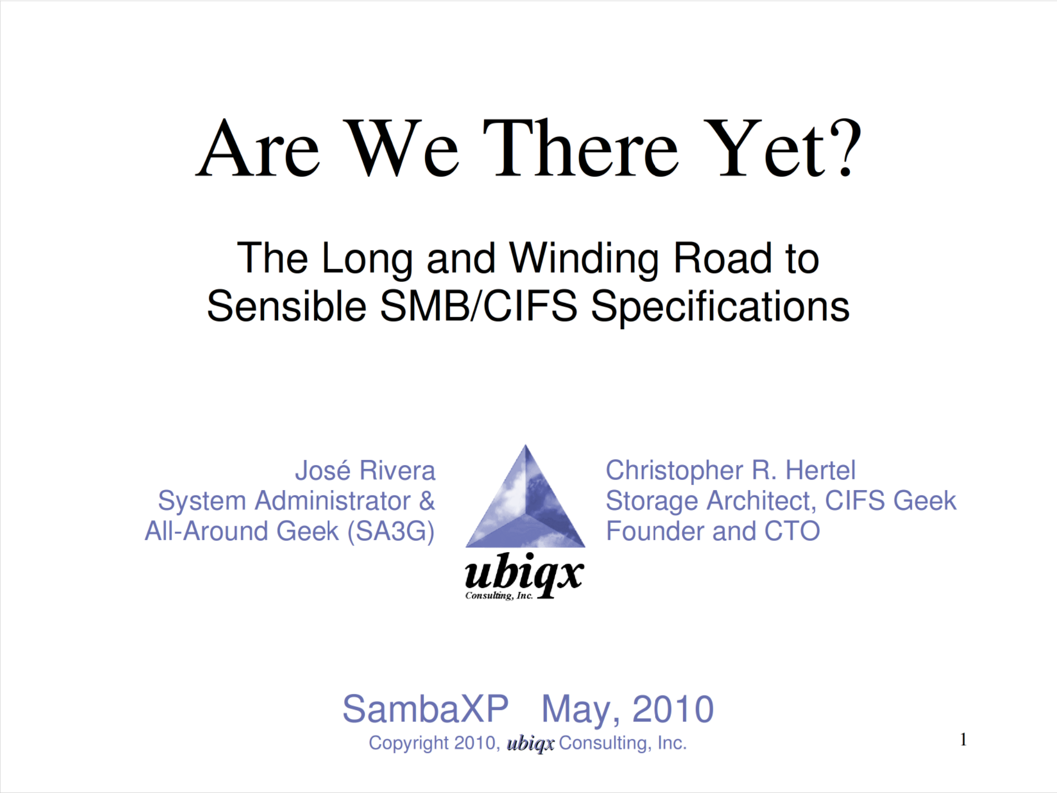


Are We There Yet?

The Long and Winding Road to Sensible SMB/CIFS Specifications

José Rivera
System Administrator &
All-Around Geek (SA3G)

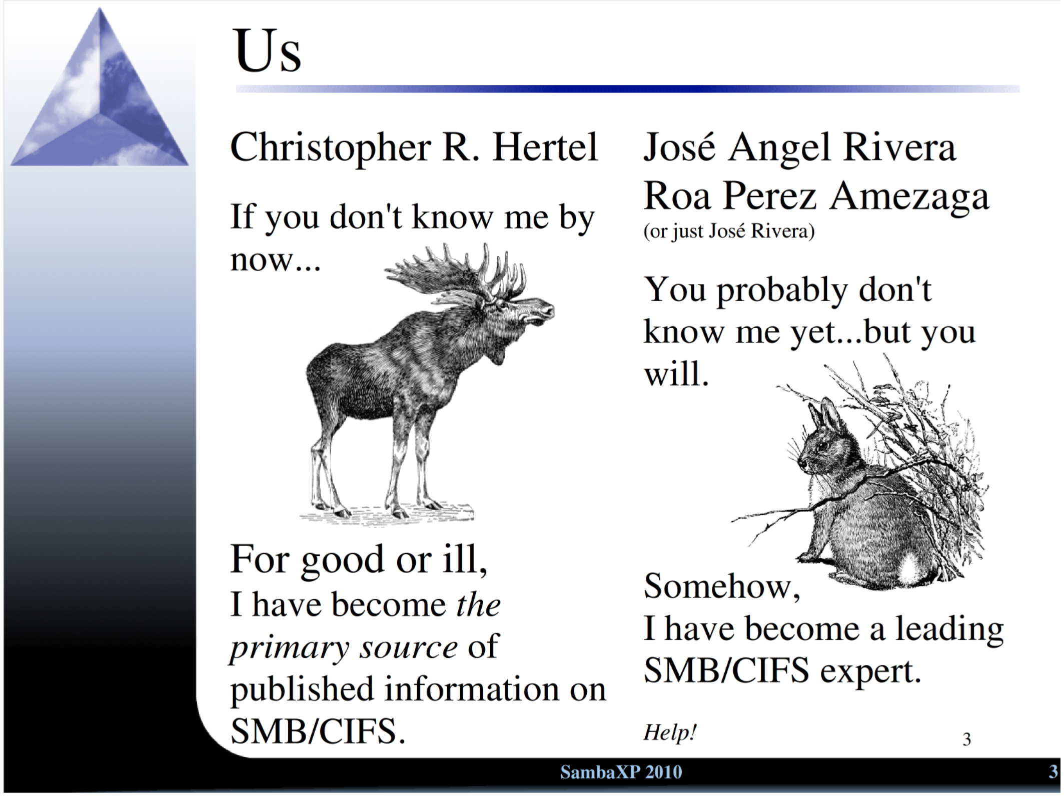
Christopher R. Hertel
Storage Architect, CIFS Geek
Founder and CTO

ubiqx Consulting, Inc.

SambaXP May, 2010

Available for parties, housewarmings, anniversaries, company outings, and events.

Introductions



Us

Christopher R. Hertel

If you don't know me by now...

For good or ill,
I have become *the primary source* of published information on SMB/CIFS.

José Angel Rivera Roa Perez Amezaga
(or just José Rivera)

You probably don't know me yet...but you will.

Somehow,
I have become a leading SMB/CIFS expert.

Help!

José was important to the project because he represented the target minimum audience: a recent CS graduate. We knew that if we could write [MS-CIFS] so that he could understand it, we were meeting that goal.

As it turned out, however, José was able to absorb and process the material quickly, so he soon transitioned from test subject to participating writer.

- We (*ubiqx*) are now *the* source for published information on SMB/CIFS. (Ouch!)
  - All your CIFS are belong to us.
  - **Implementing CIFS** is still the only implementer's guide.
  - [MS-CIFS] and [MS-SMB] are now the *official* Microsoft specifications.
- This whole exercise was a leap of faith for all concerned.
  - It worked, in part, because it was the right thing to do.
  - It worked, in part, because all parties were committed to making it work.

You

Who's here?

- Samba Developers
- Samba Supporters
- Samba Users
- Samba Resellers
- Samba Leveragers of all kinds

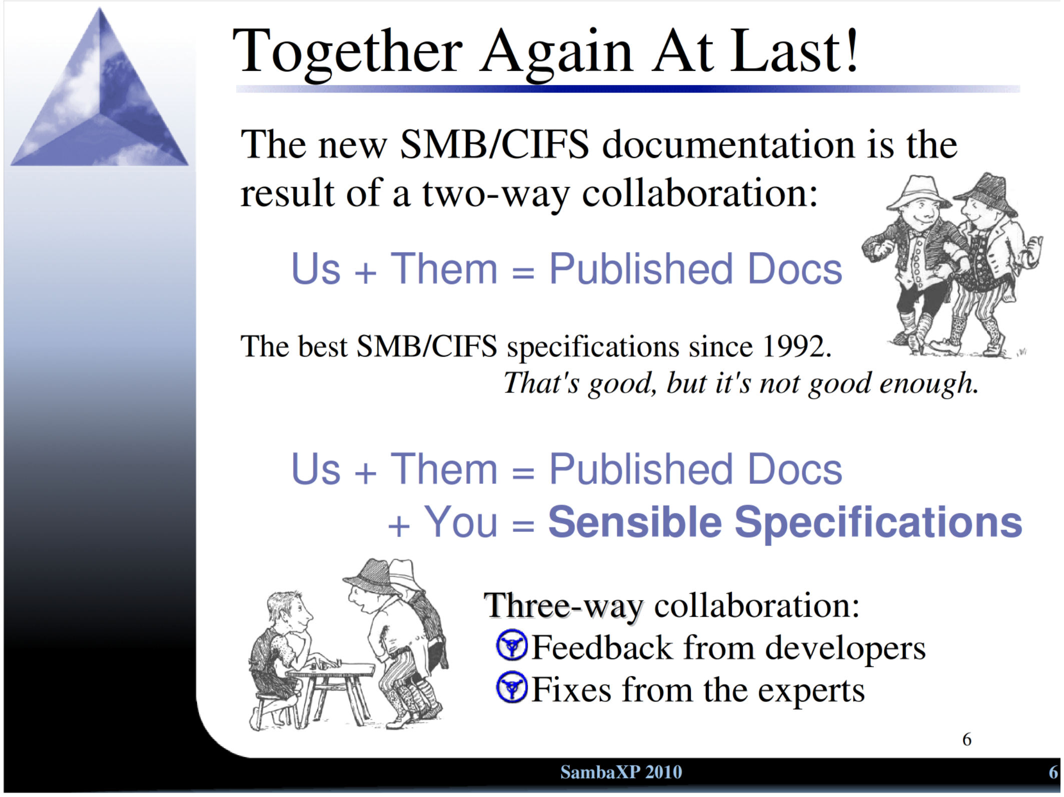
...and third-party SMB/CIFS developers who are also part of the community.

SambaXP is not just about Samba. It is also about products and services built with Samba, and about collaboration and relationships within the SMB/CIFS development community.

Them!

Microsoft

(They're here!)



Together Again At Last!

The new SMB/CIFS documentation is the result of a two-way collaboration:

Us + Them = Published Docs

The best SMB/CIFS specifications since 1992.

That's good, but it's not good enough.

Us + Them = Published Docs
+ You = **Sensible Specifications**

Three-way collaboration:

- Feedback from developers
- Fixes from the experts

6

A two-way collaboration--Us and Them--managed to produce the published specifications.

It will, however, take a three-way collaboration--Us, Them, and You--to really hammer these specifications into shape. These are *live documents*. Input from document users is critical to getting things right.

