for SMB traffic

Aurélien Aptel <aaptel@suse.com>
SUSE

Who am I?

- Aurélien Aptel
- Work in SUSE, Samba Team
- Focus on SMB kernel client aka “cifs.ko”

What is this about?

- Different debugging approaches I use
- a new tool: smbcmp

- Mostly useful to developers
- But also for administrators, to diagnose network issues

Debugging is hard

- No silver bullet
- Some approach works better than others for certain bugs
- SMB bugs
  - In client?
  - In server?
  - Both?
  - Specifications wrong?
  - Unspecified?
- Lot of possible failures
  - Goal: isolate as much as possible before digging in

Different versions: git bisect

- Setup
  - Find “good” commit
  - Find “bad” commit
- Dichotomy
  - Tries to find first bad commit
  - Checkouts intermediaries commits you can test
  - Search space divided by 2 at each step
  - N commits → O(log N) steps to determine first bad commit
  - Really powerful: 130k commits in 17 steps
- Can be automated
  - Reproduce script
    - Indicate if “good” or “bad” via the exit code
  - git bisect run myscript.sh

Different implementations

- Sometime there are no good commits or its very impractical to find
- Try different combination of servers/clients
  - Windows, samba, smbclient, cifs.ko
- Try writing a test client that only does the buggy steps
  - Samba torture test framework
  - Pike (https://github.com/emc-isilon/pike)
    - Clean, pure-python, SMB2/3 lib, with easily tweakable fields
    - Used to test SMB3 POSIX extensions (https://github.com/aaptel/pike/commits/smb3unix)
  - Microsoft has open-sourced a massive testing framework
    - https://github.com/Microsoft/WindowsProtocolTestSuites

Logs

- Samba
  - smb.conf
    - Log level = 10
    - Smblog-mode for emacs :)

```
  Unlocking key 3E8F3BDA
[ 5 2016/02/17 19:52:42.635677 ../lib/dbwrap/dbwrap.c:146 dbwrap_lock_order_state_destructor]
  release lock order 1 for /tmp/smbstate/lock/smbXsrv_open_global.tdb
[10 2016/02/17 19:52:42.635681 ../lib/dbwrap/dbwrap.c:133 debug_lock_order]
  lock order:  1:<none> 2:<none> 3:<none>
[ 5 2016/02/17 19:52:42.635694 ../source3/smbd/files.c:554 file_free]
  freed files structure 1148222956 (3 used)
[10 2016/02/17 19:52:42.635701 ../source3/smbd/smb2_server.c:2823 smbd_smb2_request_done_ex]
  smbd_smb2_request_done_ex: idx[1] status[NT_STATUS_OK] body[60] dyn[no:0] at ../source3/smbd/smb2
[10 2016/02/17 19:52:42.635707 ../source3/smbd/smb2_server.c:908 smb2_set_operation_credit]
  smb2_set_operation_credit: requested 1, charge 1, granted 1, current possible/max 386/512, total
[10 2016/02/17 19:52:42.638936 ../source3/smbd/smb2_server.c:3683 smbd_smb2_io_handler]
  smbd_smb2_request idx[1] of 5 vectors
[10 2016/02/17 19:52:42.638959 ../source3/smbd/smb2_server.c:656 smb2_validate_sequence_number]
  smb2_validate_sequence_number: clearing id 334 (position 334) from bitmap
[10 2016/02/17 19:52:42.638965 ../source3/smbd/smb2_server.c:2139 smbd_smb2_request_dispatch]
  smbd_smb2_request_dispatch: opcode[SMB2_OP_CREATE] mid = 334
[ 4 2016/02/17 19:52:42.638974 ../source3/smbd/uid.c:384 change_to_user]
 Skipping user change - already user
[10 2016/02/17 19:52:42.638986 ../source3/smbd/smb2_create.c:502 smbd_smb2_create_send]
  smbd_smb2_create: name[test.txt]
[10 2016/02/17 19:52:42.639003 ../source3/smbd/smb2_create.c:930 smbd_smb2_create_send]
  smbd_smb2_create_send: open execution phase
[ 5 2016/02/17 19:52:42.639010 ../source3/smbd/filename.c:276 unix_convert]
  unix_convert called on file "test.txt"
[10 2016/02/17 19:52:42.639017 ../source3/smbd/statcache.c:283 stat_cache_lookup]
  stat_cache_lookup: lookup succeeded for name [TEST.TXT] -> [test.txt]
[10 2016/02/17 19:52:42.639027 ../source3/smbd/vfs.c:1160 check_reduced_name]
  check_reduced_name: check_reduced_name [test.txt] [/tmp/snapdir]
[10 2016/02/17 19:52:42.639036 ../source3/smbd/vfs.c:1220 check_reduced_name]
  check_reduced_name realpath [test.txt] -> [/tmp/snapdir/test.txt]
[ 5 2016/02/17 19:52:42.639041 ../source3/smbd/vfs.c:1307 check_reduced_name]
  check_reduced_name: test.txt reduced to /tmp/snapdir/test.txt
[10 2016/02/17 19:52:42.639046 ../source3/smbd/open.c:4977 create_file_default]
  create_file: access_mask = 0x10e0080 file_attributes = 0x0, share_access = 0x3, create_dispositic
[10 2016/02/17 19:52:42.639054 ../source3/smbd/open.c:4452 create_file_unixpath]
  create_file_unixpath: access_mask = 0x10e0080 file_attributes = 0x0, share_access = 0x3, create_c
[10 2016/02/17 19:52:42.639061 ../source3/smbd/open.c:4507 create_file_unixpath]
  create_file_unixpath: open on test.txt failed - SEC_FLAG_SYSTEM_SECURITY denied.
[10 2016/02/17 19:52:42.639066 ../source3/smbd/open.c:4796 create_file_unixpath]
  create_file_unixpath: NT_STATUS_PRIVILEGE_NOT_HELD
[10 2016/02/17 19:52:42.639076 ../source3/smbd/open.c:5074 create_file_default]
  create_file: NT_STATUS_PRIVILEGE_NOT_HELD
```

