Microsoft

April 17-18, 2024
sambaXP Conference

Microsoft Interoperability Track

Microsoft

FileServer TestSuite Updates 2024

Obaro Ogbo
Senior Software Engineer

Microsoft

Agenda

Overview
What's New
Feedback

Windows Protocols Test Suites Overview

What is the Windows Protocols Test Suites?

Collection of Test Suites

Each Test Suite evaluates the implementation of a family of protocols

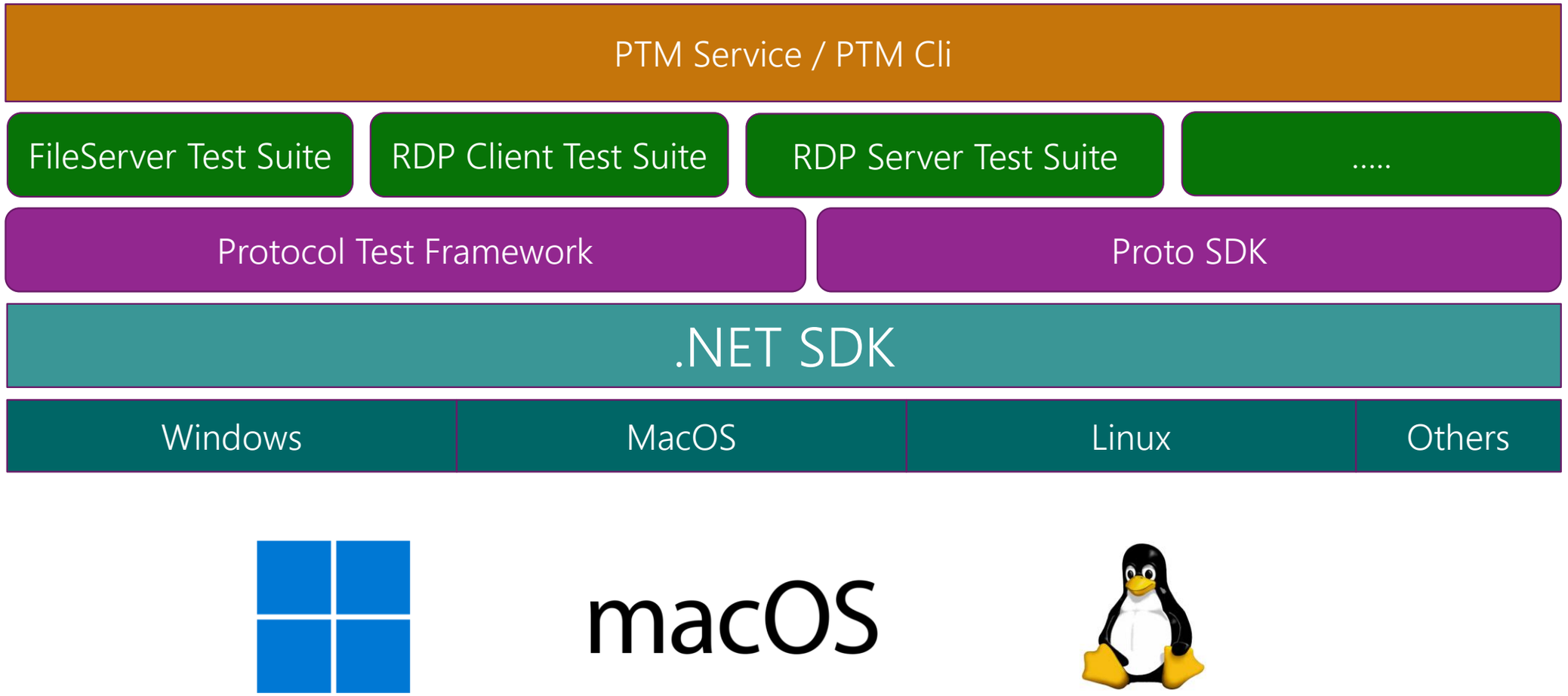
- FileServer, RDP Server, RDP Client, Active Directory, etc

Evaluates whether a protocol implementation meets certain interoperability requirements.

- Originally developed for in-house testing of Microsoft Open Specifications.
- Used to test/verify Windows behavior.
- Also used to test 3rd-party implementations.

Do not cover every protocol requirement, and do not certify an implementation, but can be a useful indication of interoperability.

