Unveiling Network Errors: A Deep Dive into ICMP 'Destination Unreachable' Messages

Johannes Weber
@webernetz

Let me introduce myself

- Johannes Weber

- Network Security Consultant @SVA
- Firewalls ;)
- IPv6 & Security
- DNS & Security
- Blog: https://weberblog.net
- X: @webernetz
- LinkedIn

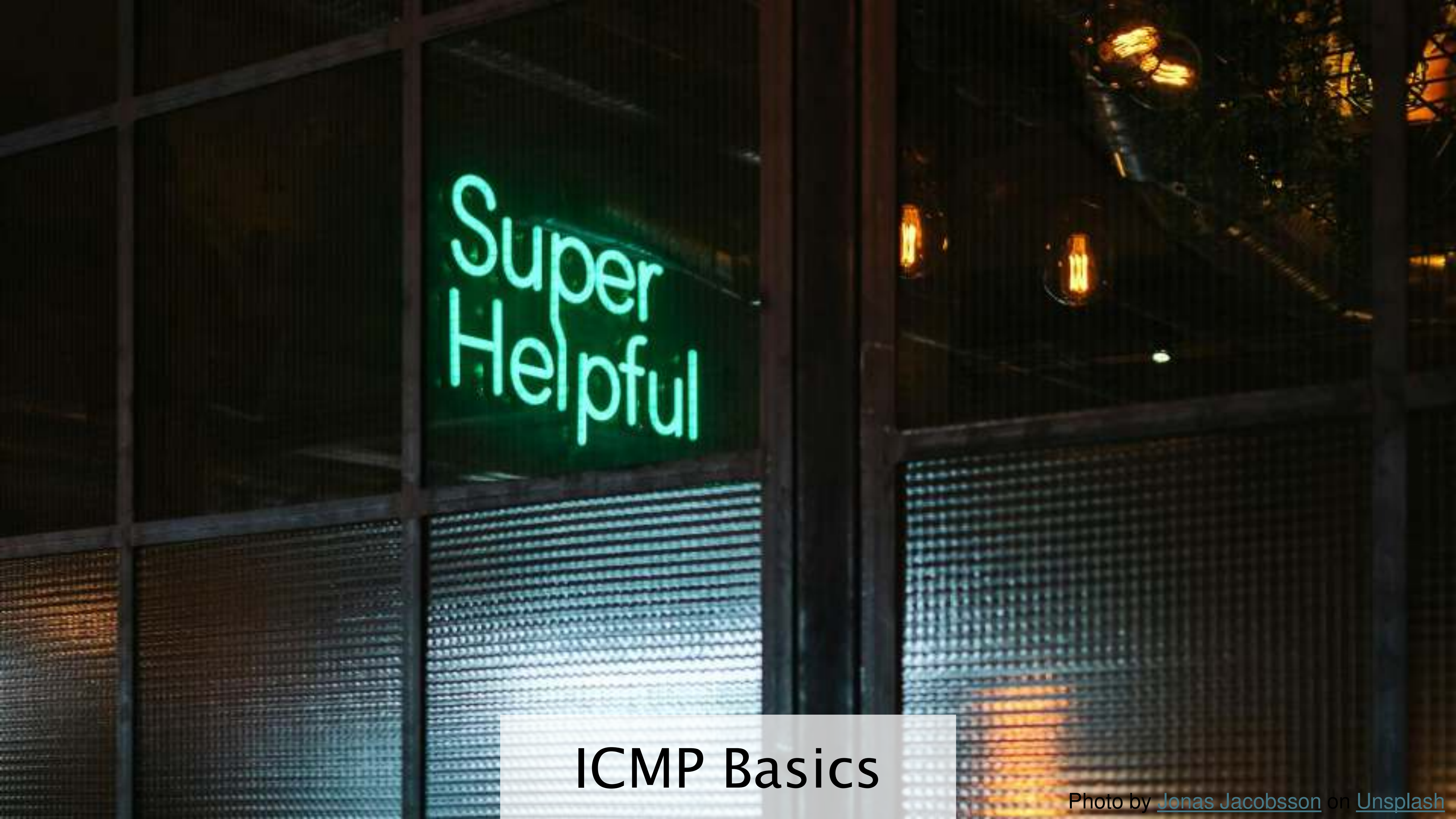
Motivation

- The unlifted network treasure: **ICMP**

- **Troubleshooting**: Where does this problem come from?

- **Housekeeping**: Do I see any (upcoming) problems?

- ICMP Basics
- Destination Unreachables
- within Wireshark
- How to capture?
- Case study: NTP Pool
- Challenges ;)



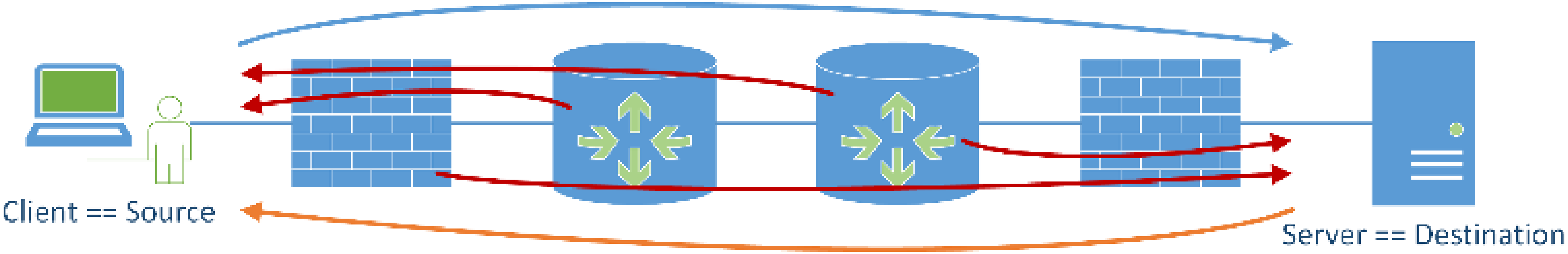
ICMP Basics

Photo by Jonas Jacobsson on Unsplash

ICMP Basics

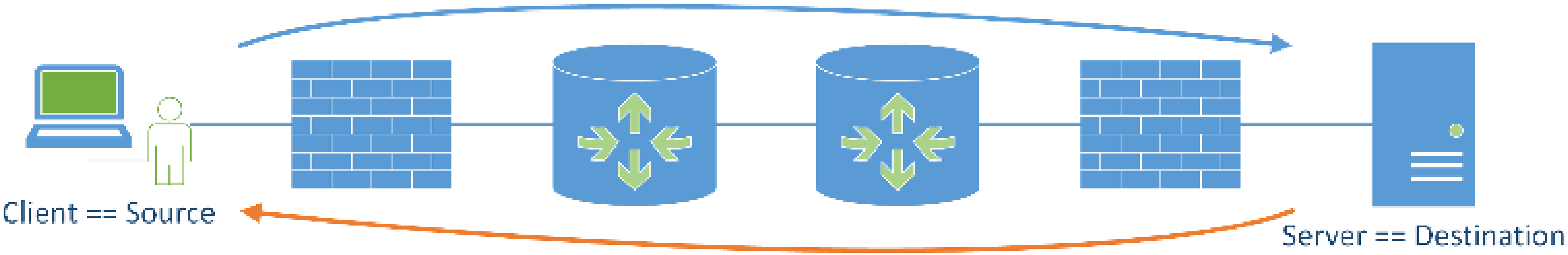
- Internet Control Message Protocol
- echo request & echo reply aka **Ping** (128|129, v4: 0|8)
- IPv6 [mandatory!]: RS & RA, NS & NA aka **NDP**
- error messages:
  - hop limit exceeded (routing loop) aka **traceroute**
  - packet too big aka **PMTU**
  - → destination unreachable ←
  - parameter problem
- MLD, redirect, …

