SHARKFEST 2015

WIRESHARK DEVELOPER AND USER CONFERENCE

#SharkFest15

Zen and the art of packet Capturing...

sake.blok@SYN-bit.nl

About me...

- Capturing packets since '99
- Used SnifferPro, but changed to Ethereal quickly :-)
- Lots of bug chasing on Alteon and F5 ADC's
- Missed some features, so started developing in2006
- Became core developer in 2007
- Started SYN-bit in 2009
- Focus on network troubleshooting and providing wireshark/protocol trainings
- And some ADC consultancy (F5 BigIPs, iRules, etc)
- Scuba diving / Arthouse movies

Why capture packets?

- Learn
- Develop
- Solve
- Monitor
- Detect

Capture challenges...

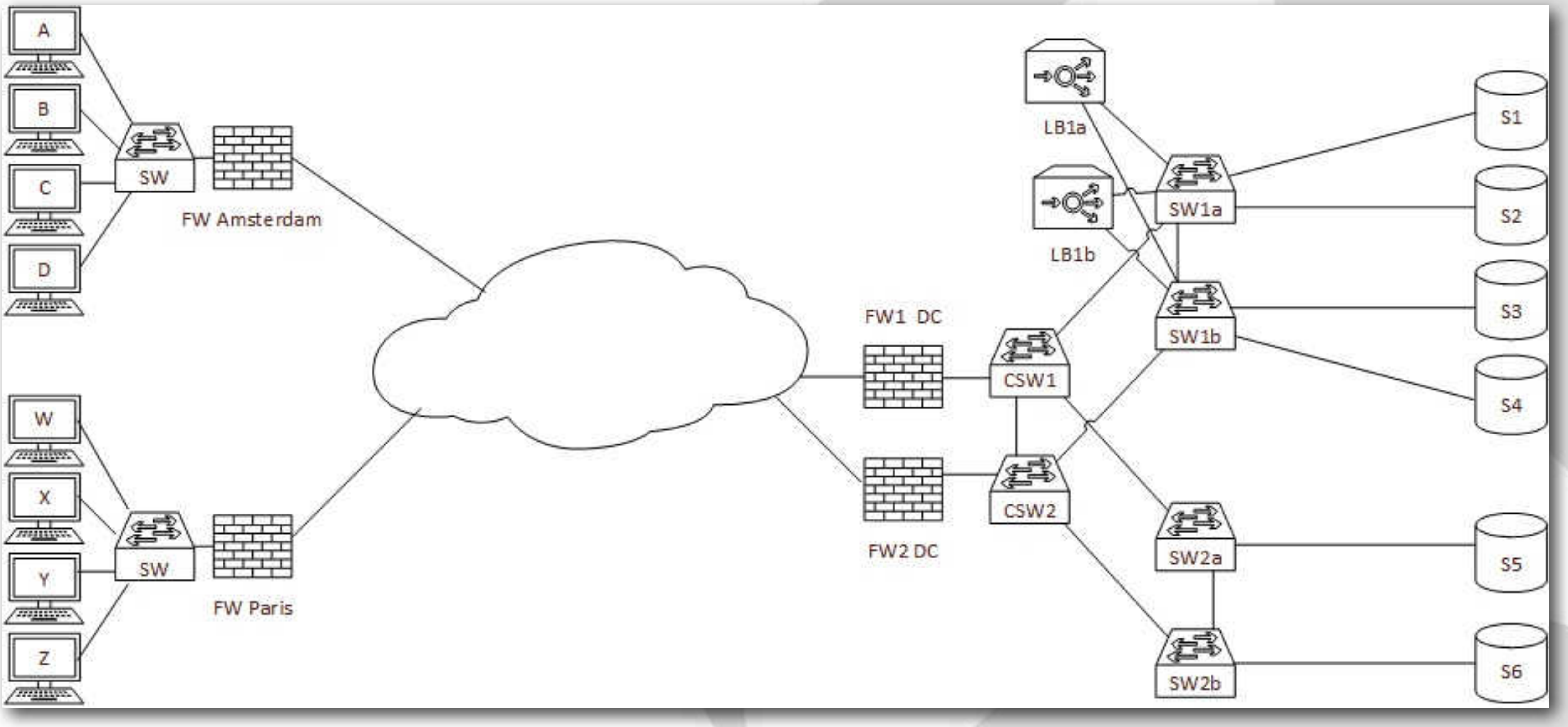
- Where to get the packets
- How to get the packets
- How to timestamp the packets
- How to filter the packets
- How to save the packets

Zen and the art of packet Capturing...

- Where to capture the packets
- How to capture the packets
- Timestamping
- Capture filters
- Saving packets
- Q&A