Windows Protocol Test Suites Architecture



Windows Protocol Test Suites

Open sourced on GitHub since 2016
https://github.com/Microsoft/WindowsProtocolTestSuites
https://github.com/Microsoft/ProtocolTestFramework

Category	Test Suite Name	Protocols
File Sharing	FileServer	MS-SMB2, MS-FSRVP, MS-SWN, MS-DFSC, MS-RSVD, MS-SQOS, MS-FSA, MS-HVRS
	MS-SMB	MS-SMB
	MS-SMBD	MS-SMBD
	BranchCache	MS-PCCRC, MS-PCCRR, MS-PCCRTP, MS-PCHC, MS-CCROD
Security	Kerberos	MS-KILE, MS-KKDCP, MS-PAC
	MS-AZOD	MS-AZOD
Active Directory	ADFamily	MS-ADTS-LDAP, MS-ADTS-PublishDC, MS-ADTS-Schema, MS-ADTS-Security, MS-APDS, MS-DRSR, MS-FRS2, MS-LSAD, MS-LSAT, MS-NRPC, MS-SAMR
	MS-ADOD	MS-ADOD
Remote Desktop	RDP Client/RDP Server	MS-RDPBCGR, MS-RDPEDISP, MS-RDPEDYC, MS-RDPEGFX, MS-RDPEGT, MS-RDPEI, MS-RDPEMT, MS-RDPEUDP, MS-RDPEUSB, MS-RDPEVOR, MS-RDPRFX, MS-RDPELE
BYOD	MS-ADFSPIP	MS-ADFSPIP
XCA	MS-XCA	MS-XCA

FileServer Test Suite Scope

Protocols

- [MS-SMB2]:Server Message Block (SMB) Protocol Versions 2 and 3
- [MS-FSA]: File System Algorithms
- [MS-SQOS]: Storage Quality of Service Protocol
- [MS-SWN]: Service Witness Protocol
- [MS-RSVD]: Remote Shared Virtual Disk Protocol
- [MS-FSRVP]: File Server Remote VSS Protocol
- [MS-DFSC]: Distributed File System (DFS): Referral Protocol
- [MS-HVRS]: Hyper-V Remote Storage Profile

Messages

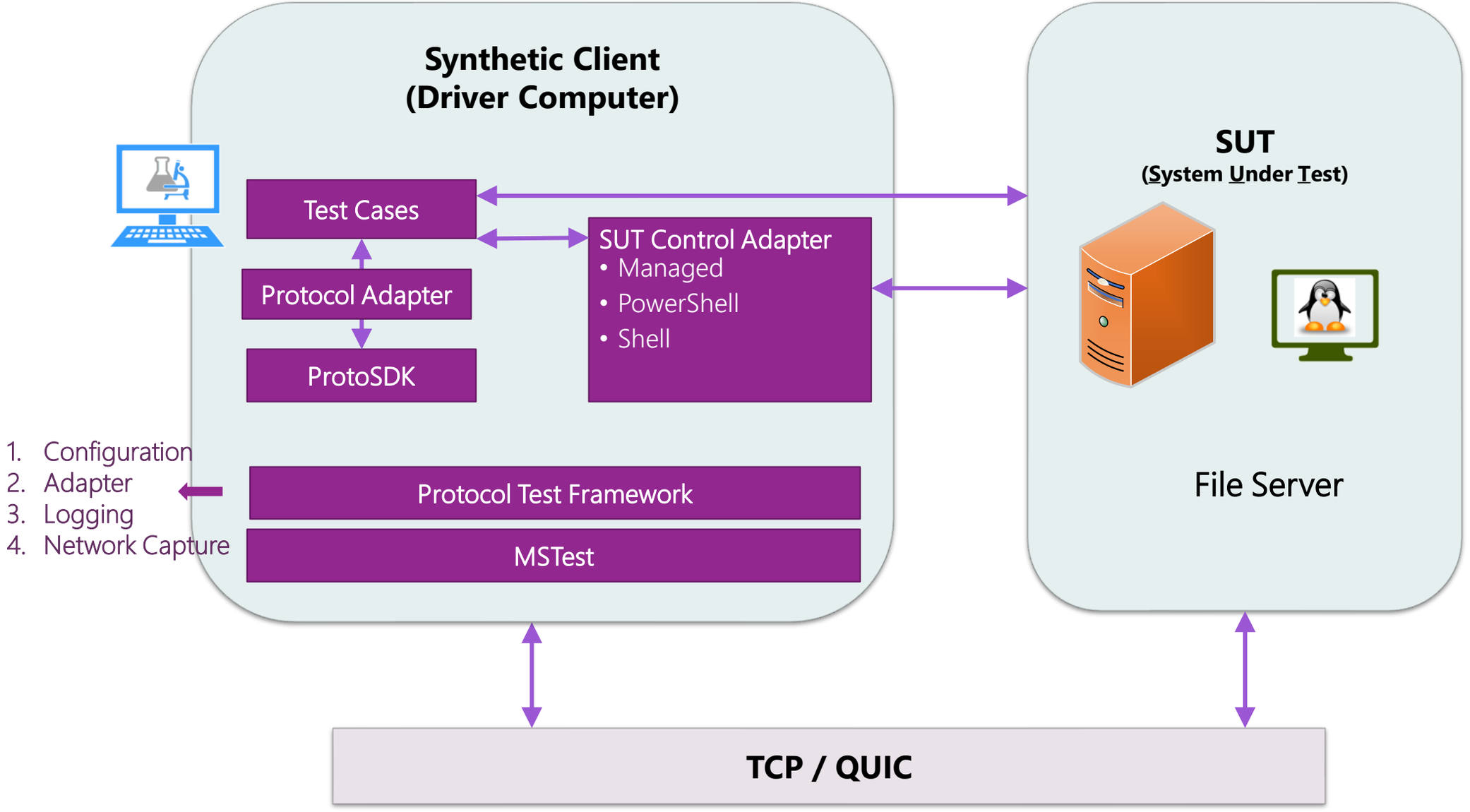
- Negotiate (Negotiate Contexts)
- Session Setup
- Tree Connect/Disconnect
- Create/Close (Create Contexts)
- IOCTL
- Query Directory
- Query Info/Set Info
- File Access
- Leasing