ICMP Destination Unreachable



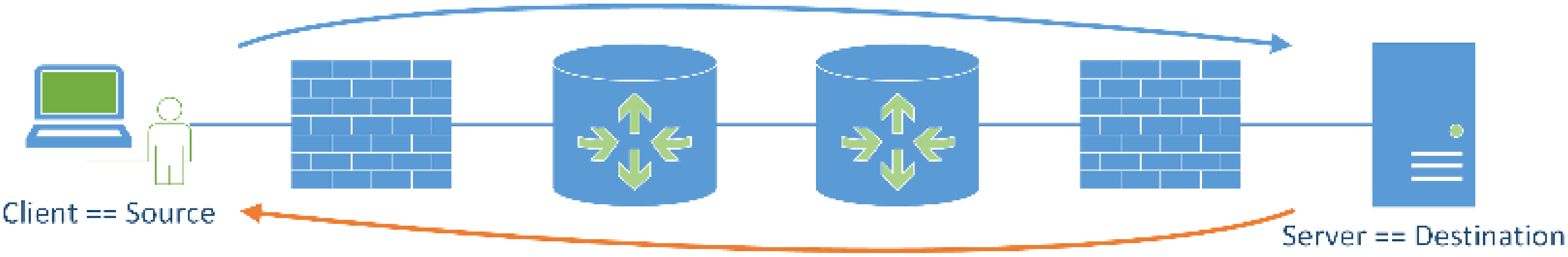
- RFC 4443: A Destination Unreachable message SHOULD be generated **by a router**, or by the IPv6 layer in the **originating node**, in response to a **packet that cannot be delivered** to its destination address for reasons other than congestion.
- IPv6: Type 1; IPv4: Type 3
- Initial packet client -> server
  Answering packet server -> client

ICMP Destination Unreachable



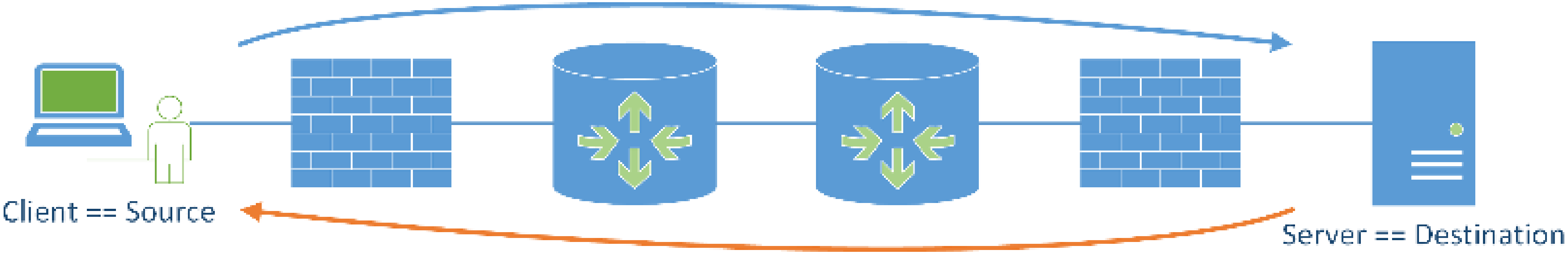
- Code 0 [v4: 0]: no route to destination
  - only in the default-free zone (without default routes)
  - c->s: typing error, false AAAA/A record, no BGP announcement
  - c<-s: spoofed source address, no BGP announcement

ICMP Destination Unreachable



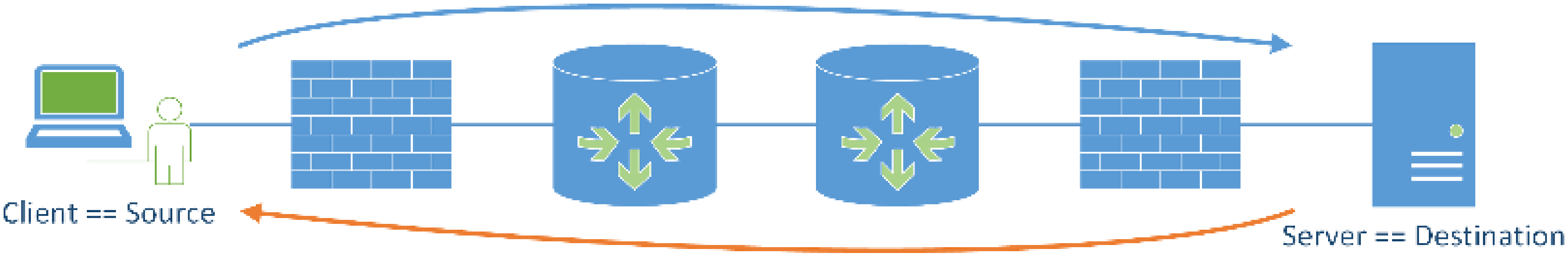
- Code 1 [v4: 13]: communication with destination administratively prohibited
  - aka firewall
  - that is: UDP -> DNS or NTP
  - (TCP has RST)
  - c->s: normal firewall behaviour
  - c<-s: missing stateful firewall?!?, answer longer than UDP session timeout (30s) e.g. DNS

ICMP Destination Unreachable



- Code 3 [v4: 1]: address unreachable
  - layer 2 address not resolvable, NS/ARP sent but INCOMPLETE
  - c->s: client link down, wrong address, false AAAA/A record
  - c<-s: link down (with short neighbour cache timeout), spoofed source address

ICMP Destination Unreachable



- Code 4 [v4: 3]: port unreachable
  - if transport protocol has no listener
  - that is: UDP -> DNS or NTP
  - (TCP has RST)
  - c->s: server isn't listening, port scan [1]
  - c<-s: client has closed source port too early
  - heavily related to defective NTP clients [2]

[1] https://weberblog.net/nmap-packet-capture/

[2] https://www.linkedin.com/pulse/how-use-ntp-pool-heiko-gerstung/

ICMP Destination Unreachable

- Code 5 & 6 are subsets of code 1: administratively prohibited
- Code 5: source address failed ingress/egress policy
  - filtering
  - c->s: normal firewall behaviour, **source** address filtering
  - c<-s: missing stateful firewall?!?, **source** address filtering

- Code 6: reject route to destination
  - route to destination is „reject“ rather than firewall policy
  - c->s: why putting server in a network that is rejected?
  - c<-s: how should clients be able to communicate at all?

