DNS and related attacks

CSE 548 Spring 2026

jedimaestro@asu.edu

Outline

- IP and IP fragmentation basics
- Review: On-path *vs.* in-path *vs.* off-path
- Birthday attacks
  - Example: Wagner Sacramento’s birthday attack on DNS (2002)
- Dan Kaminsky’s DNS poisoning attack (2008) (concurrency)
- Side channel attacks (information theory)
  - Example: Fragmentation attack
- Solution: signatures
  - Important ingredient for signatures: *extended Euclidean algorithm*

Where do Internet standards come from?

- IETF = Internet Engineering Task Force
- RFC = Request for Comments
  - MUST, MUST NOT, SHOULD, SHOULD NOT, MAY (RFC 2119)
- “The only laws on the Internet are assembly and RFCs” -- Phrack 65
  - Assembly is an abstraction
  - RFCs are not always followed
    - Often ambiguous

IP reassembly

- Routers (or endhosts, if they want) can break IP packets up into fragments that the receiver has to reassemble
- Ambiguity in the way overlapping IP fragments are put back together into an IP packet
- All of the following images were plagiarized from:

  https://www.sans.org/reading-room/whitepapers/detection/ip-fragment-reassembly-scapy-33969

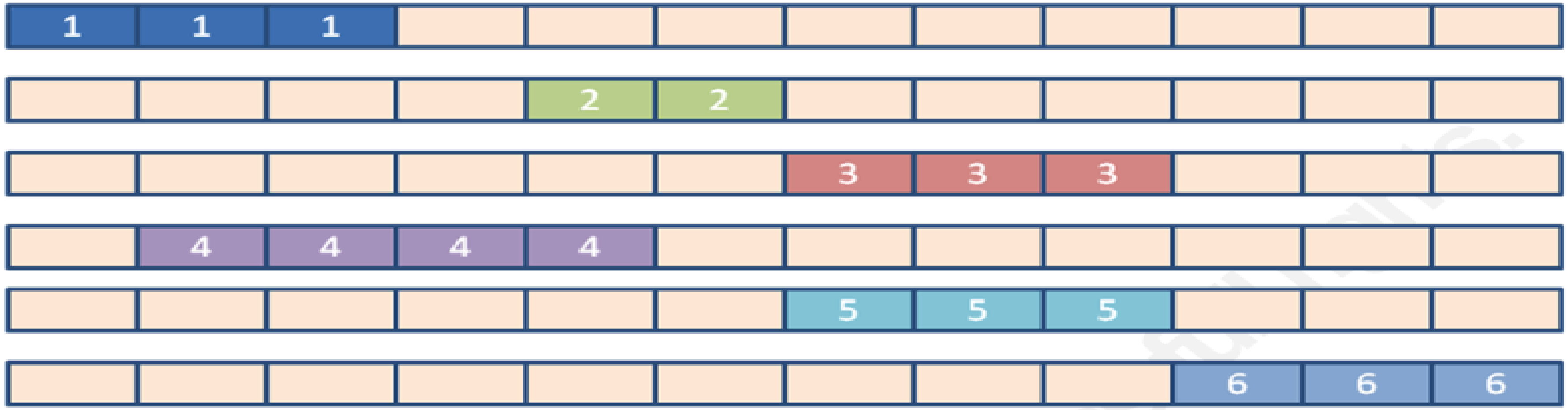


Figure 1: 6 Fragmented Packets (Shankar & Paxson, 2003)(Novak, 2005)

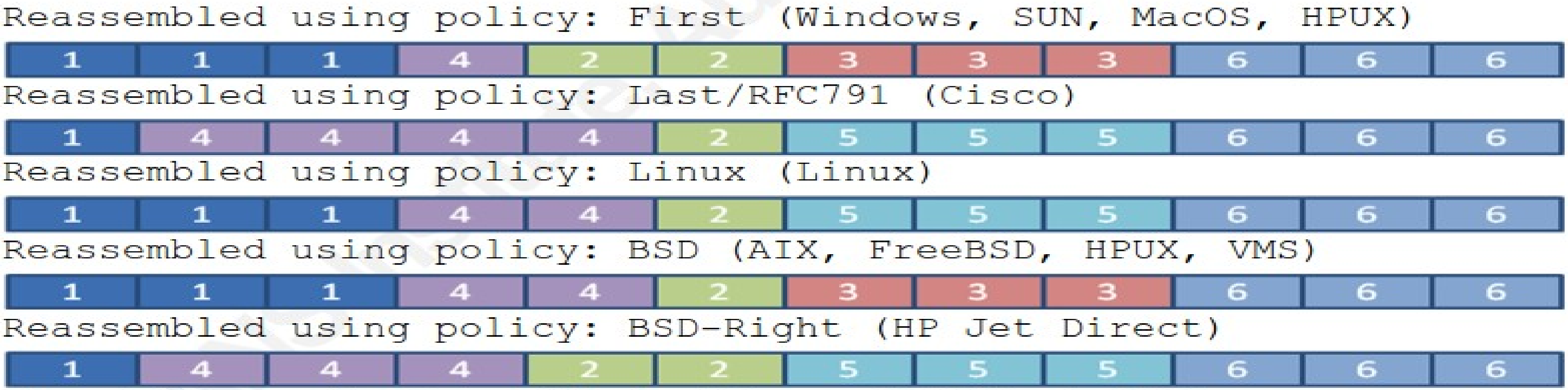


Figure 2: 5 Reassembly Methods (Shankar & Paxson, 2003)(Novak, 2005)

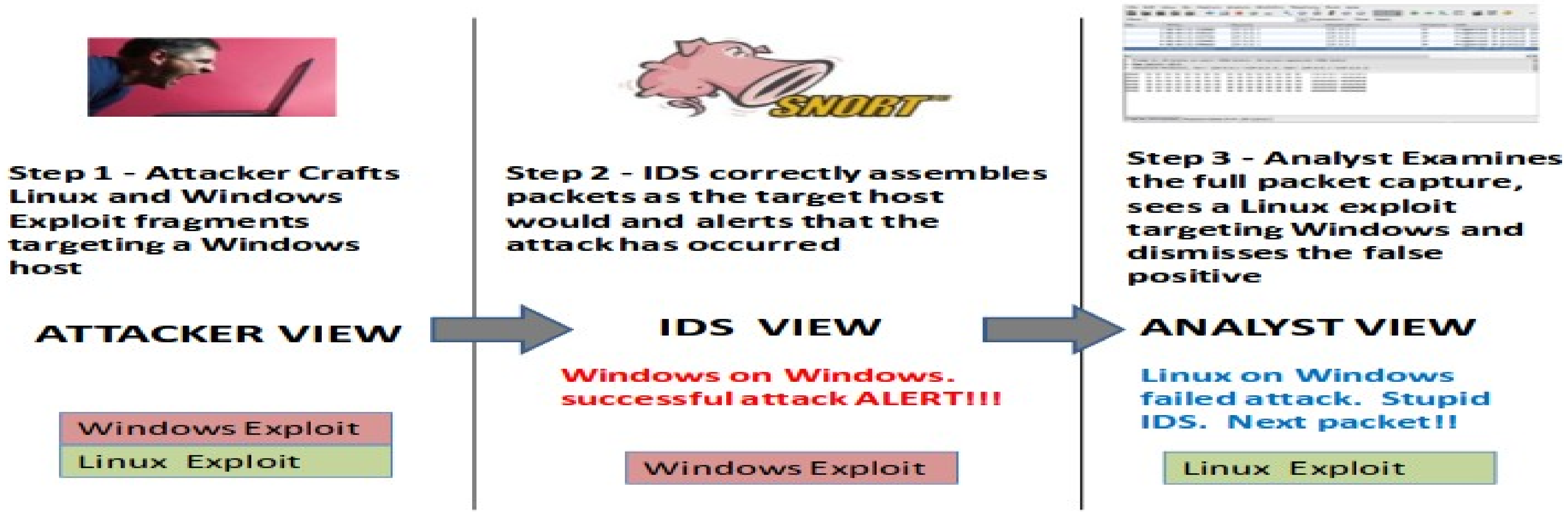


Figure 3: Views of the attacker, IDS and analyst

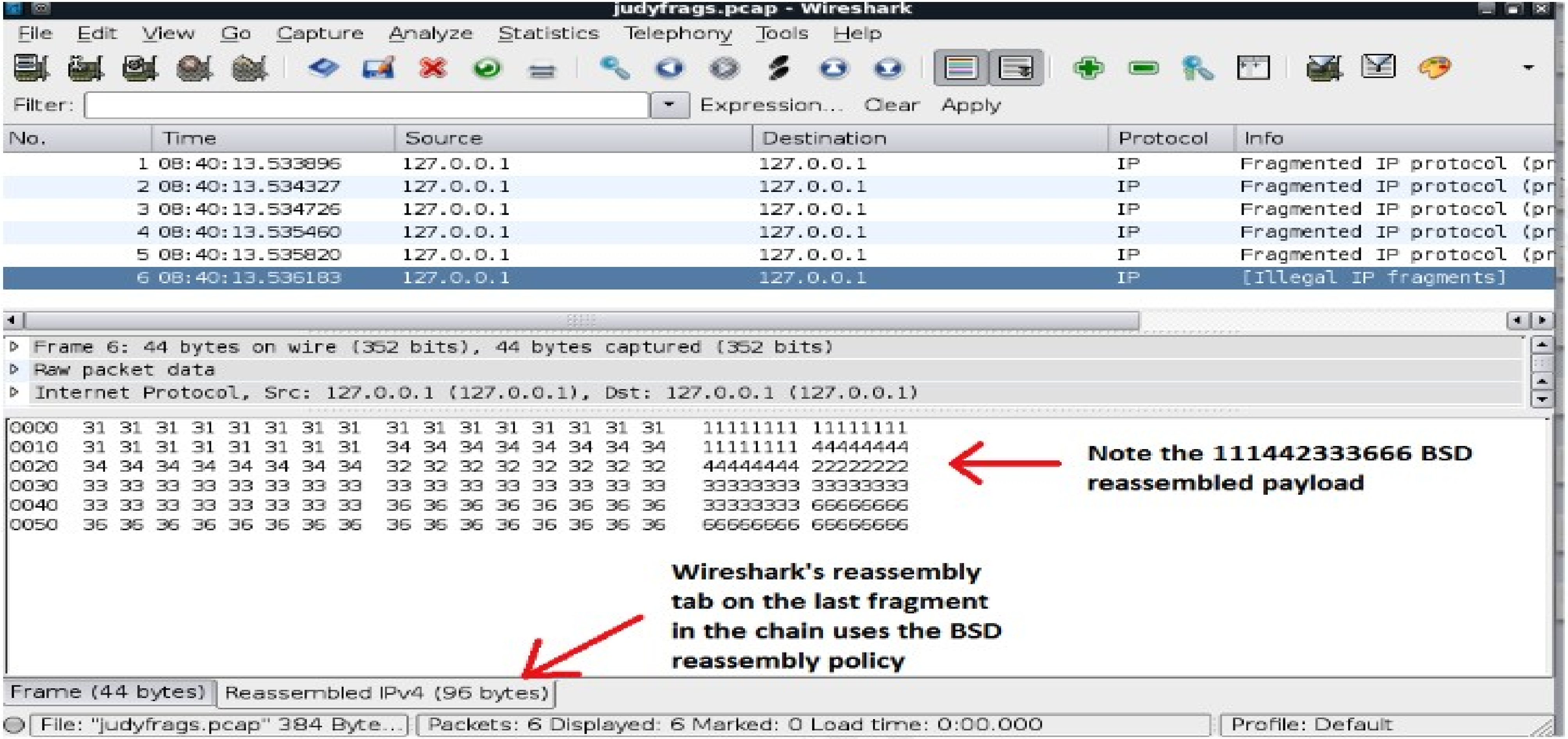
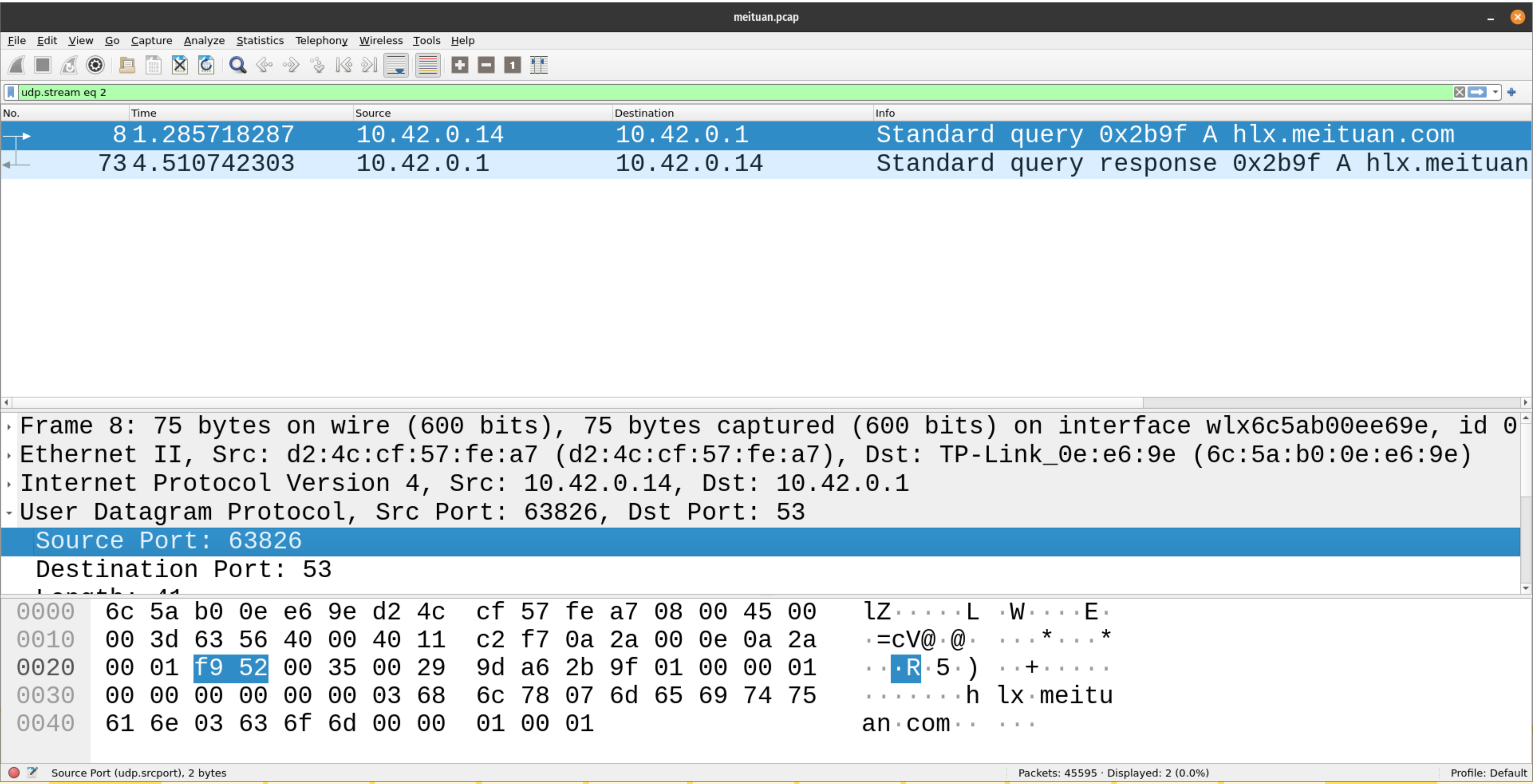