- Dialect 2.002
- Dialect 2.1
- Dialect 3.0
- Dialect 3.02
- Dialect 3.11

FileServer Test Suite Approach

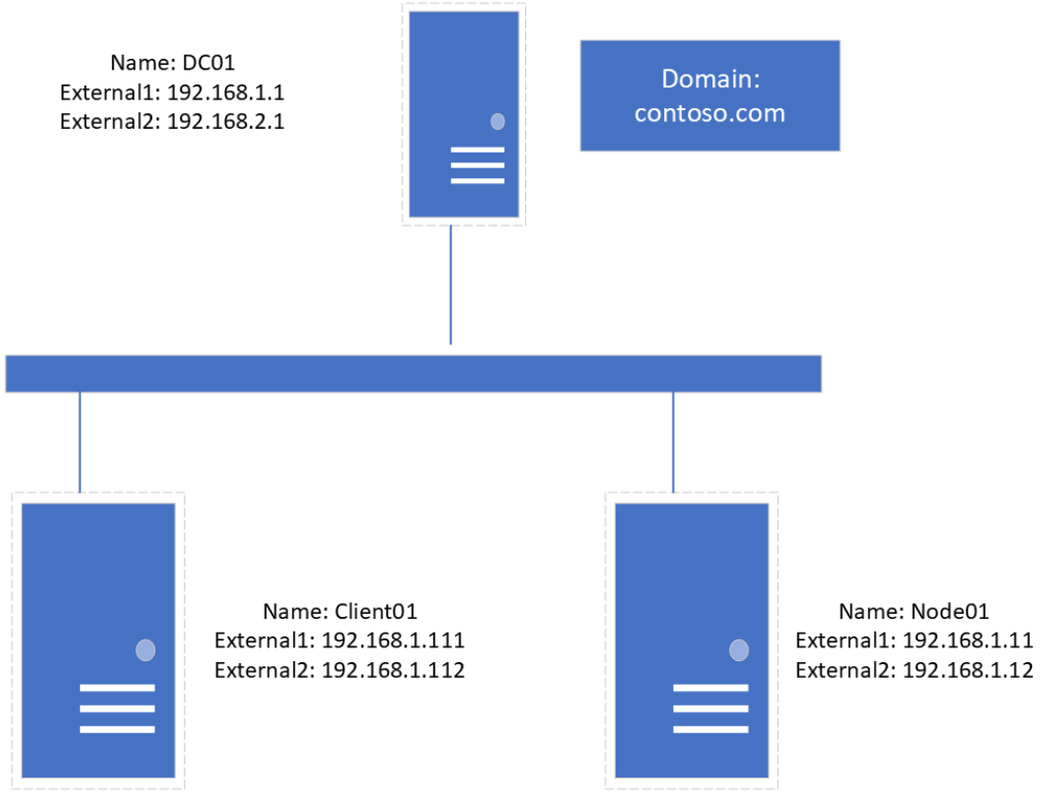
- **Traditional Test**
  - Covers basic functionalities and scenarios
  - All new test cases
- **Model-based Test**
  - 2K+ test cases generated by 13 models
  - Still supported, no more expanded

