SMB 3.0
(Because 3 > 2)

David Kruse
Microsoft

SMB 2.2 (Per SDC 2011)

- Multichannel
- SMB over RDMA
- Scale-Out Awareness
- Persistent Handles
- Witness Notification Protocol
- Clustered Client Failover
- Directory Leasing
- Branch Cache v2
- Support for Storage Features (TRIM, etc)
- SNIA offers video recordings at http://snia.org/events/storage-developer2011/2011presentations#cifs_smb_smb2

Problem Space

Availability

- Enable transparent client recovery in the presence of
  - Network Failure
  - Server Failure
- Minimize failover time to reduce application stalls

Performance

- Enable clients to aggregate available bandwidth across adapters transparently
- Continue to increase efficiency on high bandwidth networks

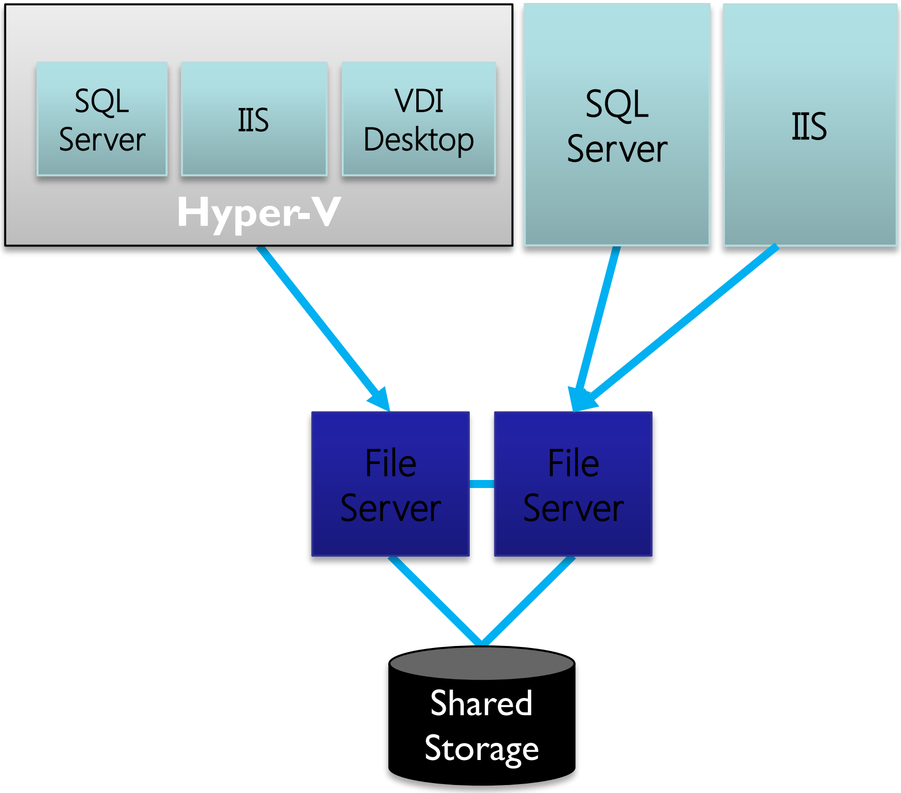
Traffic Reduction

- Continue improving user perceived latency when working in a WAN environment

2012 Storage Developer Conference. © Microsoft Corporation. All Rights Reserved.

Remote File Storage for Server Applications

- What is it?
  - Server applications storing their data files on SMB file shares (UNC paths)
  - Examples:
    - Hyper-V: Virtual Hard Disks (VHD), config., snapshots
    - SQL Server: Database and log files
- What is the value?
  - Easier provisioning – shares instead of LUNs
  - Easier management – shares instead of LUNs
  - Flexibility – dynamic server relocation
  - Leverage network investments – no need for specialized storage networking infrastructure or knowledge
  - Lower cost – Acquisition and Operation cost



SMB3 is…

- First class storage
  - Item by item, a storage solution that can match the capabilities of traditional block solutions

2012 Storage Developer Conference. © Microsoft Corporation. All Rights Reserved.

Who Speaks What?