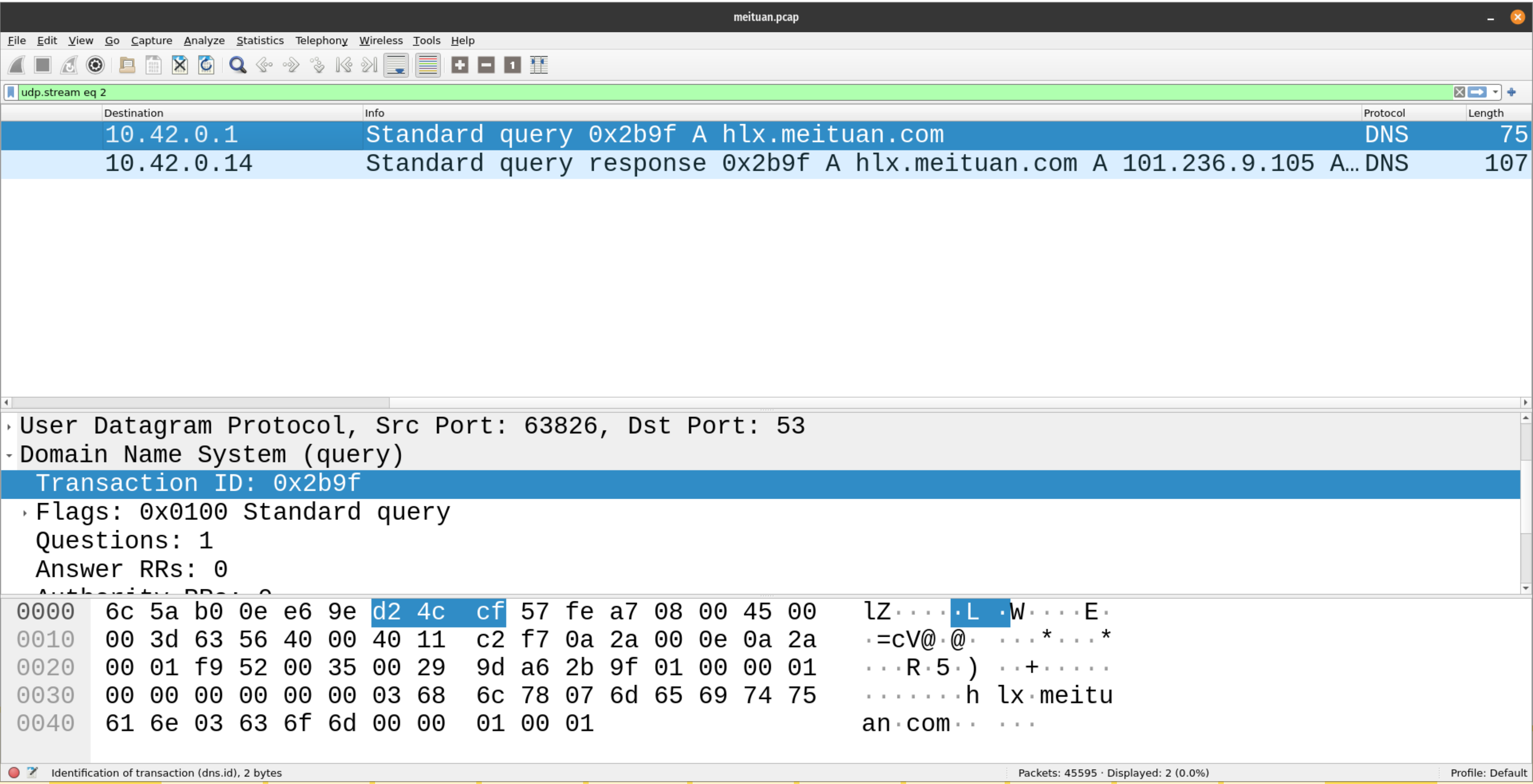
Figure 4: Wireshark uses BSD reassembly technique

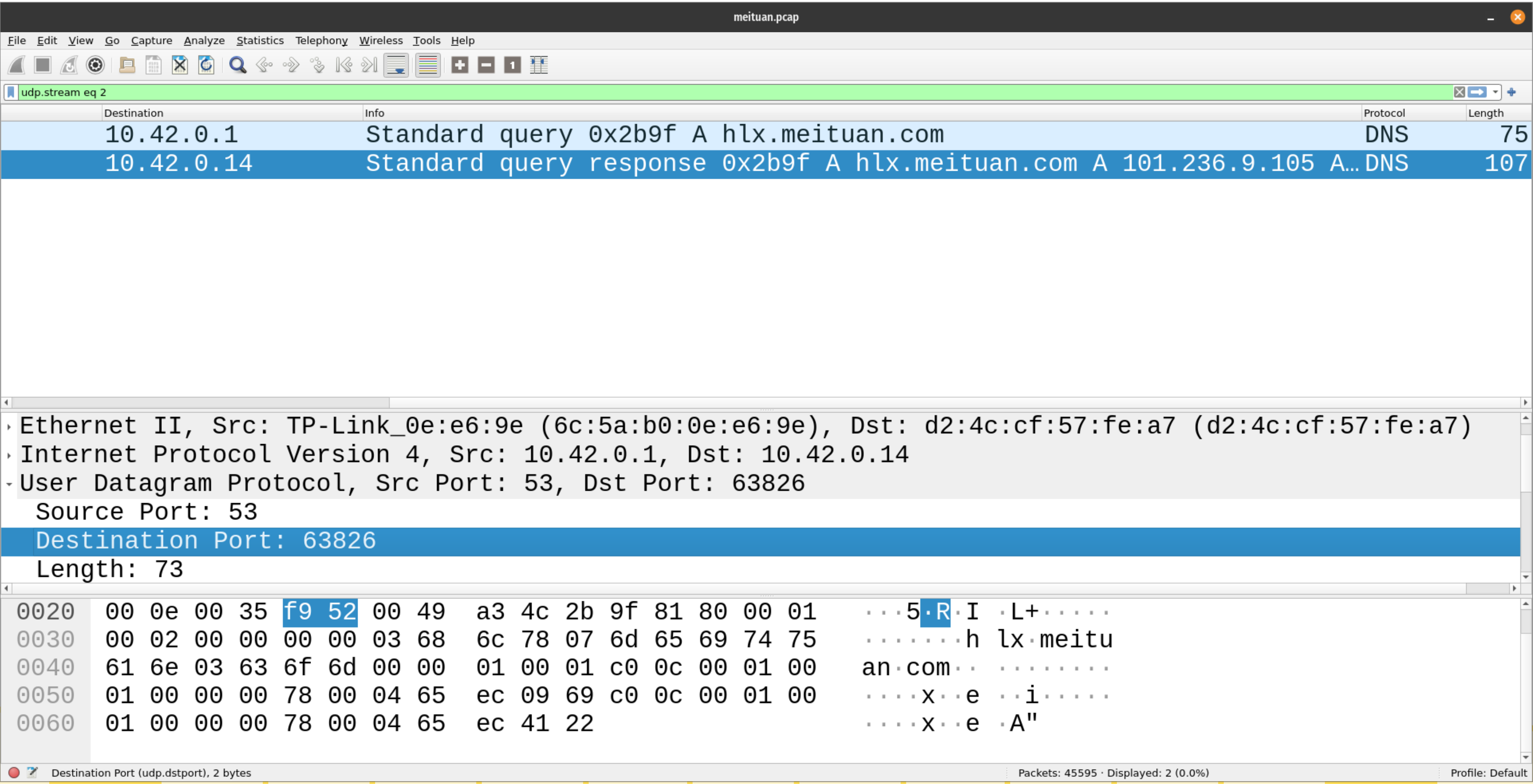
IPID → IP header (layer 3)

