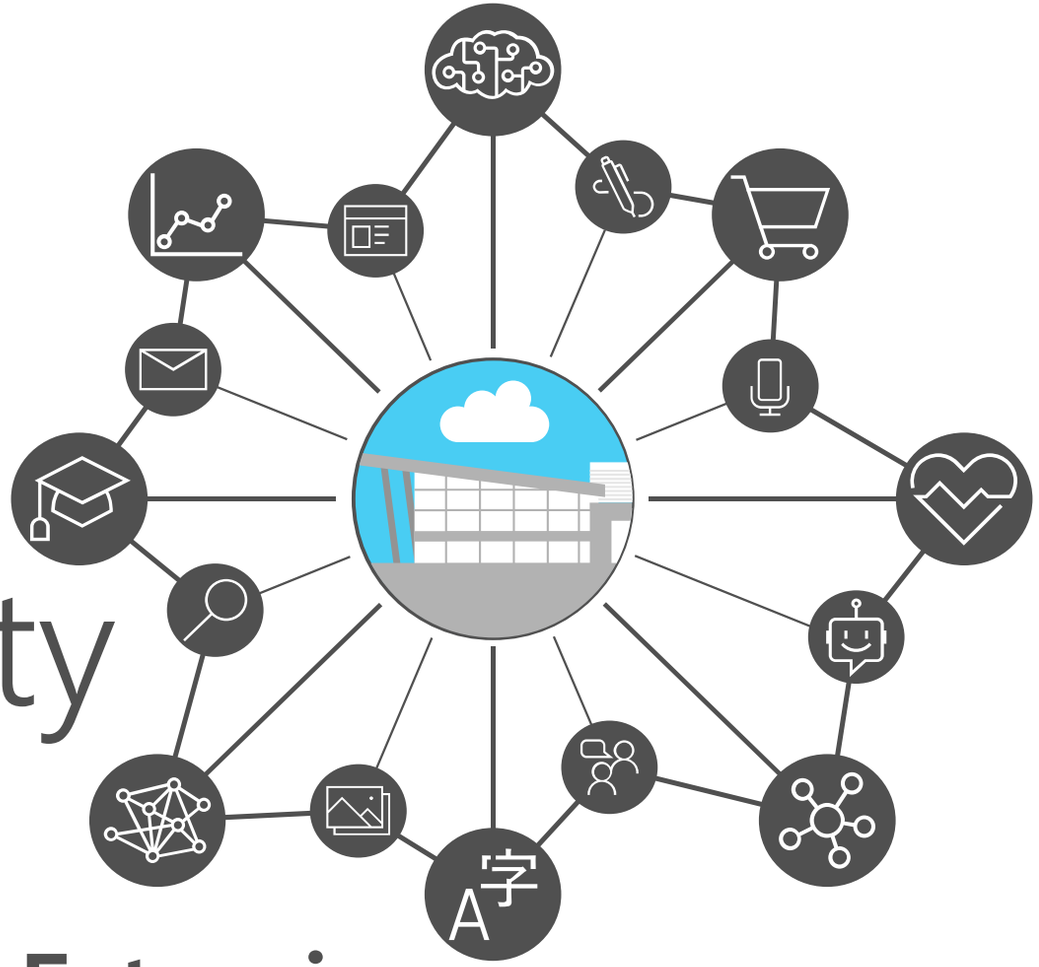
Microsoft



Redmond Interoperability Plugfest 2018

State of the SMB3.1.1 POSIX Extensions

Steve French
Azure Storage – Microsoft

Jeremy Allison
Google

+

SMB 3.11

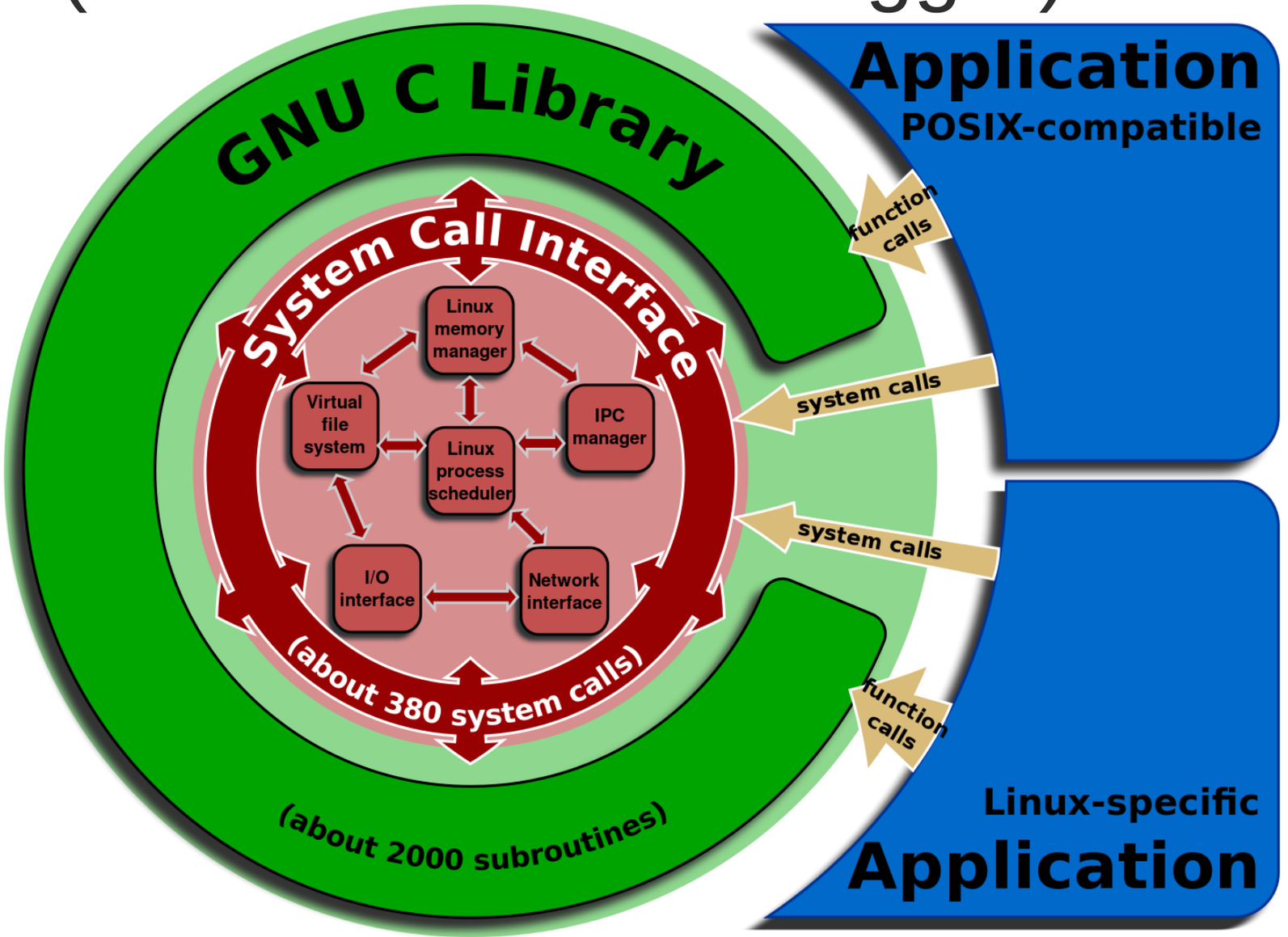
Legal Statement

- This work represents the views of the author(s) and does not necessarily reflect the views of Microsoft or Google
- Linux is a registered trademark of Linus Torvalds.
- Other company, product, and service names may be trademarks or service marks of others.