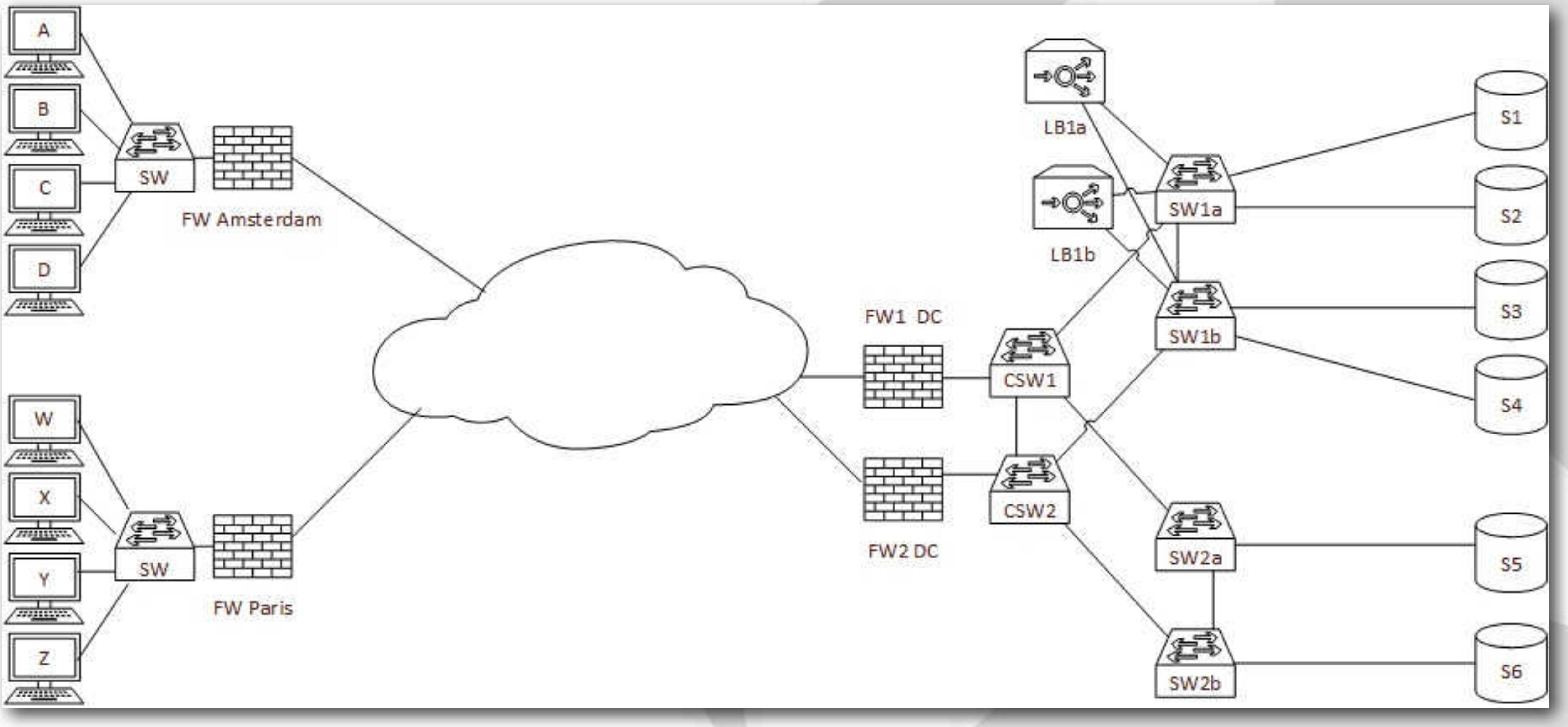
Where would you capture if...

User A complains that something is not working in application on server S5?



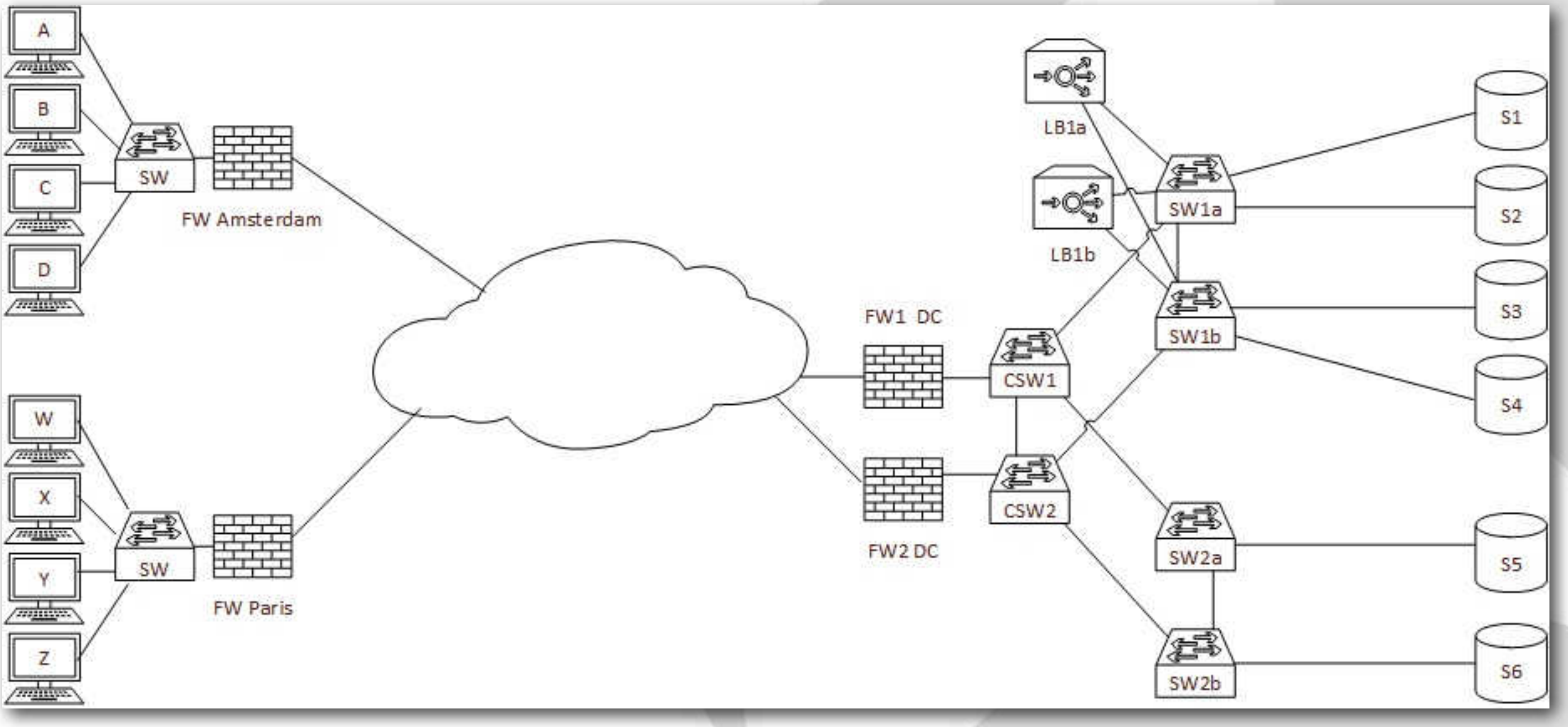
Where would you capture if...

