Custom LUA dissectors to the rescue in RCA

How to write a LUA dissector for a proprietary protocol to assist in troubleshooting

Sake Blok
relational therapist for computer systems

sake.blok@SYN-bit.nl

9 November 2017
Hands-on files:

http://www.SYN-bit.nl/files/sf17eu-lua.zip
Custom LUA dissectors to the rescue in root cause analysis
Application and network troubleshooting

Protocol and packet analysis

Training (Wireshark, TCP, SSL)

www.SYN-bit.nl
Houston, we have a problem…

- Trading platform
- Connections being reset
- Logfiles point to network
- Software supplier blames the network (duh!)
- Custom protocol, so just TCP analysis possible
- What now?
So how does the application work?
PDU’s coming in

Internet

Client 1

Client 2

Client 3

DMZ

Multiplexer

Server VLAN

Module A

Module B

Module C

Module D

Module E
PDU's going out

Internet

DMZ

Server VLAN

PDU's going out

Client 1

Internet

DMZ

Server VLAN

Multiplexer

Module A

Module B

Module C

Module D

Module E

1 ➔ A

2 ➔ A

1 ➔ B

2 ➔ B

3 ➔ A

1 ➔ C

2 ➔ C

3 ➔ C

1 ➔ E

3 ➔ E

Client 2

Client 3

Custom LUA dissectors to the rescue in root cause analysis
... plus Control PDU’s
2014-06-05 09:30:12,755 NOTICE - Closing clientConnection: 2652 from: Trader
2014-06-05 09:30:48,043 NOTICE - Creating clientConnection: 2656 user: XXXXXX0013 on: Trader
2014-06-05 09:31:58,978 NOTICE - Shutdown connection from: Trader
2014-06-05 09:31:58,978 WARN - ModuleServer Trader lost connection to multiplexer on 192.168.0.1:2107
2014-06-05 09:37:40,252 NOTICE - Creating clientConnection: 1129 user: APPSERVER1 on: Trader
2014-06-05 09:42:03,556 WARN - Connection lost in receive loop: Trader
2014-06-05 09:42:03,556 NOTICE - Shutdown connection from: Trader
2014-06-05 09:42:03,556 WARN - ModuleServer Trader lost connection to multiplexer on 192.168.0.1:2107
Multiplexer Log

2014-06-05 09:34:41,006 NOTICE - Logging in clientConnection: 2678 user: XXXXXX0009 to: MarketConfiguration
2014-06-05 09:34:41,006 NOTICE - Logging in clientConnection: 2678 user: XXXXXXX0009 to: FinancialModule
2014-06-05 09:34:52,816 NOTICE - Disabling client logins on port: '443'
2014-06-05 09:34:52,816 NOTICE - Client accept loop stopped
2014-06-05 09:34:53,580 NOTICE - Logging in clientConnection: 2671 user: XXXXXX0000 to: Trader
2014-06-05 09:34:53,580 NOTICE - Logging in clientConnection: 2671 user: XXXXXXX0000 to: MarketConfiguration
2014-06-05 09:34:56,622 NOTICE - Logging in clientConnection: 2679 user: XXXXXX0021 to: BackOffice
2014-06-05 09:34:56,825 NOTICE - Logging in clientConnection: 2679 user: XXXXXX0021 to: Trader
2014-06-05 09:34:56,825 NOTICE - Logging in clientConnection: 2679 user: XXXXXXX0021 to: MarketConfiguration
2014-06-05 09:34:56,825 NOTICE - Logging in clientConnection: 2679 user: XXXXXXX0021 to: FinancialModule

2014-06-05 09:35:01,458 sss 127.000.000.001 Operator 2014-06-05 09:35:01.458 Multiplexer stopped.
Stopped logging 2014-06-05 09:35:01,458
Started logging 2014-06-05 09:35:16,060
2014-06-05 09:35:32,924 NOTICE - Received start id: 2001
2014-06-05 09:35:32,940 NOTICE - Received security parameters
Stopped logging 2014-06-05 09:41:26,864
Started logging 2014-06-05 09:42:46,004
2014-06-05 09:42:50,107 NOTICE - Received start id: 2001
2014-06-05 09:42:50,138 NOTICE - Received security parameters
2014-06-05 09:43:28,531 NOTICE - Enabling client logins on port: '443'
2014-06-05 09:43:28,531 NOTICE - Client accept loop started
Custom LUA dissectors to the rescue in root cause analysis

Timeline

2014-06-05 09:31:58,978 WARN - ModuleServer Trader lost connection to multiplexer on 192.168.0.1:2107

2014-06-05 09:34:52,816 NOTICE - Disabling client logins on port: '443'
2014-06-05 09:34:52,816 NOTICE - Client accept loop stopped

2014-06-05 09:35:01,458 NOTICE - Disabling client logins on port: '443'
2014-06-05 09:35:01,458 NOTICE - Client accept loop stopped

2014-06-05 09:35:32,924 NOTICE - Enabling client logins on port: '443'
2014-06-05 09:35:32,924 NOTICE - Client accept loop started

Packet timestamps don’t match

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Delta</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>31362</td>
<td>09:31:50.156965</td>
<td>0.000559</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TCP</td>
<td>1434</td>
<td>[TCP Zer0Window] 2107 → 1195 [ACK] Seq=397458 Ack=21251242 Win=0 Len=1380</td>
</tr>
<tr>
<td>31363</td>
<td>09:31:50.156965</td>
<td>0.000084</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TCP</td>
<td>1434</td>
<td>[TCP Zer0Window] 2107 → 1195 [ACK] Seq=398838 Ack=21251242 Win=0 Len=1380</td>
</tr>
<tr>
<td>31364</td>
<td>09:31:50.156965</td>
<td>0.000084</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TCP</td>
<td>1434</td>
<td>[TCP Zer0Window] 2107 → 1195 [ACK] Seq=400218 Ack=21251242 Win=0 Len=1380</td>
</tr>
<tr>
<td>31365</td>
<td>09:31:50.156965</td>
<td>0.000084</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TCP</td>
<td>1434</td>
<td>[TCP Zer0Window] 2107 → 1195 [ACK] Seq=401598 Ack=21251242 Win=0 Len=1380</td>
</tr>
<tr>
<td>31366</td>
<td>09:31:50.156965</td>
<td>0.000084</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TCP</td>
<td>875</td>
<td>[TCP Zer0Window] 2107 → 1195 [PSH, ACK] Seq=402878 Ack=21251242 Win=0 Len=821</td>
</tr>
<tr>
<td>31367</td>
<td>09:31:50.156965</td>
<td>0.000084</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TCP</td>
<td>1434</td>
<td>[TCP Zer0Window] 2107 → 1195 [ACK] Seq=403799 Ack=21251242 Win=0 Len=821</td>
</tr>
<tr>
<td>31368</td>
<td>09:31:50.156965</td>
<td>0.000084</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TCP</td>
<td>60</td>
<td>1195 → 2107 [ACK] Seq=21251242 Ack=403799 Win=7741440 Len=0</td>
</tr>
<tr>
<td>31369</td>
<td>09:31:50.156965</td>
<td>3.729581</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TCP</td>
<td>60</td>
<td>1195 → 2107 [PSH, ACK] Seq=148616 Ack=180000 Win=130816 Len=11</td>
</tr>
<tr>
<td>31370</td>
<td>09:31:50.156965</td>
<td>0.202224</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TCP</td>
<td>60</td>
<td>1195 → 2107 [ACK] Seq=180900 Ack=148627 Win=130816 Len=0</td>
</tr>
<tr>
<td>31371</td>
<td>09:31:50.156965</td>
<td>0.079015</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TCP</td>
<td>60</td>
<td>1195 → 2107 [PSH, ACK] Seq=148627 Ack=180000 Win=130816 Len=11</td>
</tr>
<tr>
<td>31372</td>
<td>09:31:50.156965</td>
<td>0.201788</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TCP</td>
<td>60</td>
<td>1195 → 2107 [PSH, ACK] Seq=148638 Ack=180000 Win=130816 Len=11</td>
</tr>
<tr>
<td>31373</td>
<td>09:31:50.156965</td>
<td>0.516220</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TCP</td>
<td>60</td>
<td>1195 → 2107 [PSH, ACK] Seq=148649 Ack=180000 Win=130816 Len=11</td>
</tr>
<tr>
<td>31374</td>
<td>09:31:50.156965</td>
<td>0.217015</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TCP</td>
<td>60</td>
<td>1195 → 2107 [PSH, ACK] Seq=148660 Ack=180000 Win=130816 Len=0</td>
</tr>
<tr>
<td>31375</td>
<td>09:31:50.156965</td>
<td>4.121218</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TCP</td>
<td>60</td>
<td>1195 → 2107 [PSH, ACK] Seq=148671 Ack=180000 Win=130816 Len=11</td>
</tr>
<tr>
<td>31376</td>
<td>09:31:50.156965</td>
<td>0.215595</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TCP</td>
<td>60</td>
<td>1195 → 2107 [PSH, ACK] Seq=148682 Ack=180000 Win=130816 Len=0</td>
</tr>
<tr>
<td>31377</td>
<td>09:32:03.343767</td>
<td>3.901880</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TCP</td>
<td>66</td>
<td>1592 → 2107 [SYN] Seq=0 Win=8192 Len=0 MSS=1380 WS=256 SACK_PERM=1</td>
</tr>
<tr>
<td>31378</td>
<td>09:32:03.344095</td>
<td>0.000328</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TCP</td>
<td>66</td>
<td>1592 → 1592 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1</td>
</tr>
<tr>
<td>31379</td>
<td>09:32:03.344383</td>
<td>0.000288</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TCP</td>
<td>60</td>
<td>1592 → 2107 [ACK] Seq=1 Ack=1 Win=131972 Len=0</td>
</tr>
<tr>
<td>31380</td>
<td>09:32:03.344877</td>
<td>0.000494</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TCP</td>
<td>63</td>
<td>1592 → 2107 [PSH, ACK] Seq=1 Ack=1 Win=131972 Len=9</td>
</tr>
<tr>
<td>31381</td>
<td>09:32:03.346625</td>
<td>0.004748</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TCP</td>
<td>71</td>
<td>2107 → 1170 [PSH, ACK] Seq=148660 Ack=130816 Len=17</td>
</tr>
<tr>
<td>31382</td>
<td>09:32:03.544694</td>
<td>0.195069</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TCP</td>
<td>60</td>
<td>2107 → 1592 [ACK] Seq=1 Ack=10 Win=131972 Len=0</td>
</tr>
</tbody>
</table>