There are too many bugs and omissions in Leach/Naik and in the SNIA CIFS TR. The protocol is too big to get it all right in one go. Community feedback and fixes are the only way we can catch the remaining bugs.

The 1992 specifications to which we are referring are the X/Open c195 and c209 documents:

[XOPEN-IPC] (c195)
X/Open CAE Specification
IPC Mechanisms for SMB
December 1991, X/Open Company Limited
ISBN: 1 872630 28 6

[XOPEN-SMB] (c209)
X/Open CAE Specification
Protocols for X/Open PC Interworking: SMB, Version 2
September 1992, X/Open Company Limited
ISBN: 1 872630 45 6

Note that these are actual protocol specifications published by a *bone fide* standards organization. They are the only actual SMB standards; they cover SMB from the Core Protocol through to LAN Mangager 2.0. They do not cover LAN Manager 2.1 or NT LAN Manager.

The SNIA CIFS document is a Technical Reference, not a specification, but it was an improvement over the unfinished Leach/Naik drafts.

Where to Start

For those unfamiliar with ***Microsoft** Open Specifications*:

- http://www.microsoft.com/openspecifications/

For SMB/CIFS and SMB2:

[MS-CIFS]

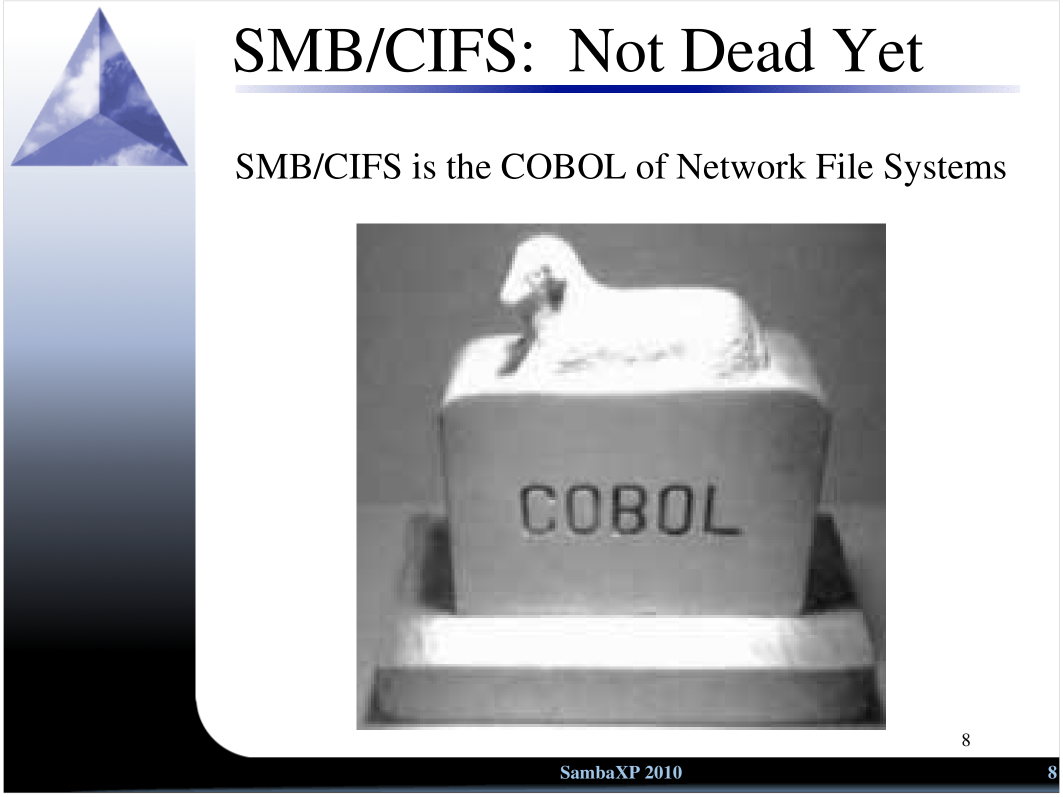
- http://msdn.microsoft.com/en-us/library/ee442092.aspx

[MS-SMB]

- http://msdn.microsoft.com/en-us/library/cc246231.aspx

[MS-SMB2] *(not our fault, we didn't write it)*

- http://msdn.microsoft.com/en-us/library/cc246482.aspx



SMB/CIFS: Not Dead Yet

SMB/CIFS is the COBOL of Network File Systems

It's not dead yet.

SMB/CIFS is fading away, though, isn't it? Won't SMB2 replace it in time?

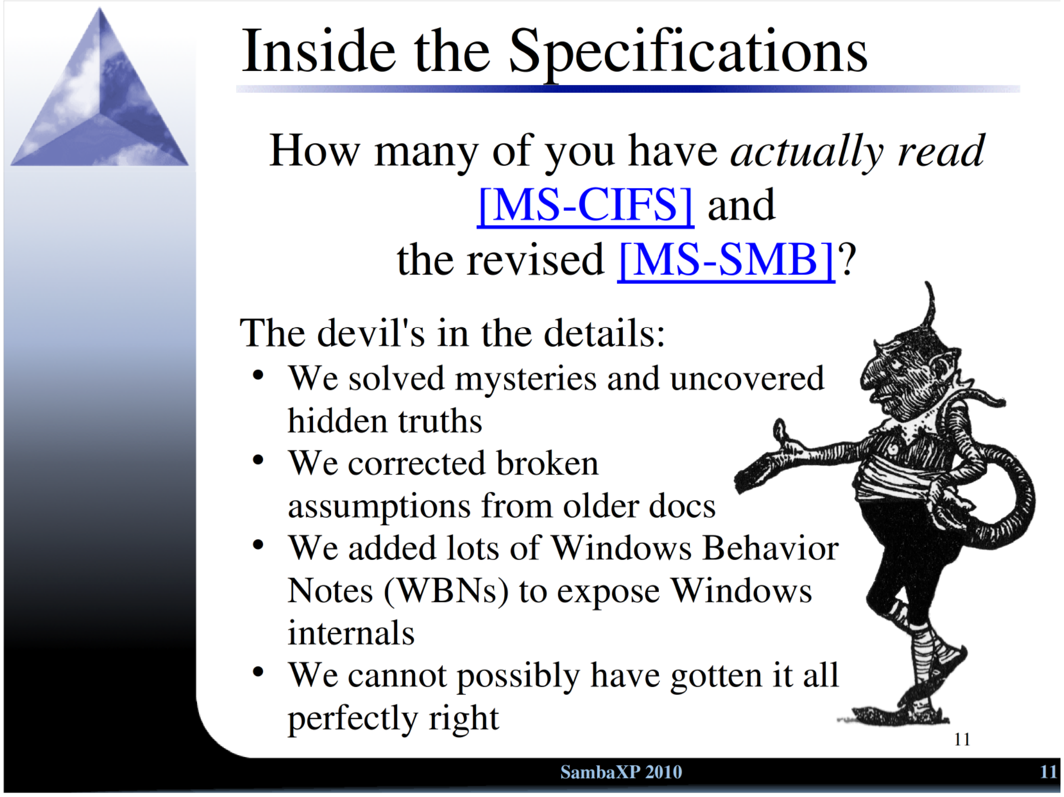
- We can hope so.
- Consider all of the NAS devices being produced and sold, particularly at the low end.
- Consider all of the Windows XP systems (and even Windows 98, etc.) still in use.
- Consider that we still see OS/2 questions on the Samba-Technical mailing list.

Inside the Specs

The Taming of the Template

Inside the Specifications

How many of you have *actually read* [MS-CIFS] and the revised [MS-SMB]?



Inside the Specifications

How many of you have *actually read* [MS-CIFS] and the revised [MS-SMB]?

The devil's in the details:

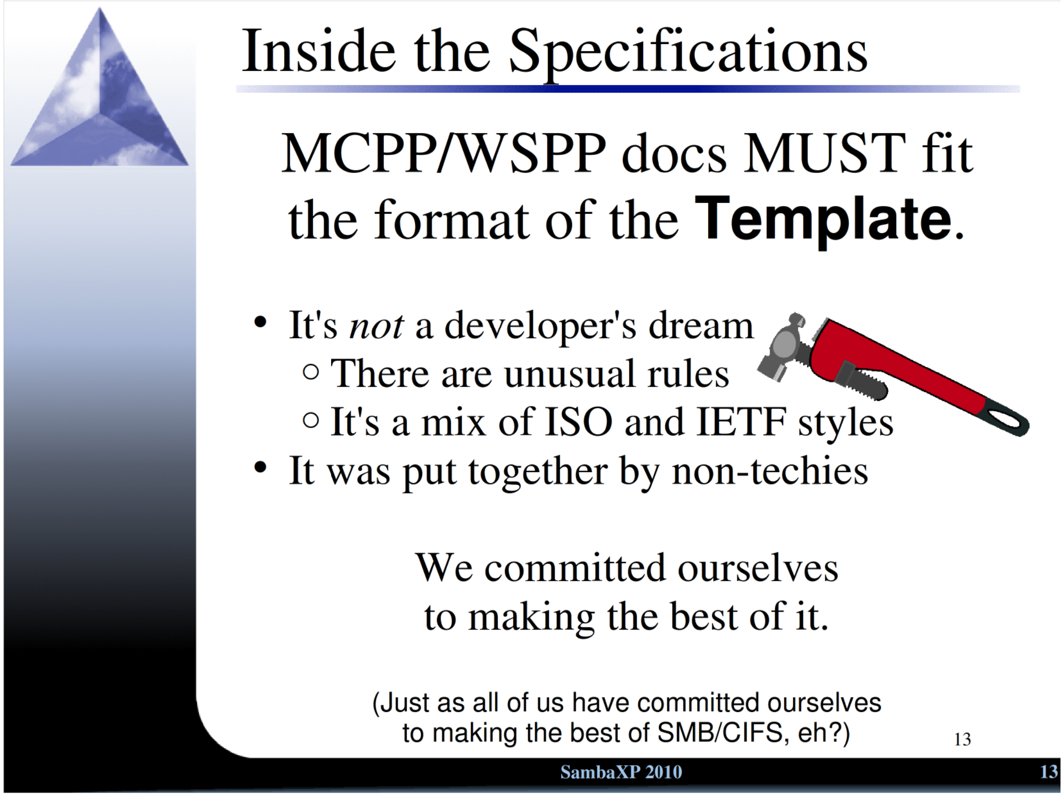
- We solved mysteries and uncovered hidden truths
- We corrected broken assumptions from older docs
- We added lots of Windows Behavior Notes (WBNs) to expose Windows internals
- We cannot possibly have gotten it all perfectly right

Compare against the older NT LAN Manager docs (Leach/Naik and SNIA CIFS). The newer docs provide much more depth.