FileServer Test Suite Infrastructure

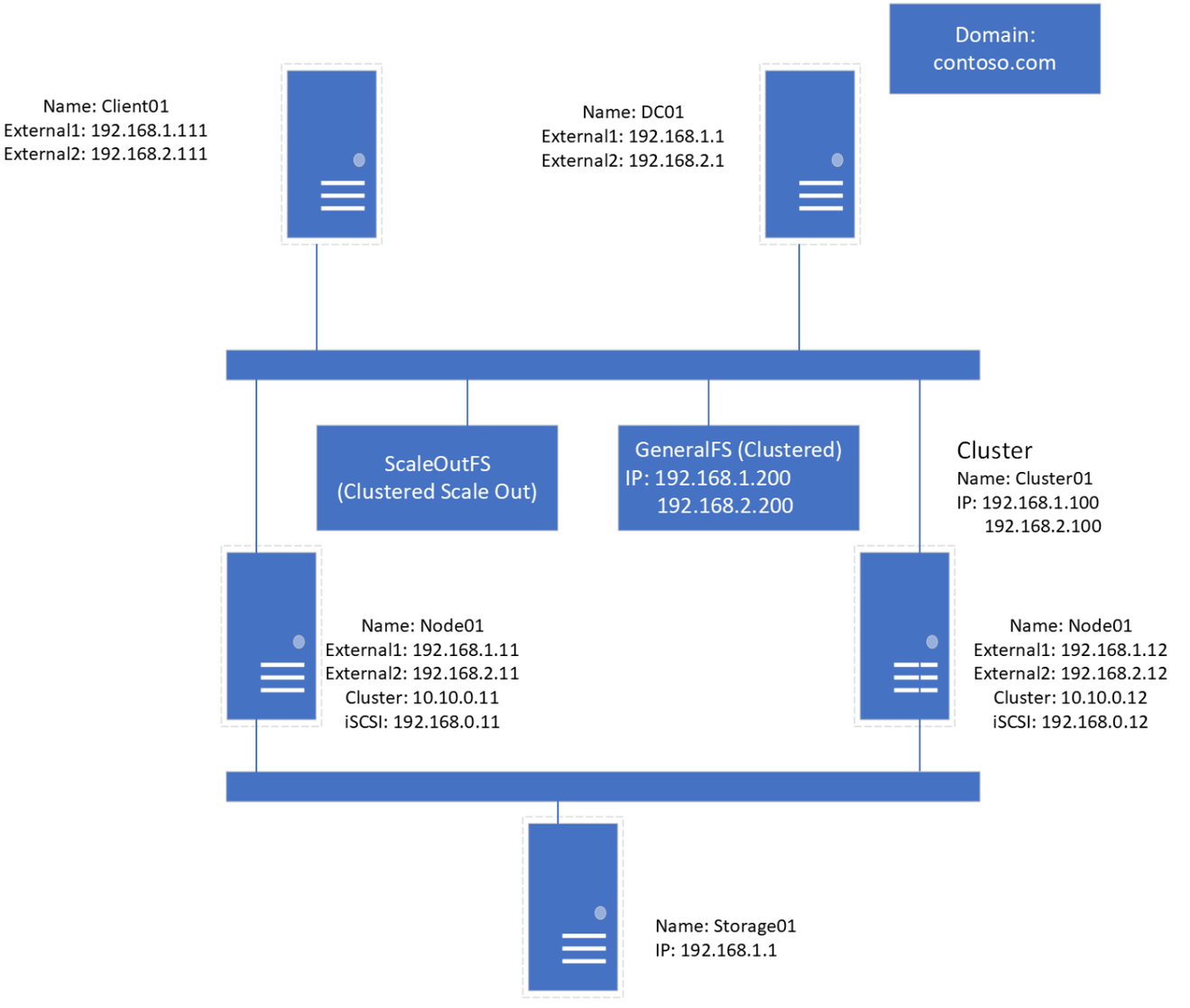


FileServer Test Suite Environment

FileServer Standard



FileServer Cluster



Microsoft

What's New (Github Release 4.24.4.0)