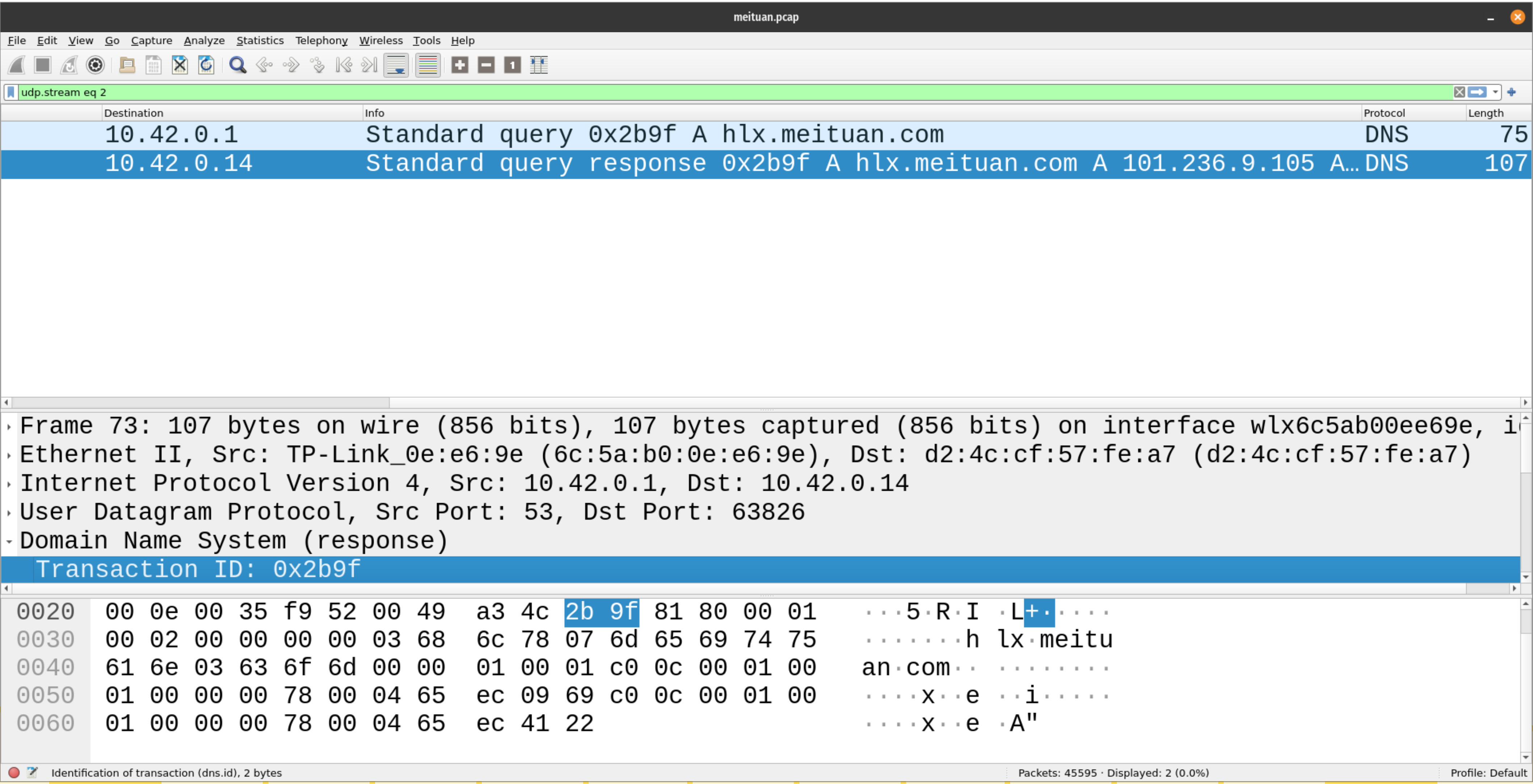
Source port → UDP header (layer 4)

TXID → DNS request and response (layer 7)


meituan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
udp.stream eq 2
No. Time Source Destination Info
8 1.285718287 10.42.0.14 10.42.0.1 Standard query 0x2b9f A hlx.meituan.com
73 4.510742303 10.42.0.1 10.42.0.14 Standard query response 0x2b9f A hlx.meituan
Frame 8: 75 bytes on wire (600 bits), 75 bytes captured (600 bits) on interface wlx6c5ab00ee69e, id 0
Ethernet II, Src: d2:4c:cf:57:fe:a7 (d2:4c:cf:57:fe:a7), Dst: TP-Link_0e:e6:9e (6c:5a:b0:0e:e6:9e)
Internet Protocol Version 4, Src: 10.42.0.14, Dst: 10.42.0.1
User Datagram Protocol, Src Port: 63826, Dst Port: 53
Source Port: 63826
Destination Port: 53
0000 6c 5a b0 0e e6 9e d2 4c cf 57 fe a7 08 00 45 00 lZ·····L ·W····E·
0010 00 3d 63 56 40 00 40 11 c2 f7 0a 2a 00 0e 0a 2a ·=cV@·@· ···*···*
0020 00 01 f9 52 00 35 00 29 9d a6 2b 9f 01 00 00 01 ···R·5·) ··+·····
0030 00 00 00 00 00 00 03 68 6c 78 07 6d 65 69 74 75 ·······h lx·meitu
0040 61 6e 03 63 6f 6d 00 00 01 00 01 an·com·· ···
Source Port (udp.srcport), 2 bytes
Packets: 45595 · Displayed: 2 (0.0%)
Profile: Default


meituan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
udp.stream eq 2
Destination Info Protocol Length
10.42.0.1 Standard query 0x2b9f A hlx.meituan.com DNS 75
10.42.0.14 Standard query response 0x2b9f A hlx.meituan.com A 101.236.9.105 A… DNS 107
User Datagram Protocol, Src Port: 63826, Dst Port: 53
Domain Name System (query)
Transaction ID: 0x2b9f
Flags: 0x0100 Standard query
Questions: 1
Answer RRs: 0
0000 6c 5a b0 0e e6 9e d2 4c cf 57 fe a7 08 00 45 00 lZ····L·W····E·
0010 00 3d 63 56 40 00 40 11 c2 f7 0a 2a 00 0e 0a 2a ·=cV@·@····*···*
0020 00 01 f9 52 00 35 00 29 9d a6 2b 9f 01 00 00 01 ···R·5·)··+·····
0030 00 00 00 00 00 00 03 68 6c 78 07 6d 65 69 74 75 ·······hlx·meitu
0040 61 6e 03 63 6f 6d 00 00 01 00 01 an·com·····
Identification of transaction (dns.id), 2 bytes
Packets: 45595 · Displayed: 2 (0.0%)
Profile: Default


meituan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
udp.stream eq 2
Destination Info Protocol Length
10.42.0.1 Standard query 0x2b9f A hlx.meituan.com DNS 75
10.42.0.14 Standard query response 0x2b9f A hlx.meituan.com A 101.236.9.105 A… DNS 107
Ethernet II, Src: TP-Link_0e:e6:9e (6c:5a:b0:0e:e6:9e), Dst: d2:4c:cf:57:fe:a7 (d2:4c:cf:57:fe:a7)
Internet Protocol Version 4, Src: 10.42.0.1, Dst: 10.42.0.14
User Datagram Protocol, Src Port: 53, Dst Port: 63826
Source Port: 53
Destination Port: 63826
Length: 73
0020 00 0e 00 35 f9 52 00 49 a3 4c 2b 9f 81 80 00 01 ···5·R·I ·L+·····
0030 00 02 00 00 00 00 03 68 6c 78 07 6d 65 69 74 75 ·······h lx·meitu
0040 61 6e 03 63 6f 6d 00 00 01 00 01 c0 0c 00 01 00 an·com·· ········
0050 01 00 00 00 78 00 04 65 ec 09 69 c0 0c 00 01 00 ····x··e ··i·····
0060 01 00 00 00 78 00 04 65 ec 41 22 ····x··e ·A"
Destination Port (udp.dstport), 2 bytes
Packets: 45595 · Displayed: 2 (0.0%)
Profile: Default


meituan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
udp.stream eq 2
Destination Info Protocol Length
10.42.0.1 Standard query 0x2b9f A hlx.meituan.com DNS 75
10.42.0.14 Standard query response 0x2b9f A hlx.meituan.com A 101.236.9.105 A… DNS 107
Frame 73: 107 bytes on wire (856 bits), 107 bytes captured (856 bits) on interface wlx6c5ab00ee69e, i
Ethernet II, Src: TP-Link_0e:e6:9e (6c:5a:b0:0e:e6:9e), Dst: d2:4c:cf:57:fe:a7 (d2:4c:cf:57:fe:a7)
Internet Protocol Version 4, Src: 10.42.0.1, Dst: 10.42.0.14
User Datagram Protocol, Src Port: 53, Dst Port: 63826
Domain Name System (response)
Transaction ID: 0x2b9f
0020 00 0e 00 35 f9 52 00 49 a3 4c 2b 9f 81 80 00 01 ···5·R·I ·L+·····
0030 00 02 00 00 00 00 03 68 6c 78 07 6d 65 69 74 75 ·······h lx·meitu
0040 61 6e 03 63 6f 6d 00 00 01 00 01 c0 0c 00 01 00 an·com·· ········
0050 01 00 00 00 78 00 04 65 ec 09 69 c0 0c 00 01 00 ····x··e ··i·····
0060 01 00 00 00 78 00 04 65 ec 41 22 ····x··e ·A"
Identification of transaction (dns.id), 2 bytes
Packets: 45595 · Displayed: 2 (0.0%)
Profile: Default

jedi@tortuga: ~

```
jedi@tortuga:~$ dig @8.8.8.8 ns meituan.com | head -n 21

; <<>> DiG 9.18.39-0ubuntu0.22.04.2-Ubuntu <<>> @8.8.8.8 ns meituan.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59326
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;meituan.com.                   IN      NS

;; ANSWER SECTION:
meituan.com.            21600   IN      NS      ns3.dnsv5.com.
meituan.com.            21600   IN      NS      ns4.dnsv5.com.

;; Query time: 1029 msec
;; SERVER: 8.8.8.8#53(8.8.8.8) (UDP)
;; WHEN: Mon Feb 02 09:58:19 MST 2026
;; MSG SIZE  rcvd: 82
jedi@tortuga:~$
```

jedi@tortuga: ~

```
jedi@tortuga:~$ dig @8.8.8.8 ns hlx.meituan.com | head -n 21

; <<>> DiG 9.18.39-0ubuntu0.22.04.2-Ubuntu <<>> @8.8.8.8 ns hlx.meituan.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30382
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;hlx.meituan.com.               IN      NS

;; ANSWER SECTION:
hlx.meituan.com.        180     IN      CNAME   bi-hreport.vip.meituan.com.

;; AUTHORITY SECTION:
meituan.com.            180     IN      SOA     ns3.dnsv5.com. enterprise3dnsadm
in.dnspod.com. 1770016267 3600 180 1209600 180

;; Query time: 45 msec
;; SERVER: 8.8.8.8#53(8.8.8.8) (UDP)
jedi@tortuga:~$
```

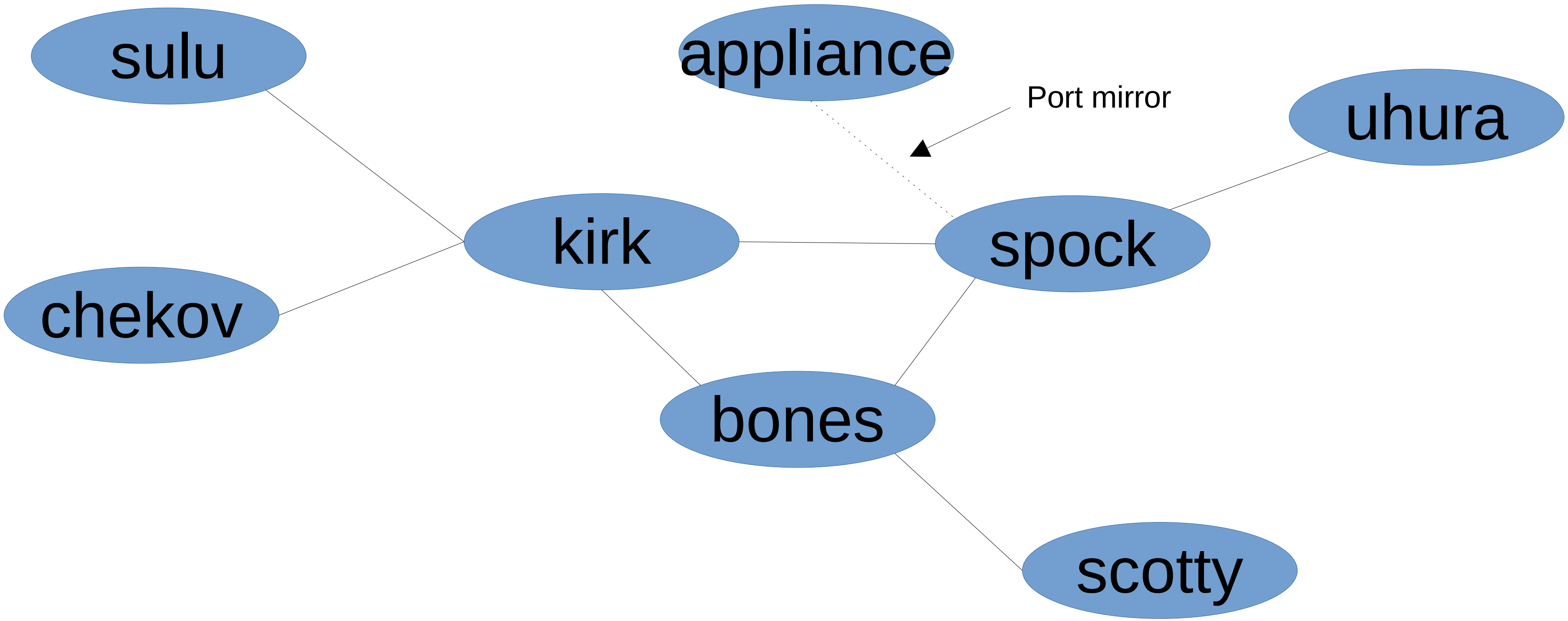
What stops me from saying `www.facebook.com` is a common name for `www.breakpointingbad.com`?

```
    The primary goal of bailiwick checking is to prevent a malicious
    nameserver authoritative for example.com from providing (and a
    resolver from caching) records for other-example.net, or worse, for
    TLDs or the root zone.
```