Inside the Specifications

How to Read MCPP/WSPP Documents

Hot cup of strong tea recommended.



Inside the Specifications

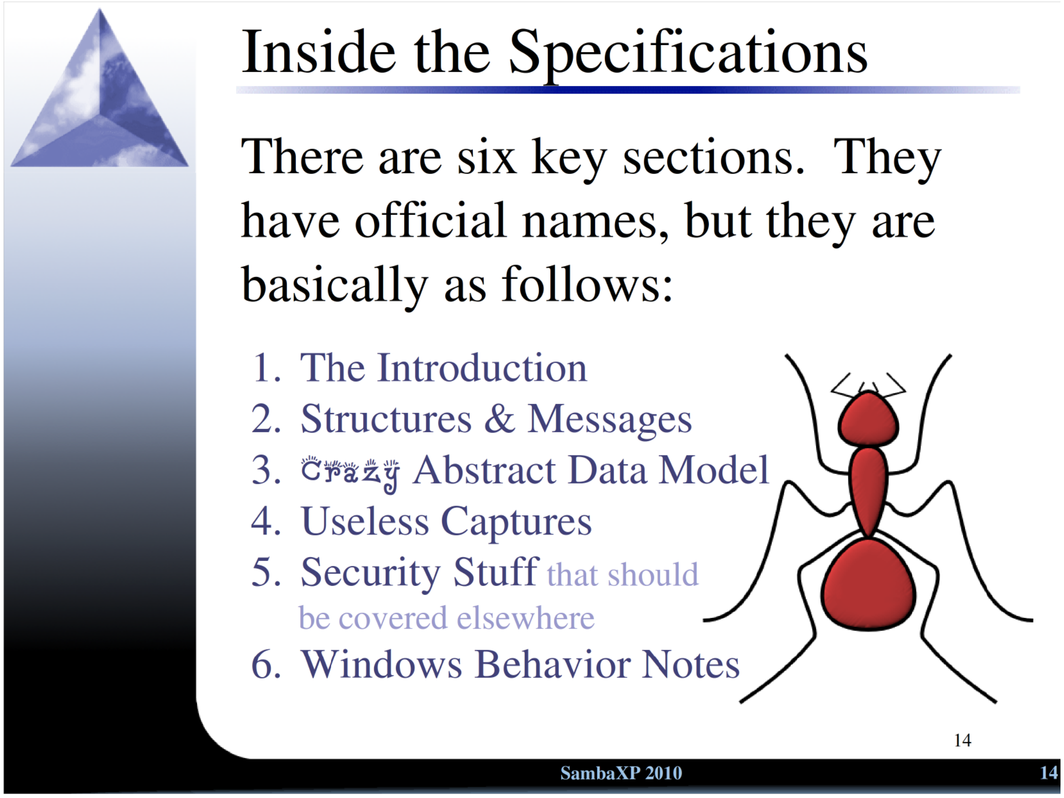
MCPP/WSPP docs MUST fit the format of the **Template**.

- It's *not* a developer's dream
  - There are unusual rules
  - It's a mix of ISO and IETF styles
- It was put together by non-techies

We committed ourselves
to making the best of it.

(Just as all of us have committed ourselves
to making the best of SMB/CIFS, eh?)

There are actually several templates. We used the “block template”.

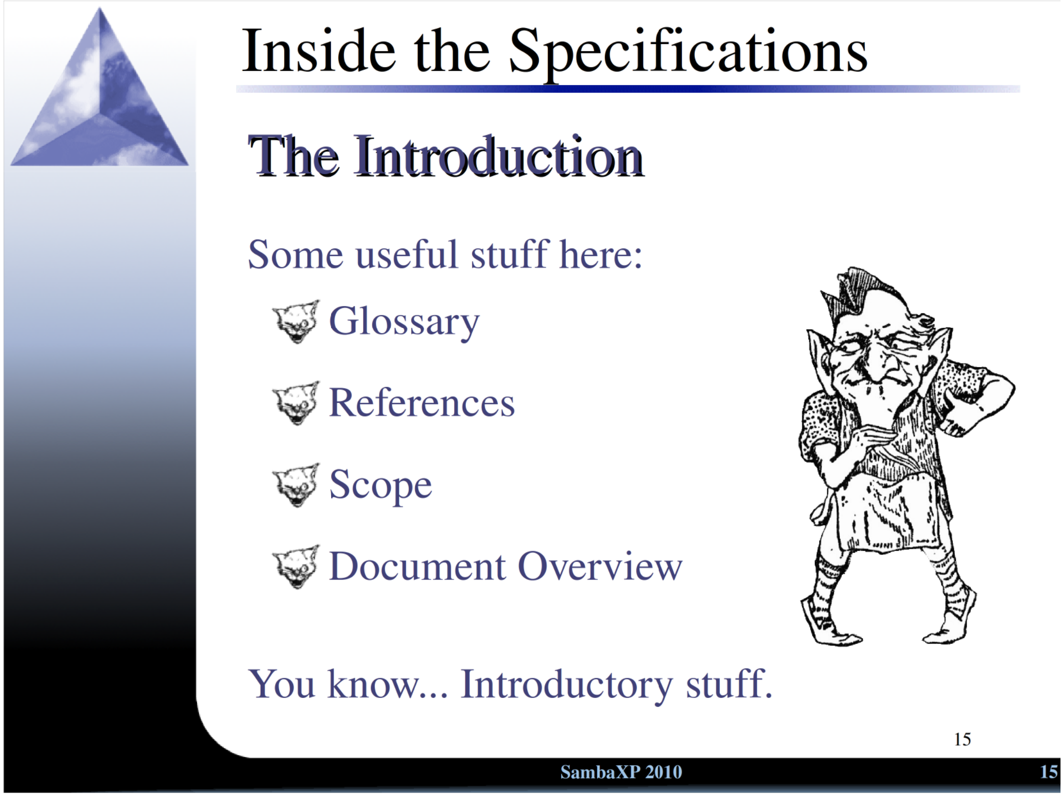


Inside the Specifications

There are six key sections. They have official names, but they are basically as follows:

1. The Introduction
2. Structures & Messages
3. Crazy Abstract Data Model
4. Useless Captures
5. Security Stuff that should be covered elsewhere
6. Windows Behavior Notes

Well, there's also section 7 which covers document changes...



Inside the Specifications

The Introduction

Some useful stuff here:

- Glossary
- References
- Scope
- Document Overview

You know... Introductory stuff.

This slide is fairly sparse, but how much detail do you really need to introduce an Introduction?

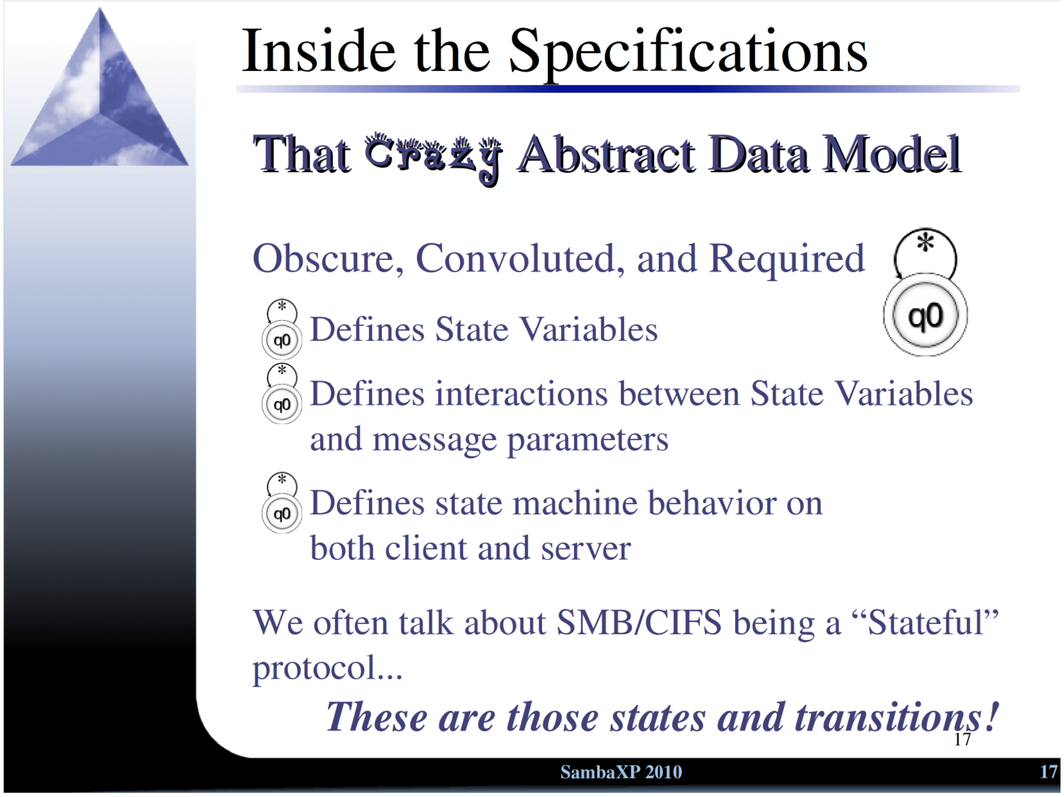
Inside the Specifications

Structures and Messages

Lots of useful stuff here:

- Transport Overview
  - References to Transport docs.
- Defined Constants
  - Error Codes, Command Codes, etc.
- Basic SMB structures (InfoLevels, etc.)
- Per-Command/Subcommand Message Layout
  - Field Definitions

Syntactic details and lots of basic relationships between fields—the stuff that most geeks want.