FileServer TestSuites Updates

Windows Protocol TestSuites Release v4.24.4.0

https://github.com/Microsoft/WindowsProtocolTestSuites/releases/tag/4.24.4.0

FileServer Test Suites

- Update to dotnet 8
- New Tests
- Bug fixes

Protocol Test Manager Service

- Expand custom test categories
- Other bug fixes

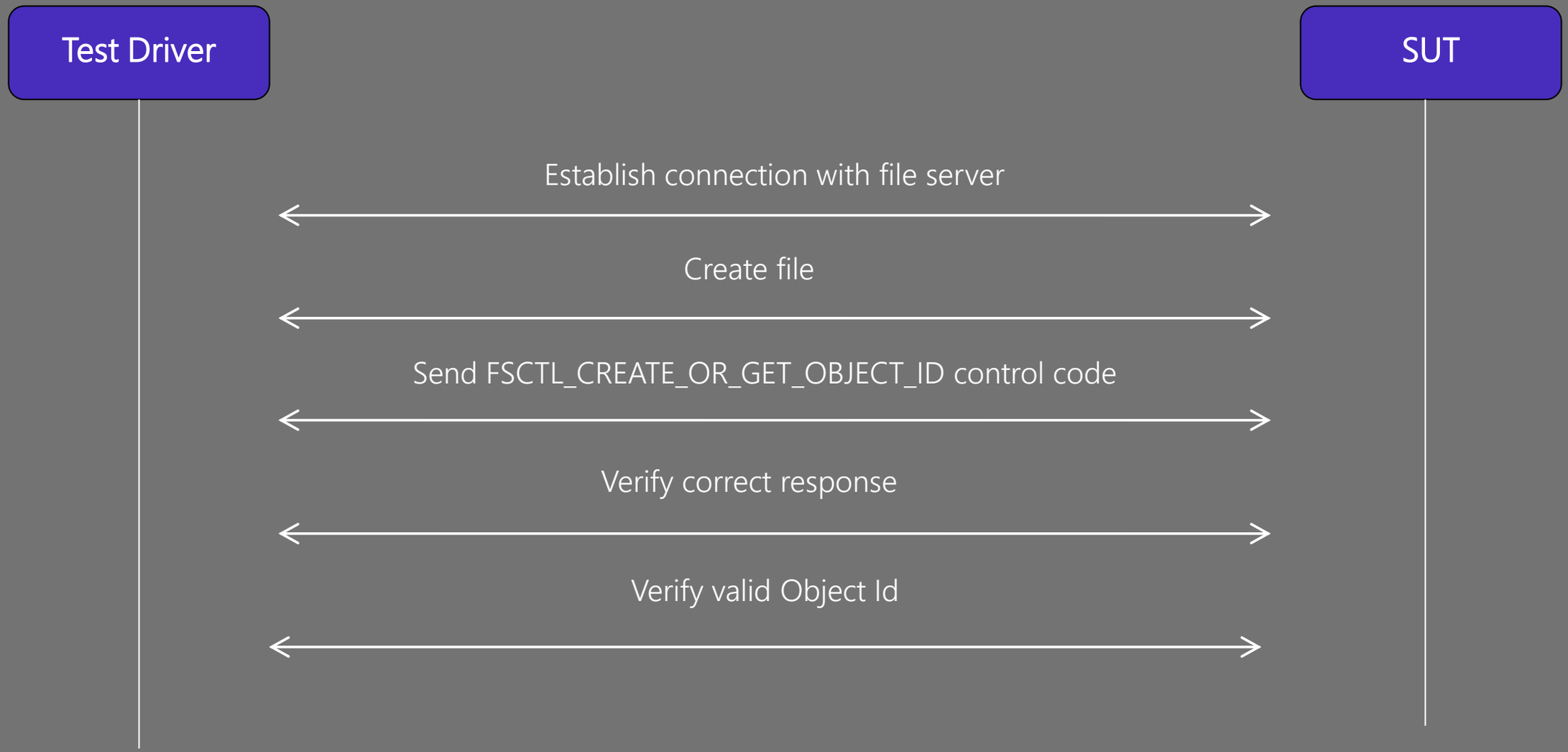
Protocol Test Framework

- Update to dotnet 8
- Bug fixes

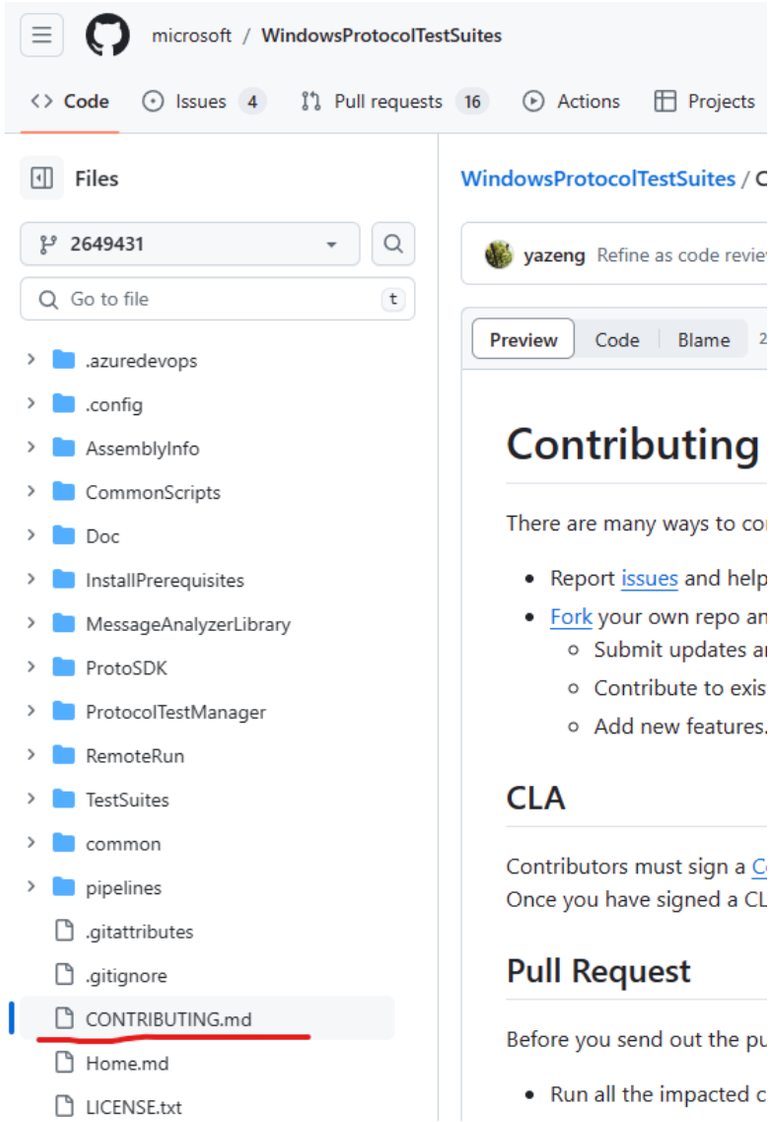
New Tests for FSCTL_CREATE_OR_GET_OBJECT_ID

- Tests algorithm for FSCTL_CREATE_OR_GET_OBJECT_ID on a given file system
  - Detailed description available in [MS-FSA] and [MS-FSCC]
    https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-fsa
    https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-fscc
- Verify File Object has valid Object Id after invoking on object

FSCTL_CREATE_OR_GET_OBJECT_ID test sample



Contact Us - GitHub



Open an Issue

https://www.github.com/Microsoft/WindowsProtocolTestSuites/issues

Create a Pull Request

https://github.com/microsoft/WindowsProtocolTestSuites/pulls

Consult CONTRIBUTING.md

For questions and/or feedback

Please email us
testsuitehelp@microsoft.com

Open an issue on Github
https://www.github.com/Microsoft/WindowsProtocolTestSuites/issues

Microsoft

Overview of SMB2 Dissectors on Wireshark

Adedeji Adeloye
Software Engineer

Microsoft

Agenda

- What this session is not about
- Why Wireshark?
- Our Goal
- Contributions so far
- Demo
- Open Discussion

What this session is not about…

- Introduction to Wireshark
- How to use Wireshark
- Wireshark vs. other tools
- How Windows protocols work

Why Wireshark?