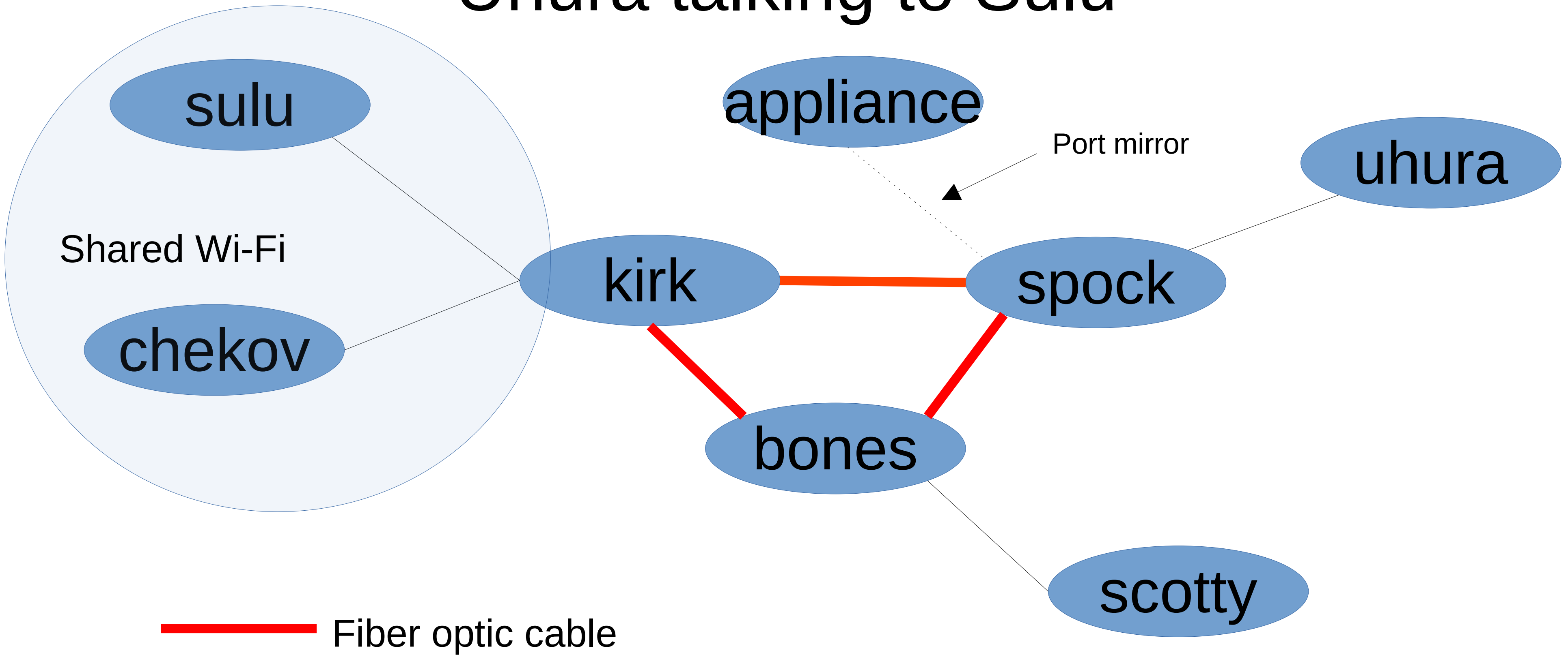
https://datatracker.ietf.org/doc/draft-qiu-dnsop-enhanced-bailiwick/#:~:text=The%20primary%20goal%20of%20bailiwick,TLDs%20or%20the%20root%20zone.

Introduced in the mid 90’s

Uhura talking to Sulu



Uhura talking to Sulu



sulu == DNS client, uhura == DNS server

- kirk and spock are in-path
- appliance is on-path
  - Gets a copy of the packets from the port mirror on kirk
- chekov is on-path
  - Shared Wi-Fi with sulu, kirk has a wireless interface and two fiber optic interfaces
- scotty and bones are off-path

On-path attack

- Need to respond faster than the DNS server
  - Not hard, 3 seconds (example above) is an eternity
  - Maybe DoS the DNS server
- Need to get the TXID and source port correct
  - Trivial, just read them from the packet

In-path attack

- ~~Need to respond faster than the DNS server~~
  - ~~Not hard, 3 seconds (example above) is an eternity~~
  - ~~Maybe DoS the DNS server~~
- Need to get the TXID and source port correct
  - Trivial, just read them from the packet
- Just don’t forward the request to the DNS server
  - Or, do and then modify the response on its way back

Off-path attack

- Need to respond faster than the DNS server
  - ~~Not hard, 3 seconds (example above) is an eternity~~
  - ~~Maybe~~ DoS the DNS server
- Need to get the TXID and source port correct
  - **Not easy**, being off path means you're *blind* to these values
  - Guessing might work ($2^{16} * 2^{16} = 2^{32}$)
    - Side channels and birthday attacks even better
- Need to know what was queried and when
  - Cache poisoning (you know these things because you caused it)

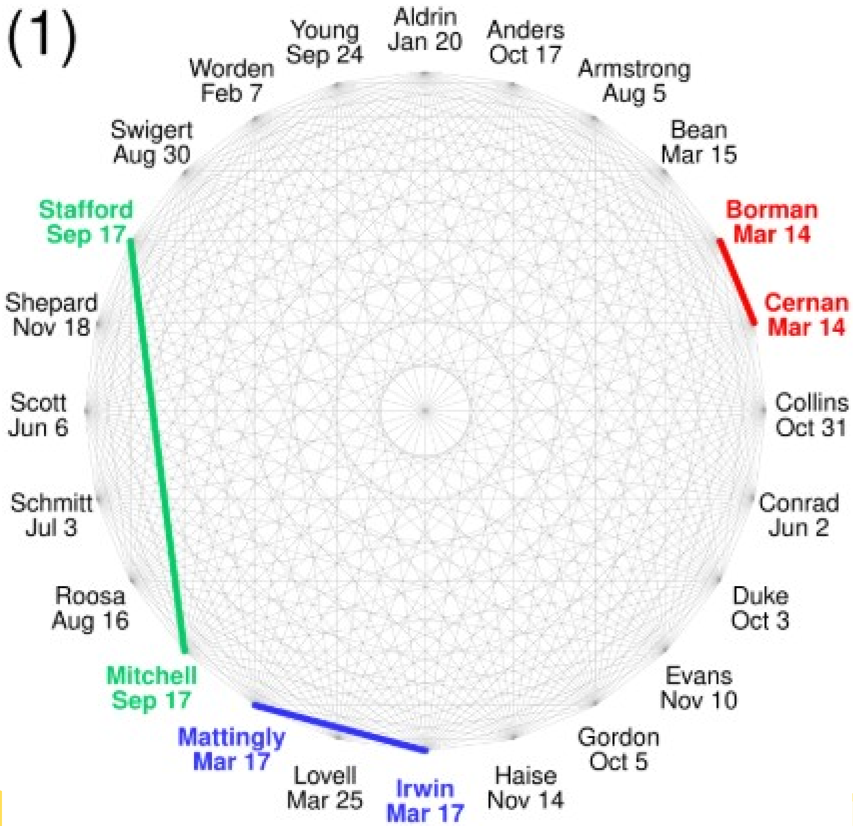
Birthday Attacks

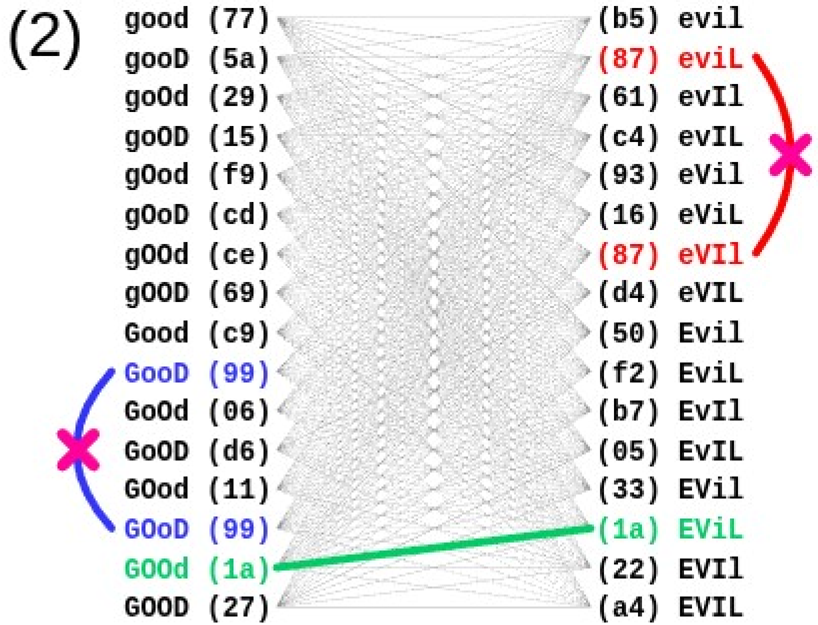
- https://www.kb.cert.org/vuls/id/457
- 2002