Inside the Specifications

That Crazy Abstract Data Model

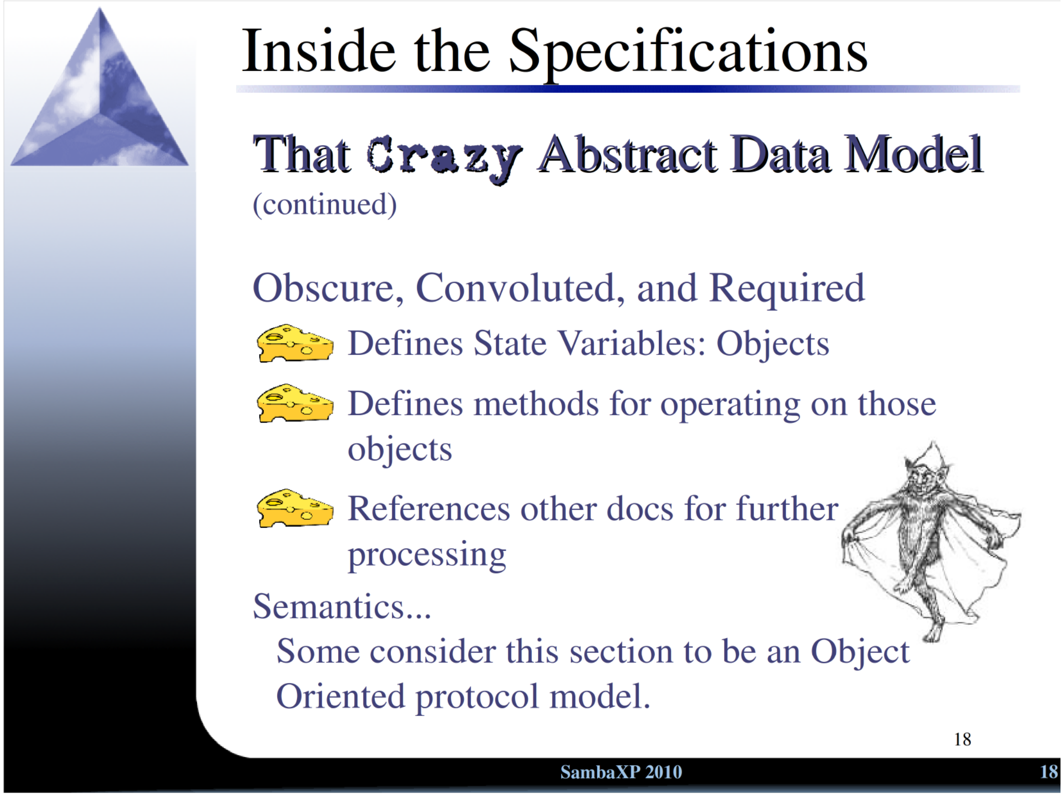
Obscure, Convoluted, and Required

- Defines State Variables
- Defines interactions between State Variables and message parameters
- Defines state machine behavior on both client and server

We often talk about SMB/CIFS being a “Stateful” protocol...

These are those states and transitions!

Other than source code itself, this is the first attempt (of which we are aware) to formally define the state information required by CIFS.



Inside the Specifications

That Crazy Abstract Data Model

(continued)

Obscure, Convoluted, and Required

- Defines State Variables: Objects
- Defines methods for operating on those objects
- References other docs for further processing

Semantics...

Some consider this section to be an Object Oriented protocol model.

18

SambaXP 2010 18

Many of those who support these documents see them from an Object Oriented perspective:

- Section 1 provides the initial definitions and required references
- Section 2 defines data types
- Section 3 provides the methods

Under this model, [MS-SMB] is a descendant of [MS-CIFS].

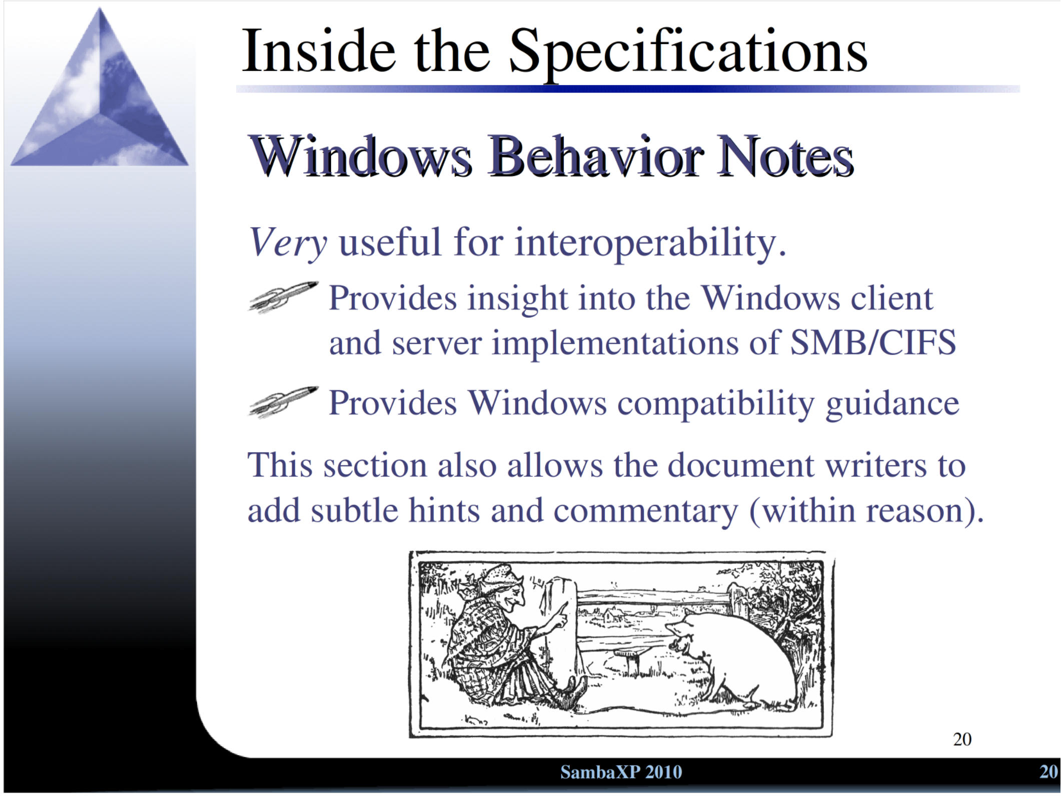
Inside the Specifications

Useless Captures, and Redundant Security Stuff

Important to the Template!

- Developers can grab their own captures
- Security information should be well described elsewhere

...but these do not get in the way and may prove useful to someone, somewhere, somehow, some day.



Inside the Specifications

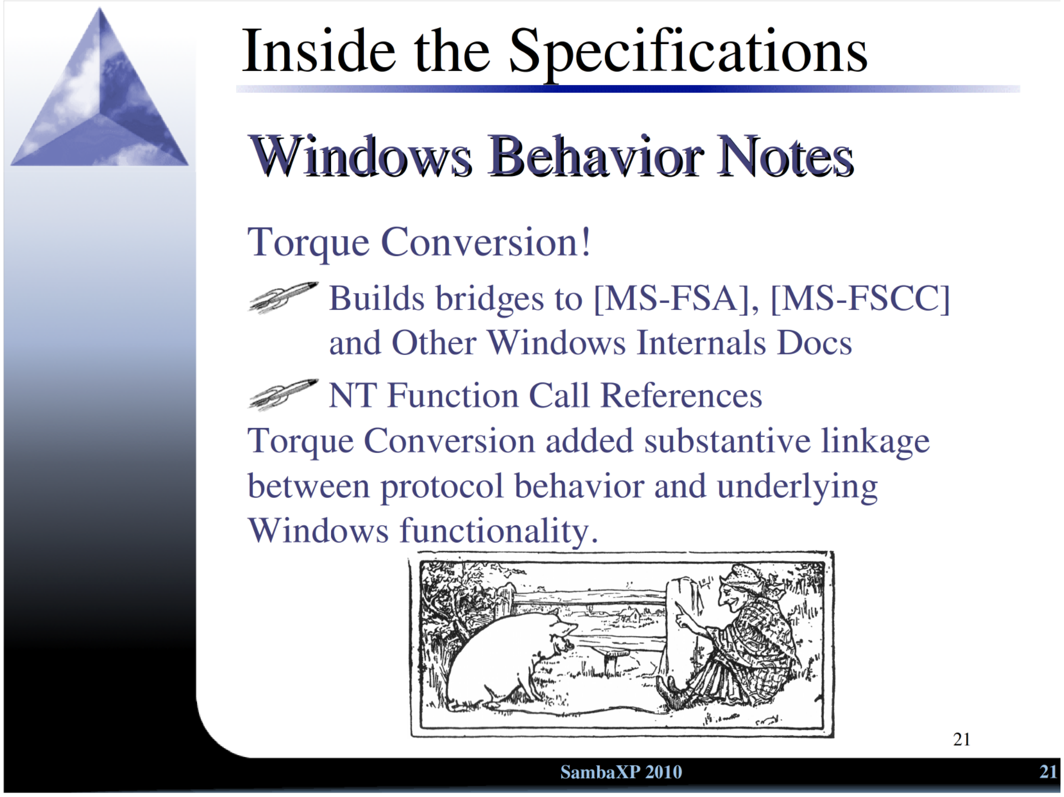
Windows Behavior Notes

Very useful for interoperability.

- Provides insight into the Windows client and server implementations of SMB/CIFS
- Provides Windows compatibility guidance

This section also allows the document writers to add subtle hints and commentary (within reason).

This is the section that answers the question: “How does Windows do this?”



Inside the Specifications

Windows Behavior Notes

Torque Conversion!