Outline

- What is POSIX?
- Why do these extensions matter?
- Demo
- What if we don't have them?
  - What works?
  - Some history: CIFS Extensions
  - Alternatives
- Some details
- What if Linux continues to extend, to improve?

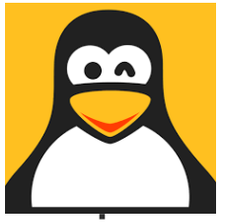
POSIX != Linux
(Linux API is much bigger)



Linux is BIG

- Currently 293 Linux syscalls!
- vs
- About 100 POSIX API calls

Motivations for Extensions



- Linux Apps work!
  - Case sensitivity e.g. is required for the kernel to build on Linux
  - (And Linux and other posix-like operating systems want posix behavior for files whether on premise or in cloud)
- Improve common situations where customers have Linux and Windows and Mac clients accessing the same data
- Deprecation of CIFS – make sure extensions work with most secure, most optimal SMB3.1.1 dialect

What could you try today?

- For obvious reasons, these experimental changes are not turned on by default so …
  - With current mainline Linux (4.18-rc)
  - You must mount with “vers=3.1.1”
  - AND also specify new mount option “posix” and turn off remapping of reserved characters (ie append “nomapposix”)
  - Only a few limited protocol features (posix open context request) can be tried but although small change it is VERY useful and enough to experiment with and test various apps
- JRA has a tree on samba.org (git.samba.org/jra/samba/.git in branch “master-smb2”) with prototype server code

Example

- On the client:
  - “mount –t smb3 //<address>/<share> /mnt -o username=<user>,password=<pass>, vers=3.1.1,posix,mfsymlinks,nomapposix,noperm
- On the server add to smb.conf
  - “mangled names = no”
  - “directory mask = 07777”
  - “create mask = 07777”

Note the new mount option “posix” vs “nounix” (in default SMB3.1.1 mount)

```
root@Ubuntu-17-Virtual-Machine:~/cifs-2.6# cat /proc/mounts | grep cifs
//localhost/test-no-posix /mnt1 cifs rw,relatime,vers=3.1.1,cache=strict,username=testuser,domain=,uid=0,noforc
euid,gid=0,noforcegid,addr=127.0.0.1,file_mode=0755,dir_mode=0755,soft,nounix,serverino,mapposix,rsize=1048576,
wsize=1048576,echo_interval=60,actimeo=1 0 0
//localhost/test /mnt cifs rw,relatime,vers=3.1.1,cache=strict,username=testuser,domain=,uid=0,noforceuid,gid=0
,noforcegid,addr=127.0.0.1,file_mode=0755,dir_mode=0755,soft,posix,posixpaths,serverino,mapposix,rsize=1048576,
wsize=1048576,echo_interval=60,actimeo=1 0 0
root@Ubuntu-17-Virtual-Machine:~/cifs-2.6# cat /proc/fs/cifs/DebugData
Display Internal CIFS Data Structures for Debugging
---------------------------------------------------
CIFS Version 2.12
Features: dfs fscache lanman posix spnego xattr acl
Active VFS Requests: 0
Servers:
Number of credits: 16 Dialect 0x311 posix
1) Name: 127.0.0.1 Uses: 2 Capability: 0x300047 Session Status: 1       TCP status: 1
        Local Users To Server: 1 SecMode: 0x1 Req On Wire: 0
        Shares:
        0) IPC: \\127.0.0.1\IPC$ Mounts: 1 DevInfo: 0x0 Attributes: 0x0
        PathComponentMax: 0 Status: 1 type: 0
        Share Capabilities: None        Share Flags: 0x0
        tid: 0x4f5511db Maximal Access: 0x1f00a9

        1) \\localhost\test Mounts: 1 DevInfo: 0x20 Attributes: 0x1006f
        PathComponentMax: 255 Status: 1 type: DISK
        Share Capabilities: None Aligned, Partition Aligned,    Share Flags: 0x0
        tid: 0x8579c31d Optimal sector size: 0x200      Maximal Access: 0x1f01ff

        2) \\localhost\test-no-posix Mounts: 1 DevInfo: 0x20 Attributes: 0x1006f
        PathComponentMax: 255 Status: 1 type: DISK
        Share Capabilities: None Aligned, Partition Aligned,    Share Flags: 0x0
        tid: 0x1813a493 Optimal sector size: 0x200      Maximal Access: 0x1f01ff

        MIDs:
```

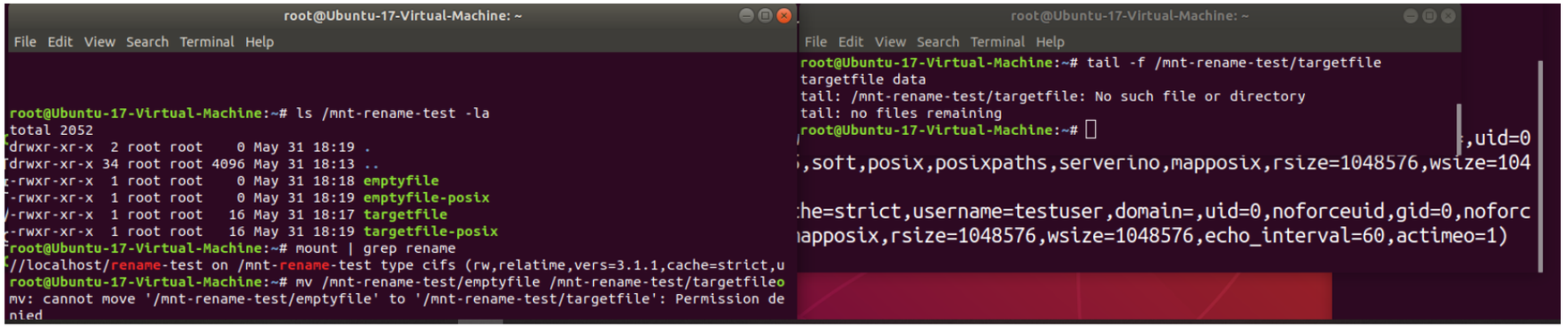
Mode bits on create and case sensitivity work!