Custom LUA dissectors to the rescue in root cause analysis

#sf17eu • Estoril, Portugal
The Blaming Game

• Zero Window on Multiplexer
  ‣ But why does the Trader module report it did not receive a response while it is still receiving lots of data?

• TCP/RST from APPSERVER1
  ‣ But not at the time of the log message!

• The network is the problem
  ‣ But what is? And how to convince the software vendor?
  ‣ Client (APPSERVER1) or server (Multiplexer) side problem?
LUA to the rescue...

- Different LUA dissection types
- Skeleton Dissector
- Parsing the PDU header
- Reassembly

Starting point: https://wiki.wireshark.org/Lua
LUA in Wireshark

- Lua can be used to write dissectors, post-dissectors and taps.
  - Although it's possible to write dissectors in Lua, Wireshark dissectors are written in C, as C is several times faster than Lua. Lua is ok for prototyping dissectors, during Reverse Engineering you can use your time for finding out how things work instead of compiling and debugging your C dissector.
  - Post-dissectors are dissectors meant to run after every other dissector has run. They can add items to the dissection tree so they can be used to create your own extensions to the filtering mechanism.
  - Taps are used to collect information after the packet has been dissected.
- So for our purpose, a normal Lua dissector is the best fit
• LUA has shipped with the Windows version of Wireshark since 0.99.4.
• Availability on other platforms varies.
  ‣ Check: “Help -> About Wireshark”
How to execute?

- CLI option: `-X lua_script:<script-file>`
- In the plugin directories
  - Need extension .lua
  - Personal Plugins: `/Users/sake/.config/wireshark/plugins`  
    - dissector plugins
  - Global Plugins: `/Applications/Wireshark...ents/PlugIns/wireshark`  
    - dissector plugins
Hello World!

```
sake@MacSake:~$ cat ~/.config/wireshark/plugins/hello.lua
-- hello.lua
-- Lua's implementation of D. Ritchie's hello world program.
print("hello world!")
sake@MacSake:~$
```

```
sake@MacSake:~$ tshark -r icmp.pcap -c 1
   1   0.000000   172.16.0.34
! 8.8.8.8      ICMP 98 Echo (ping) request  id=0xfa0e, seq=0/0, ttl=64
sake@MacSake:~$
```

```
sake@MacSake:~$ tshark -r icmp.pcap -c 1 -X lua_script:hello.lua
hello world!
   1   0.000000   172.16.0.34
! 8.8.8.8      ICMP 98 Echo (ping) request  id=0xfa0e, seq=0/0, ttl=64
sake@MacSake:~$
```

```
sake@MacSake:~$ mv hello.lua ~/.config/wireshark/plugins/
sake@MacSake:~$ tshark -r icmp.pcap -c 1
hello world!
   1   0.000000   172.16.0.34
! 8.8.8.8      ICMP 98 Echo (ping) request  id=0xfa0e, seq=0/0, ttl=64
sake@MacSake:~$
```
Writing: trade.lua

- Write a basic LUA dissector
- Get a protocol description
- Add some reverse engineering
  - Interpret packet data
  - Write code
  - Test
  - Rinse
  - Repeat
PDU format

- Each packet has a 5 byte header:
  - 1: sender
    ‣ List of module ID’s provided
  - 2-3: total size including header
  - 4-5: receiver
    ‣ In case of control packets: The receiver will be ff:ff
- The header is followed by a keyword and a null termination
- Optional data
Examples

- Keepalive packets from module (DocumentSystem) to multiplexer contain the following bytes:
  - 25:00:0a:ff:ff:50:69:6e:67:00

- The Multiplexer replies with:
  - 01:00:0a:ff:ff:50:6f:6e:67:00

- Roundtrip packets from the Multiplexer to module(DocumentSystem) contain the following bytes:

- The module replies with:
Example Drilldown

25                                      Module ID
00:17                                    Size (Big endian)
ff:ff                                    Receiver
00                                    Keyword termination
07                                      Size of following data
1b:88:aa:db:46:b6:0c                    Microsecond count (Little endian)
01.lua: Register the dissector

do
  local trade = Proto("trade","TRADE");

  function trade.dissector(tvb,pinfo,tree)
    end

  local tcp_encap_table = DissectorTable.get("tcp.port")
  tcp_encap_table:add(2107,trade)
end
Result of 01.lua
local trade = Proto("trade","TRADE");

function trade.dissector(tvb,pinfo,tree)
    -- info("Entering dissector for frame " .. pinfo.number)
    pinfo.cols.protocol = "TRADE"
end

local tcp_encap_table = DissectorTable.get("tcp.port")
tcp_encap_table:add(2107,trade)
### Result of 02.lua