- Builds bridges to [MS-FSA], [MS-FSCC] and Other Windows Internals Docs
- NT Function Call References

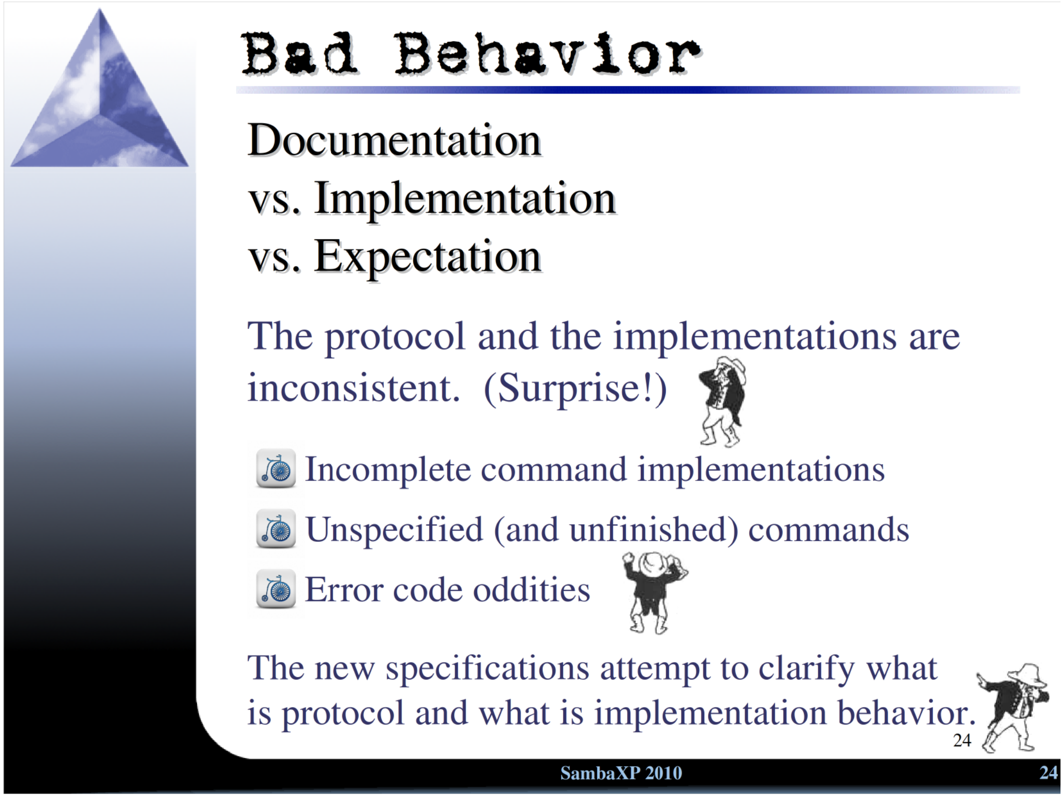
Torque Conversion added substantive linkage between protocol behavior and underlying Windows functionality.

I think that it was Tridge who coined the term “Torque Conversion”, but it might have been Jim Pinkerton from Microsoft.

PUPPY!

Bad Behavior

23



Bad Behavior

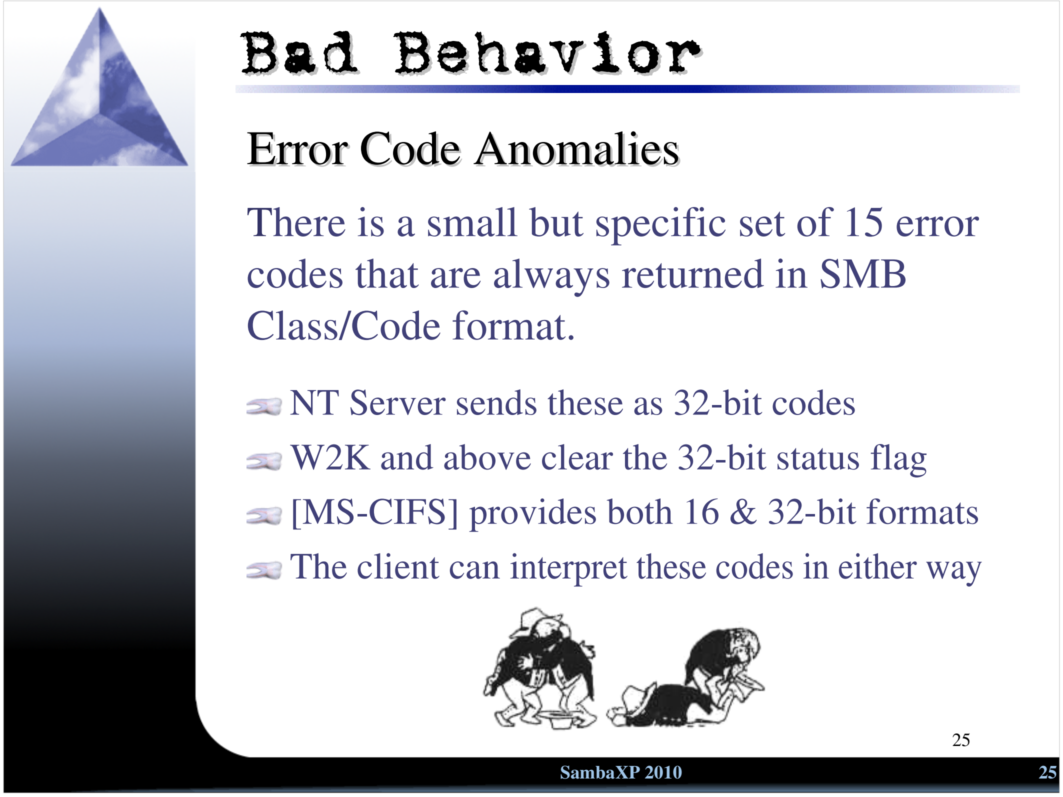
Documentation
vs. Implementation
vs. Expectation

The protocol and the implementations are inconsistent. (Surprise!)

- Incomplete command implementations
- Unspecified (and unfinished) commands
- Error code oddities

The new specifications attempt to clarify what is protocol and what is implementation behavior.

“In theory, theory and practice are the same. In practice, they're not.”



Bad Behavior

Error Code Anomalies

There is a small but specific set of 15 error codes that are always returned in SMB Class/Code format.

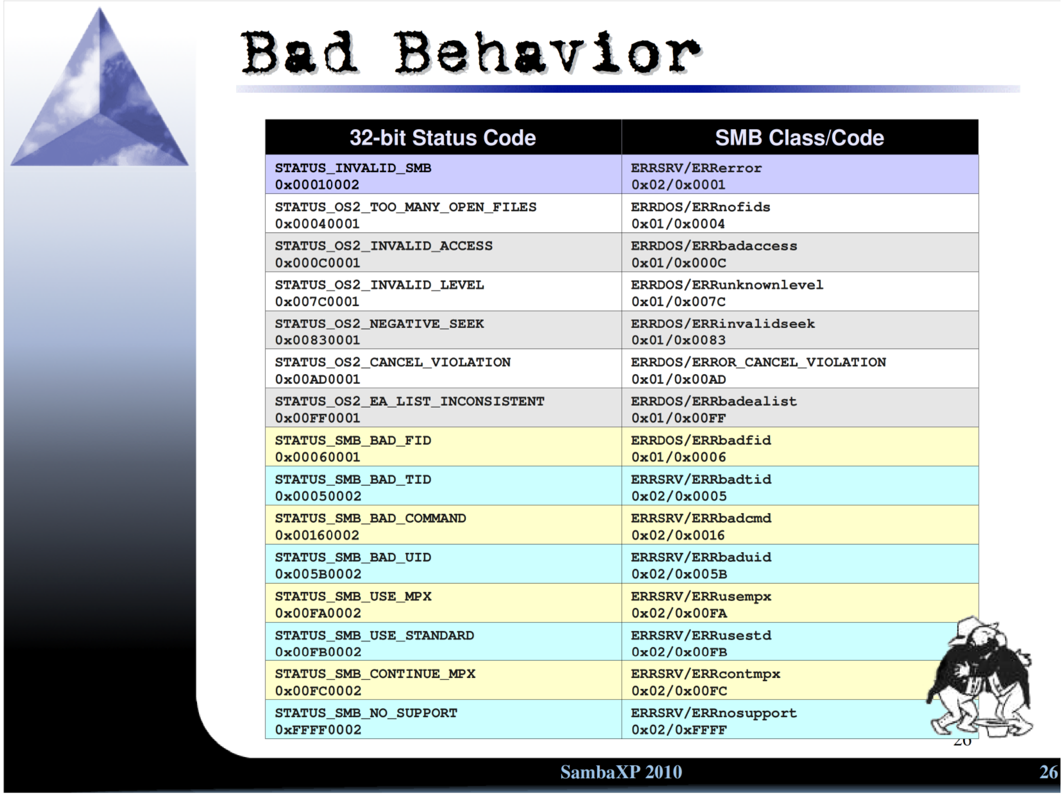
- NT Server sends these as 32-bit codes
- W2K and above clear the 32-bit status flag
- [MS-CIFS] provides both 16 & 32-bit formats
- The client can interpret these codes in either way

For this set of 15 codes...

If 32-bit status codes have been negotiated, and the SMB request has the 32-bit status flag value set (indicating that it wants a 32-bit status in the response), Windows NT sets the 32-bit flag in the response (does not change the flag setting).

Windows 2000 and above clear the 32-bit flag in the response, even if 32-bit status values were negotiated and requested.

So which is it? Are these 32-bit values or Class/Code pairs?



Bad Behavior

32-bit Status Code	SMB Class/Code
STATUS_INVALID_SMB 0x00010002	ERRSRV/ERRerror 0x02/0x0001
STATUS_OS2_TOO_MANY_OPEN_FILES 0x00040001	ERRDOS/ERRnofids 0x01/0x0004
STATUS_OS2_INVALID_ACCESS 0x000C0001	ERRDOS/ERRbadaccess 0x01/0x000C
STATUS_OS2_INVALID_LEVEL 0x007C0001	ERRDOS/ERRunknownlevel 0x01/0x007C
STATUS_OS2_NEGATIVE_SEEK 0x00830001	ERRDOS/ERRinvalidseek 0x01/0x0083
STATUS_OS2_CANCEL_VIOLATION 0x00AD0001	ERRDOS/ERROR_CANCEL_VIOLATION 0x01/0x00AD
STATUS_OS2_EA_LIST_INCONSISTENT 0x00FF0001	ERRDOS/ERRbadealist 0x01/0x00FF
STATUS_SMB_BAD_FID 0x00060001	ERRDOS/ERRbadfid 0x01/0x0006
STATUS_SMB_BAD_TID 0x00050002	ERRSRV/ERRbadtid 0x02/0x0005
STATUS_SMB_BAD_COMMAND 0x00160002	ERRSRV/ERRbadcmd 0x02/0x0016
STATUS_SMB_BAD_UID 0x005B0002	ERRSRV/ERRbaduid 0x02/0x005B
STATUS_SMB_USE_MPX 0x00FA0002	ERRSRV/ERRusempx 0x02/0x00FA
STATUS_SMB_USE_STANDARD 0x00FB0002	ERRSRV/ERRusestd 0x02/0x00FB
STATUS_SMB_CONTINUE_MPX 0x00FC0002	ERRSRV/ERRcontmpx 0x02/0x00FC
STATUS_SMB_NO_SUPPORT 0xFFFF0002	ERRSRV/ERRnosupport 0x02/0xFFFF

26

SambaXP 2010 26

Okay, so there's no way anyone's going to be able to read that chart from the projection screen. Sorry.