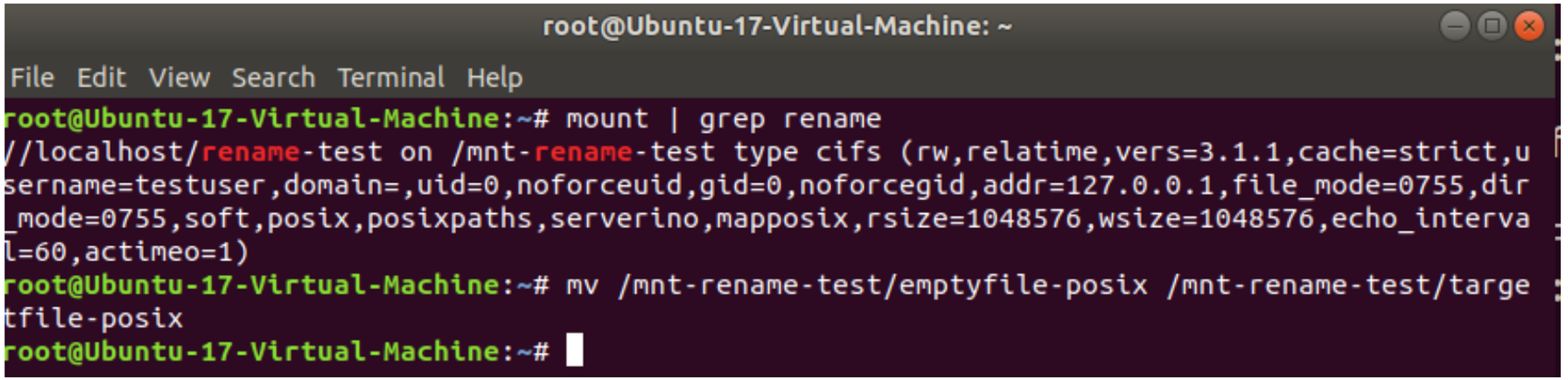
```
root@Ubuntu-17-Virtual-Machine:/mnt# ~/create-4-files-with-mode-test
root@Ubuntu-17-Virtual-Machine:/mnt# cd /mnt1
root@Ubuntu-17-Virtual-Machine:/mnt1# ~/create-4-files-with-mode-test
root@Ubuntu-17-Virtual-Machine:/mnt1# ls /test /test-no-posix -la
/test:
total 12
drwxrwxrwx  3 root     root     4096 May 31 16:55 .
drwxr-xr-x 32 root     root     4096 May 31 16:46 ..
-rwx------  1 testuser testuser    0 May 31 16:55 0700
-rwxrwx---  1 testuser testuser    0 May 31 16:55 0770
-rwxrwxr-x  1 testuser testuser    0 May 31 16:55 0775
drwxr-xr-x  2 sfrench  sfrench  4096 Mar 24 10:34 tmp

/test-no-posix:
total 8
drwxrwxrwx  2 root     root     4096 May 31 16:55 .
drwxr-xr-x 32 root     root     4096 May 31 16:46 ..
-rwxrw-r--  1 testuser testuser    0 May 31 16:55 0700
-rwxrw-r--  1 testuser testuser    0 May 31 16:55 0770
-rwxrw-r--  1 testuser testuser    0 May 31 16:55 0775
root@Ubuntu-17-Virtual-Machine:/mnt1# mkdir UPPER
root@Ubuntu-17-Virtual-Machine:/mnt1# touch upper
root@Ubuntu-17-Virtual-Machine:/mnt1# cd /mnt
root@Ubuntu-17-Virtual-Machine:/mnt# mkdir UPPER
root@Ubuntu-17-Virtual-Machine:/mnt# touch upper
root@Ubuntu-17-Virtual-Machine:/mnt# ls /test /test-no-posix
/test:
0700  0770  0775  tmp  upper  UPPER

/test-no-posix:
0700  0770  0775  UPPER
```

Mode bits on mkdir works!

```
root@smf-Thinkpad-P51:~/cifs-2.6# mount -t smb3 //127.0.0.1/scratch /mnt -o username=testuser,
,vers=3.11,posix
root@smf-Thinkpad-P51:~/cifs-2.6# umask 0000
root@smf-Thinkpad-P51:~/cifs-2.6# mkdir /mnt/0774 -m 0774
root@smf-Thinkpad-P51:~/cifs-2.6# mkdir /mnt/0770 -m 0770
root@smf-Thinkpad-P51:~/cifs-2.6# mkdir /mnt/0444 -m 0444
root@smf-Thinkpad-P51:~/cifs-2.6# ls /scratch -la
total 20
drwxrwxrwx  5 root     root     4096 Jun 15 20:42 .
drwxr-xr-x 35 root     root     4096 Jun 15 20:39 ..
dr--r--r--  2 testuser testuser 4096 Jun 15 20:42 0444
drwxrwx---  2 testuser testuser 4096 Jun 15 20:42 0770
drwxrwxr--  2 testuser testuser 4096 Jun 15 20:42 0774
root@smf-Thinkpad-P51:~/cifs-2.6# cat /proc/version
Linux version 4.17.0+ (sfrench@smf-Thinkpad-P51) (gcc version 7.3.0 (Ubuntu 7.3.0-16ubuntu3))
```

Rename works with POSIX extensions!