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Delta</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>09:10:04.368322</td>
<td>0.000000</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>1181 → 2107 [PSH, ACK] Seq=1 Ack=1 Win=138016 Len=10</td>
</tr>
<tr>
<td>2</td>
<td>09:10:04.368410</td>
<td>0.000088</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>1184 → 2107 [PSH, ACK] Seq=1 Ack=1 Win=129792 Len=10</td>
</tr>
<tr>
<td>3</td>
<td>09:10:04.368413</td>
<td>0.000083</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>1153 → 2107 [PSH, ACK] Seq=1 Ack=1 Win=138016 Len=10</td>
</tr>
<tr>
<td>4</td>
<td>09:10:04.562429</td>
<td>0.202016</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TCP</td>
<td>60</td>
<td>2107 → 1184 [ACK] Seq=1 Ack=11 Win=130816 Len=0</td>
</tr>
<tr>
<td>5</td>
<td>09:10:04.562433</td>
<td>0.000040</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TCP</td>
<td>60</td>
<td>2107 → 1153 [ACK] Seq=1 Ack=11 Win=130816 Len=0</td>
</tr>
<tr>
<td>6</td>
<td>09:10:04.562436</td>
<td>0.000038</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TCP</td>
<td>60</td>
<td>2107 → 1181 [ACK] Seq=1 Ack=11 Win=130816 Len=0</td>
</tr>
<tr>
<td>7</td>
<td>09:10:04.609538</td>
<td>0.047102</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>1536 → 2107 [PSH, ACK] Seq=1 Ack=1 Win=129536 Len=10</td>
</tr>
<tr>
<td>8</td>
<td>09:10:04.609921</td>
<td>0.000383</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>1177 → 2107 [PSH, ACK] Seq=1 Ack=1 Win=130560 Len=10</td>
</tr>
<tr>
<td>9</td>
<td>09:10:04.609974</td>
<td>0.000530</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>1192 → 2107 [PSH, ACK] Seq=1 Ack=1 Win=130384 Len=10</td>
</tr>
<tr>
<td>10</td>
<td>09:10:04.610060</td>
<td>0.000869</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>1200 → 2107 [PSH, ACK] Seq=1 Ack=1 Win=130384 Len=10</td>
</tr>
<tr>
<td>11</td>
<td>09:10:04.610172</td>
<td>0.001112</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>1203 → 2107 [PSH, ACK] Seq=1 Ack=1 Win=130384 Len=10</td>
</tr>
<tr>
<td>12</td>
<td>09:10:04.610245</td>
<td>0.000737</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>1256 → 2107 [PSH, ACK] Seq=1 Ack=1 Win=130384 Len=10</td>
</tr>
<tr>
<td>13</td>
<td>09:10:04.735016</td>
<td>0.124771</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>1188 → 2107 [PSH, ACK] Seq=1 Ack=1 Win=130816 Len=10</td>
</tr>
</tbody>
</table>

- Frame 8: 64 bytes on wire (512 bits), 64 bytes captured (512 bits)
- Ethernet II, Src: 02:00:00:00:00:01 (02:00:00:00:00:01), Dst: 02:00:00:00:00:02
- Internet Protocol Version 4, Src: 18.0.0.1, Dst: 192.168.0.1
- Transmission Control Protocol, Src Port: 1177, Dst Port: 2107, Seq: 1, Ack: 1, Len 64
local trade = Proto("trade","TRADE");

function trade.dissector(tvb,pinfo,tree)
    -- info("Entering dissector for frame " .. pinfo.number)
    pinfo.cols.protocol = "TRADE"

    local subtree

    subtree = tree:add(trade,tvb(0,tvb:len()),"Trade PDU : ")

end

local tcp_encap_table = DissectorTable.get("tcp.port")
tcp_encap_table:add(2107,trade)
**Result of 03.lua**

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Delta</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>09:10:04,360322</td>
<td>0.000000</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td></td>
<td>64</td>
<td>1184 → 2107 [PSH, ACK] Seq=1 Ack=1 Win=138816 Len=10</td>
</tr>
<tr>
<td>2</td>
<td>09:10:04,360410</td>
<td>0.000088</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td></td>
<td>64</td>
<td>1184 → 2107 [PSH, ACK] Seq=1 Ack=1 Win=129792 Len=10</td>
</tr>
<tr>
<td>3</td>
<td>09:10:04,360413</td>
<td>0.000033</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td></td>
<td>64</td>
<td>1155 → 2107 [PSH, ACK] Seq=1 Ack=1 Win=138816 Len=10</td>
</tr>
<tr>
<td>4</td>
<td>09:10:04,562429</td>
<td>0.202016</td>
<td>192.168.0.1</td>
<td>TCP</td>
<td></td>
<td>60</td>
<td>2107 → 1184 [ACK] Seq=1 Ack=11 Win=138816 Len=0</td>
</tr>
<tr>
<td>5</td>
<td>09:10:04,562433</td>
<td>0.000004</td>
<td>192.168.0.1</td>
<td>TCP</td>
<td></td>
<td>60</td>
<td>2107 → 1553 [ACK] Seq=1 Ack=11 Win=138560 Len=0</td>
</tr>
<tr>
<td>6</td>
<td>09:10:04,562436</td>
<td>0.000003</td>
<td>192.168.0.1</td>
<td>TCP</td>
<td></td>
<td>60</td>
<td>2107 → 1181 [ACK] Seq=1 Ack=11 Win=135552 Len=0</td>
</tr>
<tr>
<td>7</td>
<td>09:10:04,609538</td>
<td>0.047102</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td></td>
<td>64</td>
<td>1536 → 2107 [PSH, ACK] Seq=1 Ack=1 Win=129536 Len=10</td>
</tr>
<tr>
<td>8</td>
<td>09:10:04,609921</td>
<td>0.003333</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td></td>
<td>64</td>
<td>1177 → 2107 [PSH, ACK] Seq=1 Ack=1 Win=138816 Len=10</td>
</tr>
<tr>
<td>9</td>
<td>09:10:04,609974</td>
<td>0.000533</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td></td>
<td>64</td>
<td>1155 → 2107 [PSH, ACK] Seq=1 Ack=1 Win=138816 Len=10</td>
</tr>
<tr>
<td>10</td>
<td>09:10:04,610050</td>
<td>0.000005</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td></td>
<td>64</td>
<td>1155 → 2107 [PSH, ACK] Seq=1 Ack=1 Win=138816 Len=10</td>
</tr>
<tr>
<td>11</td>
<td>09:10:04,610172</td>
<td>0.000112</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td></td>
<td>64</td>
<td>1155 → 2107 [PSH, ACK] Seq=1 Ack=1 Win=138816 Len=10</td>
</tr>
<tr>
<td>12</td>
<td>09:10:04,610245</td>
<td>0.000073</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td></td>
<td>64</td>
<td>1155 → 2107 [PSH, ACK] Seq=1 Ack=1 Win=138816 Len=10</td>
</tr>
<tr>
<td>13</td>
<td>09:10:04,735016</td>
<td>0.124771</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td></td>
<td>64</td>
<td>1188 → 2107 [PSH, ACK] Seq=1 Ack=1 Win=138816 Len=10</td>
</tr>
</tbody>
</table>

Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits)
Ethernet II, Src: 02:00:00:00:00:01 (02:00:00:00:00:00), Dst: 02:00:00:00:00:02 (02:00:00:00:00:00)
Internet Protocol Version 4, Src: 10.0.0.1, Dst: 192.168.0.1
Transmission Control Protocol, Src Port: 1181, Dst Port: 2107, Seq: 1, Ack: 1, Len: 10

**Trade PDU:**
04.lua: Add a field

```lua
sake@MacSake:~$ diff -b 03.lua 04.lua
2a3,5
>     local f = trade.fields
> >     f.sender     = ProtoField.uint8("trade.sender","Sender",base.DEC)
9a13,20
>         local offset = 0
>         local info = ""
> >         subtree = tree:add(trade,tvb(offset,tvb:len()),"Trade PDU : ")
> >         -- Add sender
> >         subtree:add(f.sender,tvb(offset,1))
> >         offset = offset + 1
11c22,23
<         subtree = tree:add(trade,tvb(0,tvb:len()),"Trade PDU : ")
---
>         -- Tell the calling dissector how many bytes we dissected
>         return tvb:len()
```
Result of 04.lua