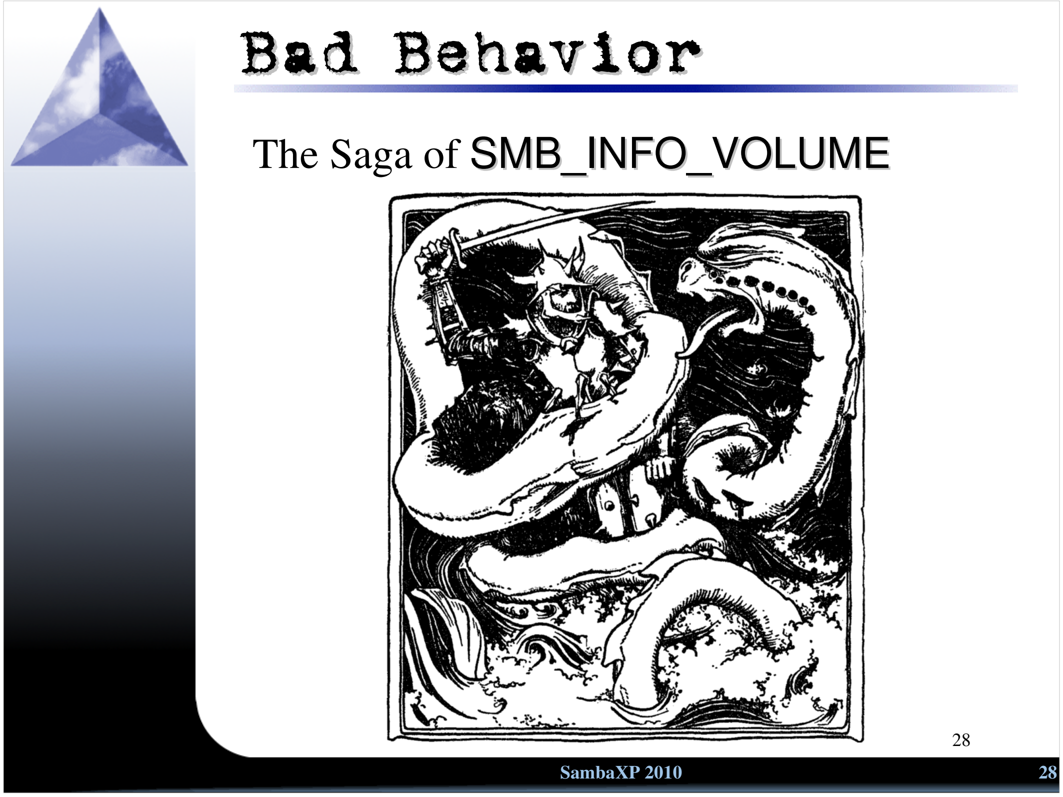
Internally, the Windows SMB server handles *all* status values as 32-bit codes. If the client has negotiated 16-bit class/code pairs, translation is handled just before the response packet is sent out the door.

This set of 15 codes represents the only status values for which there is no 32-bit mapping. Internally, these status values are represented using a block of reserved 32-bit status codes (from the set reserved for vendor use). Those internal codes are not allowed out onto the wire.

So... for this set of 15 status values, the old-style class/code pair is sent on the wire even if 32-bit status codes have been negotiated. Thing is... the wire formats do not collide with any other 32-bit status code, so you can interpret them either way.

These codes are commonly returned by Trans2 calls, and vintage file I/O calls such as SMB_COM_OPEN, SMB_COM_CREATE, and SMB_COM_SEEK. It is likely that it was expected that they would only be returned to downlevel clients.

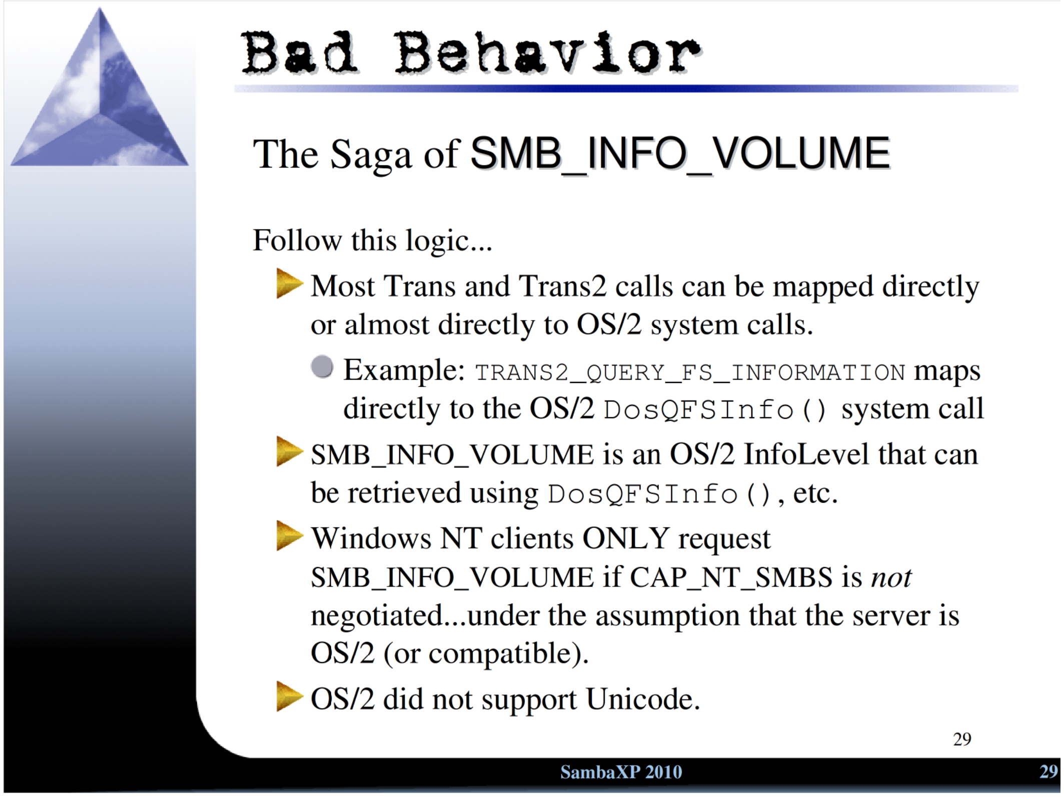
I threw this image into the slide deck for no particular reason.



Bad Behavior

The Saga of SMB_INFO_VOLUME

This saga won't be all that new to Samba developers.



Bad Behavior

The Saga of SMB_INFO_VOLUME

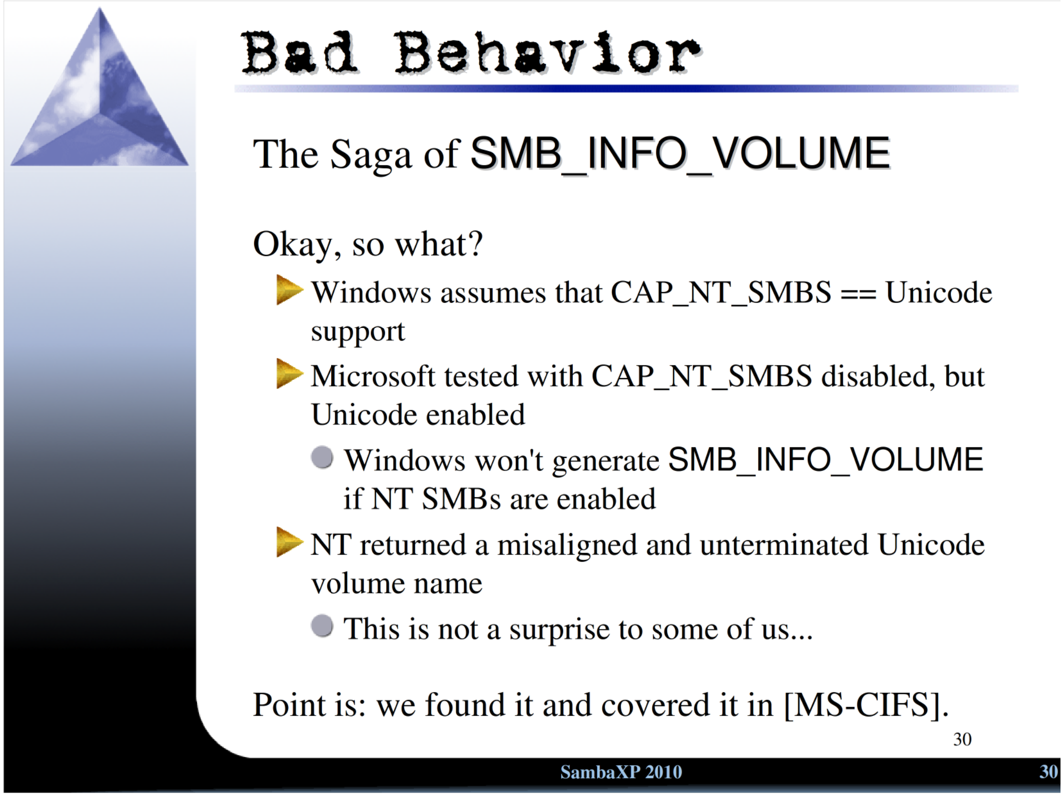
Follow this logic...

- Most Trans and Trans2 calls can be mapped directly or almost directly to OS/2 system calls.
  - Example: `TRANS2_QUERY_FS_INFORMATION` maps directly to the OS/2 `DosQFSInfo()` system call
- SMB_INFO_VOLUME is an OS/2 InfoLevel that can be retrieved using `DosQFSInfo()`, etc.
- Windows NT clients ONLY request SMB_INFO_VOLUME if CAP_NT_SMBS is *not* negotiated...under the assumption that the server is OS/2 (or compatible).
- OS/2 did not support Unicode.

Trans and Trans2 calls really do map to OS/2 calls, except in cases in which NT overwrote an existing (and unused) Trans2 call with a new one. Yes, that happened.

Windows NT, however, adds support for additional (NT-specific) InfoLevels in the existing calls.

NT also only supports the older InfoLevel requests if needed, or if they were unsure. NT, for instance, does not support TRANS2_SET_FS_INFORMATION (W2K does, but only for NT pass-through InfoLevels).



