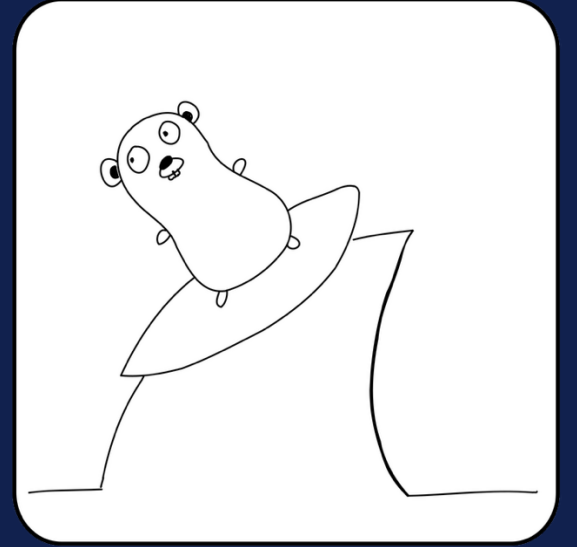


Wirego

One Wireshark plugin to rule them all

Benoît Girard



NETWORK PROTOCOL REVERSE



M6000 boot detect load.pcap

Restart current capture

Apply a display filter ... <#>/>

No.	Time	Source	Destination	Protocol	Length	Info
21	35.078559	192.168.1.125	192.168.1.249	TCP	60	1026 → 1026 [SYN] Seq=0 Win=8192 Len=0 MSS=1460
22	35.079972	192.168.1.249	192.168.1.125	TCP	60	1026 → 1026 [SYN, ACK] Seq=0 Ack=1 Win=4096 Len=0 MSS=1460
23	35.080288	192.168.1.125	192.168.1.249	TCP	60	1026 → 1026 [ACK] Seq=1 Ack=1 Win=8760 Len=0
24	35.137853	192.168.1.249	192.168.1.125	TCP	61	1026 → 1026 [PSH, ACK] Seq=1 Ack=1 Win=4096 Len=7
25	35.297569	192.168.1.125	192.168.1.249	TCP	60	1026 → 1026 [ACK] Seq=1 Ack=8 Win=8753 Len=0
26	35.808286	192.168.1.125	192.168.1.255	Protocol nam...	110	Info example
27	36.094327	192.168.1.125	192.168.1.249	UDP	118	1027 → 1024 Len=76
28	36.100045	192.168.1.125	192.168.1.249	TCP	72	1026 → 1026 [PSH, ACK] Seq=1 Ack=8 Win=8753 Len=18
29	36.111002	192.168.1.249	192.168.1.125	TCP	146	1026 → 1026 [PSH, ACK] Seq=8 Ack=19 Win=4096 Len=92
30	36.111485	192.168.1.125	192.168.1.249	TCP	245	1026 → 1026 [PSH, ACK] Seq=19 Ack=100 Win=8661 Len=191
31	36.122860	192.168.1.249	192.168.1.125	TCP	60	1026 → 1026 [ACK] Seq=100 Ack=210 Win=4096 Len=0
32	36.141731	192.168.1.249	192.168.1.125	TCP	154	1026 → 1026 [PSH, ACK] Seq=100 Ack=210 Win=4096 Len=100

> Frame 30: 245 bytes on wire (1960 bits), 245 bytes captured (1960 bits) on interface 0

> Ethernet II, Src: Advantech_42:20:13 (00:d0:c9:42:20:13), Dst: TcGroup_00:0c:29:00:00:00

> Internet Protocol Version 4, Src: 192.168.1.125, Dst: 192.168.1.249

> Transmission Control Protocol, Src Port: 1026, Dst Port: 1026, Seq: 19, Len: 191

> Data (191 bytes)

Data: 0002000ef000201f00464706780000002af70002000ef000201f004647067

[Length: 191]

0000 00 01 66 04 5e 30 00 d0 c9 42 20 13 00 00 45 00 ...f...B...E...

0010 00 e7 0e 00 40 00 00 06 67 4a c0 a8 01 7d c0 a8 ...@...gJ...)

0020 01 f9 04 02 04 02 00 00 8e 49 25 d9 74 64 50 18 ...I...tdP...

0030 21 d5 ec d0 00 00 00 02 00 0e f0 00 20 1f 00 46 ...!.....F...

0040 47 06 78 00 00 00 2a f7 00 02 00 0e f0 00 20 1f ...G x...*.....