Some users in Paris complain complain about the performance of the loadbalanced application on server S1, S2, S3, S4?



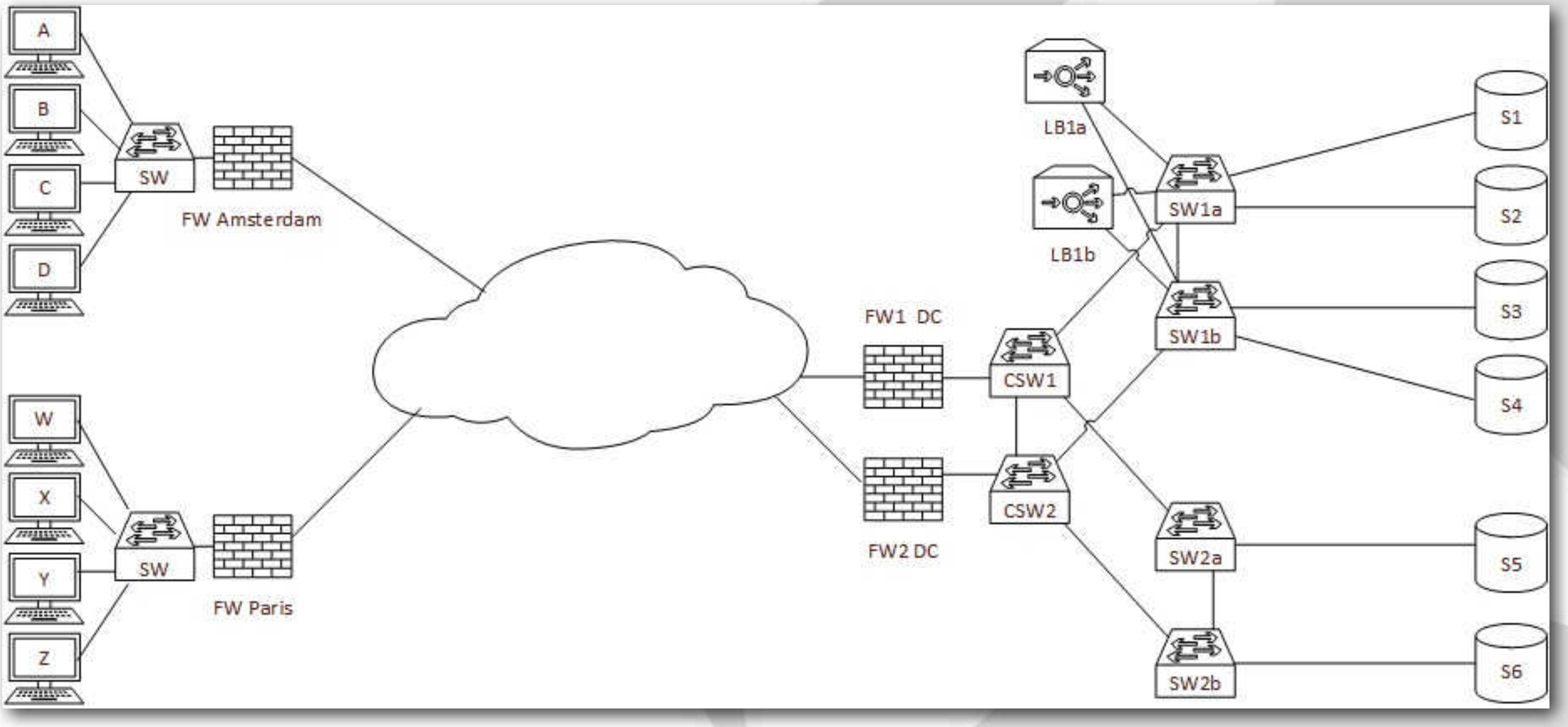
Where would you capture if...

All users complain about the performance of an application on server S6?



Where would you capture if...

All users in Amsterdam complain about slow performance on every application?



Some guidelines...