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Delta</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>09:10:04,300322</td>
<td>0.000000</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>1181 → 2187 [PSH, ACK] Seq=1 Ack=1 Win=130616 Len=10</td>
</tr>
<tr>
<td>2</td>
<td>09:10:04,300410</td>
<td>0.000008</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>1184 → 2187 [PSH, ACK] Seq=1 Ack=1 Win=129792 Len=10</td>
</tr>
<tr>
<td>3</td>
<td>09:10:04,300413</td>
<td>0.000003</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TCP</td>
<td>64</td>
<td>1153 → 2153 [PSH, ACK] Seq=1 Ack=1 Win=130616 Len=10</td>
</tr>
<tr>
<td>4</td>
<td>09:10:04,562429</td>
<td>0.20216</td>
<td>192.168.0.1</td>
<td>192.168.0.1</td>
<td>TCP</td>
<td>64</td>
<td>2107 → 1184 [ACK] Seq=1 Ack=11 Win=130616 Len=0</td>
</tr>
<tr>
<td>5</td>
<td>09:10:04,562433</td>
<td>0.000004</td>
<td>192.168.0.1</td>
<td>192.168.0.1</td>
<td>TCP</td>
<td>64</td>
<td>2107 → 1553 [ACK] Seq=1 Ack=11 Win=130560 Len=0</td>
</tr>
<tr>
<td>6</td>
<td>09:10:04,562436</td>
<td>6.000000</td>
<td>192.168.0.1</td>
<td>192.168.0.1</td>
<td>TCP</td>
<td>64</td>
<td>2107 → 1181 [ACK] Seq=1 Ack=11 Win=133552 Len=0</td>
</tr>
<tr>
<td>7</td>
<td>09:10:04,609538</td>
<td>0.47102</td>
<td>192.168.0.1</td>
<td>192.168.0.1</td>
<td>TCP</td>
<td>64</td>
<td>1536 → 2107 [PSH, ACK] Seq=1 Ack=11 Win=129536 Len=10</td>
</tr>
<tr>
<td>8</td>
<td>09:10:04,609921</td>
<td>0.000083</td>
<td>192.168.0.1</td>
<td>192.168.0.1</td>
<td>TCP</td>
<td>64</td>
<td>1177 → 2107 [PSH, ACK] Seq=1 Ack=11 Win=130560 Len=10</td>
</tr>
<tr>
<td>9</td>
<td>09:10:04,609974</td>
<td>0.000053</td>
<td>192.168.0.1</td>
<td>192.168.0.1</td>
<td>TCP</td>
<td>64</td>
<td>1192 → 2107 [PSH, ACK] Seq=1 Ack=11 Win=130804 Len=10</td>
</tr>
<tr>
<td>10</td>
<td>09:10:04,610060</td>
<td>0.000086</td>
<td>192.168.0.1</td>
<td>192.168.0.1</td>
<td>TCP</td>
<td>64</td>
<td>1200 → 2107 [PSH, ACK] Seq=1 Ack=11 Win=130804 Len=10</td>
</tr>
<tr>
<td>11</td>
<td>09:10:04,610172</td>
<td>0.000112</td>
<td>192.168.0.1</td>
<td>192.168.0.1</td>
<td>TCP</td>
<td>64</td>
<td>1203 → 2107 [PSH, ACK] Seq=1 Ack=11 Win=130804 Len=10</td>
</tr>
<tr>
<td>12</td>
<td>09:10:04,610245</td>
<td>0.000073</td>
<td>192.168.0.1</td>
<td>192.168.0.1</td>
<td>TCP</td>
<td>64</td>
<td>1546 → 2107 [PSH, ACK] Seq=1 Ack=11 Win=129792 Len=10</td>
</tr>
<tr>
<td>13</td>
<td>09:10:04,735816</td>
<td>0.124771</td>
<td>192.168.0.1</td>
<td>192.168.0.1</td>
<td>TCP</td>
<td>64</td>
<td>1188 → 2107 [PSH, ACK] Seq=1 Ack=11 Win=130816 Len=10</td>
</tr>
</tbody>
</table>

Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits)
Ethernet II, Src: 02:00:00:00:00:00 (02:00:00:00:00:00), Dst: 02:00:00:00:00:00
Internet Protocol Version 4, Src: 10.0.0.1, Dst: 192.168.0.1
Transmission Control Protocol, Src Port: 1181, Dst Port: 2187, Seq: 1, Ack: 1, Len: 64
Transmission Control Protocol, Src Port: 2187, Dst Port: 1181, Seq: 1, Ack: 1, Len: 64
Trade PDU:
Sender: 11
-- Add sender
local sender = tvb(offset,1):uint()
info = tostring(sender)

subtree:add(f.sender,tvb(offset,1))
offset = offset + 1

subtree:append_text(info)
pinfo.cols.info = info
Result of 05.lua

Custom LUA dissectors to the rescue in root cause analysis
local modules = {
    [1] = 'Multiplexer',
    [7] = 'Tradesite',
    [93] = 'Zep',
    [99] = 'Scripting'
}

f.sender = ProtoField.uint8("trade.sender","Sender",base.DEC,modules)

-- Add sender
local sender = tvb(offset,1):uint()  
local mod = modules[sender]
assert( mod ~= nil, "Unknown Sender!" )
info = mod
Result of 06.lua

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Delta</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>09:10:04,360322</td>
<td>0.000000</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>BackOffice</td>
</tr>
<tr>
<td>2</td>
<td>09:10:04,360410</td>
<td>0.000088</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>MarketConfiguration</td>
</tr>
<tr>
<td>3</td>
<td>09:10:04,360413</td>
<td>0.000131</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>Scripting</td>
</tr>
<tr>
<td>4</td>
<td>09:10:04,562429</td>
<td>0.202161</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TCP</td>
<td>64</td>
<td>Settlement</td>
</tr>
<tr>
<td>5</td>
<td>09:10:04,562433</td>
<td>0.000040</td>
<td>192.168.0.1</td>
<td>192.168.0.1</td>
<td>TCP</td>
<td>64</td>
<td>TranquilizerUpdate</td>
</tr>
<tr>
<td>6</td>
<td>09:10:04,562436</td>
<td>0.000010</td>
<td>192.168.0.1</td>
<td>192.168.0.1</td>
<td>TCP</td>
<td>64</td>
<td>DocumentSystem</td>
</tr>
<tr>
<td>7</td>
<td>09:10:04,690538</td>
<td>0.047102</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>DocumentSystem</td>
</tr>
<tr>
<td>8</td>
<td>09:10:04,699921</td>
<td>0.000383</td>
<td>192.168.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>DocumentSystem</td>
</tr>
<tr>
<td>9</td>
<td>09:10:04,699974</td>
<td>0.000053</td>
<td>192.168.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>DocumentSystem</td>
</tr>
<tr>
<td>10</td>
<td>09:10:04,610060</td>
<td>0.000060</td>
<td>192.168.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>DocumentSystem</td>
</tr>
<tr>
<td>11</td>
<td>09:10:04,610172</td>
<td>0.000112</td>
<td>192.168.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>DocumentSystem</td>
</tr>
<tr>
<td>12</td>
<td>09:10:04,610245</td>
<td>0.000073</td>
<td>192.168.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>DocumentSystem</td>
</tr>
<tr>
<td>13</td>
<td>09:10:04,735016</td>
<td>0.124771</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>BackOffice</td>
</tr>
</tbody>
</table>

Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits)
Ethernet II, Src: 00:00:00:00:00:01 (02:00:00:00:00:01), Dst: 02:00:00:00:00:00
Internet Protocol Version 4, Src: 10.0.0.1, Dst: 192.168.0.1
Transmission Control Protocol, Src Port: 1337, Dst Port: 2187, Seq: 1, Ack: 1, Len: 64 bytes
Trade PDU: BackOffice
07.lua: Add more fields

sake@MacSake:~/Dropbox/sharkfest/2017eu/anonymized$ diff 06.lua 07.lua
23a24,25
>     f.len        = ProtoField.uint16("trade.length","Length",base.DEC)
>     f.receiver   = ProtoField.uint16("trade.receiver","Receiver",base.DEC)
44a47,57
>         -- Add length
>         subtree:add(f.len,tvb(offset,2))
>         offset = offset + 2
>
>         -- Add receiver
>         local receiver = tvb(offset,2):uint()
>         info = info .. " -> " .. receiver
>         subtree:add(f.receiver,tvb(offset,2))
>         offset = offset + 2
>
sake@MacSake:~/Dropbox/sharkfest/2017eu/anonymized$
Result of 07.lua
08.lua: Differentiate type of packets