ICMP Destination Unreachable

- Internet Control Message Protocol version 6 (ICMPv6) Parameters: https://www.iana.org/assignments/icmpv6-parameters/icmpv6-parameters.xml
- Internet Control Message Protocol (ICMP) Parameters: https://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml

Wireshark & tshark

Photo by Gerald Schömbs on Unsplash

The Ultimate PCAP

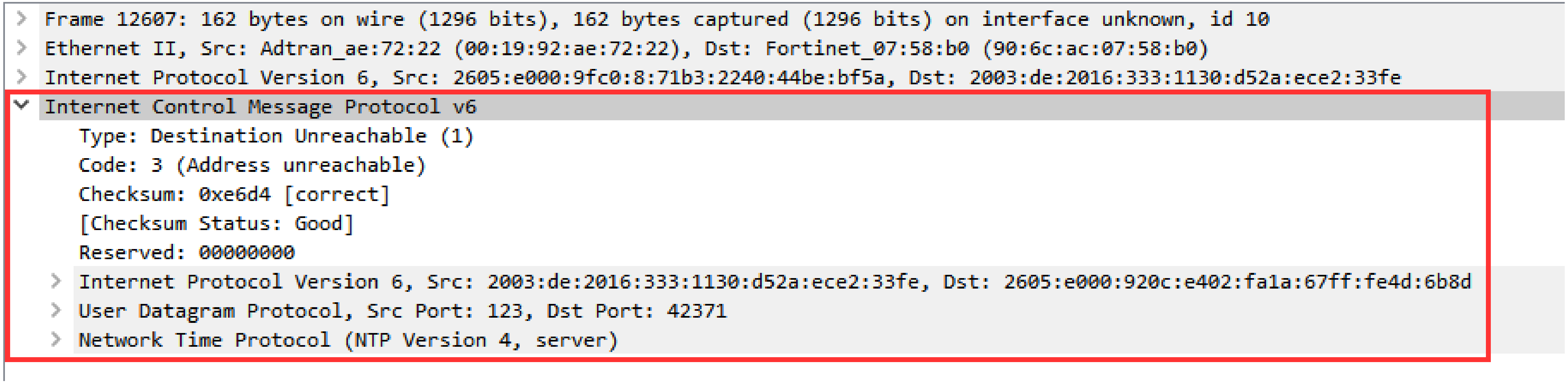
- The Ultimate PCAP: https://weberblog.net/the-ultimate-pcap/
- 80+ protocols in one single PCAP file
- 100+ options & stuff

ICMP Destination Unreachable in Wireshark

- `icmpv6.type eq 1 or icmp.type eq 3`
- [Live]

How to match sessions to errors?

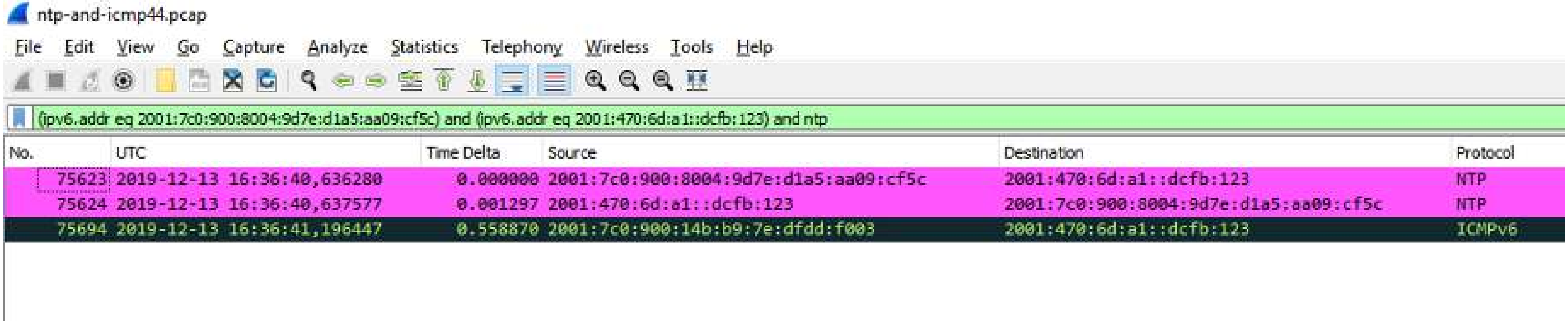
- ICMP error messages contain „as much of invoking packet as possible”
- [Live: `udp.stream == 247`]



```
> Frame 12607: 162 bytes on wire (1296 bits), 162 bytes captured (1296 bits) on interface unknown, id 10
> Ethernet II, Src: Adtran_ae:72:22 (00:19:92:ae:72:22), Dst: Fortinet_07:58:b0 (90:6c:ac:07:58:b0)
> Internet Protocol Version 6, Src: 2605:e000:9fc0:8:71b3:2240:44be:bf5a, Dst: 2003:de:2016:333:1130:d52a:ece2:33fe
v Internet Control Message Protocol v6
     Type: Destination Unreachable (1)
     Code: 3 (Address unreachable)
     Checksum: 0xe6d4 [correct]
     [Checksum Status: Good]
     Reserved: 00000000
   > Internet Protocol Version 6, Src: 2003:de:2016:333:1130:d52a:ece2:33fe, Dst: 2605:e000:920c:e402:fa1a:67ff:fe4d:6b8d
   > User Datagram Protocol, Src Port: 123, Dst Port: 42371
   > Network Time Protocol (NTP Version 4, server)
```

How to match sessions to errors?

- Wireshark‘s display filter automatically shows corresponding errors!