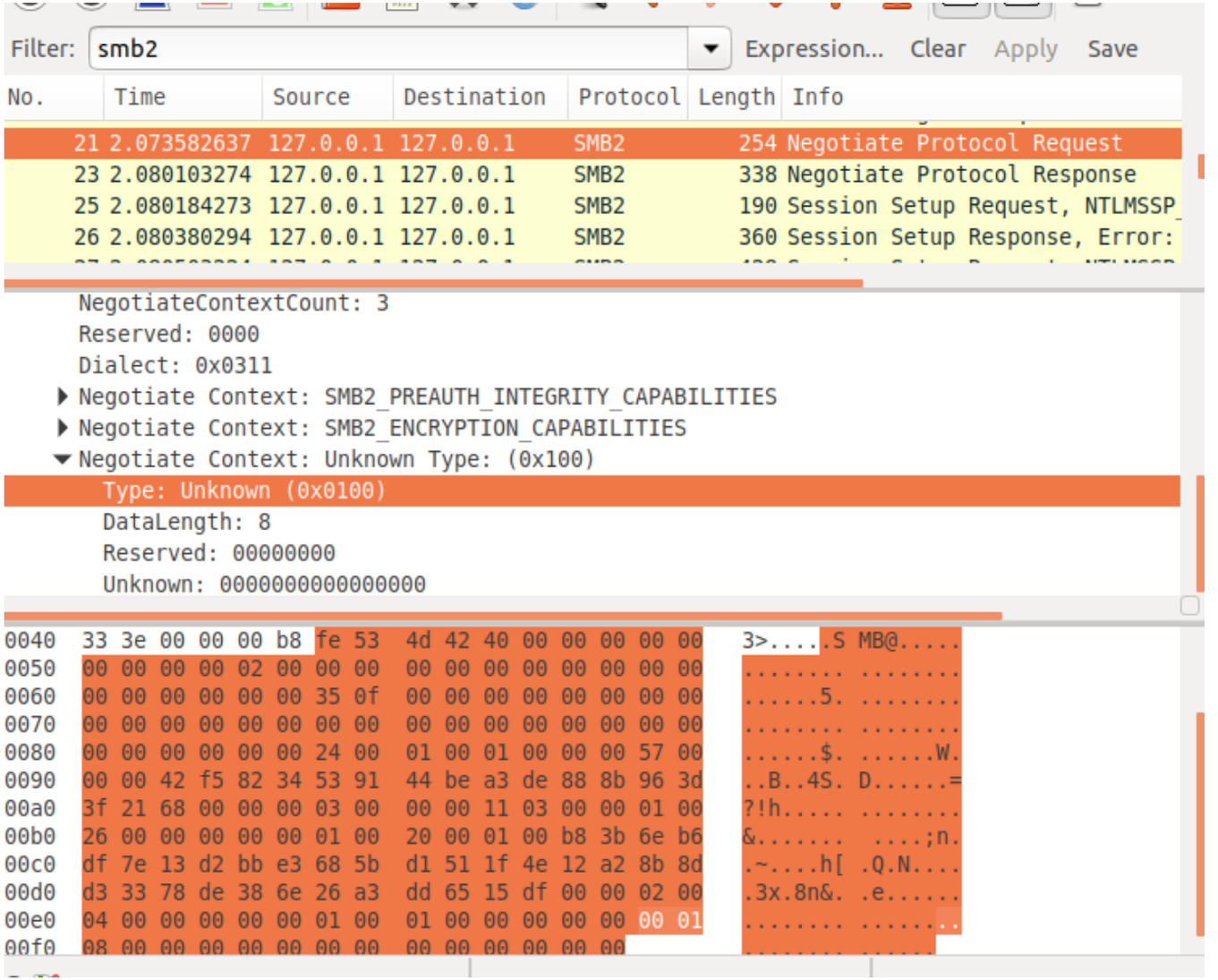
Statfs (“stat –f”) without POSIX extensions:

```
root@smf-Thinkpad-P51:~/cifs-2.6# cat /proc/mounts | grep cifs
//localhost/scratch /mnt cifs rw,relatime,vers=3.0,cache=strict,username=testuser,domai
,uid=0,noforceuid,gid=0,noforcegid,addr=127.0.0.1,file_mode=0755,dir_mode=0755,soft,nou
x,serverino,mfsymlinks,noperm,rsize=1048576,wsize=1048576,echo_interval=60,actimeo=1 0
root@smf-Thinkpad-P51:~/cifs-2.6# stat -f /mnt
  File: "/mnt"
    ID: 0        Namelen: 4096     Type: smb2
Block size: 1024       Fundamental block size: 1024
Blocks: Total: 234804176  Free: 28323720   Available: 28323720
Inodes: Total: 0          Free: 0
root@smf-Thinkpad-P51:~/cifs-2.6# stat -f /scratch
  File: "/scratch"
    ID: e94471edc7140504 Namelen: 255     Type: ext2/ext3
Block size: 4096       Fundamental block size: 4096
Blocks: Total: 58701044   Free: 10080212   Available: 7080929
Inodes: Total: 14983168   Free: 13901548
```

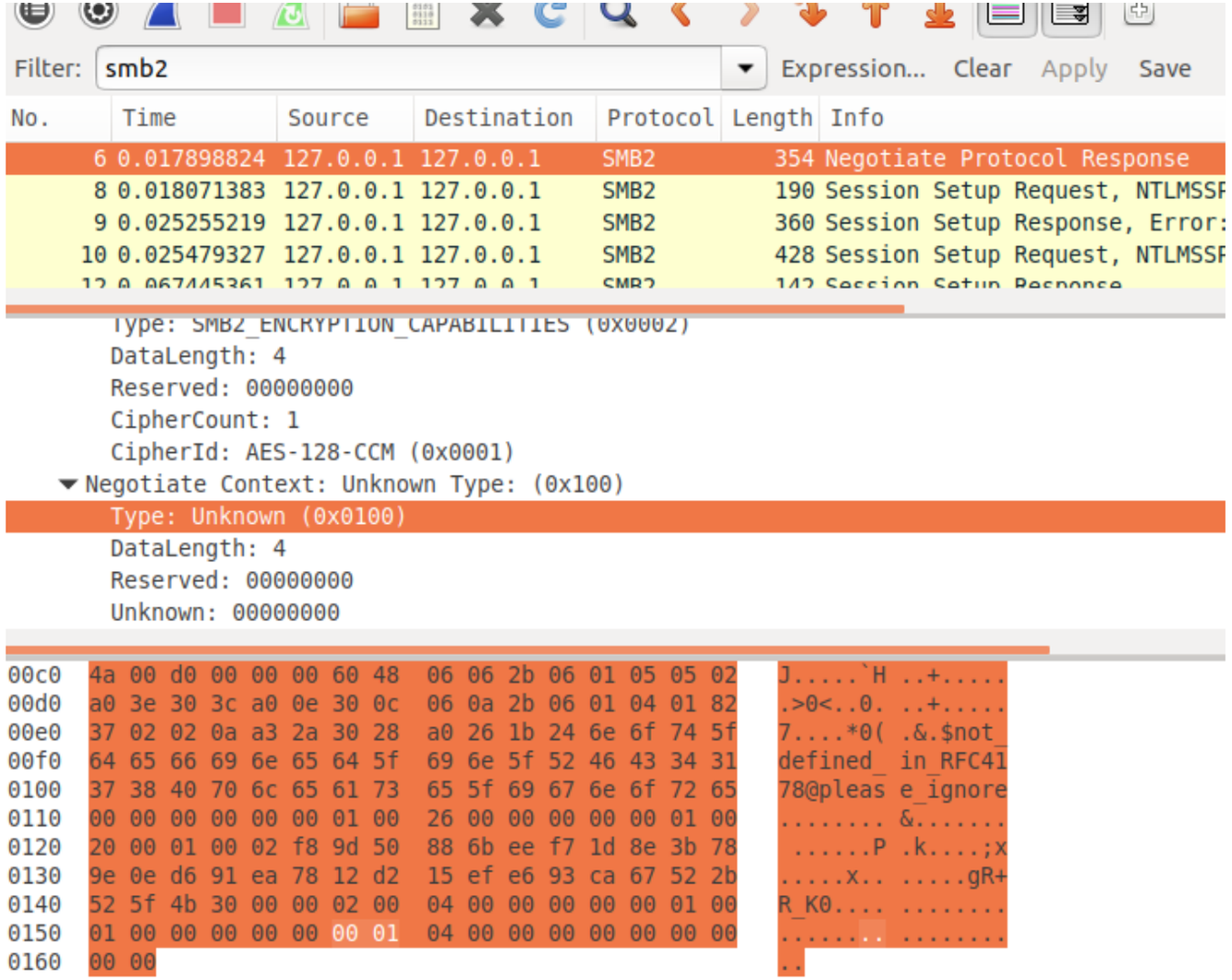
Statfs (“stat –f”) with POSIX extensions – works!

```
root@smf-Thinkpad-P51:~/cifs-2.6# cat /proc/mounts | grep smb3
//127.0.0.1/scratch /mnt1 smb3 rw,relatime,vers=3.1.1,cache=strict,username=testuse
in=,uid=0,noforceuid,gid=0,noforcegid,addr=127.0.0.1,file_mode=0755,dir_mode=0755,s
six,posixpaths,serverino,mapposix,noperm,rsize=1048576,wsize=1048576,echo_interval=
imeo=1 0 0
root@smf-Thinkpad-P51:~/cifs-2.6# stat -f /mnt1
  File: "/mnt1"
    ID: 0        Namelen: 4096     Type: smb2
Block size: 4096       Fundamental block size: 4096
Blocks: Total: 58701044   Free: 10080249   Available: 7080966
Inodes: Total: 14983168   Free: 13901538
root@smf-Thinkpad-P51:~/cifs-2.6# stat -f /scratch
  File: "/scratch"
    ID: e94471edc7140504 Namelen: 255     Type: ext2/ext3
Block size: 4096       Fundamental block size: 4096
Blocks: Total: 58701044   Free: 10080127   Available: 7080844
Inodes: Total: 14983168   Free: 13901536
```