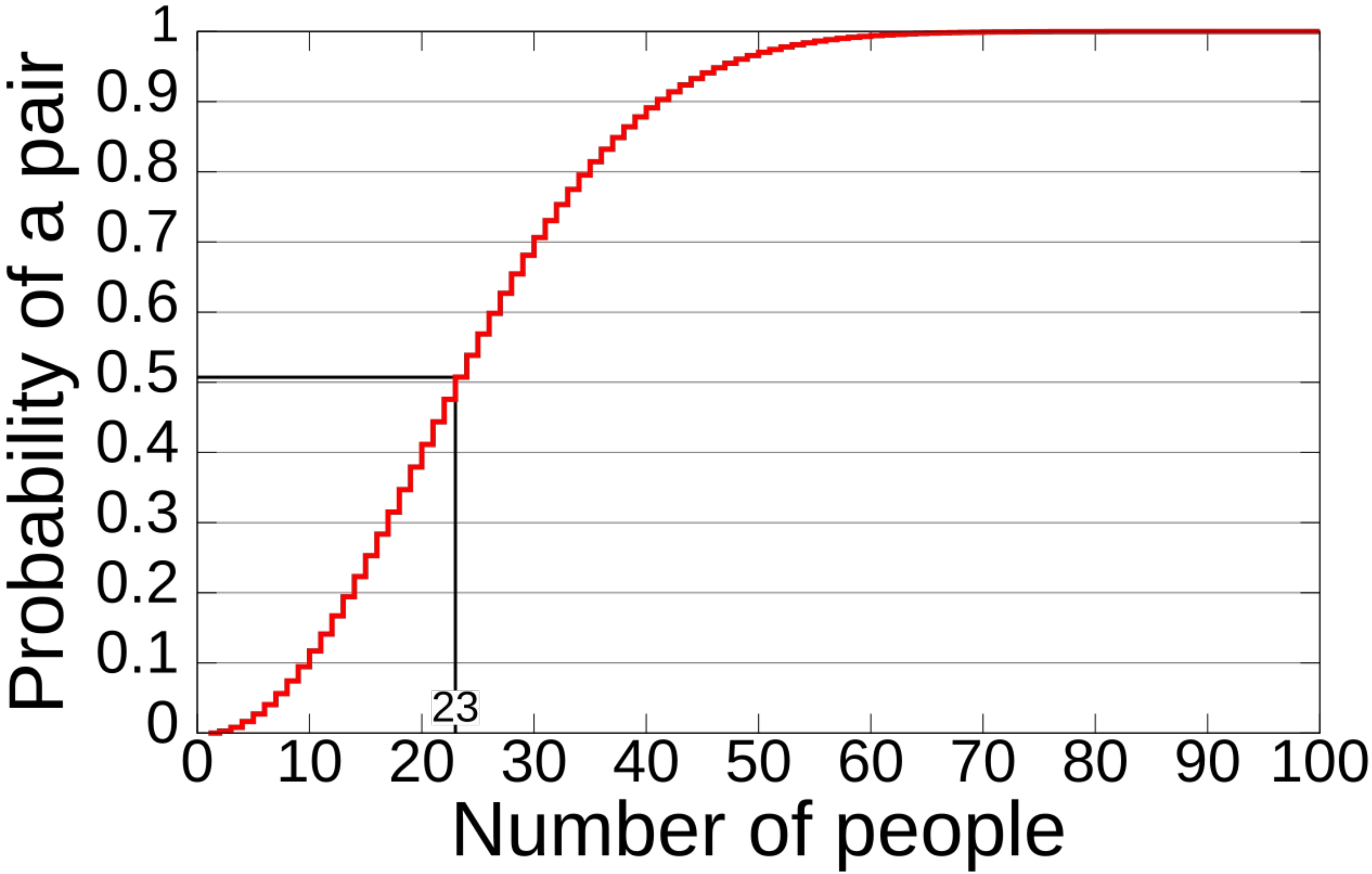
If the attacker has to guess...	...and is limited to the following number of open requests...	...it will take the following number of packets to achieve a 50% success rate (includes both requests and responses)
TID only (16bits)	1	32.7 k (2^{15})
TID only (16bits)	4	10.4 k
TID only (16bits)	200	427
TID only (16bits)	unlimited	426
TID and port (32 bits)	1	2.1 billion (2^{31})
TID and port (32 bits)	4	683 million
TID and port (32 bits)	200	15 million
TID and port (32 bits)	unlimited	109 k

Table 1: Number of packets required to reach 50% success probability for various numbers of open queries

https://en.wikipedia.org/wiki/Birthday_attack






Probability of a pair
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
23
0
10
20
30
40
50
60
70
80
90
100
Number of people

Solution to the specific birthday attack on DNS above… Don’t allow multiple queries for the same domain at the same time.

Dan Kaminsky’s attack (2008)

- https://www.blackhat.com/presentations/bh-jp-08/bh-jp-08-Kaminsky/BlackHat-Japan-08-Kaminsky-DNS08-BlackOps.pdf

DNS is distributed

- Three possible answers to any question
  - "Here's your answer"
  - "Go away"
  - "I don't know, ask that guy over there"
    - This is delegation. You start with a request, and then get bounced around all over the place.
    - 13 root servers: "www.foo.com? I don't know, go ask the com server, it's at 1.2.3.4"
    - Com server: "www.foo.com? I don't know, go ask the foo.com server, it's at 2.3.4.5"
    - Foo.com server: "www.foo.com? Yeah, that's at 3.4.5.6."

jedi@tortuga: ~

```
jedi@tortuga:~$ dig www.breakpointingbad.com

; <<>> DiG 9.18.39-0ubuntu0.22.04.2-Ubuntu <<>> www.breakpointingbad.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33334
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;www.breakpointingbad.com.      IN      A

;; ANSWER SECTION:
www.breakpointingbad.com. 1799  IN      A       149.28.240.117

;; Query time: 58 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Mon Feb 02 10:21:40 MST 2026
;; MSG SIZE  rcvd: 69

jedi@tortuga:~$
```

jedi@tortuga: ~

```
jedi@tortuga:~$ dig www.kodak.com | head -n 21

; <<>> DiG 9.18.39-0ubuntu0.22.04.2-Ubuntu <<>> www.kodak.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 48823
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;www.kodak.com.                 IN      A

;; ANSWER SECTION:
www.kodak.com.          260     IN      CNAME   web-prod-fd-bmhsdwc9gnbgdres.a01
.azurefd.net.
web-prod-fd-bmhsdwc9gnbgdres.a01.azurefd.net. 20 IN CNAME mr-a01.tm-azurefd.net.
mr-a01.tm-azurefd.net.  9       IN      A       150.171.109.147

;; Query time: 14 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Mon Feb 02 10:25:33 MST 2026
;; MSG SIZE  rcvd: 148
jedi@tortuga:~$
```

- If the bad guy can reply 100 times before the good guy returns, that 65536 to 1 advantage drops to 655 to 1.
  - Alas…still long odds. And when he loses, he has to wait the TTL. That could be 655 days – almost 2 years!
  - Or maybe not.

Finally, the bad guy doesn't actually need to wait to try again.

- If the bad guy asks the name server to look up www.foo.com ten times, there will only be one race with the good guy
  - The first race will be lost (most likely), and then the other nine will be suppressed by the TTL
    - No new races on this name for one more day! Here, use the answer from a while ago
    - So, can we race on other names?
- If the bad guy asks the name server to look up 1.foo.com, 2.foo.com, 3.foo.com, and so on, for ten names, there will be 10 races with the good guy
  - TTL only stops repeated races for the same name!
- Eventually, the bad guy will guess the right TXID before the good guy shows up with it
  - And now…the bad guy is the proud spoofer of … 83.foo.com
  - So? He didn't *want* to poison 83.foo.com. He wanted www.foo.com

Bait and Switch

- Is it possible for a bad guy, who has won the race for 83.foo.com, to end up stealing www.foo.com as well?
  - He has three possible replies that can be associated with correctly guessed TXID
  - 1) "Here's your answer for 83.foo.com – it's 6.6.6.6"
  - 2) "I don't know the answer for 83.foo.com."
  - 3) "83.foo.com? I don't know, go ask the www.foo.com server, it's at 6.6.6.6"
    - This has to work – it's just another delegation
      - 13 root servers: "83.foo.com? I don't know, go ask the com server, it's at 1.2.3.4"
      - Com server: "83.foo.com? I don't know, go ask the foo.com server, it's at 2.3.4.5"
      - Foo.com server: "83.foo.com? I don't know, go ask the www.foo.com server, it's at 6.6.6.6"

Does bailiwick checking save us?

Does bailiwick checking save us?

No! The only “authentication” is the source port and TXID!

Solution to the Kaminsky attack… OSes now randomize source ports.

(Also, some other stuff, like 0x20 encoding: `BrEAkPoinTiNGBaD.COm`)

But, what if we didn’t have to guess the TXID or source port?

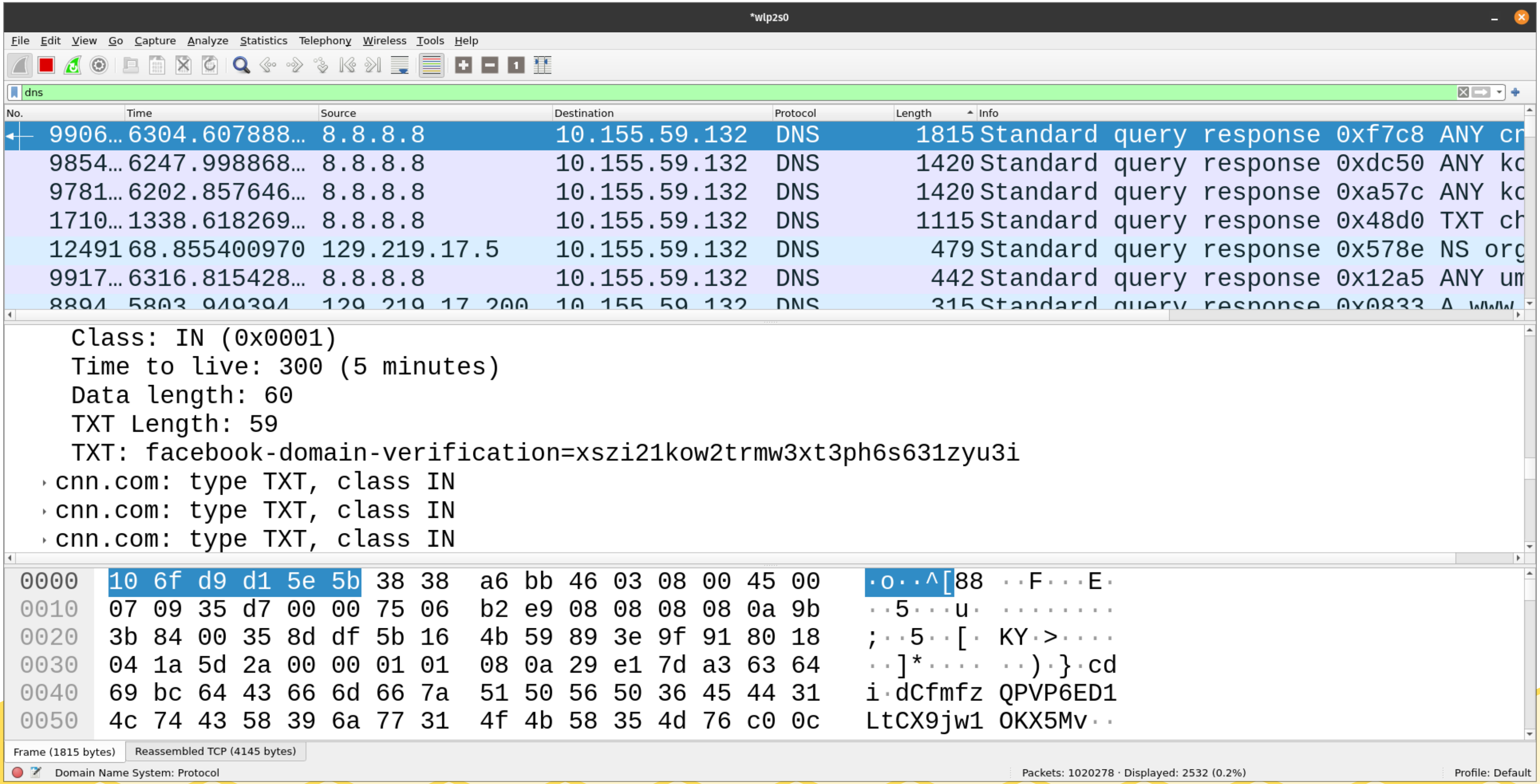
jedi@tortuga: ~/Downloads

```
; <<>> DiG 9.18.39-0ubuntu0.22.04.2-Ubuntu <<>> cnn.com any @8.8.8.8
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61262
;; flags: qr rd ra; QUERY: 1, ANSWER: 74, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;cnn.com.                       IN      ANY

;; ANSWER SECTION:
cnn.com.                60      IN      A       151.101.195.5
cnn.com.                60      IN      A       151.101.131.5
cnn.com.                60      IN      A       151.101.67.5
cnn.com.                60      IN      A       151.101.3.5
cnn.com.                21600   IN      NS      ns-378.awsdns-47.com.
cnn.com.                21600   IN      NS      ns-1652.awsdns-14.co.uk.
cnn.com.                21600   IN      NS      ns-587.awsdns-09.net.
cnn.com.                21600   IN      NS      ns-1242.awsdns-27.org.
cnn.com.                900     IN      SOA     ns-1652.awsdns-14.co.uk. awsdns-
hostmaster.amazon.com. 1 7200 900 1209600 86400
:
```