Client / Server OS	Windows 8 Windows Server 2012	Windows 7 Windows Server 2008 R2	Windows Vista Windows Server 2008	Previous versions of Windows
Windows 8 Windows Server 2012	SMB 3.0	SMB 2.1	SMB 2.0	SMB 1.0
Windows 7 Windows Server 2008 R2	SMB 2.1	SMB 2.1	SMB 2.0	SMB 1.0
Windows Vista Windows Server 2008	SMB 2.0	SMB 2.0	SMB 2.0	SMB 1.0
Previous versions of Windows	SMB 1.0	SMB 1.0	SMB 1.0	SMB 1.0

2012 Storage Developer Conference. © Microsoft Corporation. All Rights Reserved.

Who Speaks What? (Pre-release)

Windows 8	SMB 2.0+ Dialects (Client and Server)
Windows 8 Beta	SMB 2.0 SMB 2.1 ***SMB 2.2***
Windows 8 Release Candidate	SMB 2.0 SMB 2.1 ***SMB 2.2*** **SMB 3.0**
Windows 8	SMB 2.0 SMB 2.1 **SMB 3.0**

2012 Storage Developer Conference. © Microsoft Corporation. All Rights Reserved.

New Since SDC 2011

- ❒ Secure Negotiate
- ❒ SMB Encryption
- ❒ Signing Update
- ❒ Session Key Derivation
- ❒ Channel Sequence

Secure Negotiate

- Used to detect attempts to push client to an older dialect by an intermediate party
  - Defense from downgrades for security related changes or additions (encryption, signing)
- Relies on server signing an unsolicited signed request from the client
- Servers that don't understand the new FSCTL, return a *signed* error
- Client accepts *signed* error or success as successful validation
- Client and server verify dialect information matches result of negotiate

2012 Storage Developer Conference. © Microsoft Corporation. All Rights Reserved.

A Slide for Chris

- What happens if you downgrade to SMB1?
  - Secure Negotiate requires signing of the request/response pair to ensure tampering
  - SMB1 does not support unsolicited signed requests. (Signing is negotiated as a connection property)
  - Signing is disabled in most deployments, making secure negotiate difficult
  - If signing was enabled for connection, FSCTL could be implemented by server and sent by client to verify server does not support SMB2.
  - Windows clients *currently* do not issue this, and server *currently* does not respond or disconnect.

Configuring the Client

HKLM\SYSTEM\CurrentControlSet\Services\LanmanWorkstation\Parameters\RequireSecureNegotiate

0 = Disabled

1 = Required

2012 Storage Developer Conference. © Microsoft Corporation. All Rights Reserved.

Time to Ponder…

- When do we recommend organizations disable SMB1/CIFS by default?

```
PS C:\> Get-SmbSession | Select
Dialect,ClientComputerName,ClientUserName | Where-Object
{$_.Dialect –lt 2.00}
```

2012 Storage Developer Conference. © Microsoft Corporation. All Rights Reserved.

SMB Encryption

- End-to-end encryption of SMB data in flight
  - Protects data from eavesdropping/snooping attacks on untrusted networks
- Zero new deployment costs
  - No need for IPSec, specialized hardware, or WAN accelerators
- Can be configured per share or for the entire server
- Can be turned on for a variety of scenarios where data traverses untrusted networks
  - Branch Offices over WAN networks
  - Application workload over unsecured networks

SMB Encryption - Details

- Algorithm
  - AES CCM 128 bit
  - No algorithm negotiation capability
  - Will sign AND encrypt in the same step (independent of SMB Signing setting)
  - An encrypted packet is considered “signed” (Not necessary to both sign the SMB packet and then encrypt it)
- Performance
  - Overhead: Uses AES acceleration provided by most new processors. (cycles/byte is lower than SMB2 signing!)
- Impact to versions < 3.0 (Vista, XP, Win8 pre-beta)
  - If encryption is turned on, clients will get “Access Denied” errors

SMB Encryption - Details

- What packets get encrypted?
  - If Server level encryption is turned on, all request-response packets from SessionSetup onwards
  - If Share level encryption is turned on, all request-response packets from TreeConnect onwards
  - None of the unsolicited responses from server are encrypted (such as lease break notifications)
- Encryption will work the same regardless of Kerberos or NTLM auth
- Protocols:
  - SMB3 on both client and server
  - RPC over NamedPipes will use encryption if it is turned on for the server
  - Using encryption over RDMA will force Send/Recv (instead of Read/Write) mode.