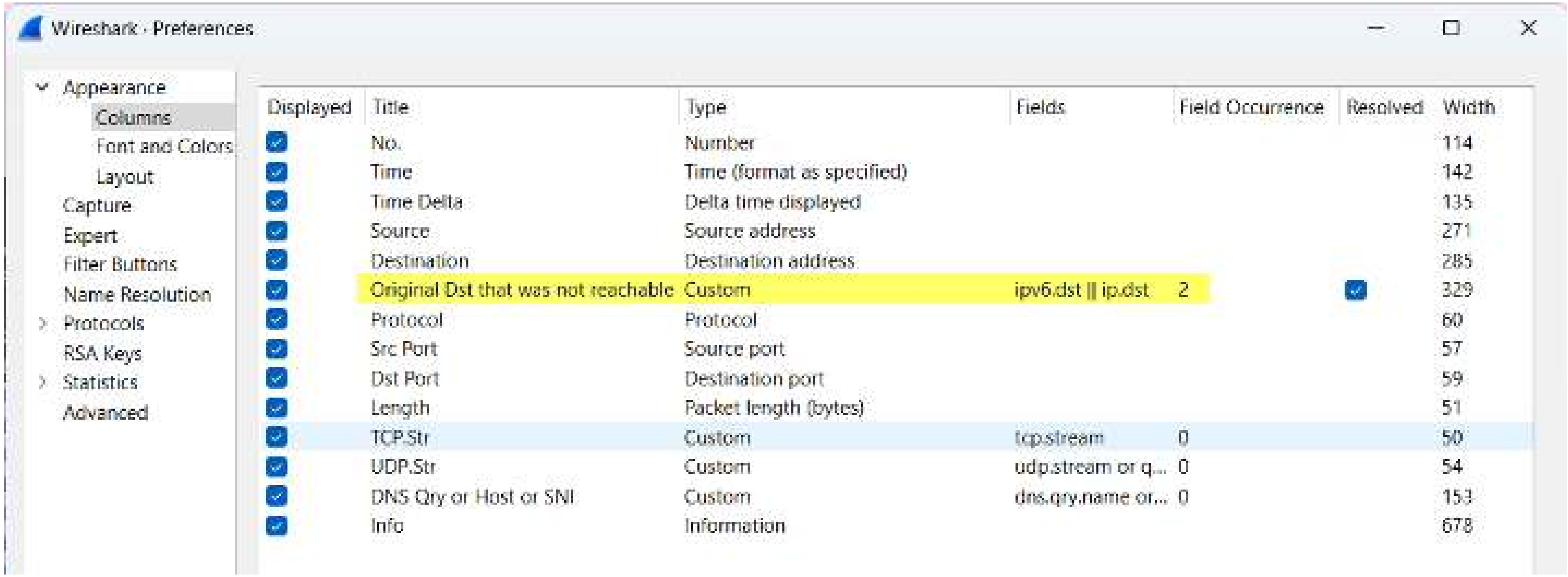
```
(ipv6.addr eq 2001:7c0:900:8004:9d7e:d1a5:aa09:cf5c) and (ipv6.addr eq 2001:470:6d:a1::dcfb:123) and ntp
```

No.	UTC	Time Delta	Source	Destination	Protocol
75623	2019-12-13 16:36:40,636280	0.000000	2001:7c0:900:8004:9d7e:d1a5:aa09:cf5c	2001:470:6d:a1::dcfb:123	NTP
75624	2019-12-13 16:36:40,637577	0.001297	2001:470:6d:a1::dcfb:123	2001:7c0:900:8004:9d7e:d1a5:aa09:cf5c	NTP
75694	2019-12-13 16:36:41,196447	0.558870	2001:7c0:900:14b:b9:7e:dfdd:f003	2001:470:6d:a1::dcfb:123	ICMPv6

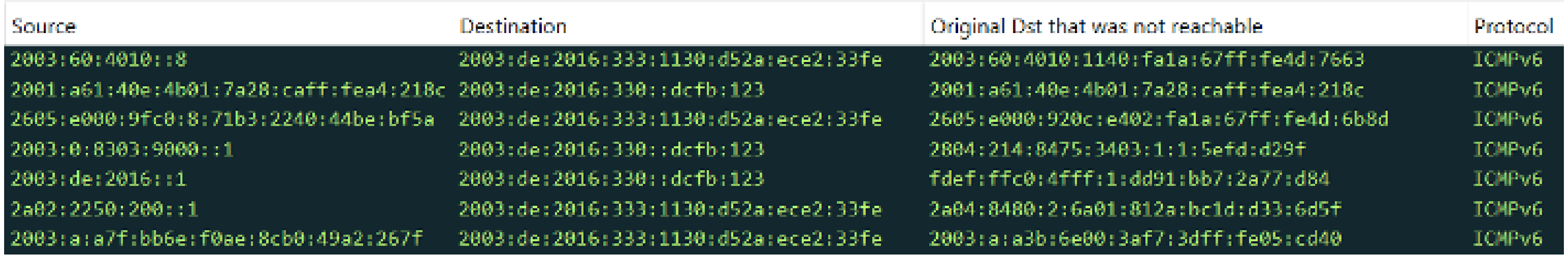
- Filtering for client/server IPv6 addresses & NTP
- Corresponding ICMPv6 from *different* source is shown as well

How to match sessions to errors?

- Custom Column: Original Dst that was not reachable



Displayed	Title	Type	Fields	Field Occurrence	Resolved	Width
☑	No.	Number				114
☑	Time	Time (format as specified)				142
☑	Time Delta	Delta time displayed				135
☑	Source	Source address				271
☑	Destination	Destination address				285
☑	Original Dst that was not reachable	Custom	ipv6.dst || ip.dst	2	☑	329
☑	Protocol	Protocol				60
☑	Src Port	Source port				57
☑	Dst Port	Destination port				59
☑	Length	Packet length (bytes)				51
☑	TCP.Str	Custom	tcp.stream	0		50
☑	UDP.Str	Custom	udp.stream or q...	0		54
☑	DNS Qry or Host or SNI	Custom	dns.qry.name or...	0		153
☑	Info	Information				678



Source	Destination	Original Dst that was not reachable	Protocol
2003:60:4010::8	2003:de:2016:333:1130:d52a:ece2:33fe	2003:60:4010:1140:fa1a:67ff:fe4d:7663	ICMPv6
2001:a61:40e:4b01:7a28:caff:fea4:218c	2003:de:2016:330::dcfb:123	2001:a61:40e:4b01:7a28:caff:fea4:218c	ICMPv6
2605:e000:9fc0:8:71b3:2240:44be:bf5a	2003:de:2016:333:1130:d52a:ece2:33fe	2605:e000:920c:e402:fa1a:67ff:fe4d:6b8d	ICMPv6
2003:0:8303:9000::1	2003:de:2016:330::dcfb:123	2804:214:8475:3403:1:1:5efd:d29f	ICMPv6
2003:de:2016::1	2003:de:2016:330::dcfb:123	fdef:ffc0:4fff:1:dd91:bb7:2a77:d84	ICMPv6
2a02:2250:200::1	2003:de:2016:333:1130:d52a:ece2:33fe	2a04:8480:2:6a01:812a:bc1d:d33:6d5f	ICMPv6
2003:a:a7f:bb6e:f0ae:8cb0:49a2:267f	2003:de:2016:333:1130:d52a:ece2:33fe	2003:a:a3b:6e00:3af7:3dff:fe05:cd40	ICMPv6

tshark

- Parse through (big) pcaps
- Display/write only relevant fields
- Additional use standard Unix tools:
  `sort | uniq | wc -l`

```
tshark -r <input-file> -Y <Display-Filter>
-T fields -e <field-to-extract> | sort | uniq -c
```

tshark

```
tshark -r ntp.pcapng -Y "icmpv6" -T fields -e
icmpv6.type -e icmpv6.code | sort | uniq -c

  385 1 0 no route to destination
  291 1 1 communication administratively prohibited
 9839 1 3 address unreachable
  367 1 4 port unreachable
    3 1 5 source address failed ingress/egress policy
    1 1 6 reject route to destination
   37 3 0 hop limit exceeded in transit
```

How to capture?