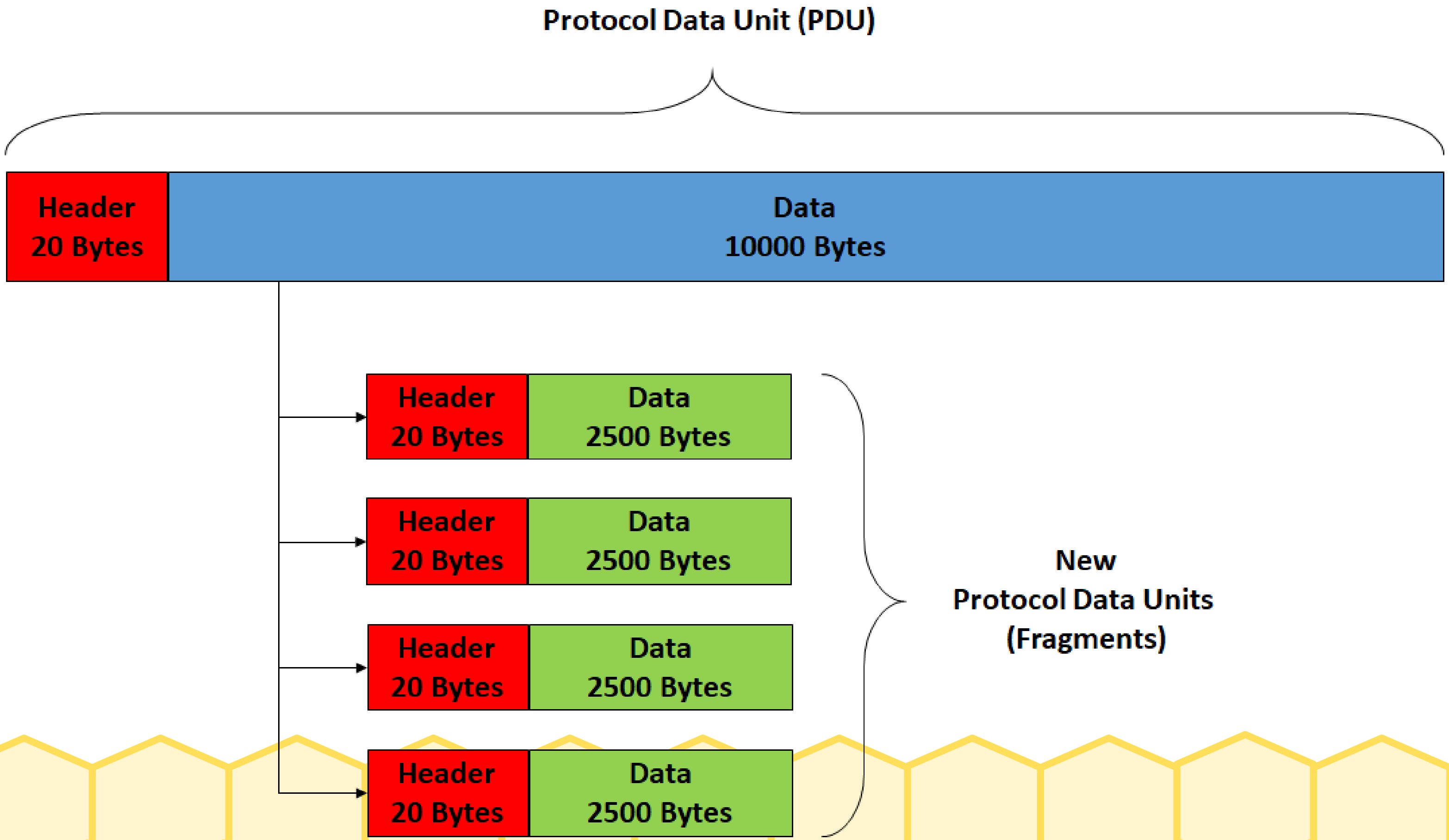
jedi@tortuga: ~/Downloads

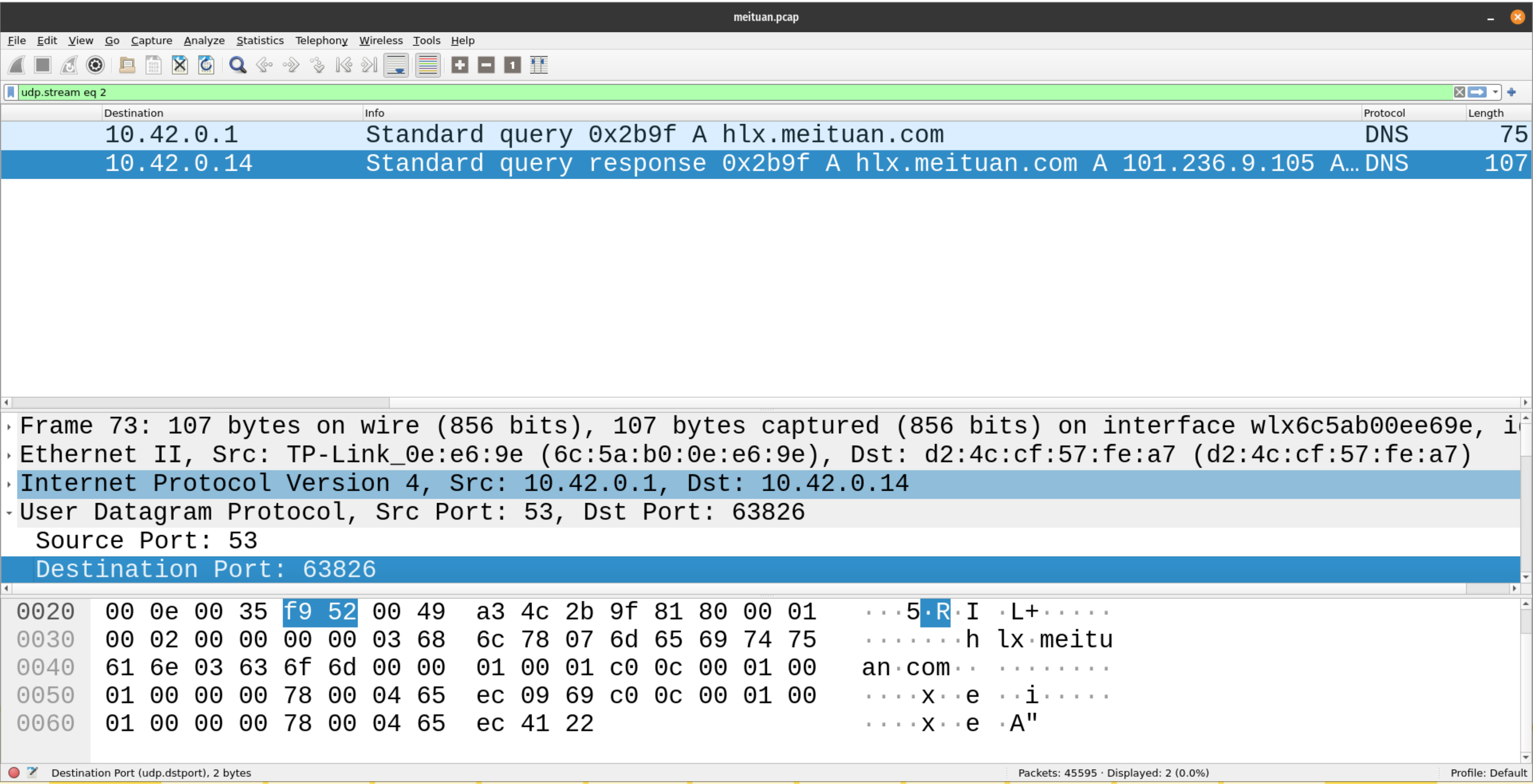
```
cnn.com.                21600   IN      NS      ns-587.awsdns-09.net.
cnn.com.                21600   IN      NS      ns-1242.awsdns-27.org.
cnn.com.                900     IN      SOA     ns-1652.awsdns-14.co.uk. awsdns-
hostmaster.amazon.com. 1 7200 900 1209600 86400
cnn.com.                300     IN      MX      10 cnn-com.mail.protection.outlo
ok.com.
cnn.com.                300     IN      TXT     "MS=ms66433104"
cnn.com.                300     IN      TXT     "228426766-4422034"
cnn.com.                300     IN      TXT     "openai-domain-verification=dv-y
GIc9wI1iK7uFqtmBqEp94Xk"
cnn.com.                300     IN      TXT     "126953328-4422040"
cnn.com.                300     IN      TXT     "adobe-sign-verification=c3dc321
7f76deddcb413a23e4e665fad"
cnn.com.                300     IN      TXT     "_globalsign-domain-verification
=2lybn8Z2GKCTHNehPEREKdz_jh5SahShpw0eRqCWjl"
cnn.com.                300     IN      TXT     "2baPGrmeo+RwsWdIdq/gIVSEWNb4tC9
mLGQu0j4l/mduqhm06T+V9vNLXsauLyH9FwMZJSRHvj/YHGKOVWRylw=="
cnn.com.                300     IN      TXT     "facebook-domain-verification=xs
zi21kow2trmw3xt3ph6s631zyu3i"
cnn.com.                300     IN      TXT     "lucidlink-verification=B9TYHWKA
XAA93NQ61ST71E7NW8"
cnn.com.                300     IN      TXT     "133461244-4422058"
cnn.com.                300     IN      TXT     "667921863-4422007"
:
```

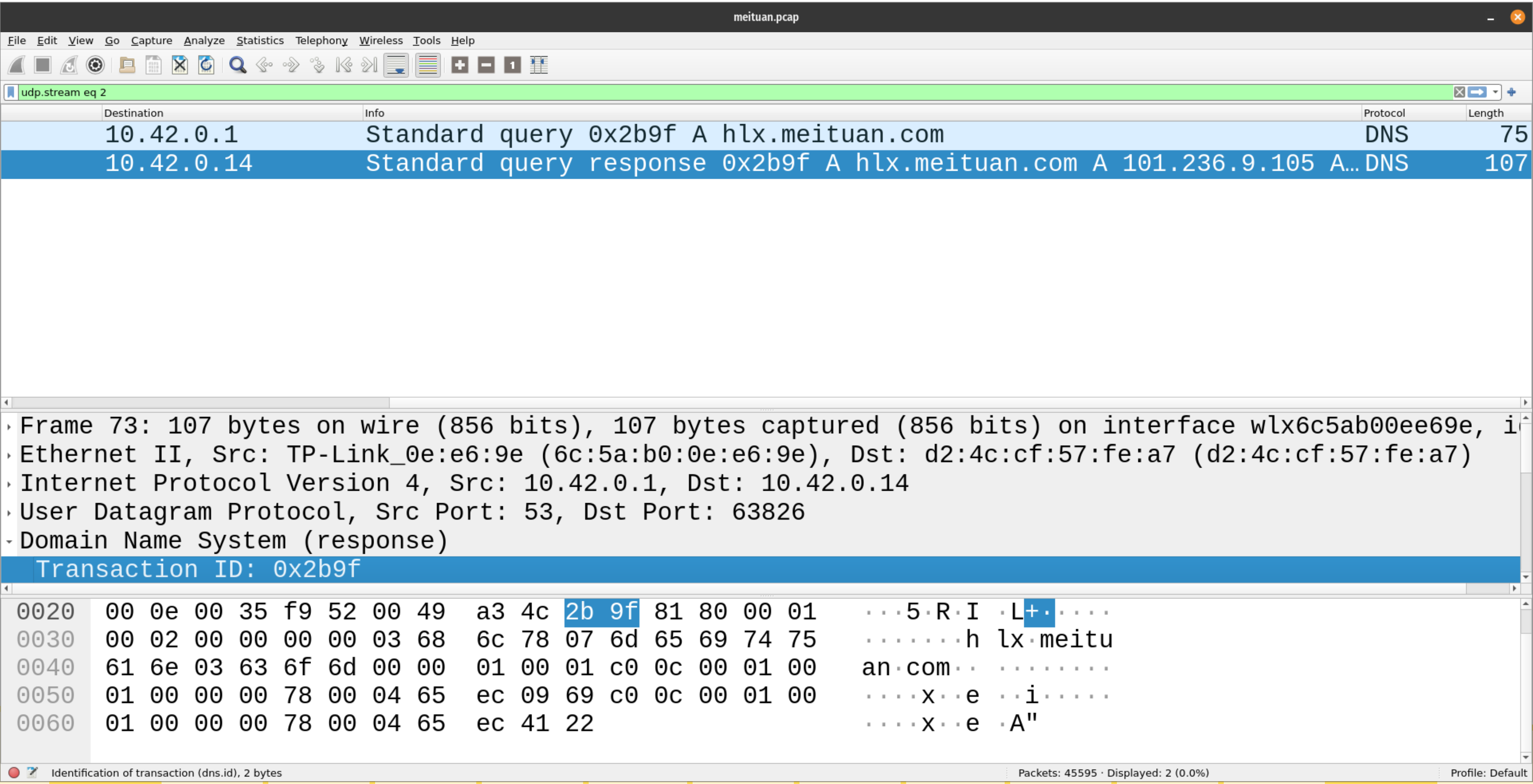
Deprecated around 2019 because of RFC 8482?



