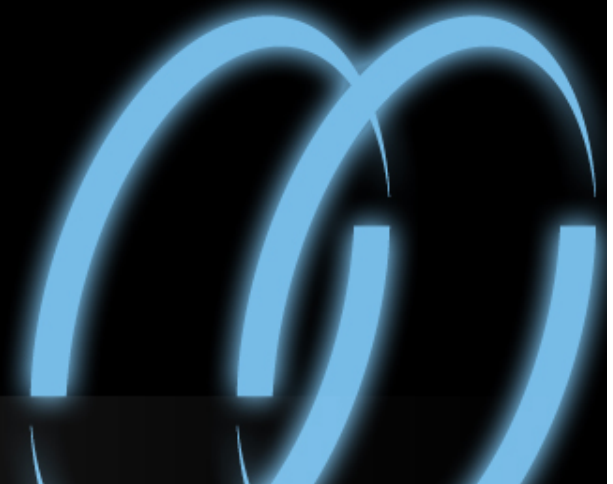


SMB1 / SMB2

A BSD Perspective

Zach Loafman, Staff Engineer
zml@freebsd.org



Preface

- Isilon produces scale-out storage
 - FreeBSD based clustered storage
- Windows interoperability is core to our survival
- But it's not just core to us:
 - Anyone wanting to replace Windows servers should care
 - NFSv4 is heavily influenced by SMB

Agenda

- SMB overview
- SMB2 differences
- SMB details from a UNIX perspective

SMB Overview

- History stretching back into DOS, WfW
- Intellectual property is very mixed:
 - Pieces created by IBM prior to Microsoft, etc.
- Microsoft uses the term CIFS to describe the portion of SMB prior to the Win2k extensions
 - SMB or SMB1 refers to SMB as it appears in Win2k and beyond
 - SMB2 first appeared in Vista

SMB Overview

- Thanks to recent EU action, Microsoft has been forced to document the client to server protocols thoroughly
 - Initially, Samba team formed a front company that licensed the IP through a patent-free agreement
 - A couple of months later, MS gave up on this model and just made the docs public. (*)
- MSDN has very useful PDFs
 - google for e.g. “msdn smb2”.

SMB Overview

- Stateful file access protocol
- Originally designed to run over NetBEUI/NetBIOS
- As of Win2k runs on TCP directly
 - Except plenty of non-Windows clients still use NetBIOS! ☹️

SMB Overview

- Each SMB connection/session is authenticated
 - Kerberos best-effort, NTLM fall-back
- Within a session, client may connect to multiple trees
 - Each tree is a share, roughly a mount
 - Separate users must use separate connections

SMB Overview

- SMB is used also used as a transport for Windows Named Pipes
- Named pipes are used as a transport for DCE/RPC
- Because of this, many local Windows services are accessible remotely – DCE/RPC is used as local IPC.

SMB1 Details

- The best way to understand SMB is to think of it as the Windows file API on the wire:
 - Windows security model
 - NTFS influence on the protocol
- Because of this, as the APIs gradually evolved, SMB1 has a cumbersome number of commands
- Most clients send a small subset, MS has gradually reduced the vocabulary spoken

