SharkFest ’18 US

Hands-on analysis of multi-point captures

Christian Landström

Airbus CyberSecurity

About me?

- Working for AIRBUS
- Reading trace files for the fun of it
- Sharkfest addict since Stanford

Why Multipoint ?

- A single measure point is not sufficient for certain network analysis tasks
- Typical scenarios for multipoint analysis
  - Assumed packet loss between client and server
  - Determining Latency
  - Investigating packet manipulation when passing certain network devices
  - Asymmetric routing
  - Link Aggregation
  - Active/Passive and Active/Active High Redundancy Solutions

Multipoint Analysis: Best Practices

- Extremely important: Document your traces as detailed as possible!
  - Especially when dealing with loads of trace files from multiple capture points
- Sync the time of your capture devices

Comparing trace files

- Comparing traces taken at multiple points at the same time is often necessary
- Major points of interest are:
  - Identify identical packets at each capture point
  - Isolate conversations and match them
  - Determine latency
  - Determine packet loss
- Can be quite time consuming unless done automatically

Identifying Packet Matches

- Find identical TCP/UDP conversations:
  - Determine client/server socket pairs
  - Create conversation filter, apply to all capture points
  - When using multiple files per location: batch job
- For other protocols, try
  - ARP: sender/target MAC and IP in the ARP header
  - ICMP: type, code, ping sequence, packet quote
  - DHCP, DNS: transaction ID
  - GenericIP: IP-ID, TTL

Isolating TCP Conversations

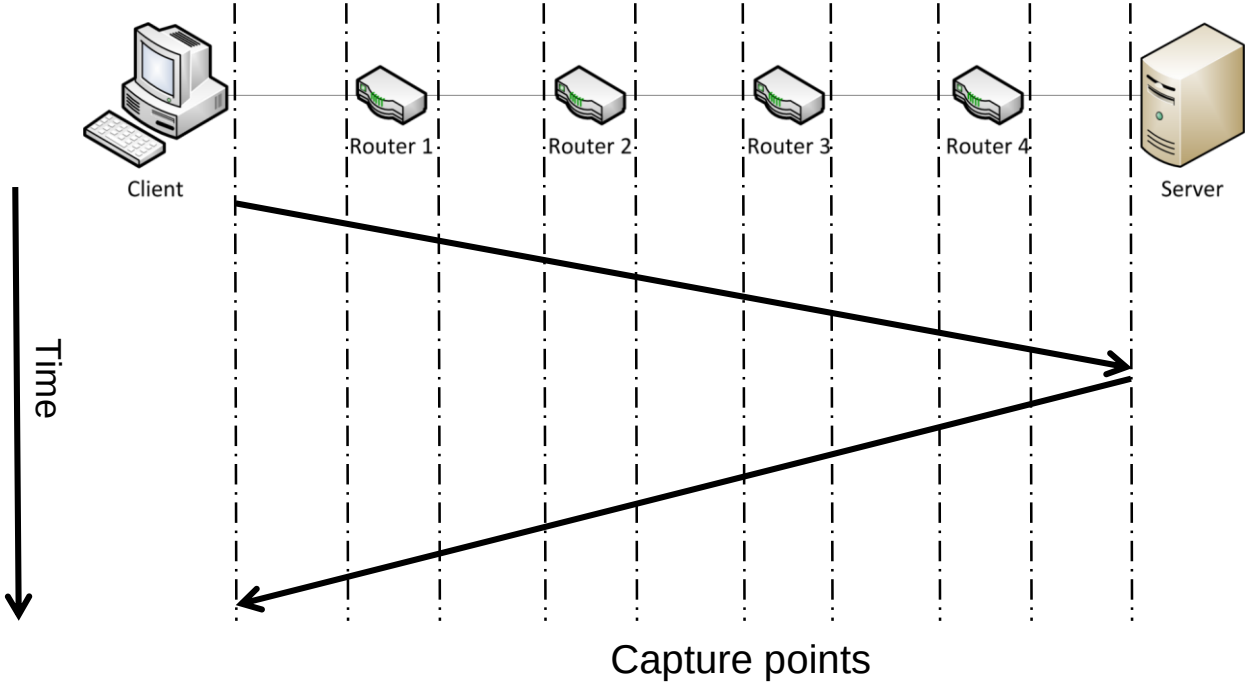
- Filter on the conversation, e.g.
  - `(ip.addr==10.0.0.1 and tcp.port==1025) and (ip.addr==10.0.0.2 and tcp.port==80)`
- Save into separate file using "Export specified packets" -> "Selected displayed packets"
- If possible: isolate initial SYN packet
  - `tcp.flags==2`
- Best Practice: deactivate relative TCP sequence numbers!