https://en.wikipedia.org/wiki/IP_fragmentation



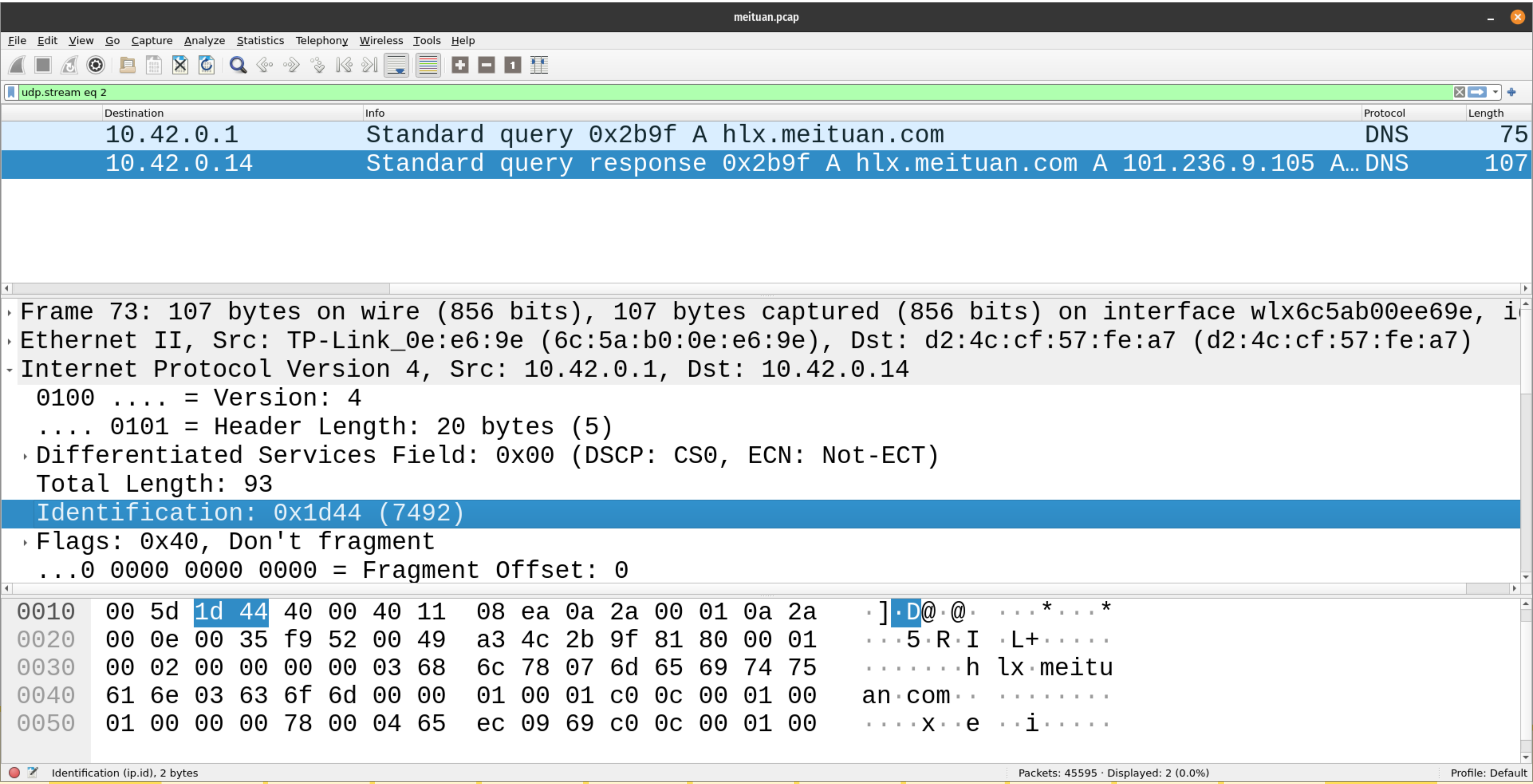

meituan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
udp.stream eq 2
Destination Info Protocol Length
10.42.0.1 Standard query 0x2b9f A hlx.meituan.com DNS 75
10.42.0.14 Standard query response 0x2b9f A hlx.meituan.com A 101.236.9.105 A… DNS 107
Frame 73: 107 bytes on wire (856 bits), 107 bytes captured (856 bits) on interface wlx6c5ab00ee69e, i
Ethernet II, Src: TP-Link_0e:e6:9e (6c:5a:b0:0e:e6:9e), Dst: d2:4c:cf:57:fe:a7 (d2:4c:cf:57:fe:a7)
Internet Protocol Version 4, Src: 10.42.0.1, Dst: 10.42.0.14
User Datagram Protocol, Src Port: 53, Dst Port: 63826
Source Port: 53
Destination Port: 63826
0020 00 0e 00 35 f9 52 00 49 a3 4c 2b 9f 81 80 00 01 ···5·R·I ·L+·····
0030 00 02 00 00 00 00 03 68 6c 78 07 6d 65 69 74 75 ·······h lx·meitu
0040 61 6e 03 63 6f 6d 00 00 01 00 01 c0 0c 00 01 00 an·com·· ········
0050 01 00 00 00 78 00 04 65 ec 09 69 c0 0c 00 01 00 ····x··e ··i·····
0060 01 00 00 00 78 00 04 65 ec 41 22 ····x··e ·A"
Destination Port (udp.dstport), 2 bytes
Packets: 45595 · Displayed: 2 (0.0%)
Profile: Default


meituan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
udp.stream eq 2
Destination Info Protocol Length
10.42.0.1 Standard query 0x2b9f A hlx.meituan.com DNS 75
10.42.0.14 Standard query response 0x2b9f A hlx.meituan.com A 101.236.9.105 A… DNS 107
Frame 73: 107 bytes on wire (856 bits), 107 bytes captured (856 bits) on interface wlx6c5ab00ee69e, i
Ethernet II, Src: TP-Link_0e:e6:9e (6c:5a:b0:0e:e6:9e), Dst: d2:4c:cf:57:fe:a7 (d2:4c:cf:57:fe:a7)
Internet Protocol Version 4, Src: 10.42.0.1, Dst: 10.42.0.14
User Datagram Protocol, Src Port: 53, Dst Port: 63826
Domain Name System (response)
Transaction ID: 0x2b9f
0020 00 0e 00 35 f9 52 00 49 a3 4c 2b 9f 81 80 00 01
0030 00 02 00 00 00 00 03 68 6c 78 07 6d 65 69 74 75
0040 61 6e 03 63 6f 6d 00 00 01 00 01 c0 0c 00 01 00
0050 01 00 00 00 78 00 04 65 ec 09 69 c0 0c 00 01 00
0060 01 00 00 00 78 00 04 65 ec 41 22
Identification of transaction (dns.id), 2 bytes
Packets: 45595 · Displayed: 2 (0.0%)
Profile: Default

https://arxiv.org/pdf/1205.4011

Fragmentation Considered Poisonous

Amir Herzberg[†] *and Haya Shulman*[‡]
Dept. of Computer Science, Bar Ilan University
[†]*amir.herzberg@gmail.com*, [‡]*haya.shulman@gmail.com*


meituan.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
udp.stream eq 2
Destination | Info | Protocol | Length
10.42.0.1 | Standard query 0x2b9f A hlx.meituan.com | DNS | 75
10.42.0.14 | Standard query response 0x2b9f A hlx.meituan.com A 101.236.9.105 A… | DNS | 107
Frame 73: 107 bytes on wire (856 bits), 107 bytes captured (856 bits) on interface wlx6c5ab00ee69e, i
Ethernet II, Src: TP-Link_0e:e6:9e (6c:5a:b0:0e:e6:9e), Dst: d2:4c:cf:57:fe:a7 (d2:4c:cf:57:fe:a7)
Internet Protocol Version 4, Src: 10.42.0.1, Dst: 10.42.0.14
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 93
Identification: 0x1d44 (7492)
Flags: 0x40, Don't fragment
...0 0000 0000 0000 = Fragment Offset: 0
0010 00 5d 1d 44 40 00 40 11 08 ea 0a 2a 00 01 0a 2a
0020 00 0e 00 35 f9 52 00 49 a3 4c 2b 9f 81 80 00 01
0030 00 02 00 00 00 00 03 68 6c 78 07 6d 65 69 74 75
0040 61 6e 03 63 6f 6d 00 00 01 00 01 c0 0c 00 01 00
0050 01 00 00 00 78 00 04 65 ec 09 69 c0 0c 00 01 00
Identification (ip.id), 2 bytes
Packets: 45595 · Displayed: 2 (0.0%)
Profile: Default

IPIDs

- Used to identify fragments and put them back together
  - Should never be repeated for a given destination
- Different strategies
  - Globally incrementing counter that wraps around at 2^{16}
  - Pick at random without replacement
  - Per-destination
  - Bucket-based
    - Can add noise

How much entropy?

- Globally incrementing counter?
- Pick at random?

15.99997798605273604449798348692167764035 70...

How much entropy?

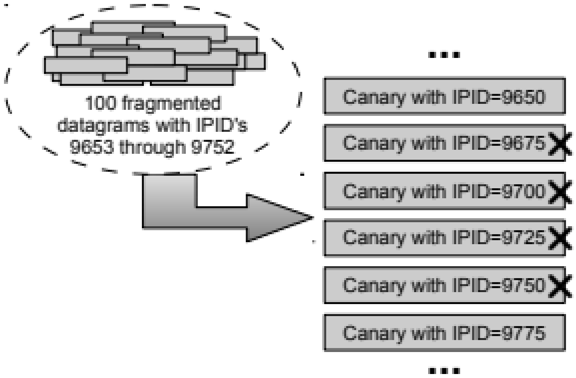
- Per-destination?
  - Think about a noisy server that is talking to other clients
- Bucket-based?

https://www.usenix.org/system/files/conference/foci14/foci14-knockel.pdf

Counting Packets Sent Between Arbitrary Internet Hosts

Jeffrey Knockel
Dept. of Computer Science
University of New Mexico
jeffk@cs.unm.edu