Photo by Jimmy Chang on Unsplash

How to capture?

- Normal filters:
  `(host 2001:db8::22 and host 2001:db8::53)`

- Always include „`or icmp or icmp6`“

```
tcpdump -i eno1 '(host 2001:db8::22 and host
2001:db8::53) or icmp6`
```

- Note: Unlike Wireshark‘s *display filter*, a **capture filter** (eBPF) will *not* match the inner ICMP IP header.

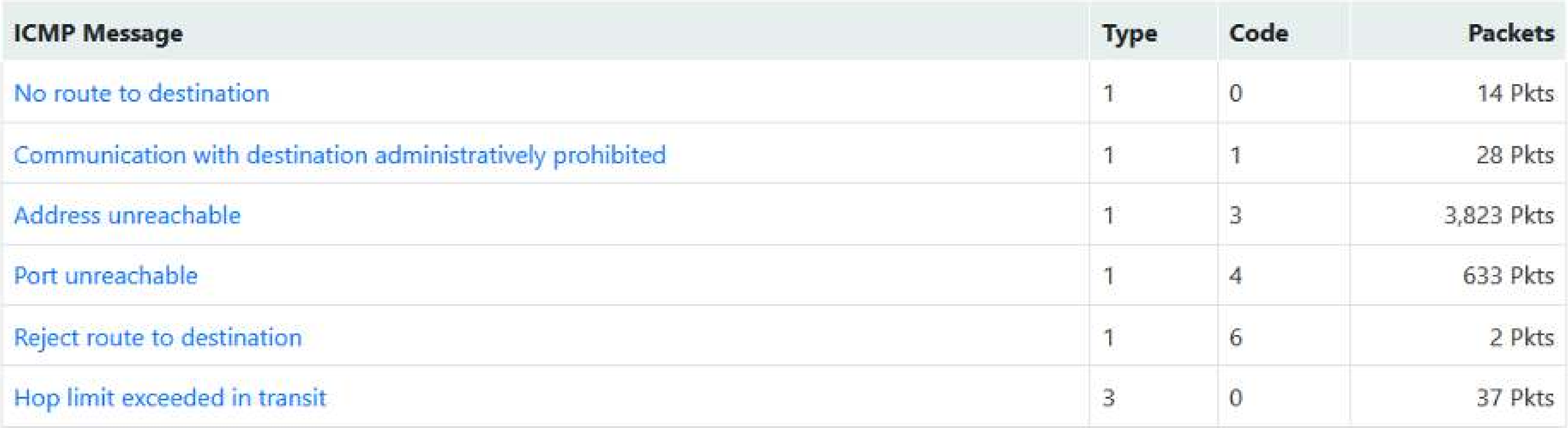
How to capture?

- Why do I have to capture by myself?
- → Neither routers nor Next-Gen Firewalls provide ICMP analysis ☹

- Do firewalls deliver ICMP packets correctly? Stateful?
- Are packet captures on firewalls capturing ICMP?

How to capture?

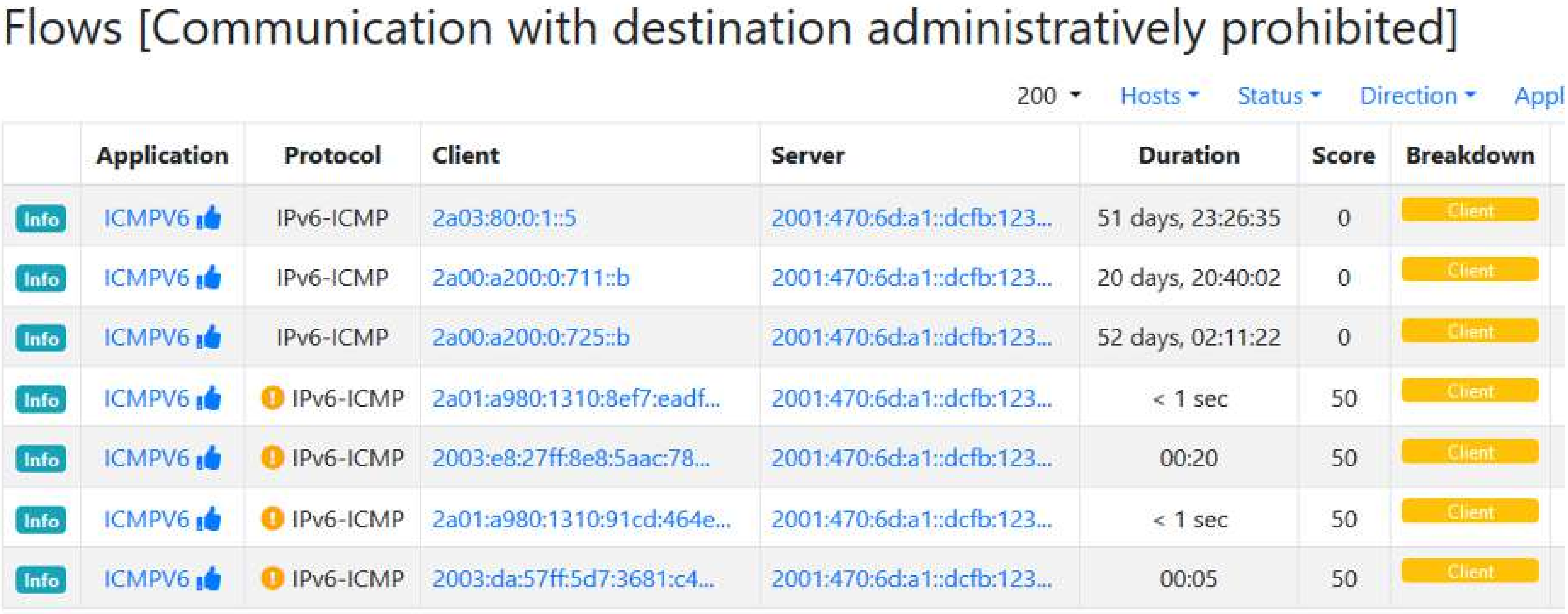
- At least **ntopng** gives a little glance



ICMP Message	Type	Code	Packets
No route to destination	1	0	14 Pkts
Communication with destination administratively prohibited	1	1	28 Pkts
Address unreachable	1	3	3,823 Pkts
Port unreachable	1	4	633 Pkts
Reject route to destination	1	6	2 Pkts
Hop limit exceeded in transit	3	0	37 Pkts