SMB1 Commands

CREATE_DIRECTORY	LOCKING_ANDX	FIND_CLOSE2
DELETE_DIRECTORY	TRANSACTION	FIND_NOTIFY_CLOSE
OPEN	TRANSACTION_SECONDARY	TREE_CONNECT
CREATE	IOCTL	TREE_DISCONNECT
CLOSE	IOCTL_SECONDARY	NEGOTIATE
FLUSH	COPY	SESSION_SETUP_ANDX
DELETE	MOVE	LOGOFF_ANDX
RENAME	ECHO	TREE_CONNECT_ANDX
QUERY_INFORMATION	WRITE_AND_CLOSE	QUERY_INFORMATION_DISK
SET_INFORMATION	OPEN_ANDX	SEARCH
READ	READ_ANDX	FIND
WRITE	WRITE_ANDX	FIND_UNIQUE
LOCK_BYTE_RANGE	NEW_FILE_SIZE	FIND_CLOSE
UNLOCK_BYTE_RANGE	CLOSE_AND_TREE_DISC	NT_TRANSACT_CREATE
CREATE_TEMPORARY	TRANS2_OPEN2	NT_TRANSACT_IOCTL
CREATE_NEW	TRANS2_FIND_FIRST2	NT_TRANSACT_SET_SECURITY_DESC
CHECK_DIRECTORY	TRANS2_FIND_NEXT2	NT_TRANSACT_NOTIFY_CHANGE
PROCESS_EXIT	TRANS2_QUERY_FS_INFORMATION	NT_TRANSACT_RENAME
SEEK	TRANS2_QUERY_PATH_INFORMATION	NT_TRANSACT_QUERY_SECURITY_DESC
LOCK_AND_READ	TRANS2_SET_PATH_INFORMATION	NT_TRANSACT_SECONDARY
WRITE_AND_UNLOCK	TRANS2_QUERY_FILE_INFORMATION	NT_CREATE_ANDX
READ_RAW	TRANS2_SET_FILE_INFORMATION	NT_CANCEL
READ_MPX	TRANS2_FSCTL	NT_RENAME
READ_MPX_SECONDARY	TRANS2_IOCTL2	OPEN_PRINT_FILE
WRITE_RAW	TRANS2_FIND_NOTIFY_FIRST	WRITE_PRINT_FILE
WRITE_MPX	TRANS2_FIND_NOTIFY_NEXT	CLOSE_PRINT_FILE
WRITE_MPX_SECONDARY	TRANS2_CREATE_DIRECTORY	GET_PRINT_QUEUE
WRITE_COMPLETE	TRANS2_SESSION_SETUP	READ_BULK
QUERY_SERVER	TRANS2_GET_DFS_REFERRAL	WRITE_BULK
SET_INFORMATION2	TRANS2_REPORT_DFS_INCONSISTENCY	WRITE_BULK_DATA
QUERY_INFORMATION2	TRANSACTION2_SECONDARY	

SMB1 Example

- The following slides are from me:
 - browsing to a server
 - picking a share
 - viewing a directory.

Initial Negotiation

Source	Destination	Protocol	Info
10.9.49.131	10.54.17.32	SMB	Negotiate Protocol Request
10.54.17.32	10.9.49.131	SMB	Negotiate Protocol Response
10.9.49.131	10.54.17.32	SMB	Session Setup AndX Request, NTLMSSP_NEGOTIATE
10.54.17.32	10.9.49.131	SMB	Session Setup AndX Response, NTLMSSP_CHALLENGE, Error: STATUS_MORE_PROCESSING_REQUIRED
10.9.49.131	10.54.17.32	SMB	Session Setup AndX Request, NTLMSSP_AUTH, User: DESKTOP\zloafman
10.54.17.32	10.9.49.131	SMB	Session Setup AndX Response

- Brief negotiation to discuss “dialect”
 - Always ‘NT LM 0.12’ in modern SMB1
 - SMB2 dialect may be offered in SMB1
- SessionSetup handles authentication and completes handshake
 - Above example is NTLM