U:%*- *smblog: ~/prog/smblog-mode/log.smbd-10* 5% (2414,0) [(Smblog)]

```
{
        struct user_struct *vuser;
        int snum = SNUM(conn);

        if (!conn) {
                DEBUG(2,("Connectio
                return(False);
        }

        vuser = get_valid_user_stru

        if ((current_user.conn ==
                   (vuser != NULL)
                   (current_user.ut
                DEBUG(4,("Skipping
                         "user\n"))
                return(True);
        }

        if (vuser == NULL) {
                /* Invalid uid ser
                DEBUG(2,("Invalid
                         (unsigned

                return false;
        }

        return change_to_user_inter
}

static bool change_to_user_by_sessi

{
        SMB_ASSERT(conn != NULL);
        SMB_ASSERT(session_info !=

        if ((current_user.conn ==
            (current_user.ut.uid ==
                DEBUG(7, ("Skipping

                return true;
        }
```

-:--- uid.c 58% (389,27)

```
SMB2 request BREAK              ... response NT_STATUS_OK
SMB2 request CREATE             ... response NT_STATUS_OK
SMB2 request CLOSE              ... response NT_STATUS_OK
SMB2 request CREATE             ... response NT_STATUS_OK
SMB2 request CLOSE              ... response NT_STATUS_OK
SMB2 request CREATE             ... response NT_STATUS_OK
SMB2 request CLOSE              ... response NT_STATUS_OK
SMB2 request CLOSE              ... response NT_STATUS_OK
SMB2 request CREATE             ... response NT_STATUS_PRIVILEGE_NOT_HELD
SMB2 request CREATE             ... response NT_STATUS_OK
SMB2 request CLOSE              ... response NT_STATUS_OK
```

U:%*- *smblog-regs: ~/prog/smblog-mode/log.smbd-10* Top (9,0) [(Smblog Regs)]