- Capture close to the client, but not ON the client
- Capture close to the server, but not ON the server
- Capture (on several points) in between the client and server to drill down the source of the problem
- Use intermediate devices like FW, LB, IDS, IPS, WAN optimizers for quick access to packets
- But use SPAN ports and/or TAPs if these devices are under suspicion

Zen and the art of packet Capturing...

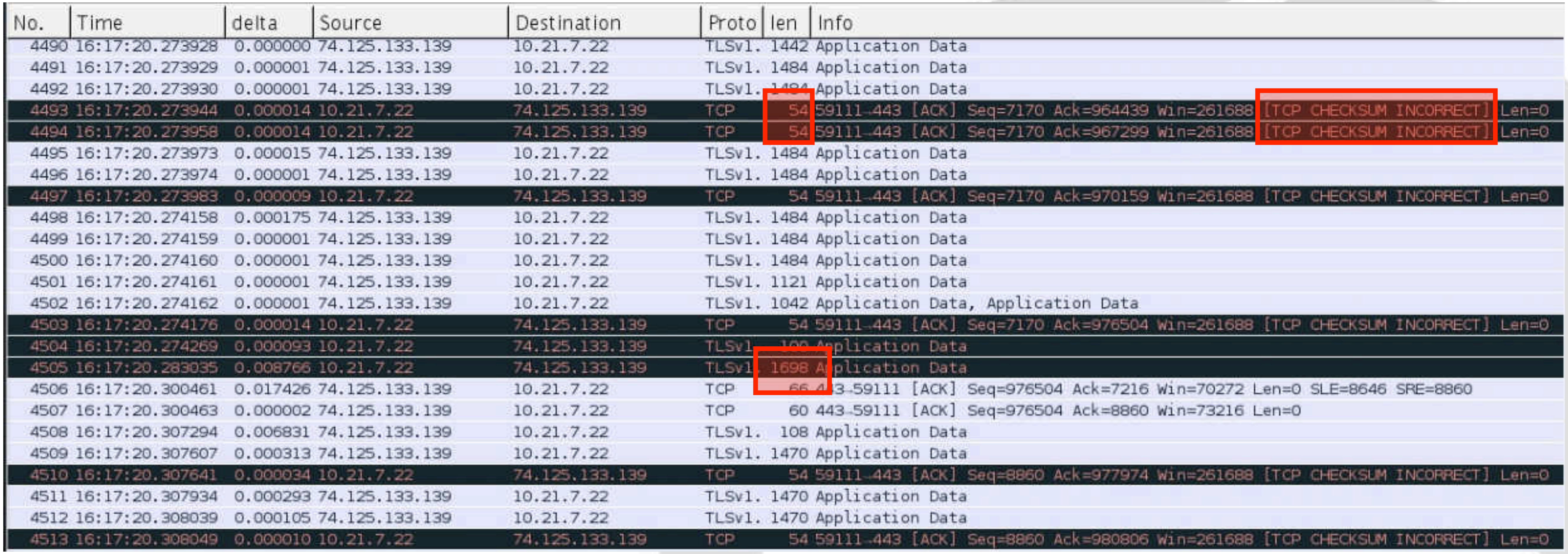
- Where to capture the packets
- How to capture the packets
  - On your own system
  - On a remote system
  - In between systems
  - In the virtual world
  - Capturing non-ethernet
  - Preventing packet discards
- Timestamping
- Capture filters
- Saving packets
- Q&A

Capture on your own system...

- Pros
  - Easy, only have to install wireshark (or other tool)
  - No need (to involve other teams) to configure things

- Cons
  - Influencing the system (which is under suspicion already)
  - You don't know which packets really are on the network or what they look like
    - Host firewalls, VPNs, etc.
    - Padding, Checksum offloading, Segmentation offloading

Artefacts of capturing on a host...