Bad Behavior

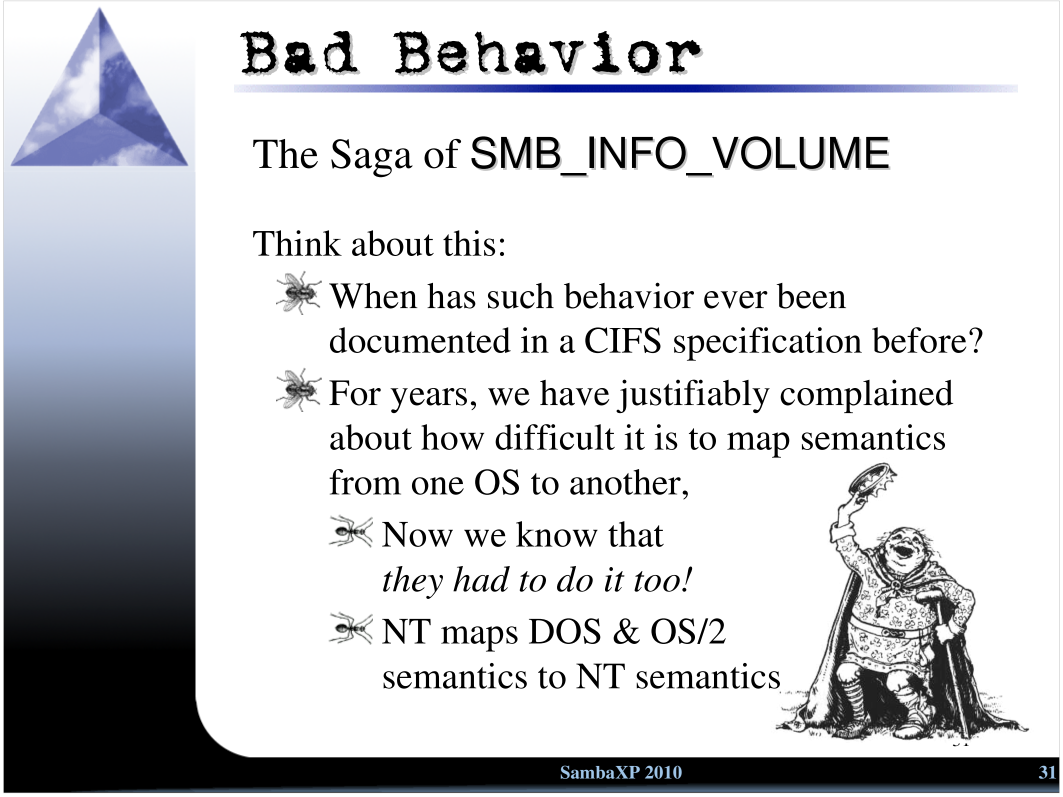
The Saga of SMB_INFO_VOLUME

Okay, so what?

- Windows assumes that CAP_NT_SMBS == Unicode support
- Microsoft tested with CAP_NT_SMBS disabled, but Unicode enabled
  - Windows won't generate SMB_INFO_VOLUME if NT SMBs are enabled
- NT returned a misaligned and unterminated Unicode volume name
  - This is not a surprise to some of us...

Point is: we found it and covered it in [MS-CIFS].

Windows clients test for CAP_NT_SMBS. If has not been negotiated, then the older SMB_INFO_VOLUME InfoLevel is requested. If it has been negotiated, then a newer NT InfoLevel is sent. The client does not check whether Unicode has been enabled or not.



Bad Behavior

The Saga of SMB_INFO_VOLUME

Think about this:

- When has such behavior ever been documented in a CIFS specification before?
- For years, we have justifiably complained about how difficult it is to map semantics from one OS to another,
  - Now we know that *they had to do it too!*
  - NT maps DOS & OS/2 semantics to NT semantics

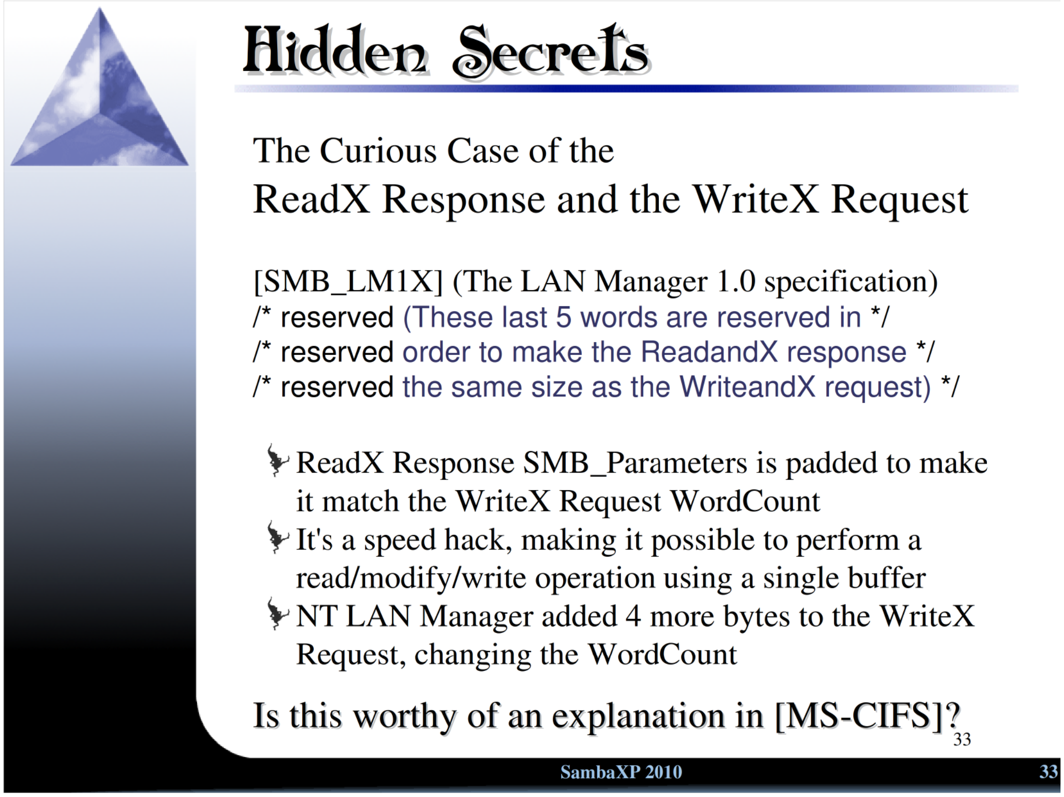
In last year's presentation, we stated that the Windows SMB implementation is a “thin layer” between the wire and the Windows OS. As we dug deeper, however, we discovered that is only true for NT SMBs and NT_Trans transactions.

For all DOS and OS/2 SMB calls, the Windows server has to translate semantics. Much of the translation is done by the server itself, though there are some emulation functions offered by the OS. These translations are very similar to the kind of semantics translations that other implementations (e.g., Samba) must perform.

A lot of information about these conversions was revealed while we were adding “Torque Conversion” to [MS-CIFS].

We originally focused on NT commands because the others had been previously documented. (See [XOPEN-SMB].)

Hidden Secrets



Hidden Secrets

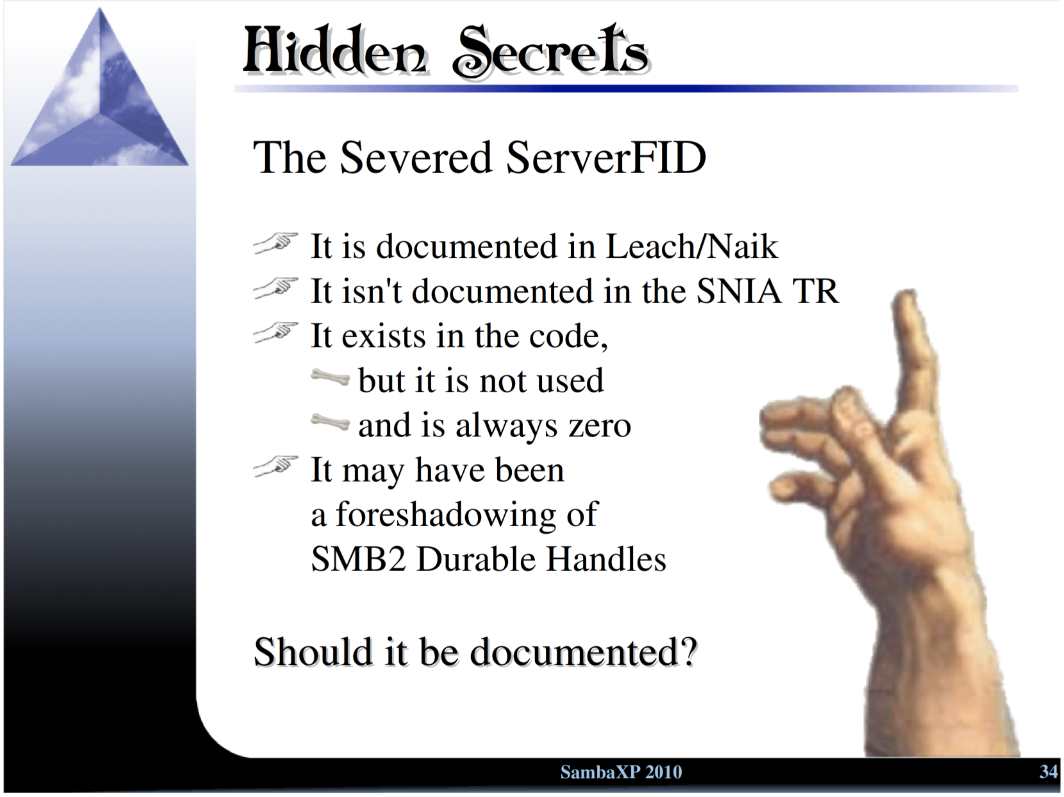
The Curious Case of the ReadX Response and the WriteX Request

[SMB_LM1X] (The LAN Manager 1.0 specification)
/* reserved (These last 5 words are reserved in */
/* reserved order to make the ReadandX response */
/* reserved the same size as the WriteandX request) */

- ReadX Response SMB_Parameters is padded to make it match the WriteX Request WordCount
- It's a speed hack, making it possible to perform a read/modify/write operation using a single buffer
- NT LAN Manager added 4 more bytes to the WriteX Request, changing the WordCount

Is this worthy of an explanation in [MS-CIFS]?

Older client code may actually rely upon this behavior, but then older client code would not negotiate NT LM 0.12.