0050 00 46 47 06 79 00 00 00 2a f7 00 02 00 0e f0 00 ...FG y...*.....

0060 20 1f 00 46 47 06 00 00 24 00 02 f7 00 02 00 0e ...FG...\$.....

0070 f0 00 20 1f 00 46 47 06 00 00 28 00 04 f7 00 02 ...FG...{(.....

0080 00 0e f0 00 20 1f 00 46 47 06 0b 00 16 00 04 f7 ...FG...G.....

0090 00 02 00 0b f0 00 20 1f 00 46 4f 01 00 00 f7 00 ...FG...FO.....

00a0 02 00 0b f0 00 20 1f 00 46 4f 01 00 00 f7 00 02 ...FG...FO.....

00b0 00 0d f0 00 20 1f 00 46 45 06 00 00 00 01 f7 00 ...FG...FE.....

00c0 02 00 0e f0 00 20 1f 00 46 47 06 00 00 24 00 02 ...FG...F\$.....

00d0 f7 00 02 00 0e f0 00 20 1f 00 46 47 06 0b 00 16 ...FG...FG.....

00e0 00 04 f7 00 02 00 0e f0 00 20 1f 00 46 47 06 0d ...FG...FG.....

00f0 00 26 00 04 f7 ...&...

Packets: 3686 · Displayed: 3686 (100.0%) · Profile: Default



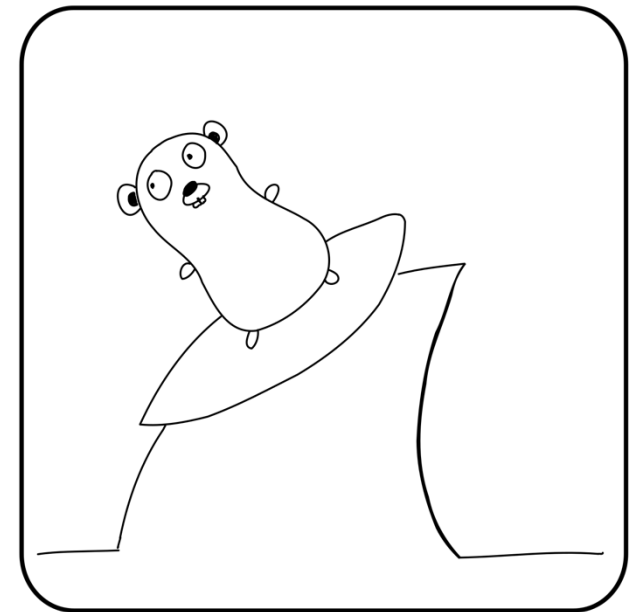
```
func (p *TCPParser) Parse(packet gopacket.Packet, ip *layers.IPv4, tcp *layers.TCP) {  
  
    if len(tcp.Payload) == 0 {  
        return  
    }  
  
    p.logs.Println("*****")  
    p.logs.Printf("[TCP Packet] RAW Payload %d bytes (0x%d)\n", len(tcp.Payload), len(tcp.Payload))  
    p.logs.Print("\n" + hex.Dump(tcp.Payload))  
  
    if (ip.SrcIP.String() == p.frameIP) && (ip.DstIP.String() == p.iconIP) {  
        p.logs.Println("-> Frame to icon (tcp)")  
        p.ParseBlocks(tcp.Payload, common.FrameToIcon)  
    } else if (ip.SrcIP.String() == p.iconIP) && (ip.DstIP.String() == p.frameIP) {  
        p.logs.Println("-> Icon to frame (tcp)")  
        p.ParseBlocks(tcp.Payload, common.IconToFrame)  
    }  
}
```



- First release in 1998
- Plugins can be developed in:
 - C
 - LUA (2006)
 - Python (support removed in 2014)