No.	Time	delta	Source	Destination	Proto	len	Info
4490	16:17:20.273928	0.000000	74.125.133.139	10.21.7.22	TLSv1.	1442	Application Data
4491	16:17:20.273929	0.000001	74.125.133.139	10.21.7.22	TLSv1.	1484	Application Data
4492	16:17:20.273930	0.000001	74.125.133.139	10.21.7.22	TLSv1.	1484	Application Data
4493	16:17:20.273944	0.000014	10.21.7.22	74.125.133.139	TCP	54	59111→443 [ACK] Seq=7170 Ack=964439 Win=261688 [TCP CHECKSUM INCORRECT] Len=0
4494	16:17:20.273958	0.000014	10.21.7.22	74.125.133.139	TCP	54	59111→443 [ACK] Seq=7170 Ack=967299 Win=261688 [TCP CHECKSUM INCORRECT] Len=0
4495	16:17:20.273973	0.000015	74.125.133.139	10.21.7.22	TLSv1.	1484	Application Data
4496	16:17:20.273974	0.000001	74.125.133.139	10.21.7.22	TLSv1.	1484	Application Data
4497	16:17:20.273983	0.000009	10.21.7.22	74.125.133.139	TCP	54	59111→443 [ACK] Seq=7170 Ack=970159 Win=261688 [TCP CHECKSUM INCORRECT] Len=0
4498	16:17:20.274158	0.000175	74.125.133.139	10.21.7.22	TLSv1.	1484	Application Data
4499	16:17:20.274159	0.000001	74.125.133.139	10.21.7.22	TLSv1.	1484	Application Data
4500	16:17:20.274160	0.000001	74.125.133.139	10.21.7.22	TLSv1.	1484	Application Data
4501	16:17:20.274161	0.000001	74.125.133.139	10.21.7.22	TLSv1.	1121	Application Data
4502	16:17:20.274162	0.000001	74.125.133.139	10.21.7.22	TLSv1.	1042	Application Data, Application Data
4503	16:17:20.274176	0.000014	10.21.7.22	74.125.133.139	TCP	54	59111→443 [ACK] Seq=7170 Ack=976504 Win=261688 [TCP CHECKSUM INCORRECT] Len=0
4504	16:17:20.274269	0.000093	10.21.7.22	74.125.133.139	TLSv1.	100	Application Data
4505	16:17:20.283035	0.008766	10.21.7.22	74.125.133.139	TLSv1.	1698	Application Data
4506	16:17:20.300461	0.017426	74.125.133.139	10.21.7.22	TCP	66	443→59111 [ACK] Seq=976504 Ack=7216 Win=70272 Len=0 SLE=8646 SRE=8860
4507	16:17:20.300463	0.000002	74.125.133.139	10.21.7.22	TCP	60	443→59111 [ACK] Seq=976504 Ack=8860 Win=73216 Len=0
4508	16:17:20.307294	0.006831	74.125.133.139	10.21.7.22	TLSv1.	108	Application Data
4509	16:17:20.307607	0.000313	74.125.133.139	10.21.7.22	TLSv1.	1470	Application Data
4510	16:17:20.307641	0.000034	10.21.7.22	74.125.133.139	TCP	54	59111→443 [ACK] Seq=8860 Ack=977974 Win=261688 [TCP CHECKSUM INCORRECT] Len=0
4511	16:17:20.307934	0.000293	74.125.133.139	10.21.7.22	TLSv1.	1470	Application Data
4512	16:17:20.308039	0.000105	74.125.133.139	10.21.7.22	TLSv1.	1470	Application Data
4513	16:17:20.308049	0.000010	10.21.7.22	74.125.133.139	TCP	54	59111→443 [ACK] Seq=8860 Ack=980806 Win=261688 [TCP CHECKSUM INCORRECT] Len=0

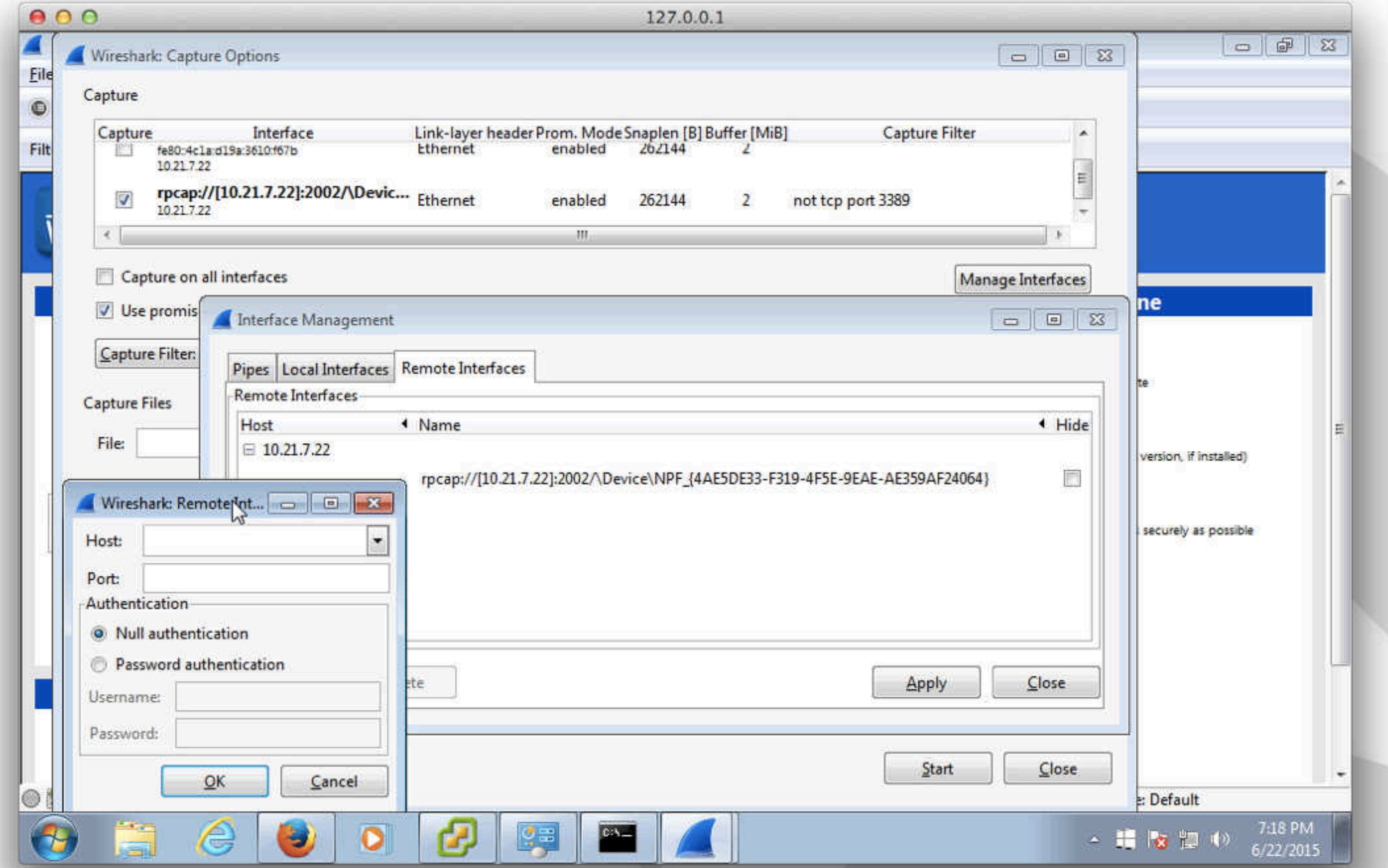
Remotely with rpcapd...