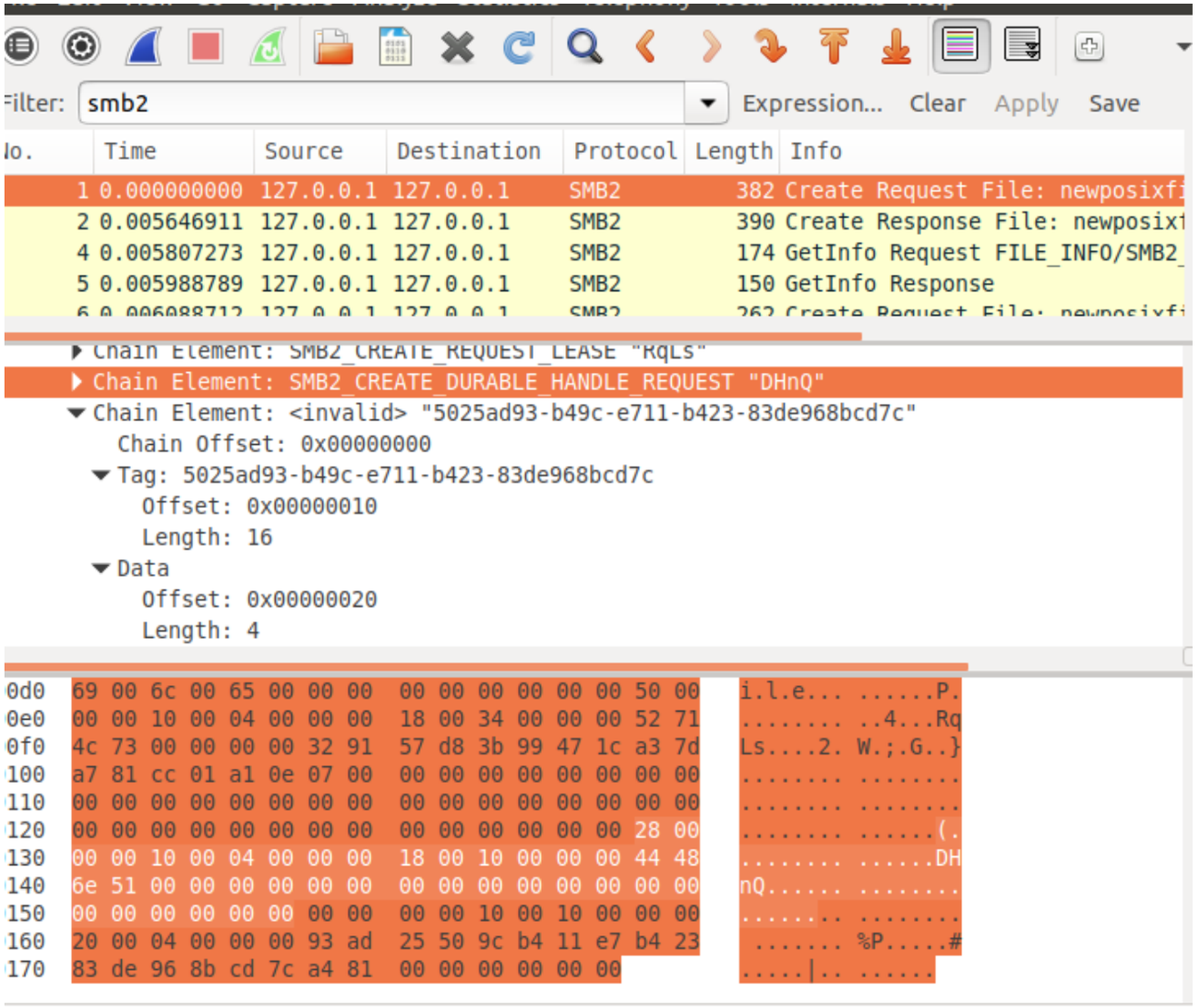
Details – Negotiate Request (w/POSIX)