Logs

- Samba
  - smb.conf
    - Log level = 10
    - Smblog-mode for emacs :)
- Kernel
  - ```
    echo 1 > /proc/fs/cifs/cifsFYI
    echo 8 > /proc/sys/kernel/printk
    echo 1 > /sys/module/dns_resolver/parameters/debug
    echo "module cifs +p" > /sys/kernel/debug/dynamic_debug/control
    echo 'file fs/cifs/* +p' > /sys/kernel/debug/dynamic_debug/control
    ```
  - ftrace / trace-cmd
    - Record call graph
      - https://jvns.ca/blog/2017/03/19/getting-started-with-ftrace/
    ```

Network capture

- Wire log
- When applicable, network trace analysis is very effective
- Wireshark!
  - smb||smb2||dns||krb4

Debugger

- Good tool but often impractical
- Breakpoints = timeouts
- Samba
  - Forks for user sessions
  - ```
    set follow-fork-mode child
    set detach-on-fork off
    ```
- Kernel
  - Qemu gdb server
  - `qemu … -s`
  - ```
    gdb -ex ‘add-auto-load-safe-path /’ \
        -ex ‘target remote :1234’ vmlinux
    ```

Debugger

- Python helper funcs in kernel.git
- Kernel cannot be compiled without optimization
  - Out of order execution
  - dreaded `<optimized out>`
  - Inline code
  - Since GCC v4.8 '-Og'
    “kernel hacking: GCC optimization for better debug experience (-Og)”
    - https://www.mail-archive.com/linux-kernel@vger.kernel.org/msg1707708.html

Code reading

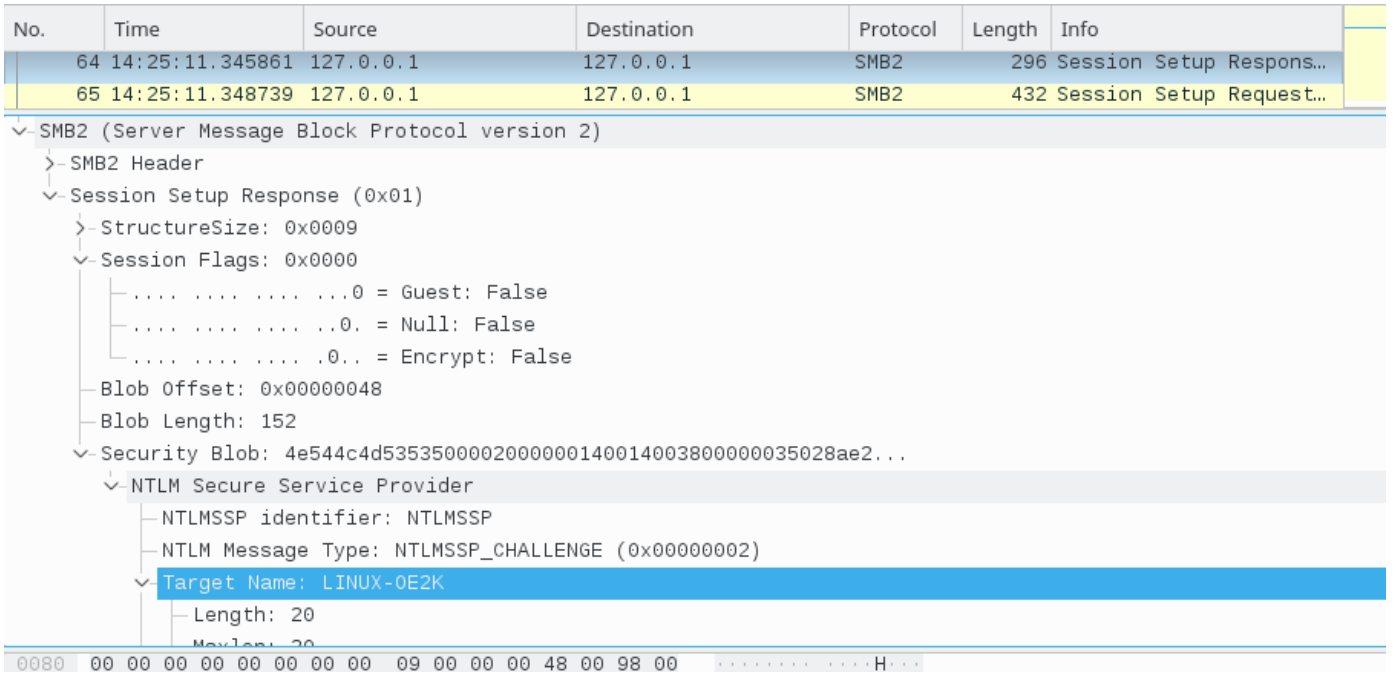
- The inevitable code/doc-reading part
  - Reading the spec one time to get an idea of how it’s supposed to work at the protocol layer
  - Finding the corresponding codepath
  - Reading source code of the relevant functions
  - Look for bug, typos, and wrong logic wrt the specs
  - Repeat
- Amount of code to grok can be very big
  - Long process, easy to miss the bug

Network capture comparison

- Get a trace of a working case
- Get a network trace of the issue
- Look hard at both traces
  - try to see what the good client/server is doing that the bad one doesn’t (or vice versa)
  - Compare packets, fields, etc

Comparing network traces

- Open both traces side by side
- Expand the little handles
- Lots of them...
  - Nested
    - Into
      - Each
        - other



Comparing network traces

- Eventually you switch to a different packet and the click-dance starts again
- Impractical for multiple reasons
  - Your index hurts
  - You skip expanding some fields because “it’s never going to be different here”
    - Until it does…
  - Your `l33t h4cker` eyes might just miss a difference
    - whitespace, caps, slash directions, flags..?
  - Some differences are false positives
    - Timestamps, random GUID, hashes, ...

Automating the comparison

- Wireshark is great…
- Would be nice to interact with it programatically
- API?
  - Not really :(
  - Tshark: text output
    - Also json and xml output
  - Also a daemon version sharkd
    - Undocumented?

tshark