SMB2 TRANSFORM_HEADER

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>20</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>30</td><td>1</td></tr>
<tr><td colspan="32">ProtocolId</td></tr>
<tr><td colspan="32">Signature</td></tr>
<tr><td colspan="32">…</td></tr>
<tr><td colspan="32">…</td></tr>
<tr><td colspan="32">…</td></tr>
<tr><td colspan="32">Nonce</td></tr>
<tr><td colspan="32">…</td></tr>
<tr><td colspan="32">…</td></tr>
<tr><td colspan="32">…</td></tr>
<tr><td colspan="32">OriginalMessageSize</td></tr>
<tr><td colspan="16">Reserved</td><td colspan="16">EncryptionAlgorithm</td></tr>
<tr><td colspan="32">SessionId</td></tr>
<tr><td colspan="32">…</td></tr>
</table>

ProtocolId (4 bytes): The protocol identifier. The value MUST be (in network order) 0xFD, 'S', 'M', and 'B'.

Signature (16 bytes): The 16-byte signature of the encrypted message generated by using **Session.EncryptionKey**.

Nonce (16 bytes): An implementation-specific value assigned for every encrypted message. This MUST NOT be reused for all encrypted messages within a session.

OriginalMessageSize (4 bytes): The size, in bytes, of the SMB2 message.

Reserved (2 bytes): This field MUST NOT be used and MUST be reserved. The client MUST set this to zero, and the server MUST ignore it on receipt.

EncryptionAlgorithm (2 bytes): The algorithm used for encrypting the SMB2 message. This field MUST be set to one of the following values: SMB2_ENCRYPTION_AES128_CCM

SessionId (8 bytes): Uniquely identifies the established session for the command.

2012 Storage Developer Conference. © Microsoft Corporation. All Rights Reserved.

Nonce Generation

- Nonce field is 16 bytes, usage is algorithm specific
- AES CCM 128 uses 11 bytes
- Sender is responsible for guaranteeing uniqueness, receiver does not validate
  - SMB2 Message ID’s prevent replay already!
- Sender implementation can partition out nonce space as needed

2012 Storage Developer Conference. © Microsoft Corporation. All Rights Reserved.

Have you seen my Keys?

- As with signing, integrity of encryption is reliant on the safety of the private session key
- Similar to disabling CIFS, turning off old authentication modes (NTLM v1) should be a priority

Developing and Supporting

- Encryption prevents visibility into network frames for debugging and diagnostics
- RDMA also can make sniffing network traffic more difficult
- Come to Paul Long’s “The Future of Protocol and SMB2/3 Analysis” : Tuesday 4:05p

SMB 3 Signing Update

- Signing algorithm moved to from HMAC-SHA256 to AES_CMAC-128 (RFC4493)
  - Takes better advantage of processor acceleration
  - On x64, algorithm takes around 3x less cycles.
    - Non-official numbers (Windows):

Algorithm	Cycles
HMAC-SHA256	6500 + 18 * (# of bytes)
AES_CMAC-128 (accelerated)	1500 + 3 * (# of bytes)