sake@MacSake:~/Dropbox/sharkfest/2017eu/anonymized$ diff -b 07.lua 08.lua
36,38d35
<         subtree = tree:add(trade,tvb(offset,tvb:len()),"Trade PDU :")
< <
<          -- Add sender
3a41,49
>         local receiver = tvb(offset+3,2):uint()
>         if receiver == 65535 then
>             subtree = tree:add(trade,tvb(offset,tvb:len()),"Trade Control PDU :")
>         else
>             subtree = tree:add(trade,tvb(offset,tvb:len()),"Trade Data PDU :")
>         end
>         -- Add sender
52,55c58
<         local receiver = tvb(offset,2):uint()
<         info = info .. " -> " .. receiver
<         subtree:add(f.receiver,tvb(offset,2))
< --
>         local t = subtree:add(f.receiver,tvb(offset,2))
57a61,69
>         if receiver == 65535 then
>             -- Dissect control PDU
>             t:append_text(" (Control)")
>             info = info .. " -> Control"
>         else
>             -- Dissect data PDU
>             info = info .. " -> " .. receiver
>         end
sake@MacSake:~/Dropbox/sharkfest/2017eu/anonymized$
Result of 08.lua
Multiple PDU’s in one packet?
repeat
  -- Add sender
  subtree:add(f.sender, tvb(offset,1))
  offset = offset + 1
  -- Add length
  -- Add receiver
  -- Add optional data
  until offset >= tvb:len() or count >= 100

if count == 1 then
  pinfo.cols.info = info
else
  pinfo.cols.info = count .. " Trade PDUs"
end
repeat
  len = tvb(offset+1,2):uint()
  -- Add sender
  subtree:add(f.sender,tvb(offset,1))
  offset = offset + 1
  len = len - 1
  -- Add length etc.

  if len > 0 then
    -- Add data field for unknown data
    local data = tvb(offset,len):bytes()
    subtree:add(f.data,tvb(offset,data:len()))
    offset = offset + data:len()
    len = len - data:len()
  end
  count = count + 1
until offset >= tvb:len() or count >= 100
Result of 09.lua

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Delta</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>51</td>
<td>09:05:09.9621</td>
<td>0.00551</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>Multiplexer -&gt; Control, len=10</td>
</tr>
<tr>
<td>52</td>
<td>09:05:09.9680</td>
<td>0.00479</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>Multiplexer -&gt; Control, len=10</td>
</tr>
<tr>
<td>53</td>
<td>09:05:09.9757</td>
<td>0.00475</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>Multiplexer -&gt; Control, len=10</td>
</tr>
<tr>
<td>54</td>
<td>09:05:09.9805</td>
<td>0.00403</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>Multiplexer -&gt; Control, len=10</td>
</tr>
<tr>
<td>55</td>
<td>09:05:09.9937</td>
<td>0.00479</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>Multiplexer -&gt; Control, len=10</td>
</tr>
<tr>
<td>56</td>
<td>09:05:10.0040</td>
<td>0.00628</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>Multiplexer -&gt; Control, len=10</td>
</tr>
<tr>
<td>57</td>
<td>09:05:10.18787</td>
<td>0.000238</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>Multiplexer -&gt; Control, len=10</td>
</tr>
<tr>
<td>58</td>
<td>09:05:10.18772</td>
<td>0.001748</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>Multiplexer -&gt; Control, len=10</td>
</tr>
<tr>
<td>59</td>
<td>09:05:10.192796</td>
<td>0.003070</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TRADE</td>
<td>64</td>
<td>Multiplexer -&gt; Control, len=10</td>
</tr>
<tr>
<td>60</td>
<td>09:05:10.295692</td>
<td>0.102896</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TCP</td>
<td>60</td>
<td>1188 -&gt; 2107 [ACK] Seq=29 Ack=29 Win=138016 Len=0</td>
</tr>
<tr>
<td>61</td>
<td>09:05:10.294354</td>
<td>0.03743</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TCP</td>
<td>60</td>
<td>1192 -&gt; 2107 [ACK] Seq=29 Ack=29 Win=138304 Len=0</td>
</tr>
<tr>
<td>62</td>
<td>09:05:10.294954</td>
<td>0.00006</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TCP</td>
<td>60</td>
<td>1184 -&gt; 2107 [ACK] Seq=29 Ack=29 Win=129792 Len=0</td>
</tr>
<tr>
<td>63</td>
<td>09:05:10.294984</td>
<td>0.00003</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TCP</td>
<td>60</td>
<td>1200 -&gt; 2107 [ACK] Seq=29 Ack=29 Win=138560 Len=0</td>
</tr>
</tbody>
</table>

Frame 57: 522 bytes on wire (4176 bits), 522 bytes captured (4176 bits)
Ethernet II, Src: 02:00:00:00:00:00 (02:00:00:00:00:00), Dst: 02:00
Internet Protocol Version 4, Src: 192.168.0.1, Dst: 10.0.0.1
Transmission Control Protocol, Src Port: 2107, Dst Port: 1170, Seq: 0
Trade Control PDU : Multiplexer -> Control, len=47
Sender: Multiplexer (1)
Length: 47
Receiver: 65535 (Control)

Data: Roundtrip For Module

Frame 58: 522 bytes on wire (4176 bits), 522 bytes captured (4176 bits)
Ethernet II, Src: 02:00:00:00:00:00 (02:00:00:00:00:00), Dst: 02:00
Internet Protocol Version 4, Src: 192.168.0.1, Dst: 10.0.0.1
Transmission Control Protocol, Src Port: 2107, Dst Port: 1170, Seq: 0
Trade Control PDU : Multiplexer -> Control, len=42
Sender: Multiplexer (1)
Length: 42
Receiver: 65535 (Control)

Data (trade.data), 42 bytes
Packets: 33471 - Displayed: 33471 (100.0%) - Load time: 0.933s Profile: sf2017-secure-dummy

Custom LUA dissectors to the rescue in root cause analysis
PDU’s spanning multiple packets

Custom LUA dissectors to the rescue in root cause analysis
local offset = pinfo.desegment_offset or 0
repeat
    if tvb:len() >= offset + 3 then
        -- we have enough bytes for the length field of the PDU
        len = tvb(offset+1,2):uint()
        if offset + len > tvb:len() then
            -- we don't have all the data we need for the full PDU yet
            pinfo.desegment_offset = offset
            pinfo.desegment_len = len - tvb:len()
            return
        end
    else
        -- we don't have all of length field yet
        pinfo.desegment_offset = offset
        pinfo.desegment_len = DESEGMENT_ONE_MORE_SEGMENT
        return
    end
end
Result of 10.lua
11.lua: Add more fields

```
sake@MacSake:~/Dropbox/sharkfest/2017eu/anonymized$ diff 10.lua 11.lua
25a26
>     f.keyword    = ProtoField.stringz("trade.keyword","Keyword")
88a90,101
>     -- Add keyword
>     local keyword = tvb(offset):stringz()
>     if keyword:len() > 0 then
>         subtree:add(f.keyword,tvb(offset,keyword:len()))
>         info = info .. ", " .. keyword
>         offset = offset + keyword:len() + 1
>         len = len - keyword:len() - 1
>     else
>         -- there was no keyword (first byte was 0x00)
>         subtree:add_expert_info(PI_MALFORMED,PI_ERROR,"There was no keyword in this control frame")
>     end
sake@MacSake:~/Dropbox/sharkfest/2017eu/anonymized$
```
Custom LUA dissectors to the rescue in root cause analysis

Result of 11.lua

Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits)
Ethernet II, Src: 02:00:00:00:00:01 (02:00:00:00:00:01), Dst: 02:00:00:00:00:02 (02:00:00:00:00:02)
Internet Protocol Version 4, Src: 10.0.0.1, Dst: 192.168.0.1
Transmission Control Protocol, Src Port: 1181, Dst Port: 2107, Seq: 1, Ack: 1, Len: 10