```
tshark -r smb3-aes-128-ccm.pcap -Y smb2
    1 ... 10.160.64.139 → 10.160.65.202 SMB2 172 Negotiate Protocol Request
    2 ... 10.160.65.202 → 10.160.64.139 SMB2 318 Negotiate Protocol Response
    3 ... 10.160.64.139 → 10.160.65.202 SMB2 190 Session Setup Request, NTLMSSP_NEGOTIATE
    4 ... 10.160.65.202 → 10.160.64.139 SMB2 318 Session Setup Response, Error: STATUS_...
    5 ... 10.160.64.139 → 10.160.65.202 SMB2 430 Session Setup Request, NTLMSSP_AUTH, User:
SUSE\administrator
    6 ... 10.160.65.202 → 10.160.64.139 SMB2 142 Session Setup Response
...
```

tshark

```
tshark -r smb3-aes-128-ccm.pcap -Y smb2 -V
Frame 1: 172 bytes on wire (1376 bits), 172 bytes captured (1376 bits) on interface 0
    Interface id: 0 (unknown)
    Encapsulation type: Ethernet (1)
    Arrival Time: May 17, 2017 12:02:16.523633000 CEST
...
    [Protocols in frame: eth:ethertype:ip:tcp:nbss:smb2]
...
SMB2 (Server Message Block Protocol version 2)
    SMB2 Header
        Server Component: SMB2
        Header Length: 64
        Credit Charge: 0
        Channel Sequence: 0
        Reserved: 0000
        Command: Negotiate Protocol (0)
        Credits requested: 2
        Flags: 0x00000000
            .... .... .... .... .... .... .... ...0 = Response: This is a REQUEST
            .... .... .... .... .... .... .... ..0. = Async command: This is a SYNC command
```

smbcmp

- First prototype in emacs
  - https://github.com/aaptel/elshark
- Moved to Python script using curses
  - Calls tshark in the background
- 2 modes
  - Single trace
    - aka curses-wireshark (summaries + details)
  - Diff traces
    - Show 2 summaries
    - Diffs the detailed output