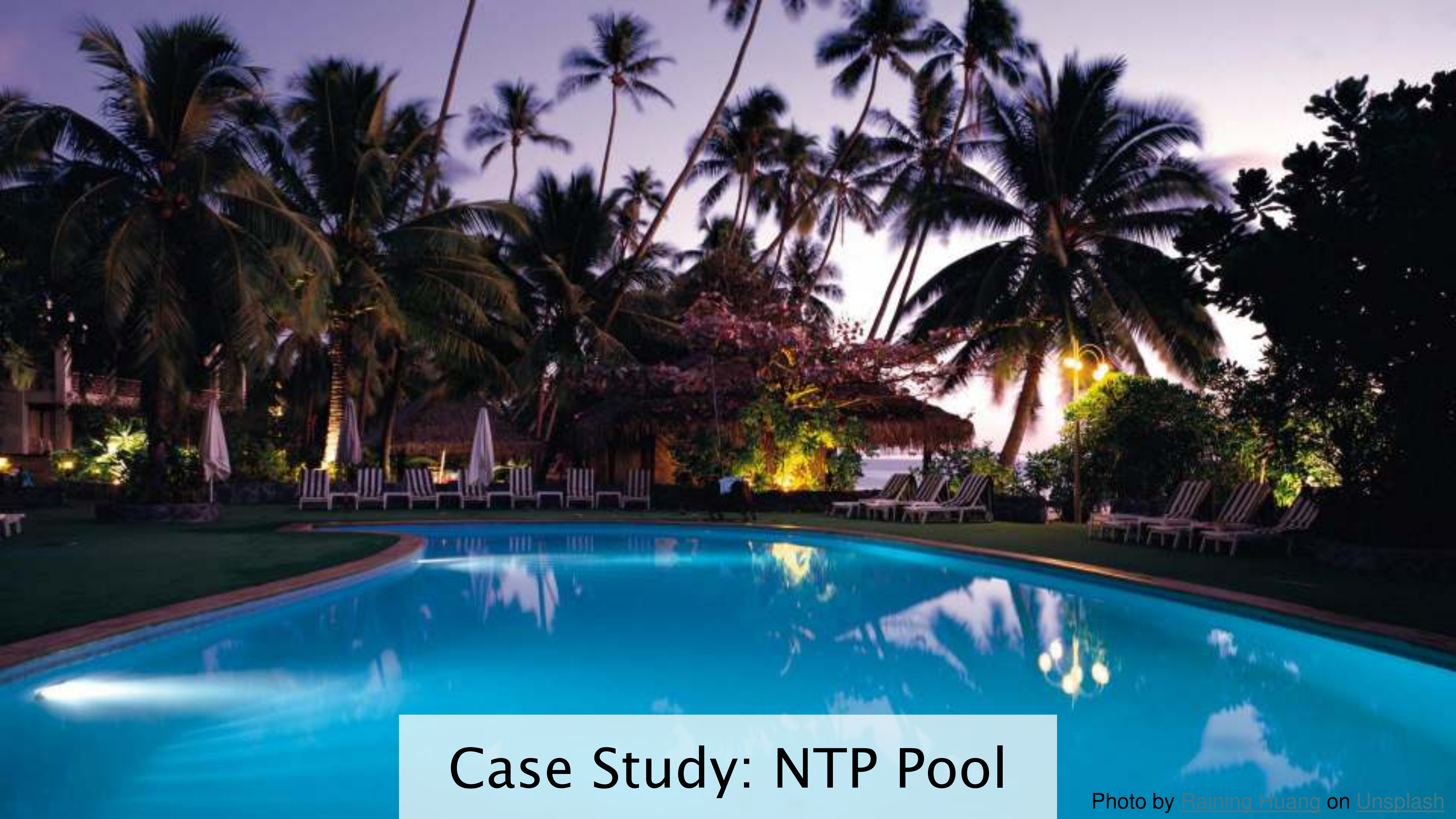
How to capture?

- At least **ntopng** gives a little glance



Flows [Communication with destination administratively prohibited]

	Application	Protocol	Client	Server	Duration	Score	Breakdown
Info	ICMPV6 👍	IPv6-ICMP	2a03:80:0:1::5	2001:470:6d:a1::dcfb:123...	51 days, 23:26:35	0	Client
Info	ICMPV6 👍	IPv6-ICMP	2a00:a200:0:711::b	2001:470:6d:a1::dcfb:123...	20 days, 20:40:02	0	Client
Info	ICMPV6 👍	IPv6-ICMP	2a00:a200:0:725::b	2001:470:6d:a1::dcfb:123...	52 days, 02:11:22	0	Client
Info	ICMPV6 👍	❗ IPv6-ICMP	2a01:a980:1310:8ef7:eadf...	2001:470:6d:a1::dcfb:123...	< 1 sec	50	Client
Info	ICMPV6 👍	❗ IPv6-ICMP	2003:e8:27ff:8e8:5aac:78...	2001:470:6d:a1::dcfb:123...	00:20	50	Client
Info	ICMPV6 👍	❗ IPv6-ICMP	2a01:a980:1310:91cd:464e...	2001:470:6d:a1::dcfb:123...	< 1 sec	50	Client
Info	ICMPV6 👍	❗ IPv6-ICMP	2003:da:57ff:5d7:3681:c4...	2001:470:6d:a1::dcfb:123...	00:05	50	Client



Case Study: NTP Pool

Photo by Raining Huang on Unsplash

Case Study: NTP Pool

- Dynamic collection of volunteer NTP servers
- Default „time server“ for many Linux distributions and infrastructure devices (routers, IoT, …)
- Meinberg M200 appliance w/ DCF77
- `ntp3.weberlab.de`

```
PZF: NORMAL OPERATION  Fri, 03.05.2019 ξ
NTP: Offs. PZF:-12us   UTC  15:24:18
```

Case Study: NTP Pool

- Availabe in zones: @, europe, de
- Pool: roughly 2x more IPv4 servers than IPv6 servers
- `pool.ntp.org`
  `0.pool.ntp.org`
  `1.pool.ntp.org`
  `2.pool.ntp.org` <- only this one has AAAA records
  `3.pool.ntp.org`
- Same for europe & de
  → 1-to-1 comparison of v4/v6 traffic is not fair
- Lab period: 66 days

Case Study: NTP Pool

	IPv6	Legacy IP
Number of NTP packets	7 M	85 M
NTP packets per second	1.24	14.9
Unique source addresses of NTP packets	0.65 M (9.3 %)	3.4 M (4 %)
NTP sources that caused ICMP errors	30 K	96 K
Percentage of failed NTP sources	**4.55 %**	**2.81 %**

- Each IPv6 source address == unique NTP client
- NOT true for legacy IP

Case Study: NTP Pool

Distribution of ICMP Errors	IPv6	Legacy IP
dest unreach, no route to dest [net unreach]	1.63 %	0.15 %
dest unreach, administrat prohibited [diverse]	0.31 %	0.14 %
dest unreach, address unreach [host unreach]	**87.51 %**	1.42 %
dest unreach, port unreach	9.82 %	**98.28 %**
time exceeded	0.73 %	0.02 %

- IPv6: 88 % address unreachable -> ND problem
- IPv4: 98 % port unreachable -> defective NTP clients