- A core part of our interoperability efforts with partners and customers is to provide test tools to assist them with implementing and testing Microsoft Windows protocols.
- For many years, Microsoft supported customers with Microsoft Message Analyzer (MMA) as a protocol parsing tool.
- Following the deprecation of MMA, and based on our research and feedback from customers, we adopted Wireshark as a replacement for MMA, to continue providing interop support to our partners and customers.
- Protocol parsers (*known as Dissectors on Wireshark*) provide users with the capability to decode raw data on the wire to human readable structures.

Microsoft

Our Goal

To contribute dissectors to Wireshark to enable partners, customers, and all users get detailed and up-to-date packet information for Windows protocols.

Overview of Contributions So Far

SMB2 Message	Description
Tree connect response flags	New flag: isolated transport flag; to indicate to the client that the server prefers communication with the share to be on a separate set of connection
Query Info	Flags dissection was updated to properly dissect flags: SL_RESTART_SCAN, SL_RETURN_SINGLE_ENTRY, SL_INDEX_SPECIFIED These were previously ignored
FSCTL_SET_INTEGRITY_INFORMATION_Ex	This is an FSA control code that is only supported on REFS file systems to request the server to set integrity information for a file or directory.
FSCTL_REFS_STREAM_SNAPSHOT_MANAGEMENT	This control code requests the server to perform a specific snapshot operation on a given data stream in a file.

Overview of Contributions So Far

SMB2 Message	Description
Server to client notification	This is a recent packet sent from the server to the client to perform implementation specific details without expecting any response
FileFullEaInformation flags	Control flags for this message was also updated
Lock response	Previously mislabeled fields were updated in the lock response message
SMB2&3 code bug fixes	We also fixed a couple of bugs in the SMB2&3 (packet-smb2.c) dissector code file in Wireshark that prevents Wireshark developers from debugging during development

Demo

fsctl_set_integrity_information_ex_success.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