- rpcap daemon available in:
    - WinPcap >= 4.0
    - linux (see: http://www.pawelko.net/?s=rpcap)

- Remote interfaces can only be used in the Windows version of Wireshark

- Run rpcapd on either one of the endpoints
- ... or just on a system (in between) to capture packets from a span port or tap

Configure a remote interface...



127.0.0.1

Wireshark: Capture Options

Capture

Capture	Interface	Link-layer header	Prom. Mode	Snaplen [B]	Buffer [MiB]	Capture Filter
☐	fe80:4c1a:d19a:3610:f67b 10.21.7.22	Ethernet	enabled	262144	2	
☑	rpcap://[10.21.7.22]:2002/\Devic... 10.21.7.22	Ethernet	enabled	262144	2	not tcp port 3389

☐ Capture on all interfaces

☑ Use promis

Capture Filter:

Capture Files

File:

Interface Management

Remote Interfaces

Host	Name	Hide
⊟ 10.21.7.22		
	rpcap://[10.21.7.22]:2002/\Device\NPF_{4AE5DE33-F319-4F5E-9EAE-AE359AF24064}	☐

Wireshark: Remote Int...

Host:

Port:

Authentication

● Null authentication

○ Password authentication

Username:

Password:

Capture on an intermediate device...

- Device types to use?
  - Security device like Firewall, IDS or IPS
  - Loadbalancer (Application Delivery Controller)
- Pros
  - Easy, tools are already in place (usually based on tcpdump)
  - These systems are placed at key network locations
- Cons
  - Influencing the system (which might be under suspicion already)
  - You don't know which packets really are on the network or what they look like
  - Some devices have limited capacity

In between things...

- Make a copy of passing packets and forward to an analyzer

- Promiscuous mode
  - In normal operation, a NIC will only forward?
    - unicast traffic for it's own mac-address
    - broadcast traffic
    - multicast traffic (to subscribed groups)

  - In promiscuous mode?
    - forward all packets that are received on the interface

(real) HUBs...

- Remember the ol'days?
  - Half duplex traffic, forwarded to each port
  - 10 or 100 Mbit/s
- Pros
  - easy to use, just place hub in between system and network
- Cons
  - Reduce speed to max 100Mbit/s
  - Turns connection into half duplex
  - Only feasible if you wan't to capture from one (or a few) system(s)

(local) SPAN port...

- SPAN (Switch Port ANalyzer)
- Also known as mirror port, analysis port, etc.
- Copies traffic from one or more source interfaces (or vlans) to the analyzer port
- Beware of packet drops
  - 1Gbit/s TX and 1 Gbit/s RX combine to 2 Gbit/s output
  - Multiple source interfaces can generate to much traffic
  - Some switches also count discards on an analysis port

Remote SPAN...

- On the source switch
  - Copy traffic from one or more source ports or vlans to a dedicated capture vlan on switch X
- Forward traffic to another switch in the network
  - Can be L2 forwarded or tunneled over IP
- On the destination switch
  - copy traffic from the remote capture to the analyzer port

Packet Brokers...

- Can multiplex traffic in any way between input and output ports
- Usually any port can be defined as input or output (or both)
- Filtering can be applied while multiplexing
    - Filter 1 vlan out of a 10 Gbit/s trunk and forward to laptop
    - Just capture one IP address or Mac address out of multiple capture sources
- Some packet brokers offer timestamping and/or port information (usually in ethernet trailers)

Cubro EX2 portable packetbroker...

- 2x 10Gbit/s SFP+ port
- 4x 1Gbit/s copper port
- Optional optic splitter on the back (SF and MM)
- Any interface can be input and/or output
- Flexible filter rules to aggregate, multiplex, etc.

- See http://www.tap-shop.net
- Sharkfest discount : 25% off with coupon code SF2015

Capturing in the virtual world...

- On the VM
- On the outside interface of the host
- Use promiscuous mode on the vSwitch
- Use port mirroring on the (distibuted) vSwitch
- Tap into the VMkernel

- See the presentation on the subject by Jasper Bongertz!

Capturing non-ethernet...

- WiFi
  - Your own traffic
  - Management frames (requires AirPcap adapter on Windows)
- Bluetooth
  - linux (your own BT traffic)
  - Ubertooth via Kismet (other peoples BT traffic)
- USB
  - load usbmon kernel module on linux
  - use USBPcap on windows
- Other protocols possible through custom hardware and drivers (and possibly extcap)

Zen and the art of packet Capturing...

- Where to capture the packets
- How to capture the packets
  - On your own system
  - On a remote system
  - In between systems
  - In the virtual world
  - Capturing non-ethernet
  - Preventing packet discards
- Timestamping
- Capture filters
- Saving packets
- Q&A

Discards at the SPAN port or TAP...