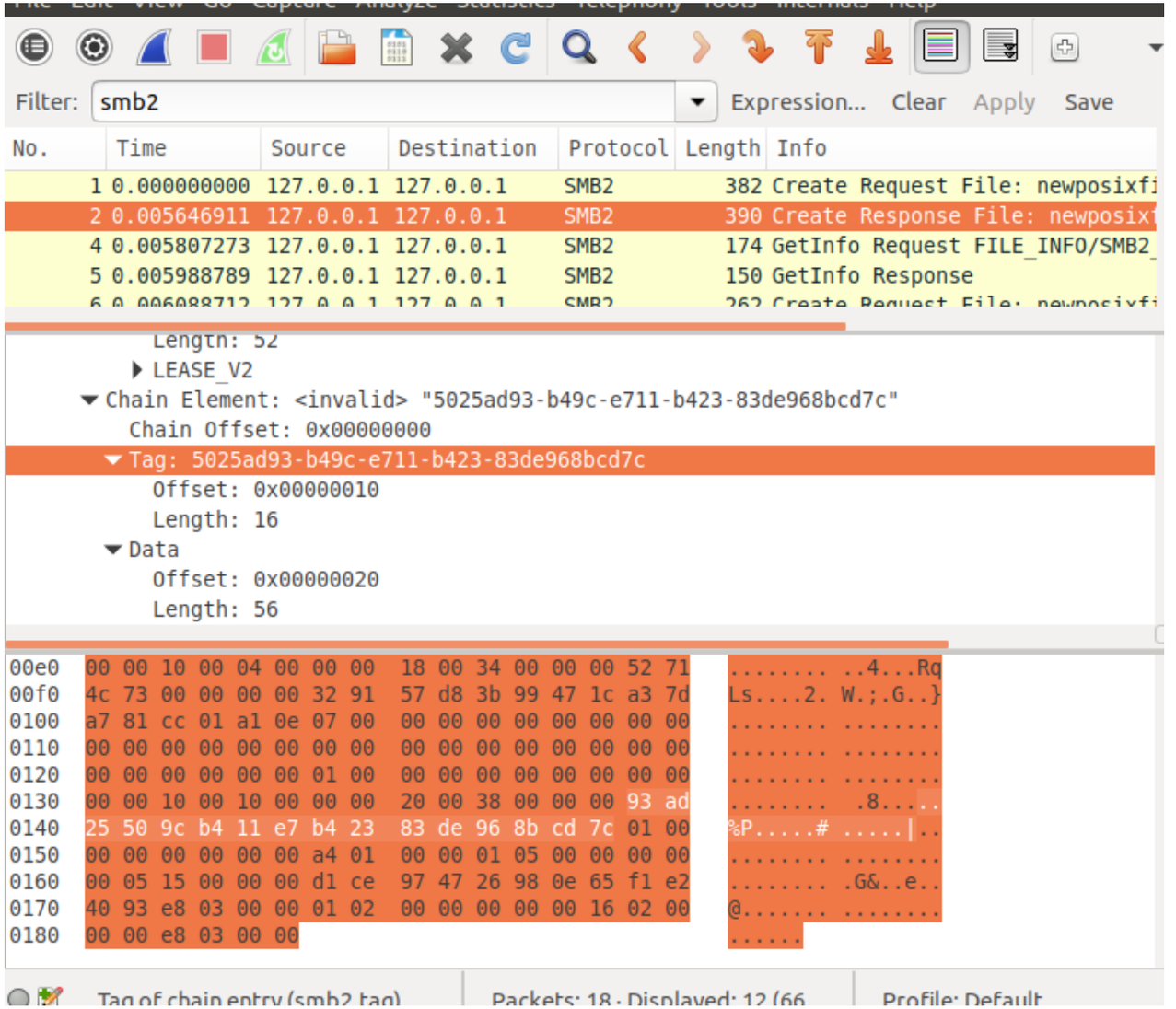
Details (continued) – Neg response



Details continued – Create (POSIX) req



Details continued – create response



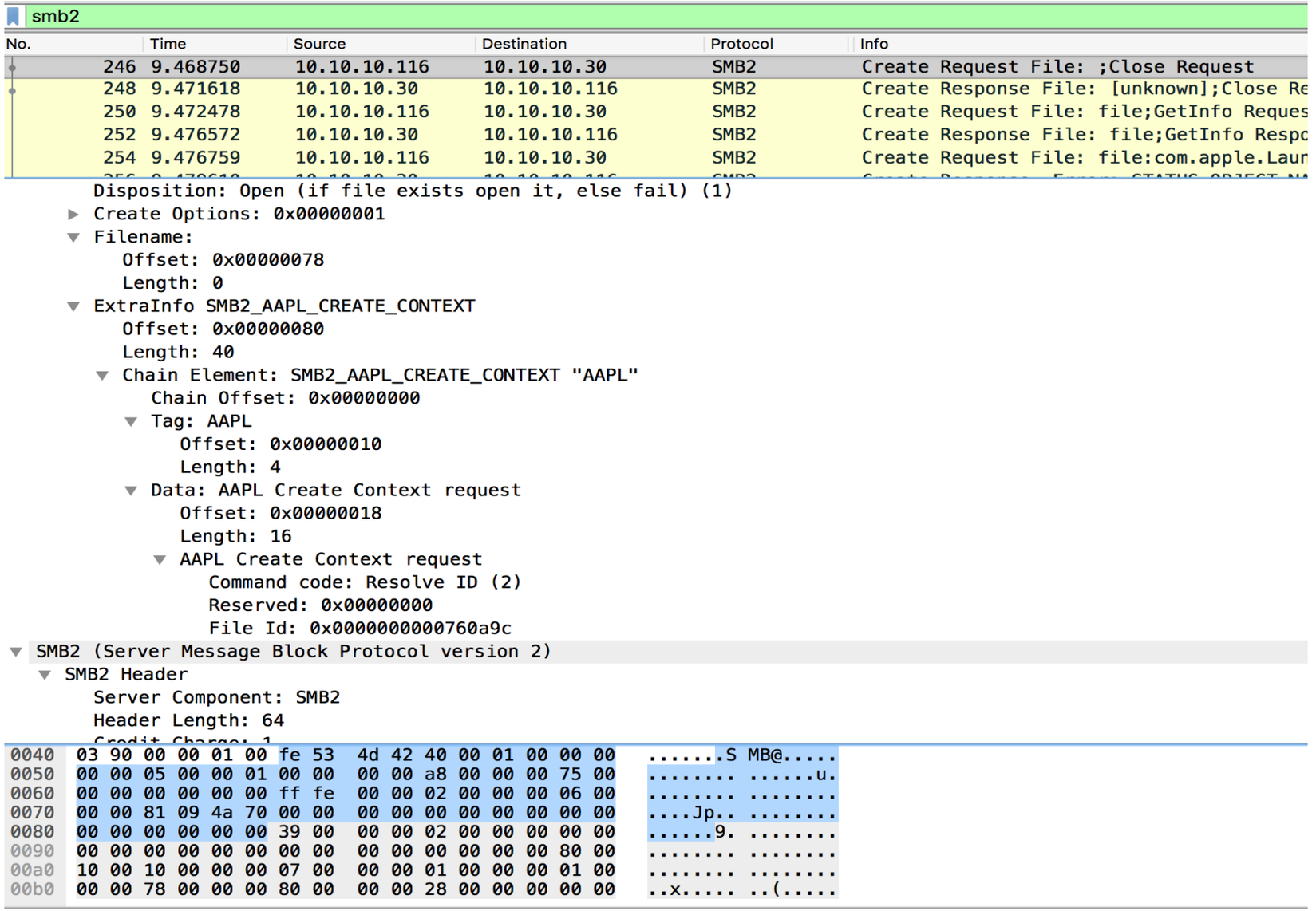
Summary of What works

- Without Extensions

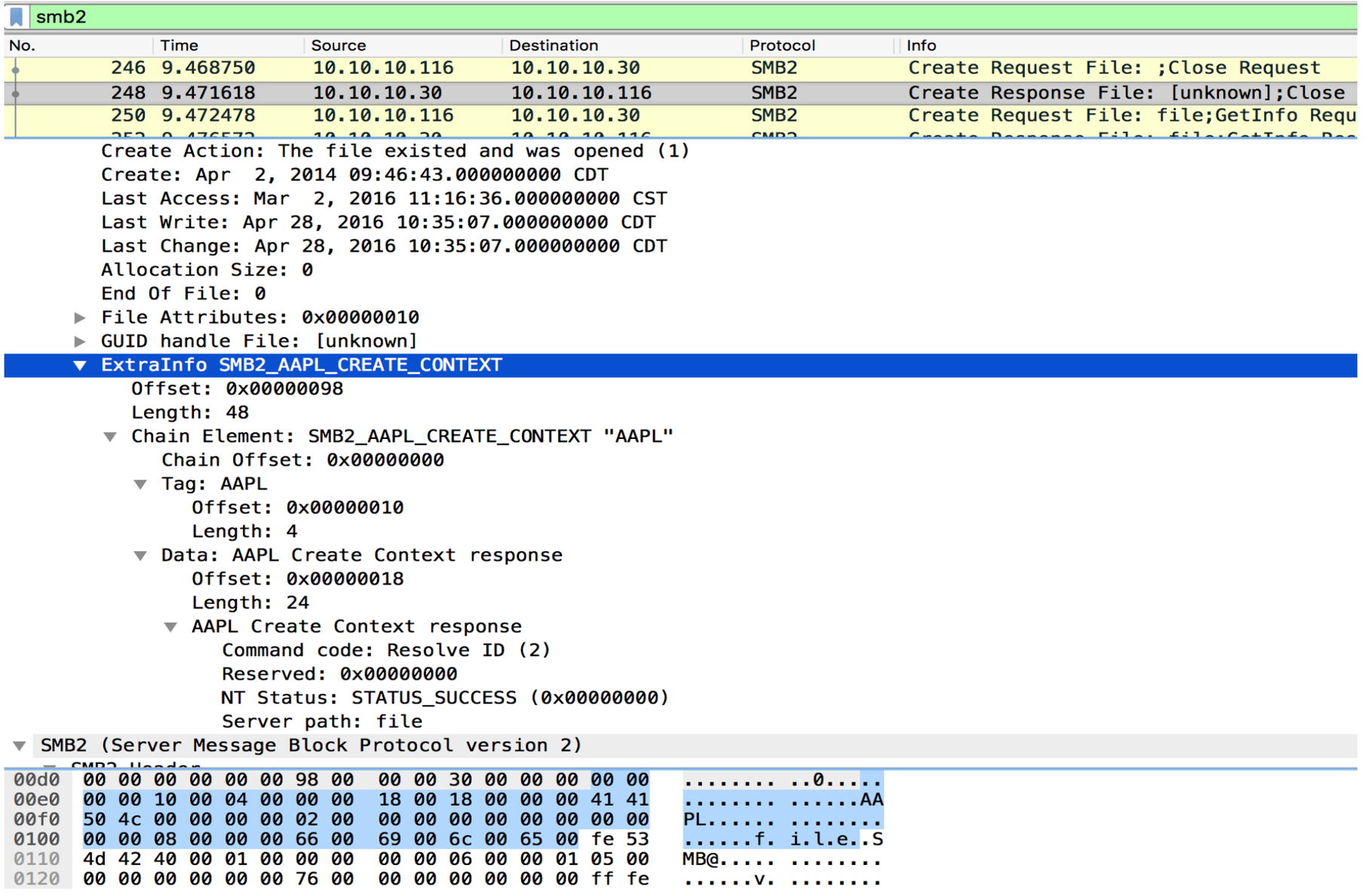
- With Extensions

Other Alternatives: AAPL

Note that Apple create context (AAPL) can be used for some of this



And the response:



CIFS Unix/POSIX Extensions

- What was wrong with what we had?
  - Remember CIFS Deprecation?
  - And not just due to WannaCry …
    - SMB3 is really good …
- Apple SMB2/SMB3 create context does handle case sensitivity, but not all POSIX compatibility issues

Client Perspective

- What about the Linux Kernel?
  - What does it really need from SMB3 to be optimal…?
  - Not just to do 'cool' things: compile kernel on SMB3 mount, boot linux (show blazing performance …!)
  - For all key features: SMB3 >= CIFS with/Unix Extensions
    - We are not asking user to go backwards
  - Can we extend them as Linux API moves
    - (Did we mention that mount API and fsinfo/statfs BOTH are changing – see Al Viro's git tree … and that statx was added last year and Linux continues to evolve ...)

The challenges of Create/Rename/Delete

The challenges of POSIX inode metadata

- What do we need to be able to return?
- What about mode bits and ACLs?

The Challenges of POSIX locking

The Challenges of POSIX FS info

Remember JRA's Server Perspective?

- Learn from the mistakes of SMB1 Unix extensions.
  - Security issues paramount.
  - Remove the possibility of server-followed symlinks
    - Break interoperability with NFS :-(, but necessary.
- Minimum Necessary Change (with apologies to Asimov's "*The End of Eternity*").
  - Fewer changes to the protocol the better.
  - Use the fact that we have experience with Samba in sharing between Windows and UNIX SMB connections.

Server Perspective Continued..

- Server-followed symlinks that the client can create have been a security disaster in Samba.
- Server-following symlinks is a useful holdover from ancient times, when admin-created symlinks gave great flexibility to setups.
  - As soon as clients gained the ability via UNIX extensions to create symlinks, disaster strikes.
  - Failed design decision to store these as real symlinks on the server filesystem.
    - Convenience for dual NFS / SMB1 servers.
- **THIS MUST NOT BE ALLOWED FOR SMB2+**

Server Perspective Continued..

- The key for SMB2 UNIX extensions is to allow simultaneous Windows and UNIX handles – using SMB2 create contexts.
  - Adding UNIX extension create context turns on POSIX behavior for this handle only.
  - Allows client code to probe for POSIX behavior – SMB2 specifies unknown create contexts are ignored.
  - The Samba server already has to handle this case in serving POSIX and non-POSIX client simultaneously.
- Leads to new Negotiate context requirement from the server.
  - That way a client can determine if a server could support POSIX behavior on a handle, but choses not to.