Named Pipe Chatter

Source	Destination	Protocol	Info
10.9.49.131	10.54.17.32	SMB	Session Setup AndX Response
10.9.49.131	10.54.17.32	SMB	Tree Connect AndX Request, Path: \\CODEBIOX.WEST.ISILON.COM\IPC\$
10.54.17.32	10.9.49.131	SMB	Tree Connect AndX Response
10.9.49.131	10.54.17.32	SMB	NT Create AndX Request, FID: 0x4266, Path: \srvsvc
10.54.17.32	10.9.49.131	SMB	NT Create AndX Response, FID: 0x4266
10.9.49.131	10.54.17.32	SMB	Trans2 Request, QUERY_FILE_INFO, FID: 0x4266, Query File Standard Info
10.54.17.32	10.9.49.131	SMB	Trans2 Response, FID: 0x4266, QUERY_FILE_INFO
10.9.49.131	10.54.17.32	DCERPC	Bind: call_id: 1, 2 context items, 1st SRVSVC V3.0
10.54.17.32	10.9.49.131	SMB	write AndX Response, FID: 0x4266, 116 bytes
10.9.49.131	10.54.17.32	SMB	Read AndX Request, FID: 0x4266, 1024 bytes at offset 0
10.9.49.131	10.54.17.32	SMB	Tree Connect AndX Request, Path: \\CODEBIOX.WEST.ISILON.COM\IFS
10.54.17.32	10.9.49.131	DCERPC	Bind_ack: call_id: 1 accept max_xmit: 4280 max_recv: 4280
10.9.49.131	10.54.17.32	SRVSVC	NetshareEnumAll request

- The above is handling share enumeration
- Connects to IPC\$
- Opens \srvsvc (NT Create)
- DCERPC bind to the named pipe

File Tree

Source	Destination	Protocol	Info
10.9.49.131	10.54.17.32	SMB	Tree Connect AndX Request, Path: \\CODEBIOX.WEST.ISILON.COM\IFS
10.54.17.32	10.9.49.131	SMB	Tree Connect AndX Response
10.9.49.131	10.54.17.32	SMB	Trans2 Request, QUERY_PATH_INFO, Query File Basic Info, Path:
10.54.17.32	10.9.49.131	SMB	Trans2 Response, QUERY_PATH_INFO
10.9.49.131	10.54.17.32	SMB	Trans2 Request, QUERY_PATH_INFO, Query File standard Info, Path:
10.54.17.32	10.9.49.131	SMB	Trans2 Response, QUERY_PATH_INFO

- And now we open the \ifs share
- And query it

File enumeration

Source	Destination	Protocol	Info
10.94.17.32	10.9.49.131	SMB	Trans2 Response, FID: 0x426a, QUERY_FILE_INFO
10.9.49.131	10.54.17.32	SMB	Trans2 Request, FIND_FIRST2, Pattern: *
10.54.17.32	10.9.49.131	SMB	Trans2 Response, FIND_FIRST2, Files: . . .snapshot xmax.pcap minhw autorun.inf pro
10.9.49.131	10.54.17.32	SMB	Close Request, FID: 0x426a
10.54.17.32	10.9.49.131	SMB	Close Response, FID: 0x426a

- Directory enumeration
- Etc.

SMB2 Differences

- SMB was rewritten completely in Vista/2k8
- Much, much smaller command set:

NEGOTIATE

LOCK

SESSION_SETUP

IOCTL

LOGOFF

CANCEL

TREE_CONNECT

ECHO

TREE_DISCONNECT

QUERY_DIRECTORY

CREATE

CHANGE_NOTIFY

CLOSE

QUERY_INFO

FLUSH

SET_INFO

READ

OPLOCK_BREAK

SMB2 Differences

- Addition of “durable” file handles means that seamless SMB2 failover is possible
- SMB2 is entirely handle based after initial CREATE call
- Compound actions in a single packet
- Credit based pipelining for in-flight packets
- Larger buffer sizes

SMB2.1 Differences

- Microsoft revamped SMB2 in Win7 / 2k8R2
- Changed opportunistic locks to more extensible “lease” mechanism
- Added resilient handles

SMB Details

- Frilly NFS? Differences are in the details.
- Next slides present SMB as seen from a UNIX/POSIX perspective.

SMB Details – Credentials

- In both Kerberos and NTLM, the server receives a Privilege Attribute Certificate (PAC) as part of authentication.
- PAC rolls up all information about the user and the group membership for said user.
 - Analogous to the cred structure.
- PAC is signed by account manager (usually the domain controller), immutable by client

SMB Details – SIDs

- PAC has Windows Security Identifiers (SIDs), not uid/gids
- SIDs are arbitrary length GUIDs
 - Composed of domain SID / relative identifier
- Example:
 - S-1-5-21-7623811015-3361044348-030300820-1013
 - In this case, 1013 is the RID, the rest is the domain
- SIDs are not typed – a given SID may be a user or group
 - This fact is particularly important in ACLs.