- Aggregating RX and TX
- Aggregating multiple source interfaces
- Capturing on one or more vlans
  - Also note that spanning both directions creates duplicates!
- Visible in port statistics of the analyzer port (at least on cisco switches)
- Packet drops on RSPAN or rpcapd
- These discards are not listed in pcapng!

Discards on the capturing host...

- At the NIC layer
  - when the system is not quick enough to handle all the interupts of the NIC and buffering on the NIC is not sufficient
  - Use netstat to view discards/drops
  - Reduce by using a faster CPU, by running less applications or use a NAC instead of a NIC

- In libpcapWinPcap
  - when libpcap/WinPcap sees the packet, but is not able to store the packets to disk
    (see dropped packet statistic in pcapng and while capturing)
  - Reduce by increasing capture buffer size (-B)

Slicing packets...

- Reduce amount of data saved to disk
  - to reduce drops
  - to reduce space needed on disk (longer captures)
- Only useful if you don't need the full payload
- Calculate slicing amount:
  - 14 bytes ethernet header (+optional 4 bytes vlan tag)
  - 20 bytes IP header (almost always)
  - 20 bytes TCP header (often, but think of TS and SACK)
  - Usually a value between 54 and 100 is useful

Zen and the art of packet Capturing...

- Where to capture the packets
- How to capture the packets
- Timestamping
- Capture filters
- Saving packets
- Q&A

Timestamping packets...

- The more accurate the timestamp, the more accurate the analysis
- Timestamps are added by libpcap/WinPcap when they are handed the packet (not when received by the NIC)
- Timestamps are stored in UTC format in pcap/pcapng (Packets are shown in wallclock time in Wireshark)
- Capturing on multiple NICs can create funny results
- Need more accuracy? Use hardware that adds timestamps on receiving the packet
(NAC, packet proker, etc.)

Drifting timestamps on windows...

- WinPcap has multiple timestamp modes (HKLM\System\CurrentControlSet\Services\NPF\TimestampMode)
- Possible values are:

  0 (default): *Timestamps generated through KeQueryPerformanceCounter, less reliable on SMP/HyperThreading machines, precision = some microseconds*
  2: *Timestamps generated through KeQuerySystemTime, more reliable on SMP/HyperThreading machines, precision = scheduling quantum (10/15 ms)*
  3: *Timestamps generated through the i386 instruction RDTSC, less reliable on SMP/HyperThreading/SpeedStep machines, precision = some microseconds*
- see: http://seclists.org/wireshark/2010/Aug/311

Zen and the art of packet Capturing...

- Where to capture the packets
- How to capture the packets
- Timestamping
- Capture filters
- Saving packets
- Q&A

Capture filters...

- Why different syntax from display filters?
- Uses the BPF engine (microkernel inside the kernel)
- Limited instructions, no jumping backwards (to prevent kernel crashes by infinite loops)
- So only filter on data of which the offset in the packet can be calculated (no looping over the data)
  - TCP port number?
  - HTTP cookie?
  - HTTP request method?
- Some keywords advance the offset, be careful
  - vlan, pppoes, mpls

Filtering guidelines...

- Don't filter when you don't need to
- Slice instead of filter if you can
- When you do need to filter:
  - exclude things you don't want to see
  - instead of select the things that you want to see

ip host 1.2.3.4

```
(000) ldh      [12]
(001) jeq      #0x800           jt 2  jf 7
(002) ld       [26]
(003) jeq      #0x1020304       jt 6  jf 4
(004) ld       [30]
(005) jeq      #0x1020304       jt 6  jf 7
(006) ret      #65535
(007) ret      #0
```

```
0000  00 12 1e bb d1 3b f8 1e df d8 87 48 08 00 45 00   .....;.....H..E.
0010  02 bd 28 fb 40 00 40 06 fd 9f c0 a8 01 2b 01 01   ..(.@.@......+Bf
0020  01 01 c3 f6 00 50 f1 b8 8d 85 db 07 fd 9e 80 18   .g...P..........
0030  ff ff ce b2 00 00 01 01 08 0a 2e b9 c5 24 03 63   ..............$.c
0040  c5 41 47 45 54 20 2f 20 48 54 54 50 2f 31 2e 31   .AGET / HTTP/1.1
```

ip host 1.2.3.4 and tcp port 80

```
(000) ldh      [12]
(001) jeq      #0x800           jt 2   jf 16
(002) ld       [26]
(003) jeq      #0x1020304       jt 6   jf 4
(004) ld       [30]
(005) jeq      #0x1020304       jt 6   jf 16
(006) ldb      [23]
(007) jeq      #0x6             jt 8   jf 16
(008) ldh      [20]
(009) jset     #0x1fff          jt 16  jf 10
(010) ldxb     4*([14]&0xf)
(011) ldh      [x + 14]
(012) jeq      #0x50            jt 15  jf 13
(013) ldh      [x + 16]
(014) jeq      #0x50            jt 15  jf 16
(015) ret      #65535
(016) ret      #0
```

```
0000  00 12 1e bb d1 3b f8 1e df d8 87 48 08 00 45 00   ......;.....H..E.
0010  02 bd 28 fb 40 00 40 06 fd 9f c0 a8 01 2b 01 01   ..(.@.@......+Bf
0020  01 01 c3 f6 00 50 f1 b8 8d 85 db 07 fd 9e 80 18   .g...P..........
0030  ff ff ce b2 00 00 01 01 08 0a 2e b9 c5 24 03 63   .............$.c
0040  c5 41 47 45 54 20 2f 20 48 54 54 50 2f 31 2e 31   .AGET / HTTP/1.1
```