  - POSIX servers may expose POSIX behaviors or deny them depending on pathname (crossing mount points).

Server Perspective Continued..

- The rest of the changes are relatively small.
- One new info level needed to cope with POSIX stat returns.
- Keep protocol as close to “native” Windows as possible.
  - Map POSIX ‘mode’ into Windows ACL encoding.
  - No POSIX ACLs – return everything as Windows ACLs.
  - No POSIX uid/gids – return everything as Windows SIDs.
    - Client systems must cope with mapping SIDs anyway.
- Filename handling (POSIX specific, case sensitive) is the largest change. No access to Windows streams.
  - If you want a Windows stream handle, open a Windows stream handle.
  - Keep USC2 encoding (no change from Windows). UTF-8 would be nice, but not strictly required so drop it.
- Allow server to associate modified behavior on a per-handle basis.


Details

Proposed SMB3 POSIX Extensions

- Negotiate Protocol
  - SMB3.1.1 (or later required)
    - POSIX Negotiate Context 0x100
    - Version is implied by the context (in case extensions are revised in the future to a version 2 or 3 …) but there is a reserved field that can be used in emergency
  - If POSIX open contexts not supported, negotiate context must be ignored
  - If POSIX open contexts supported for some files then negotiate context is returned, but server must fail opens with POSIX contexts for files where POSIX is not supported (rather than ignoring the POSIX context)
- Tree Connect – in future dialects tree connect contexts may allow more granularity in allowing servers to tell clients which shares they can't use POSIX opens on
- Case sensitivity yes/no can be exposed via existing QFS Info call

POSIX Extension Requirements

- If server returns a POSIX create context on an open:
  - It supports case sensitive names on this path
  - It supports POSIX unlink/rename semantics on this file
  - It supports advisory (POSIX) locking on this file.
    - Actually they are “OFD” not “POSIX” locks (see e.g. https://gavv.github.io/blog/file-locks/#emulating-open-file-description-locks)
  - **NEED TO VERIFY**: PATH names are not remapped (no SFU remap needed for * and \ and > and < and : …). UCS2 converted directly to UTF-8 and server supports POSIX pathnames

Other

- Hardlinks use the Windows setinfo call (already used by cifs.ko etc)
- Symlinks are client-only (opaque to server) can use “mfsymlinks” (as Mac and cifs.ko already do) or the Windows NFS symlink reparse point. Servers do not follow these symlinks (for obvious security reasons)
- Other linux extensions, e.g. fallocate are mapped to existing SMB3 operations where possible

Proposed POSIX Extensions

- Create/Open
  - New POSIX create context
    - If POSIX supported then context must be returned on all opens for which POSIX create context was sent (or open should be failed)
    - It is allowed to have POSIX and non-POSIX opens on the same file
    - It is allowed to have some files in a server which are POSIX and some which are not

POSIX open/create context resp.