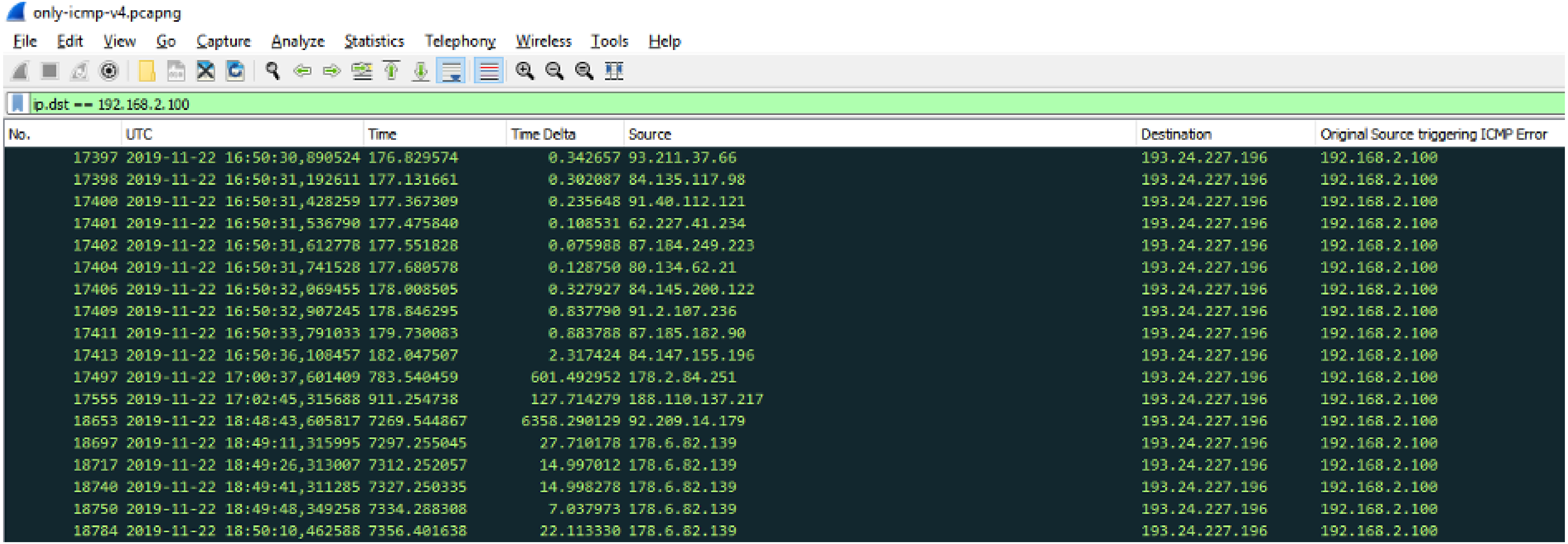
Case Study: NTP Pool

- Top v4 NTP clients causing ICMP errors

```
weberjoh@mirror:~$ tshark -r only-icmp-v4.pcapng -T fields -e
ip.dst | sort | uniq -c | sort -n -r | head -n 30
  13798 193.24.227.196,192.168.2.100
   5813 193.24.227.196,192.168.2.101
   3161 193.24.227.196,192.168.2.102
   1557 193.24.227.196,192.168.2.103
   1430 193.24.227.196,138.118.136.128
   1422 193.24.227.196,138.118.136.86
   1200 193.24.227.196,188.174.53.178
   1172 193.24.227.196,192.168.2.104
```

Case Study: NTP Pool

- Top v4 NTP clients causing ICMP errors

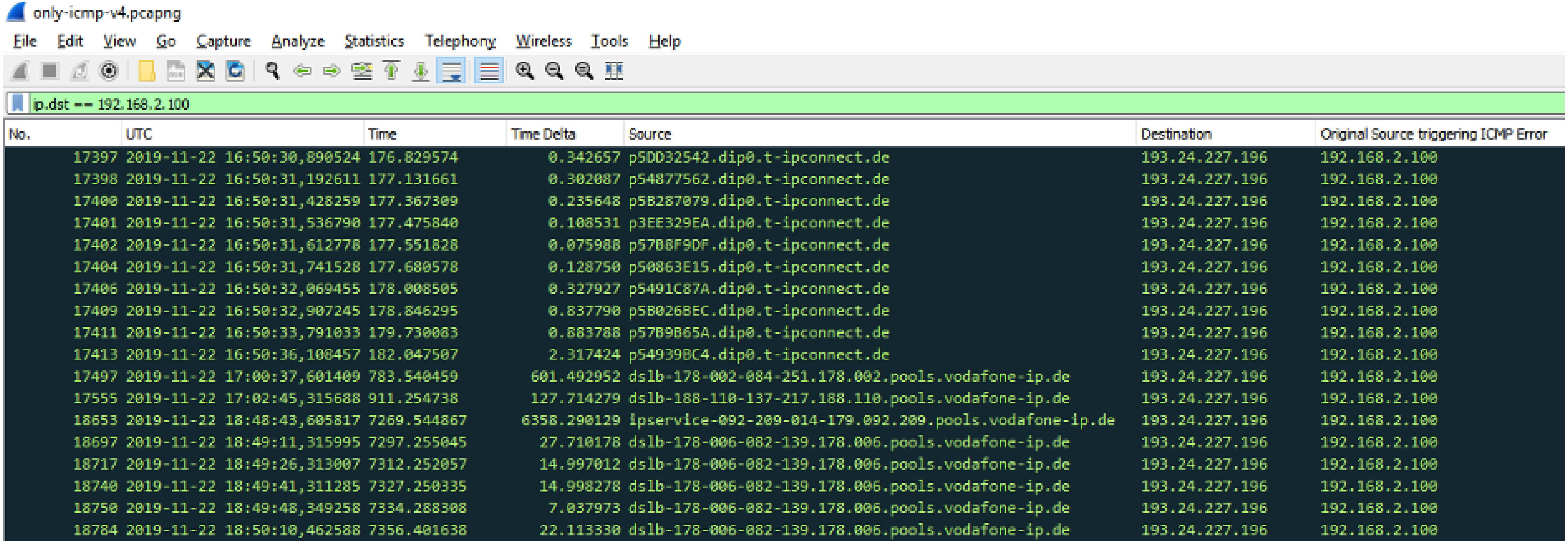

only-icmp-v4.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.dst == 192.168.2.100