Trade Control PDU: BackOffice -> Control, Ping, len=10
Sender: BackOffice (11)
Length: 10
Keyword: Ping
12.lua: Add more fields

```lua
sake@MacSake:~/Dropbox/sharkfest/2017eu/anonymized$ diff -b 11.lua 12.lua
26a27
> f.ping_clock = ProtoField.uint64("trade.ping_clock","Ping Time")
95c96
< info = info .. ", " .. keyword
---
> info = info .. ", "
97a99,114
> if len > 0 then
> if keyword == "Ping" or
> keyword == "Pong" then
> -- length (1 byte)
> -- time in microseconds (since Jan 1, 1901, see http://en.wikipedia.org/wiki/System_time)
> local clocklen = tvb(offset,1):uint()
> subtree:add_le(f.ping_clock,tvb(offset+1,clocklen))
> offset = offset + clocklen + 1
> len = len - clocklen - 1
> info = info .. keyword
> else
> info = info .. keyword .. " (Unknown keyword)"
> end
> else
> info = info .. keyword
> end
sake@MacSake:~/Dropbox/sharkfest/2017eu/anonymized$
```
Result of 12.lua
13.lua: Add more fields

```lua
elseif keyword == "Roundtrip for Module" then
    -- string with module name
    -- length (1 byte)
    -- time in milliseconds
    local mod = tvb(offset):stringz()
    subtree:add(f.mod,tvb(offset,mod:len()))
    offset = offset + mod:len() + 1
    len = len - mod:len() - 1

    local clocklen = tvb(offset,1):uint()
    assert(clocklen<=4, "clocklength too big")

    local rtt_time = tvb(offset+1,clocklen):le_uint()
    local t = subtree:add(f.rtt_time,tvb(offset+1,clocklen),rtt_time/1000)
    t:append_text(" ms")
    offset = offset + clocklen + 1
    len = len - clocklen - 1

    info = info .. "Roundtrip for Module " .. mod .. " is " .. rtt_time/1000 .. " ms"
```

```bash
sake@MacSake:~/Dropbox/sharkfest/2017eu/anonymized$ diff -b 12.lua 13.lua
27a28,29
>     f.rtt_time   = ProtoField.double("trade.rtt_time","Roundtrip Time")
>     f.mod        = ProtoField.stringz("trade.module","Module")
108a111,131
>
    elseif keyword == "Roundtrip for Module" then
        -- string with module name
        -- length (1 byte)
        -- time in milliseconds
        local mod = tvb(offset):stringz()
        subtree:add(f.mod,tvb(offset,mod:len()))
        offset = offset + mod:len() + 1
        len = len - mod:len() - 1

        local clocklen = tvb(offset,1):uint()
        assert(clocklen<=4, "clocklength too big")

        local rtt_time = tvb(offset+1,clocklen):le_uint()
        local t = subtree:add(f.rtt_time,tvb(offset+1,clocklen),rtt_time/1000)
        t:append_text(" ms")
        offset = offset + clocklen + 1
        len = len - clocklen - 1

        info = info .. "Roundtrip for Module " .. mod .. " is " .. rtt_time/1000 .. " ms"
```
Result of 13.lua

```
Frame 59: 95 bytes on wire (760 bits), 95 bytes captured (760 bits)
Ethernet II, Src: 00:00:00:00:00:02 (02:00:00:00:00:02), Dst: 02:00:00:00:00:00:01 (02:00:00:00:00:00:01)
Internet Protocol Version 4, Src: 192.168.0.1, Dst: 10.0.0.1
```

```
Sender: Multiplexer (1)
Length: 41
Receiver: 65535 (Control)
Keyword: Roundtrip for Module
Module: Tradesite
```

```
Roundtrip Time: 68,107 ms
```

```
Roundtrip Time (trade.rtt.time), 4 bytes
```

```
Packets: 33471 · Displayed: 323 (1.0%) · Load time: 0.0:0.886 · Profile: sf2017eu-dummy
```
sake@MacSake:~/Dropbox/sharkfest/2017eu/anonymized$ diff -b 13.lua 14.lua
29a30,31
>     f.conn_id    = ProtoField.uint16("trade.conn_id","Connection ID",base.DEC)
>     f.client_ip  = ProtoField.stringz("trade.client_ip","Client IP")

131a134,157
> elseif keyword == "New Connection" then
>     -- 2 bytes connection ID
>     -- string with IP address
>     local conn_id = tvb(offset,2):uint()
>     subtree:add(f.conn_id,tvb(offset,2))
>     offset = offset + 2
>     len = len - 2
>     local ip = tvb(offset):stringz()
>     subtree:add(f.client_ip,tvb(offset,ip:len()))
>     offset = offset + ip:len() + 1
>     len = len - ip:len() - 1
>     info = info .. "New Connection from " .. ip .. " (id=": .. conn_id .. ")"
>
> elseif keyword == "Close Connection" then
>     -- 2 bytes connection ID
>     local conn_id = tvb(offset,2):uint()
>     subtree:add(f.conn_id,tvb(offset,2))
>     offset = offset + 2
>     len = len - 2
>     info = info .. "Connection closed (id=": .. conn_id .. ")"
>
sake@MacSake:~/Dropbox/sharkfest/2017eu/anonymized$
Result of 14.lua

```
Keyword: New Connection

Connection ID: 2645
Client IP: 194.134.005.011
```

Custom LUA dissectors to the rescue in root cause analysis
Result of 14.lua
15.lua: Add more fields

```lua
sake@MacSake:~/Dropbox/sharkfest/2017eu/anonymized$ diff -b 14.lua 15.lua
31a32
>     f.username   = ProtoField.stringz("trade.username","Username")
157a159,195
>                         elseif keyword == "Login" then
>                             -- string with module (by sender Sylvester (7))
>                             -- 2 bytes connection ID
>                             -- string with username
>                             -- string with IP address
>                             -- "read"
>                             -- 0x00
>                             local mod
>                             if sender == 7 then
>                                 mod = tvb(offset):stringz()
>                                 subtree:add(f.mod,tvb(offset,mod:len()))
>                                 offset = offset + mod:len() + 1
>                                 len = len - mod:len() - 1
>                             end
>
>                             local conn_id = tvb(offset,2):uint()
>                             subtree:add(f.conn_id,tvb(offset,2))
>                             offset = offset + 2
>                             len = len - 2
>                             local username = tvb(offset):stringz()
>                             subtree:add(f.username,tvb(offset,username:len()))
>                             offset = offset + username:len() + 1
>                             len = len - username:len() - 1
>                             local ip = tvb(offset):stringz()
>                             subtree:add(f.client_ip,tvb(offset,ip:len()))
>                             offset = offset + ip:len() + 1
>                             len = len - ip:len() - 1
>                             if sender == 7 then
>                                 info = info .. " User " .. username .. " logged in from " .. ip .. " to " .. mod .. " (id=" .. conn_id .. ")"
>                             else
>                                 info = info .. " User " .. username .. " logged in from " .. ip .. " (id=" .. conn_id .. ")"
>                             end
>
sake@MacSake:~/Dropbox/sharkfest/2017eu/anonymized$
```
Result of 15.lua

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Delta</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>09:14:07</td>
<td>851579</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>111</td>
<td>Tradesite -&gt; Control, User XXXXX0011 logged in from ...</td>
</tr>
<tr>
<td>1246</td>
<td>09:14:08</td>
<td>852641</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>100</td>
<td>Multiplexer -&gt; Control, User XXXXX0011 logged in from ...</td>
</tr>
<tr>
<td>1249</td>
<td>09:14:08</td>
<td>863517</td>
<td>0.210876</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>714</td>
<td>5 Trade PDUs</td>
</tr>
<tr>
<td>1250</td>
<td>09:14:08</td>
<td>869751</td>
<td>0.006234</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>106</td>
<td>Multiplexer -&gt; Control, User XXXXX0011 logged in from ...</td>
</tr>
<tr>
<td>1251</td>
<td>09:14:08</td>
<td>872399</td>
<td>0.002588</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>106</td>
<td>Multiplexer -&gt; Control, User XXXXX0011 logged in from ...</td>
</tr>
<tr>
<td>1252</td>
<td>09:14:08</td>
<td>874901</td>
<td>0.002562</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>106</td>
<td>Multiplexer -&gt; Control, User XXXXX0011 logged in from ...</td>
</tr>
<tr>
<td>2364</td>
<td>09:16:14</td>
<td>1416221</td>
<td>126.3413</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>111</td>
<td>Tradesite -&gt; Control, User XXXXX0011 logged in from ...</td>
</tr>
<tr>
<td>2365</td>
<td>09:16:14</td>
<td>1418420</td>
<td>0.002199</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>106</td>
<td>Multiplexer -&gt; Control, User XXXXX0011 logged in from ...</td>
</tr>
<tr>
<td>2368</td>
<td>09:16:14</td>
<td>1626496</td>
<td>0.280876</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>714</td>
<td>5 Trade PDUs</td>
</tr>
<tr>
<td>2369</td>
<td>09:16:14</td>
<td>1627826</td>
<td>0.001330</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>106</td>
<td>Multiplexer -&gt; Control, User XXXXX0011 logged in from ...</td>
</tr>
<tr>
<td>2370</td>
<td>09:16:14</td>
<td>1629067</td>
<td>0.001241</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>106</td>
<td>Multiplexer -&gt; Control, User XXXXX0011 logged in from ...</td>
</tr>
</tbody>
</table>