Short Demo

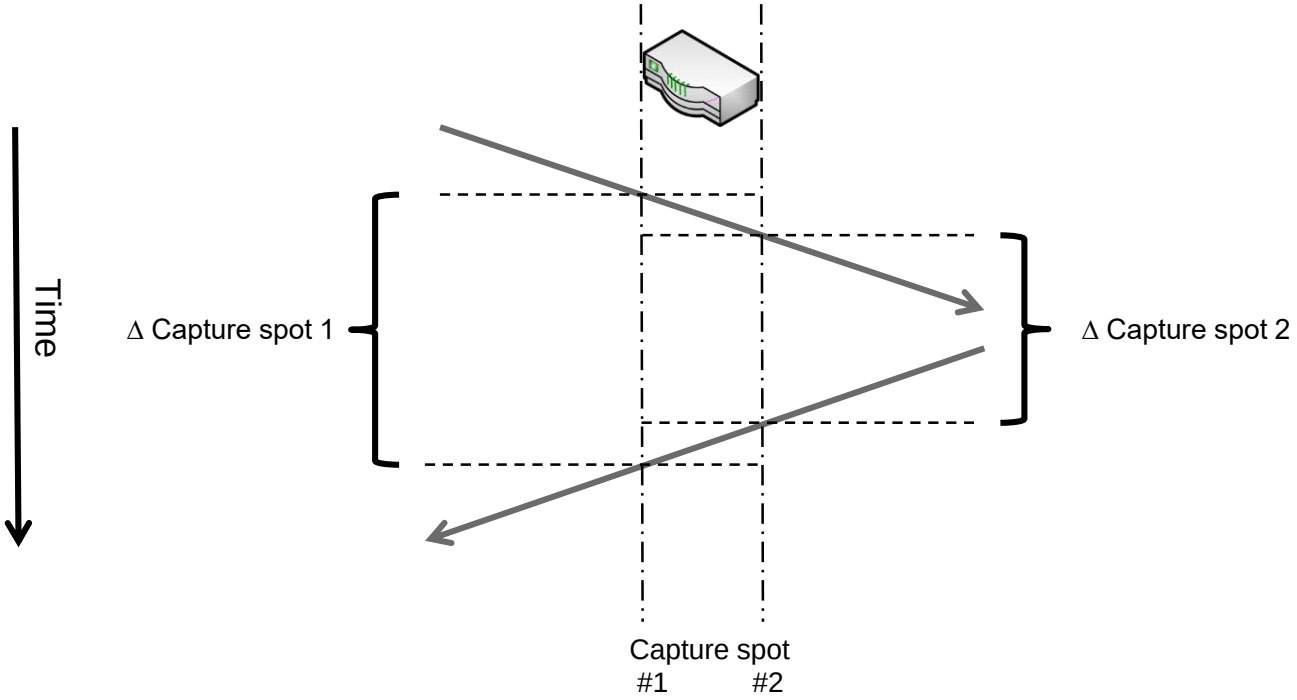
Compare Client and Server Side Trace

Determining Latency

Multipoint Captures: Latency



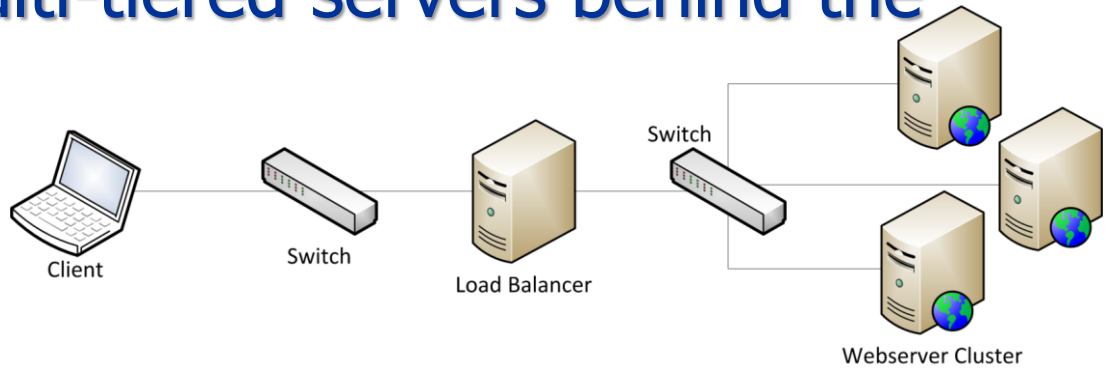
Determining Latency – Single Device



NAT, Proxy, Loadbalancer

Troublemakers: Loadbalancers

- Load balancers distribute connections to multiple identical servers
- Allows scaling the available capacity
- Example with multi-tiered servers behind the load balancer:

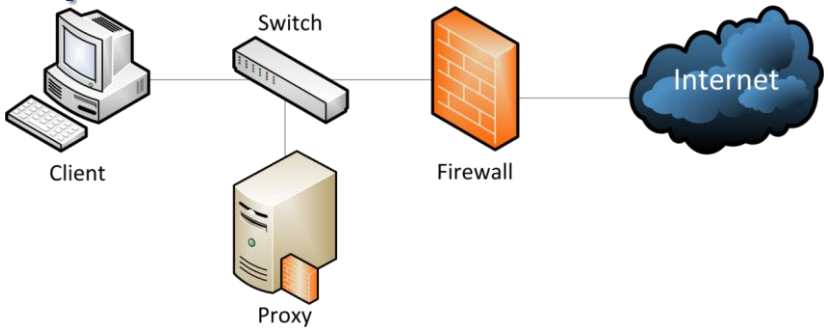


NAT Gateways

- NAT = Network Address Translation
  - Basically replaces network addresses found in packets back and forth
  - Usually relevant to layer 3, which means routers
- Typical NAT activity
  - Source NAT
  - Destination NAT

Proxy Servers

- Proxy servers separate different network and security zones
- Client requests are sent to the proxy
- The proxy fetches the requested content and delivers it to the client



Proxy Servers

Be aware of multiplexed sessions

Proxy Server: Forwarded-For

- Some proxies insert the address of the client into the request headers:

```
⊟ Hypertext Transfer Protocol
  ⊞ GET / HTTP/1.0\r\n
    Accept: text/html, application/xhtml+xml, */*\r\n
    Accept-Language: de-DE\r\n
    User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)\r\n
    Accept-Encoding: gzip, deflate\r\n
    Host: www.google.de\r\n
    [truncated] Cookie: PREF=ID=0de03f6f5ab5b026:U=acbc021047ffe581:FF=0:TM=1259593467:LM=1316903139
    Via: 1.1 localhost (squid/3.0.STABLE8)\r\n
    X-Forwarded-For: 192.168.124.100\r\n
    Cache-Control: max-age=259200\r\n
```

- Best Practice:disable "X-Forwarded-For" for security reasons
  - X-Forwarded-For will show something like „unknown"
  - Turn back on for temporary troubleshooting tasks

Time for some sharkin‘!

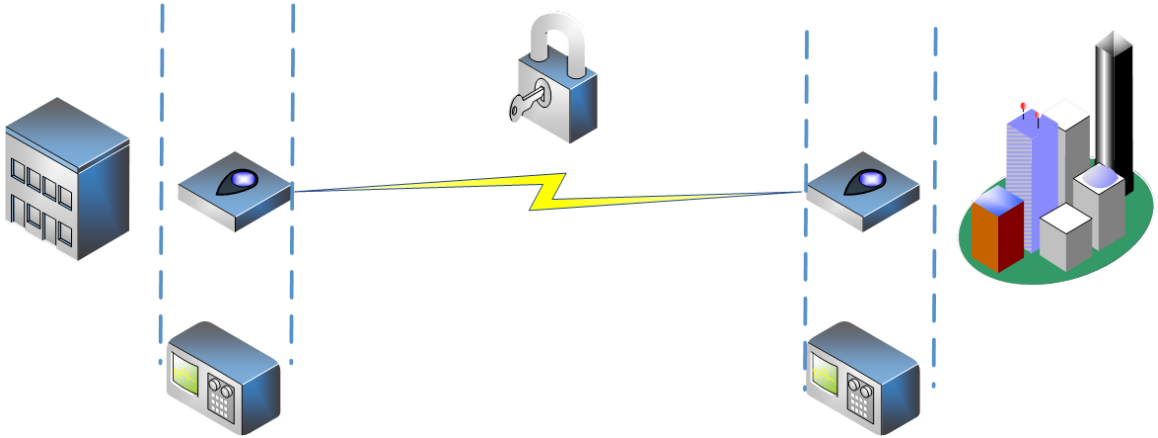
Demo #1

Client, WebServer, somewhat not sooo fast…

Demo #1


Destination: San Francisco
Origin: Copenhagen
Altitude: 37964 ft
Time to Destination: 2:41
Ground Speed: 526 mph
Air Temp: -74 F
Time at Origin: 8:48 PM
Time at Destination: 11:48 AM

Demo #2



Q&A

Mail: landi@packet-foo.com
Web: blog.packet-foo.com
Twitter: @packetfoo