Jedidiah R. Crandall
Dept. of Computer Science
University of New Mexico
crandall@cs.unm.edu



https://jedcrandall.github.io/INFOCOM2018.pdf

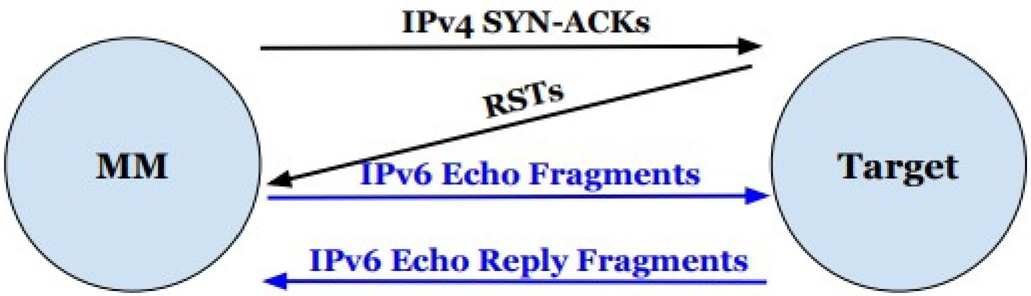


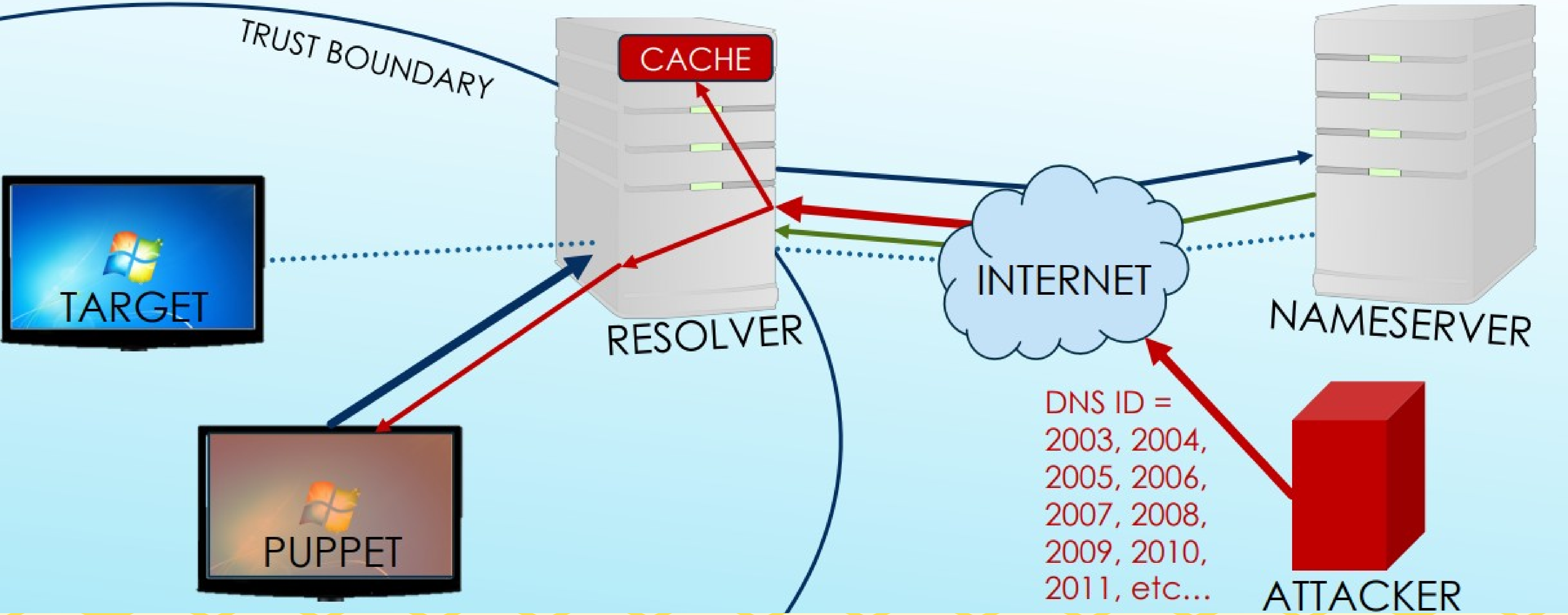
Fig. 3. IPv4 and IPv6 alias resolution.

Fragmentation attacks on Linux resolvers

- https://media.defcon.org/DEF%20CON%2027/DEF%20CON%2027%20presentations/DEFCON-27-Travis-Palmer-First-try-DNS-Cache-Poisoning-with-IPv4-and-IPv6-Fragmentation.pdf

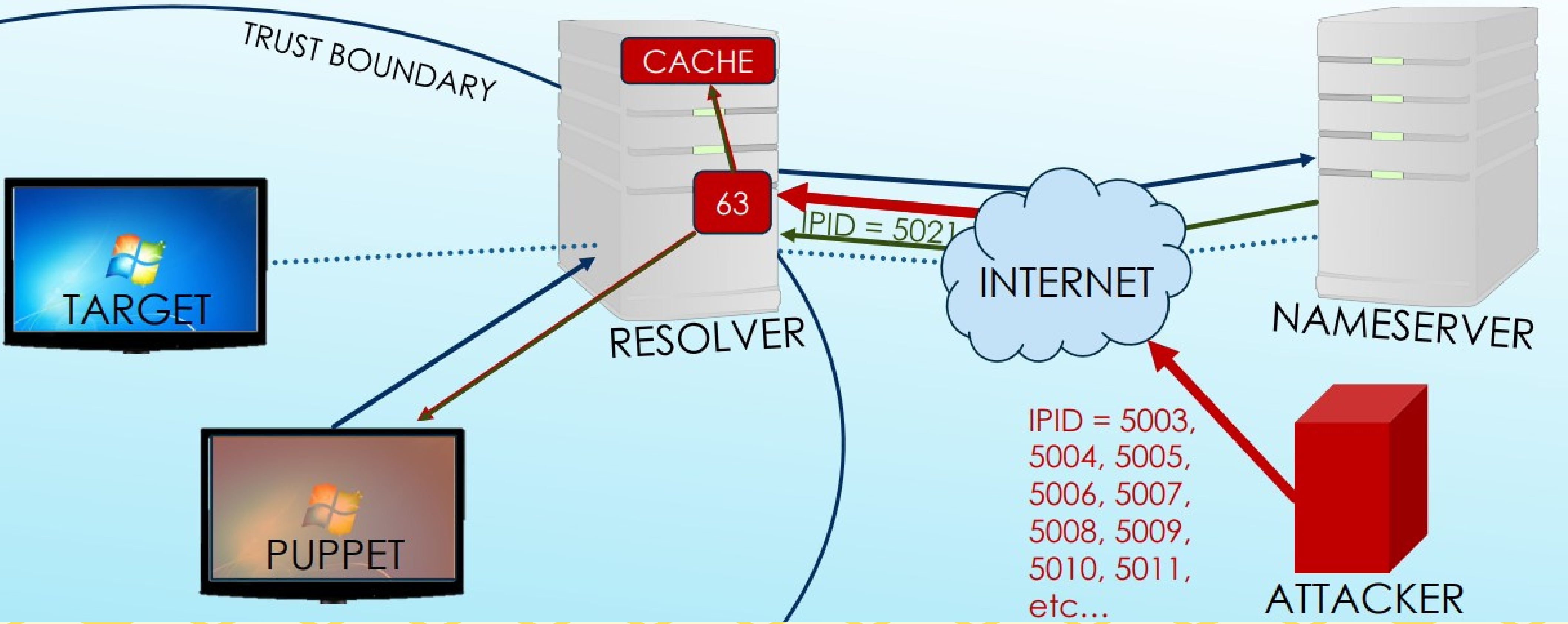
Kaminsky’s attack, assuming source port is completely predictable and you only need to guess the TXID...

IDEAL POISONING SCENARIO



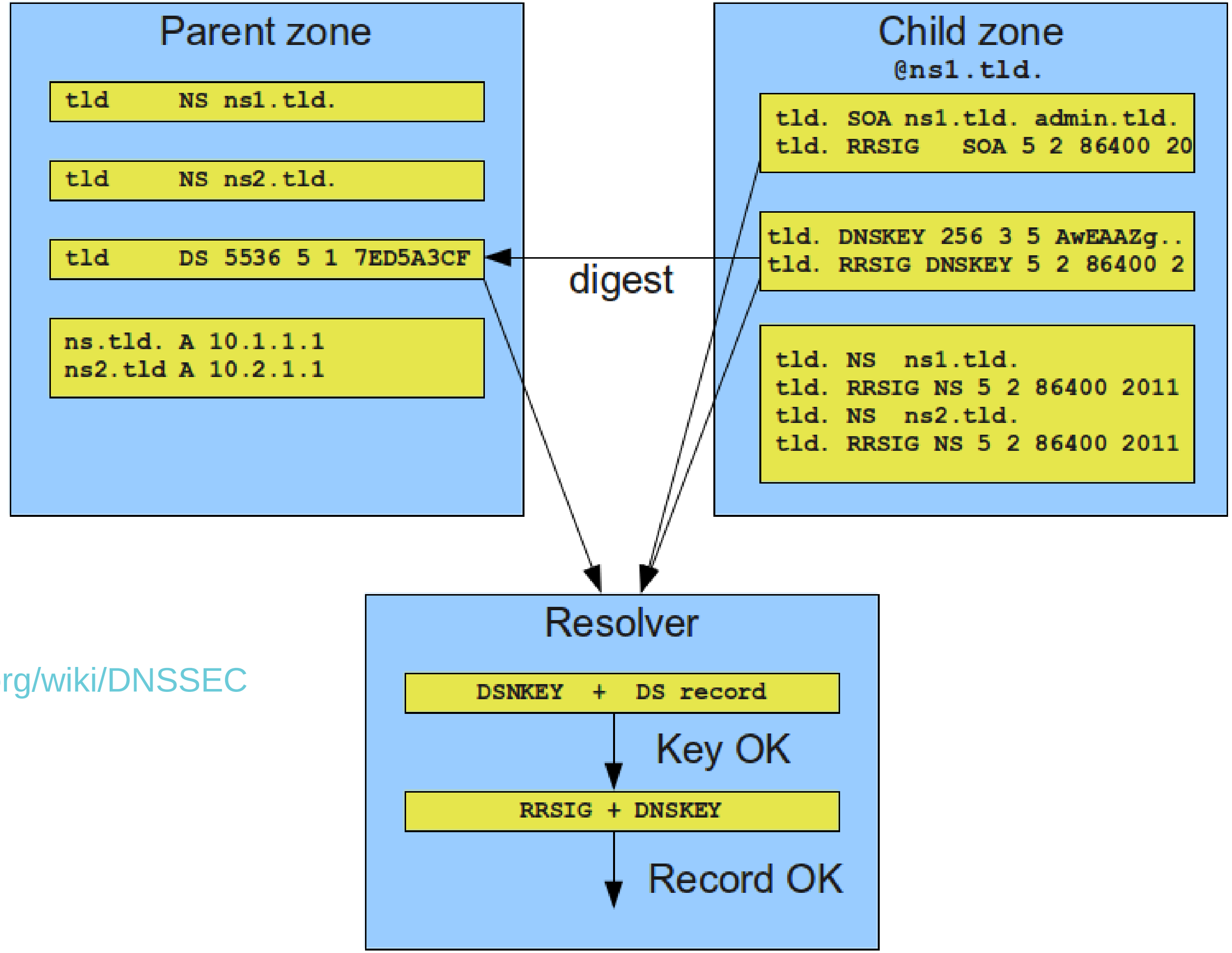
Fragmentation attacks, only need to guess IPID (TXID and source port are in existing fragment from the DNS server)...

IDEAL POISONING SCENARIO



A real solution would be a real form of authentication, like signatures...

DNSSEC



https://ru.wikipedia.org/wiki/DNSSEC

jedi@tortuga: ~

```
jedi@tortuga:~$ dig fraunhofer.de @8.8.8.8 +dnssec | head -n 18

; <<>> DiG 9.18.39-0ubuntu0.22.04.2-Ubuntu <<>> fraunhofer.de @8.8.8.8 +dnssec
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29859
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 512
;; QUESTION SECTION:
;fraunhofer.de.                 IN      A

;; ANSWER SECTION:
fraunhofer.de.          3593    IN      A       192.102.162.236
fraunhofer.de.          3593    IN      RRSIG   A 8 2 3600 20260227143958 202601
28135244 57189 fraunhofer.de. SHIGBs5CC6hV+DigBXTs0wbBb2gQlnxLVhsVnJxU8Fd6xT/Pjl
zXL1zz r2H2QdP/rgmL6OITTC/hGvfsC8QF601h5oVqSyQjwEMnKS46GXP2VHH4 4MvQAUMUABRN63t+
pqSldpuYVO0/2gjV6IDFO9ne+qCj0zxVraA4e2aY 3Po=

;; Query time: 20 msec
;; SERVER: 8.8.8.8#53(8.8.8.8) (UDP)
jedi@tortuga:~$
```

More info

- http://unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html