Hidden Secrets

The Severed ServerFID

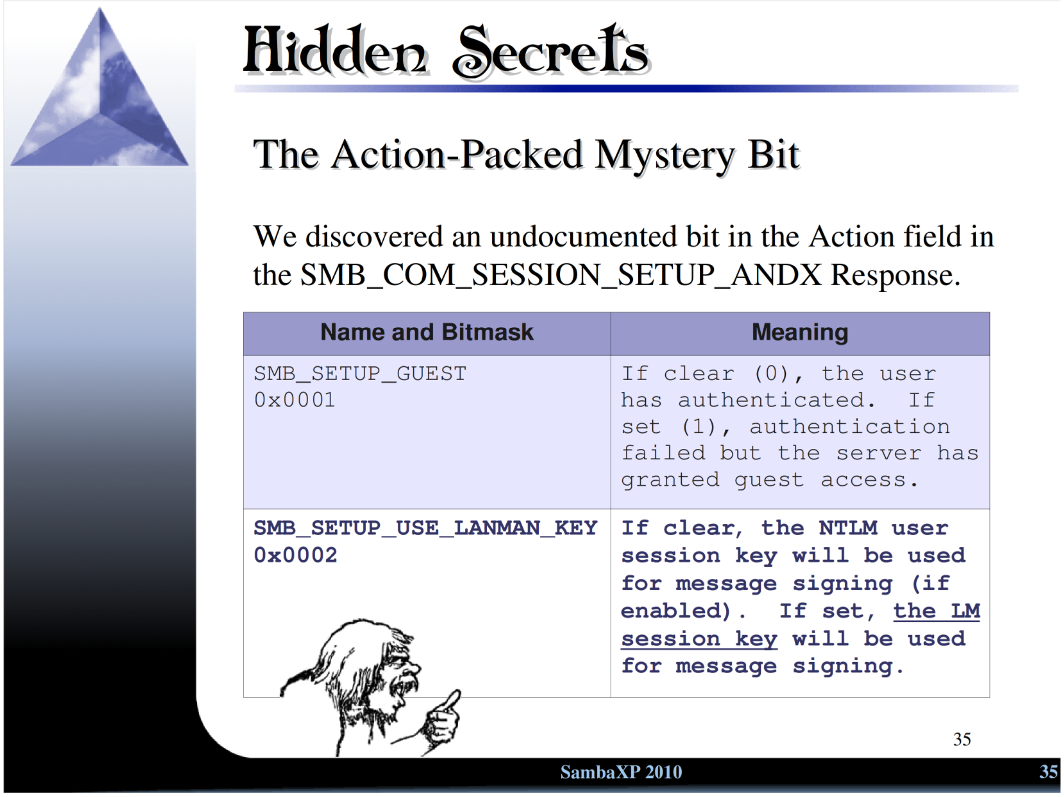
- It is documented in Leach/Naik
- It isn't documented in the SNIA TR
- It exists in the code,
  - but it is not used
  - and is always zero
- It may have been a foreshadowing of SMB2 Durable Handles

Should it be documented?

This field is currently documented in [MS-SMB], but there has been a great deal of discussion about it.

- Since it is never used, should it be documented as a simple Reserved block?
- Since it's in Leach/Naik, and in the code, shouldn't we acknowledge it?
- Does [MS-SMB] represent an older protocol with historic precedent, or should we only report wire behavior?

We believe that historic precedence should be recognized and that code quirks should be exposed, so we documented the field with a WBN explaining its non-usage.



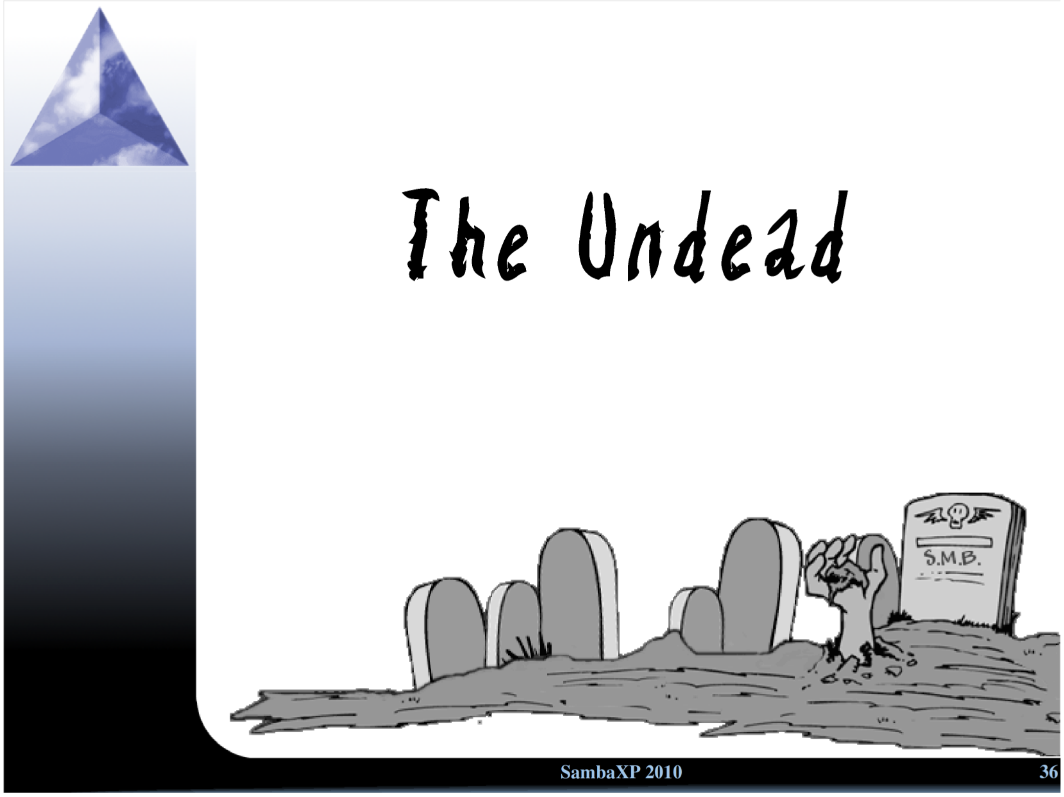
Hidden Secrets

The Action-Packed Mystery Bit

We discovered an undocumented bit in the Action field in the SMB_COM_SESSION_SETUP_ANDX Response.

Name and Bitmask	Meaning
SMB_SETUP_GUEST 0x0001	If clear (0), the user has authenticated. If set (1), authentication failed but the server has granted guest access.
SMB_SETUP_USE_LANMAN_KEY 0x0002	**If clear, the NTLM user session key will be used for message signing (if enabled). If set, the LM session key will be used for message signing.**

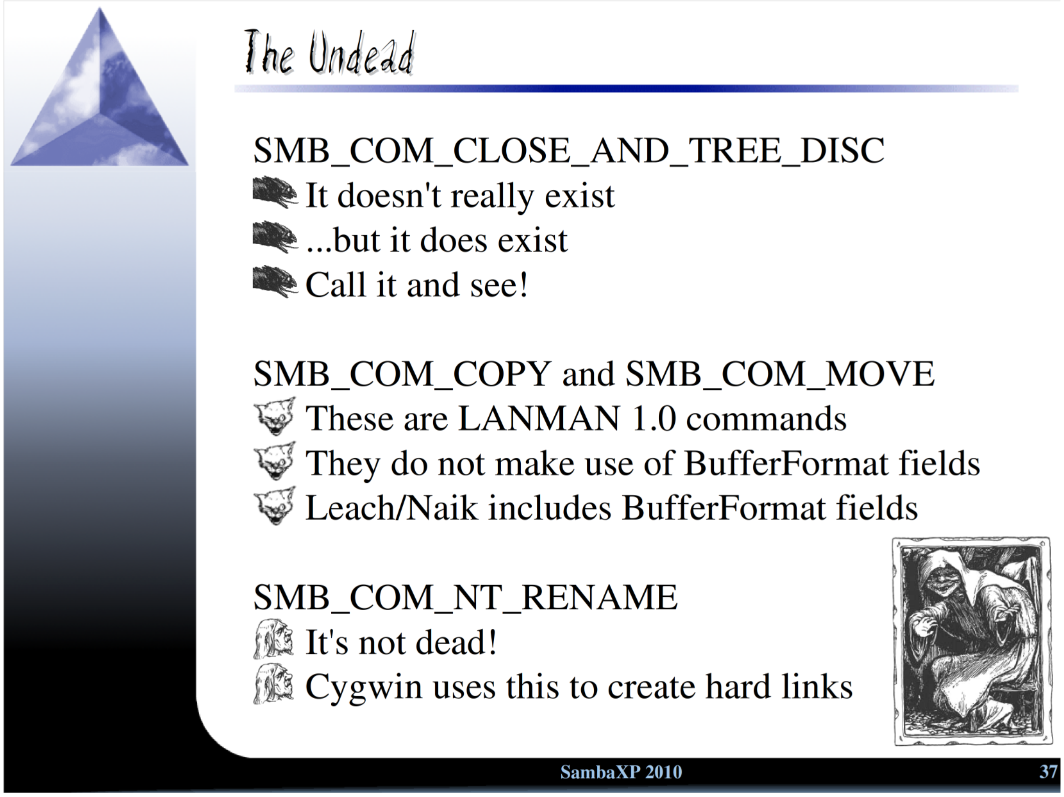
This has implications for signing, as you can imagine.



The Undead

Zombie SMBs.

Yes. Yes, they are real. Be very afraid.



The Undead

SMB_COM_CLOSE_AND_TREE_DISC

- It doesn't really exist
- ...but it does exist
- Call it and see!

SMB_COM_COPY and SMB_COM_MOVE

- These are LANMAN 1.0 commands
- They do not make use of BufferFormat fields
- Leach/Naik includes BufferFormat fields

SMB_COM_NT_RENAME

- It's not dead!
- Cygwin uses this to create hard links

We went code diving three separate times to try to figure out whether SMB_COM_NT_RENAME really worked or not. It mostly doesn't, but only the part that does is ever called.

Hard links are not officially supported in Windows NT NTFS, but they are supported in Windows 2000 and above. The Cygwin developers wanted to be able to create hard links on NT and above, and they found an (unsupported?) NT system call that worked. That call generates NT_Rename when called across the wire.

CIFS.ORG

There are many things that won't fit or don't belong in an official specification. Now we have a place to put that stuff.

The End