Frame 1245: 111 bytes on wire (888 bits), 111 bytes captured (888 bits)
Ethernet II, Src: 02:00:00:00:00:01 (02:00:00:00:00:01), Dst: 02:00:00:00:00:02 (02:00:00:00:00:02)
Internet Protocol Version 4, Src: 10.0.0.1, Dst: 192.168.0.1
Trade Control PDU: Tradesite -> Control, User XXXXX0011 logged in from 194.134.005.011 to BackOffice (id=2645), len=57
Sender: Tradesite (7)
Length: 57
Receiver: 65535 (Control)
Keyword: Login
Module: BackOffice
Connection ID: 2645
Username: XXXXX0011
Client IP: 194.134.005.011
Data: Read
16.lua: Add more fields

```lua
sake@MacSake:~/Dropbox/sharkfest/2017eu/anonymized$ diff -b 15.lua 16.lua
32a33,35 > f.module_count = ProtoField.uint16("trade.module_count","Module Count",base.DEC)
> f.client_count = ProtoField.uint16("trade.client_count","Client Count",base.DEC)
41c44 < local offset = pinfo.desegment_offset or 0
---
> local offset = 0
195a199,219 > elseif keyword == "ModuleAndClientCount" then
>     local module_count = tvb(offset,2):uint()
>     subtree:add(f.module_count,tvb(offset,2))
>     offset = offset + 2
>     len = len - 2
>     local client_count = tvb(offset,2):uint()
>     subtree:add(f.client_count,tvb(offset,2))
>     offset = offset + 2
>     len = len - 2
>     info = info .. keyword .. ", Modules: " .. module_count .. ", Clients: " .. client_count

sake@MacSake:~/Dropbox/sharkfest/2017eu/anonymized$
```
Custom LUA dissectors to the rescue in root cause analysis

Result of 16.lua
17.lua: Use fields from other layers

```lua
local tcp_seq_f = Field.new("tcp.seq")

f.greeting = ProtoField.stringz("trade.greeting","Server Greeting")

local offset = 0

-- Handle the server greeting message
local tcp_seq = tostring(tcp_seq_f())
len = tvb(1,1):uint()
if tcp_seq == "1" and (tvb:len() == len + 3) then
    subtree = tree:add(trade,tvb(0,len),"Trade Server Greeting : ")
    subtree:add(f.sender,tvb(0,1))
    subtree:add(f.len,tvb(1,1))
    subtree:add(f.greeting,tvb(2,len - 2))
    local mod = tvb(2,len - 2):stringz()
    subtree:append_text(mod)
    pinfo.cols.info = "Setting up connection to " .. mod
    return len
end
```

sake@MacSake:~/Dropbox/sharkfest/2017eu/anonymized$
Result of 17.lua

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Delta</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>31380</td>
<td>09:32:03</td>
<td>0.00000</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td></td>
<td>63 Setting up connection to Trader</td>
</tr>
<tr>
<td>32779</td>
<td>09:35:32</td>
<td>208.915</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td></td>
<td>73 Setting up connection to JackhammerUpdate</td>
</tr>
<tr>
<td>32781</td>
<td>09:35:32</td>
<td>0.00388</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td></td>
<td>75 Setting up connection to TranquilizerUpdate</td>
</tr>
<tr>
<td>32788</td>
<td>09:35:32</td>
<td>0.01358</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td></td>
<td>75 Setting up connection to TradesiteScripting</td>
</tr>
<tr>
<td>32789</td>
<td>09:35:32</td>
<td>0.00295</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td></td>
<td>71 Setting up connection to DocumentSystem</td>
</tr>
<tr>
<td>327...</td>
<td>09:35:32</td>
<td>0.00350</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td></td>
<td>67 Setting up connection to BackOffice</td>
</tr>
<tr>
<td>32802</td>
<td>09:35:32</td>
<td>0.02585</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td></td>
<td>76 Setting up connection to MarketConfiguration</td>
</tr>
<tr>
<td>32804</td>
<td>09:35:32</td>
<td>0.02103</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td></td>
<td>72 Setting up connection to FinancialModule</td>
</tr>
<tr>
<td>32805</td>
<td>09:35:32</td>
<td>0.00938</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td></td>
<td>67 Setting up connection to Settlement</td>
</tr>
<tr>
<td>32806</td>
<td>09:35:32</td>
<td>0.00945</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td></td>
<td>70 Setting up connection to Zap</td>
</tr>
<tr>
<td>32812</td>
<td>09:35:32</td>
<td>0.02397</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td></td>
<td>69 Setting up connection to TradesiteZep</td>
</tr>
</tbody>
</table>

Frame 32700: 67 bytes on wire (536 bits), 67 bytes captured (536 bits)
- Ethernet II, Src: 02:00:00:00:00:01 (02:00:00:00:00:00:01), Dst: 02:00:00:00:00:00:00
- Internet Protocol Version 4, Src: 10.0.0.1, Dst: 192.168.0.1
- Transmission Control Protocol, Src Port: 1617, Dst Port: 2187, Seq: 1

Trade Server Greeting: BackOffice
- Server: BackOffice (11) Length: 10

Server Greeting (trade.greeting), 11 bytes
Road taken

- Create a LUA dissector skeleton
- Add fields, fill info column and subtree for the basic PDU header
- Add multi-PDU support
- Add reassembly
- Dissect more parts of the PDU’s
- Now back to troubleshooting!!!
Ping-Pong?