vlan 10 or vlan 20

```
(000) ldh      [12]
(001) jeq      #0x8100         jt 2  jf 5
(002) ldh      [14]
(003) and      #0xfff
(004) jeq      #0xa            jt 10jf 5
(005) ldh      [16]
(006) jeq      #0x8100         jt 7  jf 11
(007) ldh      [18]
(008) and      #0xfff
(009) jeq      #0x14           jt 10jf 11
(010) ret      #65535
(011) ret      #0
```

vlan and (ether[14:2] & 0xfff = 10 or ether[14:2] & 0xfff = 20)

```
(000) ldh      [12]
(001) jeq      #0x8100          jt 2  jf 7
(002) ldh      [14]
(003) and      #0xfff
(004) jeq      #0xa             jt 6  jf 5
(005) jeq      #0x14            jt 6  jf 7
(006) ret      #65535
(007) ret      #0
```

TCP length > 0?

```
ip and tcp and
( total IP datagram length -
  IP header length -
  TCP header length ) > 0
```

```
ip and tcp and
( ip[2:2] -
  ((ip[0]&0x0f)<<2) -
  ((tcp[12:1]&0xf0)>>2) ) > 0
```

```
0000   00 12 1e bb d1 3b f8 1e df d8 87 48 08 00 45 00    ......;.....H..E.
0010   02 bd 28 fb 40 00 40 06 fd 9f c0 a8 01 2b 01 01    ..(.@.@......+Bf
0020   01 01 c3 f6 00 50 f1 b8 8d 85 db 07 fd 9e 80 18    .g...P..........
0030   ff ff ce b2 00 00 01 01 08 0a 2e b9 c5 24 03 63    .............$.c
0040   c5 41 47 45 54 20 2f 20 48 54 54 50 2f 31 2e 31    .AGET / HTTP/1.1
```

TCP length > 0?

```
(000) ldh      [12]
(001) jeq      #0x800           jt 2        jf 22
(002) ldb      [23]
(003) jeq      #0x6             jt 4        jf 22
(004) ldh      [16]
(005) st       M[1]
(006) ldb      [14]
(007) and      #0xf
(008) lsh      #2
(009) tax
(010) ld       M[1]
(011) sub      x
(012) st       M[5]
(013) ldxb     4*([14]&0xf)
(014) ldb      [x + 26]
(015) and      #0xf0
(016) rsh      #2
(017) tax
(018) ld       M[5]
(019) sub      x
(020) jgt      #0x0             jt 21       jf 22
(021) ret      #65535
(022) ret      #0
```

How about SIP over IP-in-IP?

```
ip proto 4
and
(
  ip[((ip[0]&0x0f)<<2)+9]=17
  or
  ip[((ip[0]&0x0f)<<2)+9]=6
)
and
(
  ip[((ip[0]&0x0f)<<2)+((ip[((ip[0]&0x0f)<<2)]&0x0f)<<2)+0:2]=5060
  or
  ip[((ip[0]&0x0f)<<2)+((ip[((ip[0]&0x0f)<<2)]&0x0f)<<2)+2:2]=5060
)
```

Zen and the art of packet Capturing...

- Where to capture the packets
- How to capture the packets
- Timestamping
- Capture filters
- Saving packets
  - On a laptop
  - Using a capture server
  - Using packet recorders
- Q&A

Capture to a laptop...

- Pros
  - Cheap hardware
  - Easy to carry

- Cons
  - Limited disk speed (can capture about 200-300 Mbit/s network traffic when saving full frames)
  - Not always possible to capture the vlan tags
  - No capture of FCS (have not seen a laptop NIC yet that does not strip the FCS)
  - Can't do 10 Gbit/s

Capture to capture server...

- Pros
  - Can stripe disks to be able so sustain higher save-rate
  - Choose a capture card that fits your needs (10Gbit/s, keep FCS and/or vlan tags, hardware timestamping, etc)

- Cons
  - Not portable
  - No real cookbooks, so needs experimenting what hardware fits the needed specifications
  - Can become expensive (depending on the specifications)

Using packet recorders...

- Pros
  - Can capture wirespeed (usually)
  - Optimized hardware
  - Some analysis and/or filtering can be done on the server, then the result can be downloaded to wireshark
    (great for remote locations)

- Cons
  - Not portable
  - Can become expensive (depending on the specifications)

Poor mans packet recorder...

- dumpcap (does packet to disk only)
- -b filesize=xxx
  create files of xxx KB
- -b files=yyy
  create yyy files (and start deleting the first when creating file yyy+1
- Combined to create a 32GB ringbuffer
  `dumpcap -i 0 -w tmp.pcapng -b filesize=32768 -b files=1024`
- Have used this for months until a problem occurred

Zen and the art of packet Capturing...

- Where to capture the packets
- How to capture the packets
- Timestamping
- Capture filters
- Saving packets
- Q&A

SHARKFEST 2015

WIRESHARK DEVELOPER AND USER CONFERENCE

#SharkFest15

FIN/ACK - ACK - FIN/ACK - ACK

sake.blok@SYN-bit.nl