SMB Details – Credentials

- In UNIX, this is usually handled differently:
 - The full identity of the client is never signed by the account manager.
 - UIDs and GIDs are not globally unique.
 - ... but are small.
- In order to cope, UNIX SMB solutions usually try to map from SID to uid/gid
 - Fraught with peril!

SMB Details – Opens

- SMB file opens (usually done these days by NtCreateAndX) are one of the most critical parts of the protocol
- A file is opened with:
 - Desired file access (open mode)
 - Sharing allowed
 - Opportunistic lock request
- Desired file access is fairly self explanatory, but much finer grained than POSIX.

SMB Details – Sharing

- Share mode is roughly inverse to a fine-grained, whole-file lock on the file
- Bitmask of “read”, “write”, “delete”, e.g.
 - “share for read/write but not delete”
 - “share for read but not write or delete”

SMB Details – Sharing

- There's no real equivalent in UNIX
- Full file locks come the closest, but are usually advisory
 - Some variants have mandatory available
 - None have the notion of “delete”
- This is required for NFSv4 interop

SMB Details – Oplocks

- Opportunistic locks are the right for the client to cache data, requested on open
- They have rather odd names, but:
 - With a batch oplock, the client may cache writes, and is given the right to handle opens locally.
 - With an exclusive oplock, the client may cache writes, but should send all opens over the wire.
 - With a level II oplock, the client may only keep a read-only cache.

SMB Details – Oplocks

- Again, no real UNIX equivalent
 - AFS and others implemented similar concepts
- Linux has an inadequate lease interface
- This is required for NFSv4 interop

SMB Details – Byte Range Locks

- Byte range locks in SMB are mandatory locks by default, unlike POSIX
- Read and write requests fail if the region is locked
- No SMB notion of advisory locks

SMB Details – Change Notify

- SMB supports a feature called “change notify” to allow registration of a listener on:
 - A single file
 - A directory but no subdirectories
 - An entire directory tree
- Change notify is advisory
 - Client should not rely on change notify for metadata cache changes
 - Change notify is not audit

SMB Details – Change Notify

- Single file, single directory level notification is present in FreeBSD and a variety of other OSes
- No one implements recursive notification
 - We do, but it's a Hard Problem™
- Single directory notification is present in NFSv4.1, but called “directory delegations”; it's more than advisory.

SMB Details

- NTFS Alternate Data Streams
 - Substreams within a file, but no separate metadata (e.g. there is only one ACL).
- Various UNIX variants have something close, but no one seems to have done it exactly the same. (Sigh.)

SMB Details

- NTFS Security Descriptors
 - SACL: “System Access Control List” describes how a file should be audited
 - DACL “Discretionary Access Control List” describes who is authorized
 - If you’re familiar with NFSv4 ACLs, they were based on NTFS ACLs

SMB Details

- Various UNIXes have implemented NFSv4 ACLs, but there are many pain points here:
 - Mode bit interaction is underspecified and should be more standardized.
 - Windows admins rarely use DENY ACEs, UNIX ACL implementations use them when imitating mode bits.
 - NTFS has a 'canonical' order, even if the protocol allows you to specify otherwise.

SMB Details

- NTFS is case-insensitive
 - This is negotiable in SMB, but in practice you want to stay insensitive to interoperate with Windows applications.
- There's no standard way to approach this problem in UNIX
 - We have a case insensitive lookup variant..
 - ..But we recently discovered we have to know the canonical name on disk at times, too.

Conclusion

- SMB is here to stay
- Interop is hard

- There are no right answers
- ... but there are some less wrong ones.

- Questions?