- __u32 number_of_hardlinks
- __u32 type; /* *e.g. REPARSE tag* */
- __u32 perms; /* *mode & ~S_IFMT* */
- struct dom_sid sid_owner;
- struct dom_sid sid_group;

SMB2/SMB3 Create Contexts

We define a new context name for this new CreateContext to distinguish it from others like MxAc and RqLs and a buffer to include POSIX Information in request and response

SMB2_CREATE_TAG_POSIX = "\x93\xAD\x25\x50\x9C\xB4\x11\xE7\xB4\x23\x83\xDE\x96\x8B\xCD\x7C"

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
Next																															
NameOffset																NameLength															
Reserved																DataOffset															
DataLength																															
Buffer (variable)																															
...																															

Proposed POSIX Infolevels

- Query/SetInfo and Query_DIR
  - Level 0x64 SMB2_FIND_POSIX_INFORMATION
  - Payload variable (Max = 216 bytes)
    - Timestamps
    - File size
    - Dos attributes
    - U64 Inode number
    - U32 device id
    - U32 zero
    - Struct posix_create_context_response

Also need to support statfs (“stat -f”) currently using FS_INFO level 100

```
+struct posix_v1_query_fs_info_response {
      /* Returned for context SMB2_POSIX_V1_STATFS_INFO */
      /* EXISTING posix extensions for fs info is good enough, note For undefined recommended transfer size return -1
in that field */
      __le32 OptimalTransferSize;  /* bsize on some os, iosize on other os */
      __le32 BlockSize; /* f_frsize, disk bytes avail based on this size */
    /* Next three fields are in terms of the block size above. If block size unknown, 4096 would be reasonable block
size for a server to report. Note that returning blocks/blocksavail removes need  to make second call (to QFSInfo
level 0x103.  UserBlockAvail is typically less than or equal to BlocksAvail,  if no distinction is made return the same
value in each */
      __le64 TotalBlocks;
      __le64 BlocksAvail;      /* bfree */
      __le64 UserBlocksAvail;   /* bavail */
      __le64 TotalFileNodes;
      __le64 FreeFileNodes;
      __le64 FileSysIdentifier;   /* fsid */
      /* NB Namelen comes from FILE_SYSTEM_ATTRIBUTE_INFO call , and flags can come from
FILE_SYSTEM_DEVICE_INFO call   */
      /* In Linux  f_type is always 0xFE 'S' 'M' 'B' since that is the fs, not the server's os – so server does not have to
return it */
```

Wireshark

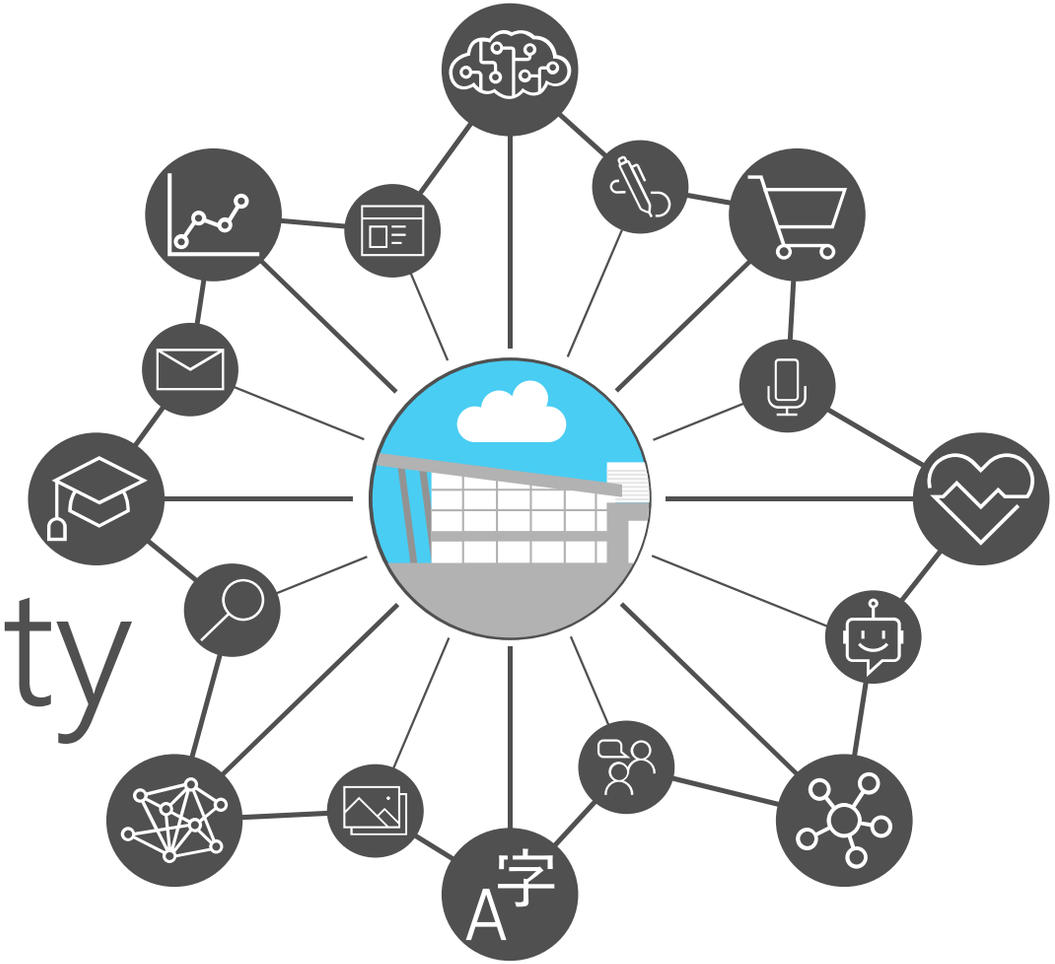
- See Aurelien’s dissector improvements
  - https://github.com/aaptel/wireshark/commits/smb3unix
  - And Pike sample test code
    - https://github.com/aaptel/pike/tree/smb3unix

POSIX Extensions – Where do we go from here?

- Continue debugging test implementations (cifs.ko and JRAs Samba POSIX test branch). Current focus on enhancing readdir this week
- Continue extending the wireshark dissectors (see Aurelien)
- Continue testing/prototyping here and additional testing at SNIA in the fall
- Continue updating the wiki with details: https://wiki.samba.org/index.php/SMB3-Linux

Microsoft

Redmond Interoperability Plugfest 2018



Thank you