No.	UTC	Time	Time Delta	Source	Destination	Original Source triggering ICMP Error
17397	2019-11-22 16:50:30,890524	176.829574	0.342657	93.211.37.66	193.24.227.196	192.168.2.100
17398	2019-11-22 16:50:31,192611	177.131661	0.302087	84.135.117.98	193.24.227.196	192.168.2.100
17400	2019-11-22 16:50:31,428259	177.367309	0.235648	91.40.112.121	193.24.227.196	192.168.2.100
17401	2019-11-22 16:50:31,536790	177.475840	0.108531	62.227.41.234	193.24.227.196	192.168.2.100
17402	2019-11-22 16:50:31,612778	177.551828	0.075988	87.184.249.223	193.24.227.196	192.168.2.100
17404	2019-11-22 16:50:31,741528	177.680578	0.128750	80.134.62.21	193.24.227.196	192.168.2.100
17406	2019-11-22 16:50:32,069455	178.008505	0.327927	84.145.200.122	193.24.227.196	192.168.2.100
17409	2019-11-22 16:50:32,907245	178.846295	0.837790	91.2.107.236	193.24.227.196	192.168.2.100
17411	2019-11-22 16:50:33,791033	179.730083	0.883788	87.185.182.90	193.24.227.196	192.168.2.100
17413	2019-11-22 16:50:36,108457	182.047507	2.317424	84.147.155.196	193.24.227.196	192.168.2.100
17497	2019-11-22 17:00:37,601409	783.540459	601.492952	178.2.84.251	193.24.227.196	192.168.2.100
17555	2019-11-22 17:02:45,315688	911.254738	127.714279	188.110.137.217	193.24.227.196	192.168.2.100
18653	2019-11-22 18:48:43,605817	7269.544867	6358.290129	92.209.14.179	193.24.227.196	192.168.2.100
18697	2019-11-22 18:49:11,315995	7297.255045	27.710178	178.6.82.139	193.24.227.196	192.168.2.100
18717	2019-11-22 18:49:26,313007	7312.252057	14.997012	178.6.82.139	193.24.227.196	192.168.2.100
18740	2019-11-22 18:49:41,311285	7327.250335	14.998278	178.6.82.139	193.24.227.196	192.168.2.100
18750	2019-11-22 18:49:48,349258	7334.288308	7.037973	178.6.82.139	193.24.227.196	192.168.2.100
18784	2019-11-22 18:50:10,462588	7356.401638	22.113330	178.6.82.139	193.24.227.196	192.168.2.100

Case Study: NTP Pool

- Top v4 NTP clients causing ICMP errors


only-icmp-v4.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.dst == 192.168.2.100

No.	UTC	Time	Time Delta	Source	Destination	Original Source triggering ICMP Error
17397	2019-11-22 16:50:30,890524	176.829574	0.342657	p5DD32542.dip0.t-ipconnect.de	193.24.227.196	192.168.2.100
17398	2019-11-22 16:50:31,192611	177.131661	0.302087	p54877562.dip0.t-ipconnect.de	193.24.227.196	192.168.2.100
17400	2019-11-22 16:50:31,428259	177.367309	0.235648	p5B287079.dip0.t-ipconnect.de	193.24.227.196	192.168.2.100
17401	2019-11-22 16:50:31,536790	177.475840	0.108531	p3EE329EA.dip0.t-ipconnect.de	193.24.227.196	192.168.2.100
17402	2019-11-22 16:50:31,612778	177.551828	0.075988	p57B8F9DF.dip0.t-ipconnect.de	193.24.227.196	192.168.2.100
17404	2019-11-22 16:50:31,741528	177.680578	0.128750	p50863E15.dip0.t-ipconnect.de	193.24.227.196	192.168.2.100
17406	2019-11-22 16:50:32,069455	178.008505	0.327927	p5491C87A.dip0.t-ipconnect.de	193.24.227.196	192.168.2.100
17409	2019-11-22 16:50:32,907245	178.846295	0.837790	p5B0268EC.dip0.t-ipconnect.de	193.24.227.196	192.168.2.100
17411	2019-11-22 16:50:33,791033	179.730083	0.883788	p57B9B65A.dip0.t-ipconnect.de	193.24.227.196	192.168.2.100
17413	2019-11-22 16:50:36,108457	182.047507	2.317424	p549398C4.dip0.t-ipconnect.de	193.24.227.196	192.168.2.100
17497	2019-11-22 17:00:37,601409	783.540459	601.492952	dslb-178-002-084-251.178.002.pools.vodafone-ip.de	193.24.227.196	192.168.2.100
17555	2019-11-22 17:02:45,315688	911.254738	127.714279	dslb-188-110-137-217.188.110.pools.vodafone-ip.de	193.24.227.196	192.168.2.100
18653	2019-11-22 18:48:43,605817	7269.544867	6358.290129	ipservice-092-209-014-179.092.209.pools.vodafone-ip.de	193.24.227.196	192.168.2.100
18697	2019-11-22 18:49:11,315995	7297.255045	27.710178	dslb-178-006-082-139.178.006.pools.vodafone-ip.de	193.24.227.196	192.168.2.100
18717	2019-11-22 18:49:26,313007	7312.252057	14.997012	dslb-178-006-082-139.178.006.pools.vodafone-ip.de	193.24.227.196	192.168.2.100
18740	2019-11-22 18:49:41,311285	7327.250335	14.998278	dslb-178-006-082-139.178.006.pools.vodafone-ip.de	193.24.227.196	192.168.2.100
18750	2019-11-22 18:49:48,349258	7334.288308	7.037973	dslb-178-006-082-139.178.006.pools.vodafone-ip.de	193.24.227.196	192.168.2.100
18784	2019-11-22 18:50:10,462588	7356.401638	22.113330	dslb-178-006-082-139.178.006.pools.vodafone-ip.de	193.24.227.196	192.168.2.100

- ICMP errors from many different routers!!! → BGP routing RFC 1918 space to different locations

Case Study: NTP Pool

- You could analyze so much more with the data
  - merge networks to ASes
  - or to BGP routers that send errors
  - or to Global Unicast Address Assignments
- You get the idea
- In the end, you'll need to investigate indiviual problems

Final Challenges

Photo by Samantha Sophia on Unsplash

Final Challenges

- The Ultimate PCAP: https://weberblog.net/the-ultimate-pcap/
- `udp.stream in {1177,1178}`
  Same query, different errors. Why?
- `udp.stream in {249,251}`
  Compare the errors. What are the differences? What's notable on the second packet?

Final Challenges

- The Ultimate PCAP: https://weberblog.net/the-ultimate-pcap/

- `udp.stream == 1178`
  Captured on the client; how far away was the firewall?

- `ip.addr == 93.197.166.58`
  How many hops into the foreign network was the originating host located?

Final Challenges

- `tshark` ☺

- Investigate all ICMPv6 destination unreachables: Which codes occur and how often?

- Investigate all ICMP destination unreachables (port unreachable) that had a destination of 194.247.5.12: List all source IP addresses. Omit doubles.

Conclusion

Photo by Thought Catalog on Unsplash

Conclusion

- ICMP provides useful information
- Analysis requires special tools & tricks
- Recommendation: capture & analyse on a regular basis

Ressources

- Shichao's Notes, ICMPv4 and ICMPv6: Internet Control Message Protocol: https://notes.shichao.io/tcpv1/ch8/
- RFC 4443, Internet Control Message Protocol (ICMPv6): https://tools.ietf.org/html/rfc4443
- Weberblog.net, Incorrect Working IPv6 NTP Clients/Networks: https://weberblog.net/incorrect-working-ipv6-ntp-clients-networks/

Feedback

A #PCAP is worth a thousand words
weberblog.net/the-ultimate-pcap

IPv6
netsec.blog
NAT