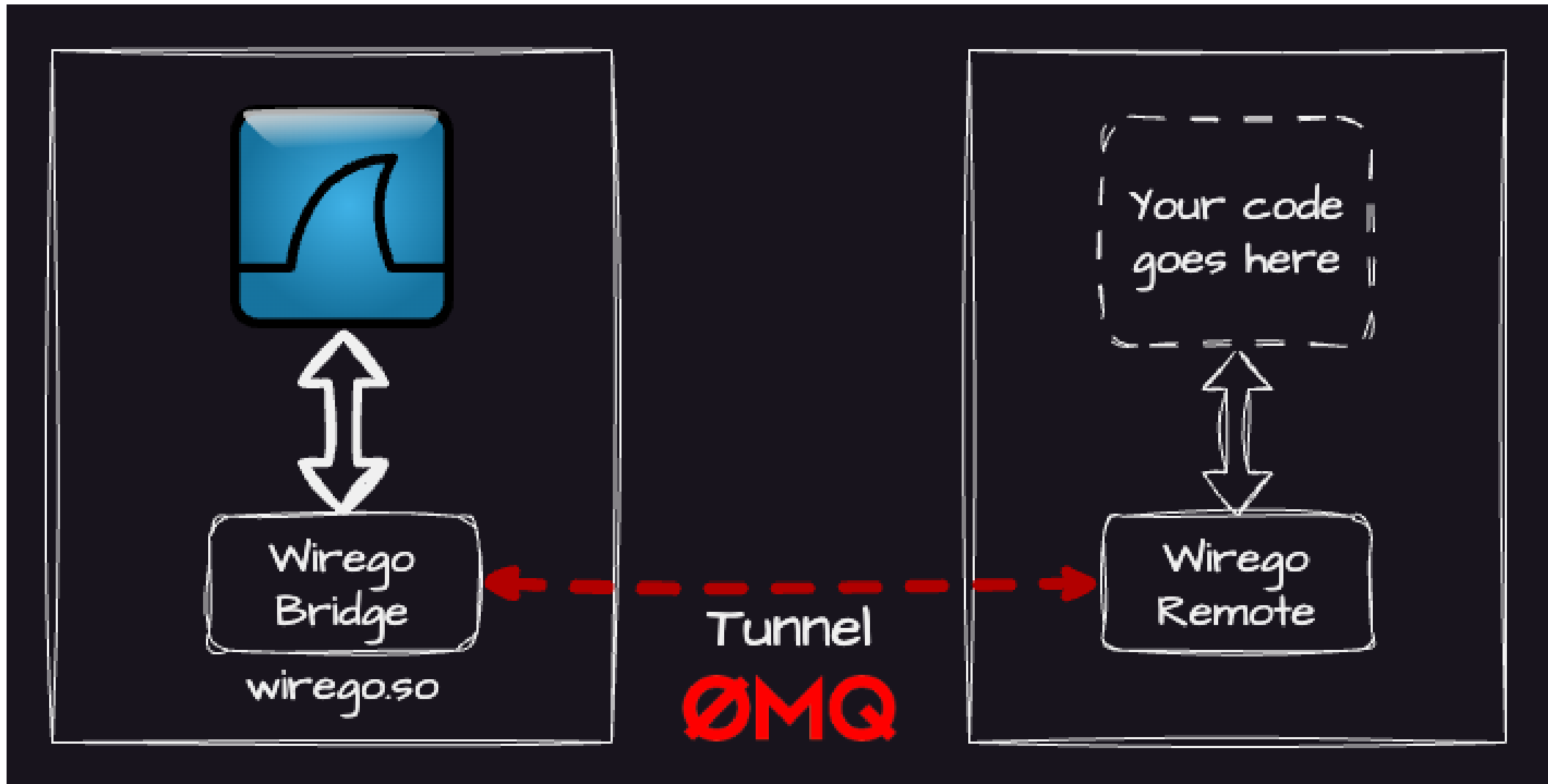
Goals:

- Develop a Wireshark plugin
- In less than 100 code lines
- Without having to build Wireshark
- Without having to read the documentation *
- In any language



* You won't read it anyway

WIREGO - ARCHITECTURE





- High performance asynchronous communication library
- Available on most platforms
- And most languages

Let's develop a Wireshark plugin for a studio recording console (SSL AWS 900)

- On port UDP 50081
- Packets format:

Offset	Size	Description
0	4	Command Code
4	4	DestCode
8	4	Desk Serial
12	4	Remote Serial
16	...	Command Data



CONCLUSION



- Wirego remote packages are already available for:
 - Go
 - Python
 - Rust (thanks to Tomasz Woszczynski)

Ressources	
Project	https://github.com/quarkslab/wirego/
Documentation	https://github.com/quarkslab/wirego/tree/main/doc
Examples	https://github.com/quarkslab/wirego/tree/main/wirego_remote/go/examples https://github.com/quarkslab/wirego/tree/main/wirego_remote/python/examples

Thank you

Contact info:

Email:

bgirard@quarkslab.com

Website:

<https://github.com/quarkslab/wirego/>



@quarkslab