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Delta</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Roundtrip Time</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>12117</td>
<td>09:26:56.806849</td>
<td>0.001147</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td>TRADE</td>
<td>64</td>
<td></td>
<td>Multiplexer -&gt; Control, Pong, len=10</td>
</tr>
<tr>
<td>12118</td>
<td>09:26:56.807115</td>
<td>0.002666</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td></td>
<td></td>
<td></td>
<td>Trader -&gt; Control, Pong, len=10</td>
</tr>
<tr>
<td>14858</td>
<td>09:27:26.813595</td>
<td>30.006844</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td></td>
<td></td>
<td></td>
<td>Multiplexer -&gt; Control, Pong, len=10</td>
</tr>
<tr>
<td>14859</td>
<td>09:27:26.816878</td>
<td>0.000119</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td></td>
<td></td>
<td></td>
<td>Trader -&gt; Control, Pong, len=10</td>
</tr>
<tr>
<td>14861</td>
<td>09:27:26.864865</td>
<td>0.408007</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td></td>
<td></td>
<td></td>
<td>Multiplexer -&gt; Control, Pong, len=10</td>
</tr>
<tr>
<td>14869</td>
<td>09:27:26.864242</td>
<td>0.001517</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td></td>
<td></td>
<td></td>
<td>Trader -&gt; Control, Pong, len=10</td>
</tr>
<tr>
<td>15898</td>
<td>09:27:26.877668</td>
<td>30.013426</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td></td>
<td></td>
<td></td>
<td>Multiplexer -&gt; Control, Pong, len=10</td>
</tr>
<tr>
<td>15899</td>
<td>09:27:26.877668</td>
<td>0.001191</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td></td>
<td></td>
<td></td>
<td>Trader -&gt; Control, Pong, len=10</td>
</tr>
<tr>
<td>15891</td>
<td>09:27:56.877571</td>
<td>0.000803</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td></td>
<td></td>
<td></td>
<td>Multiplexer -&gt; Control, Pong, len=10</td>
</tr>
<tr>
<td>15894</td>
<td>09:27:56.880857</td>
<td>0.202366</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td></td>
<td></td>
<td></td>
<td>Trader -&gt; Control, Pong, len=10</td>
</tr>
<tr>
<td>15896</td>
<td>09:27:57.921887</td>
<td>0.841830</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td></td>
<td></td>
<td></td>
<td>Multiplexer -&gt; Control, Pong, len=10</td>
</tr>
<tr>
<td>16345</td>
<td>09:28:27.937658</td>
<td>30.015171</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td></td>
<td></td>
<td></td>
<td>Trader -&gt; Control, Pong, len=10</td>
</tr>
<tr>
<td>16346</td>
<td>09:28:27.937194</td>
<td>0.000136</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td></td>
<td></td>
<td></td>
<td>Multiplexer -&gt; Control, Pong, len=10</td>
</tr>
<tr>
<td>16348</td>
<td>09:28:28.139967</td>
<td>0.202673</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td></td>
<td></td>
<td></td>
<td>Trader -&gt; Control, Pong, len=10</td>
</tr>
<tr>
<td>16349</td>
<td>09:28:28.140156</td>
<td>0.002829</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td></td>
<td></td>
<td></td>
<td>Multiplexer -&gt; Control, Pong, len=10</td>
</tr>
<tr>
<td>17114</td>
<td>09:28:58.154115</td>
<td>30.013959</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td></td>
<td></td>
<td></td>
<td>Trader -&gt; Control, Pong, len=10</td>
</tr>
<tr>
<td>17115</td>
<td>09:28:58.154966</td>
<td>0.000851</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td></td>
<td></td>
<td></td>
<td>Multiplexer -&gt; Control, Pong, len=10</td>
</tr>
<tr>
<td>17116</td>
<td>09:28:58.158058</td>
<td>0.001084</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td></td>
<td></td>
<td></td>
<td>Trader -&gt; Control, Pong, len=10</td>
</tr>
<tr>
<td>17117</td>
<td>09:28:58.159279</td>
<td>0.000229</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td></td>
<td></td>
<td></td>
<td>Multiplexer -&gt; Control, Pong, len=10</td>
</tr>
<tr>
<td>20997</td>
<td>09:29:28.168604</td>
<td>30.013235</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td></td>
<td></td>
<td></td>
<td>Trader -&gt; Control, Pong, len=10</td>
</tr>
<tr>
<td>20998</td>
<td>09:29:28.171420</td>
<td>0.002186</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td></td>
<td></td>
<td></td>
<td>Multiplexer -&gt; Control, Pong, len=10</td>
</tr>
<tr>
<td>20999</td>
<td>09:29:28.172631</td>
<td>0.001191</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td></td>
<td></td>
<td></td>
<td>Trader -&gt; Control, Pong, len=10</td>
</tr>
<tr>
<td>21000</td>
<td>09:29:28.172847</td>
<td>0.000236</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td></td>
<td></td>
<td></td>
<td>Multiplexer -&gt; Control, Pong, len=10</td>
</tr>
<tr>
<td>21165</td>
<td>09:29:58.167667</td>
<td>29.994820</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td></td>
<td></td>
<td></td>
<td>Trader -&gt; Control, Pong, len=10</td>
</tr>
<tr>
<td>21166</td>
<td>09:29:58.169923</td>
<td>0.002256</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td></td>
<td></td>
<td></td>
<td>Multiplexer -&gt; Control, Pong, len=10</td>
</tr>
<tr>
<td>21168</td>
<td>09:29:58.370326</td>
<td>0.200483</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td></td>
<td></td>
<td></td>
<td>Trader -&gt; Control, Pong, len=10</td>
</tr>
<tr>
<td>21169</td>
<td>09:29:58.379998</td>
<td>0.000672</td>
<td>192.168.0.1</td>
<td>10.0.0.1</td>
<td></td>
<td></td>
<td></td>
<td>Multiplexer -&gt; Control, Pong, len=10</td>
</tr>
</tbody>
</table>

Anonymized.pcap

(tcp.stream == 11 & (tcp.flags & (tcp.keyword in ("Ping" | "Pong"))))
The case of the missing ping!

2014-06-05 09:31:58,978 WARN - ModuleServer Trader lost connection to multiplexer on 192.168.0.1:2107

09:31:58 - 0:01:30 = ~09:30:28 and not 09:29:58 !!!
<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Delta</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>RTT</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>17114</td>
<td>09:28:58.154115</td>
<td>30.012684</td>
<td>192.168...</td>
<td>10.0.0.1</td>
<td>TRADE</td>
<td>72</td>
<td>188.409</td>
<td>Multipler -&gt; Control, Ping, len=10</td>
</tr>
<tr>
<td>17117</td>
<td>09:28:58.156279</td>
<td>0.002164</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>72</td>
<td>208.69</td>
<td>Trader -&gt; Control, Ping, len=10</td>
</tr>
<tr>
<td>17118</td>
<td>09:28:58.157637</td>
<td>0.001358</td>
<td>192.168...</td>
<td>10.0.0.1</td>
<td>TRADE</td>
<td>92</td>
<td>3.234</td>
<td>Multipler -&gt; Roundtrip for Moduel Trader</td>
</tr>
<tr>
<td>20097</td>
<td>09:28:29.168604</td>
<td>30.010967</td>
<td>192.168...</td>
<td>10.0.0.1</td>
<td>TRADE</td>
<td>72</td>
<td>3.234</td>
<td>Multipler -&gt; Control, Ping, len=10</td>
</tr>
<tr>
<td>21000</td>
<td>09:28:29.172847</td>
<td>0.004243</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>92</td>
<td>5.386</td>
<td>Multipler -&gt; Control, Roundtrip for Moduel Trader</td>
</tr>
<tr>
<td>21001</td>
<td>09:28:29.174261</td>
<td>0.001434</td>
<td>192.168...</td>
<td>10.0.0.1</td>
<td>TRADE</td>
<td>92</td>
<td>5.386</td>
<td>Multipler -&gt; Roundtrip for Moduel Trader</td>
</tr>
<tr>
<td>21168</td>
<td>09:28:29.378326</td>
<td>30.196805</td>
<td>192.168...</td>
<td>10.0.0.1</td>
<td>TRADE</td>
<td>72</td>
<td>3.234</td>
<td>Multipler -&gt; Roundtrip for Moduel Trader</td>
</tr>
<tr>
<td>21169</td>
<td>09:28:29.378998</td>
<td>0.000672</td>
<td>10.0.0.1</td>
<td>192.168.0.1</td>
<td>TRADE</td>
<td>92</td>
<td>3.234</td>
<td>Multipler -&gt; Control, Roundtrip for Moduel Trader</td>
</tr>
</tbody>
</table>

Frame 21170: 92 bytes on wire (736 bits), 92 bytes captured (736 bits)

- Ethernet II, Src: 02:00:00:00:00:00 (02:00:00:00:00:00), Dst: 02:00:00:00:00:01 (02:00:00:00:00:01)
- Internet Protocol Version 4, Src: 192.168.0.1, Dst: 10.0.0.1

Trade Control PDU: Multipler -> Control, Roundtrip for Module Trader is 188.409 ms, len=38

Sender: Multipler (1)
Length: 38
Receiver: 65535 (Control)
Keyword: Roundtrip for Module
Module: Trader
Roundtrip Time: 188.409 ms
Stop talking... PLEASE!!!
But I have so much to say…
Root Cause Analysis

- APPSERVER1 sends a lot of data and fills receive buffer of Multiplexer
- APPSERVER1 thinks it has sent a “Ping” packet, but packet never left the TCP buffer because of RWIN=0
- After 90 seconds without “Pong”, the APPSERVER1 closes the connection
- RCA: The processes on Multiplexer somehow don’t empty the receive buffer and block communication
Did LUA-dissector help?

- Understand the Ping-Pong mechanism
- Understand the RTT measurement
- Made filtering easier (or even possible)
- Gives the customer insight in their traffic in future troubleshooting events
Summary...

• Proprietary protocols are hard to analyze from hex

• LUA dissector not extremely difficult to write
  ‣ OK, after a lot of googling, so hopefully this presentation helps you from having to do the same ;-)  

• Time to find Root Cause greatly reduced
  ‣ Well, in this case, following up on the Zero-Window problem did point in the right direction. But knowing what actually happened helped in communication with the Software vendor so they could solve the issue more quickly
Q&A

You have Questions

We have Answers
Thank You!
sake.blok@SYN-bit.nl