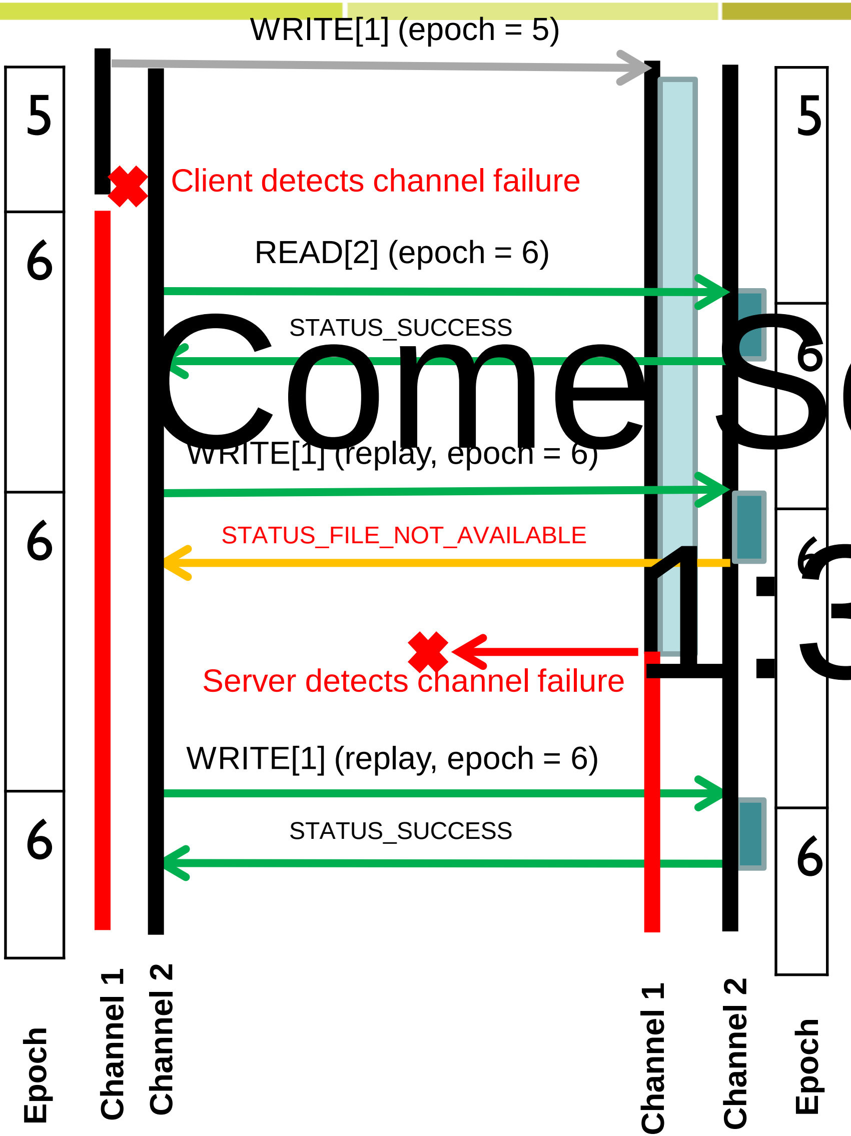
2012 Storage Developer Conference. © Microsoft Corporation. All Rights Reserved.

SMB 3 Key Derivation

- Original Session Key is fully “private”
- Separate keys derived using SP800-108 specified KDF, using HMAC-SHA256 as the PRF.
  - SMB3 Signing
  - SMB3 Encryption (Client-Server and Server-Client)
  - Session Key given to RPC

2012 Storage Developer Conference. © Microsoft Corporation. All Rights Reserved.

Channel Sequence (Object Epoch Barrier)


WRITE[1] (epoch = 5)
5
5
Client detects channel failure
6
READ[2] (epoch = 6)
STATUS_SUCCESS
6
WRITE[1] (replay, epoch = 6)
6
STATUS_FILE_NOT_AVAILABLE
6
Server detects channel failure
WRITE[1] (replay, epoch = 6)
STATUS_SUCCESS
6
6
Epoch
Channel 1
Channel 2
Channel 1
Channel 2
Epoch

CLIENT SERVER

- Lightweight compared to full replay detection.
- Guarantees that all previous "instances" of an I/O are drained before the replay is executed.
- Client maintains 16-bit channel epoch number.
  - Incremented on a network failure.
  - Sent to server via unused Status field.
- Server fails "state changing" "non-replay" requests with stale epoch numbers.
- Server fails "state changing" "replay" requests when there are outstanding operations with older epoch numbers.
- New error - STATUS_FILE_NOT_AVAILABLE avoids blocking on the server and tells client to retry.
- Server can do epoch check at "handle" granularity.
- Come to "CA SMB" talk this afternoon for more info

Come See Us at 1:30!

2012 Storage Developer Conference. © Microsoft Corporation. All Rights Reserved.

QUESTIONS?