smb || smb2

No.	Time	Source	Destination	Protocol	Length	Info
35	0.418755	192.168.0.252	20.253.153.46	SMB2	232	Negotiate Protocol Request
44	0.776677	20.253.153.46	192.168.0.252	SMB2	294	Negotiate Protocol Response
51	0.900491	192.168.0.252	20.253.153.46	SMB2	240	Session Setup Request
58	1.305212	20.253.153.46	192.168.0.252	SMB2	153	Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED
60	1.334947	192.168.0.252	20.253.153.46	SMB2	213	Session Setup Request, NTLMSSP_NEGOTIATE
64	1.697698	20.253.153.46	192.168.0.252	SMB2	385	Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
67	1.773801	192.168.0.252	20.253.153.46	SMB2	639	Session Setup Request, NTLMSSP_AUTH, User: \ServerAdmin
69	2.321282	20.253.153.46	192.168.0.252	SMB2	139	Session Setup Response
70	2.357882	192.168.0.252	20.253.153.46	SMB2	186	Tree Connect Request Tree: \\20.253.153.46\SMBReFSShare
80	2.751019	20.253.153.46	192.168.0.252	SMB2	138	Tree Connect Response
94	2.763534	192.168.0.252	20.253.153.46	SMB2	194	Create Request File: jsjirwed
103	3.092580	20.253.153.46	192.168.0.252	SMB2	210	Create Response File: jsjirwed
104	3.104081	192.168.0.252	20.253.153.46	SMB2	194	Ioctl Request FILE_SYSTEM Function:0x00e0 File: jsjirwed
108	3.405190	20.253.153.46	192.168.0.252	SMB2	170	Ioctl Response FILE_SYSTEM Function:0x00e0 File: jsjirwed
109	3.417882	192.168.0.252	20.253.153.46	SMB2	146	Close Request File: jsjirwed
110	3.800545	20.253.153.46	192.168.0.252	SMB2	182	Close Response
111	3.810428	192.168.0.252	20.253.153.46	SMB2	194	Create Request File: jsjirwed
113	4.154007	20.253.153.46	192.168.0.252	SMB2	210	Create Response File: jsjirwed
114	4.157548	192.168.0.252	20.253.153.46	SMB2	155	SetInfo Request FILE_INFO/SMB2_FILE_DISPOSITION_INFO File: jsjirwed
115	4.578895	20.253.153.46	192.168.0.252	SMB2	124	SetInfo Response
116	4.582142	192.168.0.252	20.253.153.46	SMB2	146	Close Request File: jsjirwed
117	4.975204	20.253.153.46	192.168.0.252	SMB2	182	Close Response
125	5.336342	192.168.0.252	20.253.153.46	SMB	115	Negotiate Protocol Request
130	5.731255	20.253.153.46	192.168.0.252	SMB2	228	Negotiate Protocol Response
131	5.746288	192.168.0.252	20.253.153.46	SMB2	216	Negotiate Protocol Request

```
> Frame 104: 194 bytes on wire (1552 bits), 194 bytes captured (1552 bits) on interface \Device\NPF_{8B6829A1-8842-4D83-AEAA-BFAD97B24C
> Ethernet II, Src: Intel_07:f4:f4 (f4:a4:75:07:f4:f4), Dst: f2:25:de:56:b4:e3 (f2:25:de:56:b4:e3)
> Internet Protocol Version 4, Src: 192.168.0.252, Dst: 20.253.153.46
> Transmission Control Protocol, Src Port: 64640, Dst Port: 445, Seq: 1381, Ack: 996, Len: 140
> NetBIOS Session Service
v SMB2 (Server Message Block Protocol version 2)
  > SMB2 Header
  v Ioctl Request (0x0b)
    > StructureSize: 0x0039
      Reserved: 0000
    > Function: Unknown (0x00090380)
    > GUID handle File: jsjirwed
      Max Ioctl In Size: 16
      Max Ioctl Out Size: 16
    > Flags: 0x00000001
      Reserved: 00000000
      Blob Offset: 0x00000000
      Blob Length: 0
      Out Data: NO DATA
      Blob Offset: 0x00000078
      Blob Length: 16
    v In Data
        Unknown: 01000000010000000100000000000000
```

```
0000  f2 25 de 56 b4 e3 f4 a4  75 07 f4 f4 08 00 45 00   ·%·V····  u·····E·
0010  00 b4 b4 d1 40 00 80 06  00 00 c0 a8 00 fc 14 fd   ····@···  ········
0020  99 2e fc 80 01 bd ee c4  67 5f c3 0c 5d c0 50 18   ········  g_··]·P·
0030  00 fd 70 76 00 00 00 00  00 88 fe 53 4d 42 40 00   ··pv····  ···SMB@·
0040  01 00 00 00 00 00 0b 00  01 00 08 00 00 00 00 00   ········  ········
0050  00 00 06 00 00 00 00 00  00 00 ff fe 00 00 01 00   ········  ········
0060  00 00 45 00 00 00 00 70  00 00 6b d7 ea e1 74 33   ··E····p  ··k···t3
0070  e0 a3 79 c9 da 4e bc 06  86 f7 39 00 00 00 80 03   ··y··N··  ··9·····
0080  09 00 14 00 00 00 1c 00  00 00 01 00 00 00 1c 00   ········  ········
0090  00 00 78 00 00 00 10 00  00 00 10 00 00 00 00 00   ··x·····  ········
00a0  00 00 00 00 00 00 10 00  00 00 01 00 00 00 00 00   ········  ········
00b0  00 00 01 00 00 00 01 00  00 00 01 00 00 00 00 00   ········  ········
00c0  00 00                                              ··
```

fsctl_set_integrity_information_ex_success.pcapng

Packets: 202 · Displayed: 40 (19.8%)

Profile: Default

92°F Partly sunny

Search

5:27 AM 3/25/2024

Open Discussion

- SMB3 decryption on Wireshark?
- Other Windows protocols you would like us to contribute to?

Microsoft

Try out our contributions:

https://www.wireshark.org/download/automated/

To download and start testing with our contributions, go to Index of /download/automated/win64 (wireshark.org) and download the latest installer with version number ***4.3.0rc0-1317-gxxxxxxxxxxxx-x64.exe*** and above.

Got questions or feedback?

Please email us:

winwiresharkhelp@microsoft.com

Thank you!