Data Formats For DNS Telemetry

Ladislav Lhotka • lhotka@nic.cz • 27 November 2019

CZ DOMAIN REGISTRY

Agenda

- DNS telemetry
- Motivation
- Discussion of DNS traffic capture formats:
  - PCAP
  - Parquet
  - C-DNS
- Experimental comparison of DNS trace sizes
- Conclusions
- CZ.NIC work & plans

DNS telemetry

Collection of data about DNS queries and responses on (or near) an authoritative or recursive server, and its automatic transmission to a remote system for further processing.

Possible uses:

- monitoring and analysis of server operation
- attack forensics
- Day In The Life (DITL)

  https://www.dns-oarc.net/oarc/data/ditl

Implementation strategies:

- directly in the DNS server process
- in another process or another machine

Motivation for this study

- Telemetry processing load may affect DNS server operation, but also the quality of telemetry data itself (e.g. timestamp precision).
- Telemetry data consume network capacity.
- DNS is the critical service – telemetry must be stopped at some point if the system is under attack. However, we don't want to give in too early.

We need *robust, flexible and lightweight* DNS telemetry.

Size of the telemetry data is not the only criterion, but it is important.

PCAP format

Global header

- magic number, format version
- time zone offset
- accuracy of timestamps
- max length of captured packets
- data link type

Packet header

- timestamp (µs or ns)
- number of saved packet octets
- actual packet length

PcapNg

- traces from multiple interfaces in the same file
- selectable time units
- extensible (embedded comments, metadata)

https://github.com/pcapng/pcapng

PCAP is not efficient

Some packet data is (mostly) useless:

- link layer – omitted by some tools (*dnscap*)
- transport layer – especially with TCP
- DNS responses repeat the queries
- contents of resource records may be reconstructed off-line

It is impossible to select a subset of relevant data, e.g. useful for the Entrada schema: https://entrada.sidnlabs.nl/datamodel/table_dns/

```
0000   b0 26 80 1a 3c 3c 48 df 37 27 37 d8 08 00 45 00
0010   01 a1 e6 a5 00 00 40 11 a4 72 c2 00 0d 01 c2 e4
0020   5c 4e 00 35 8b d9 01 8d 2c 05 b0 f8 80 00 00 01
0030   00 00 00 05 00 05 04 4d 41 49 4c 07 70 61 72 4f
0040   6c 6f 64 02 43 5a 00 00 01 00 01 07 70 61 72 6f
0050   6c 6f 64 02 63 7a 00 00 02 00 01 00 00 0e 10 00
0060   0b 03 6e 73 32 04 73 6b 6f 6b c0 29 c0 21 00 02
0070   00 01 00 00 0e 10 00 05 02 6e 73 c0 3b c0 21 00
0080   02 00 01 00 00 0e 10 00 0f 03 6e 73 33 06 73 6b
0090   6f 6b 63 7a 02 65 75 00 20 44 4f 41 51 35 41 4a
00a0   43 4d 32 54 35 4c 51 4c 54 34 48 35 53 4e 52 4f
00b0   30 35 34 4b 4f 45 50 54 36 c0 29 00 32 00 01 00
00c0   00 03 84 00 25 01 00 00 0a 08 1e e1 a1 64 fe 00
00d0   7f 48 14 6e 15 b3 01 4e 8a 55 0e f0 53 cc e1 1c
00e0   09 a9 90 8d 2a 14 dd 00 01 20 c0 6e 00 2e 00 01
00f0   00 00 03 84 00 56 00 32 0d 02 00 00 03 84 5c ea
0100   f6 e8 5c d9 4f 75 2d b3 02 63 7a 00 52 2c 78 3f
0110   0c 12 eb e2 2c 8d 71 63 14 66 30 c2 f0 32 96 cc
0120   c2 c0 b2 10 9a d0 61 95 56 8d 44 5a fb 44 a3 0c
0130   29 19 b8 f4 21 8a 8f d3 87 e8 b3 80 f4 3d 26 b4
0140   36 50 c2 aa 3e 53 b1 c7 37 c3 0b 49 c0 4e 00 01
0150   00 01 00 00 0e 10 00 04 51 00 e9 ec c0 4e 00 1c
0160   00 01 00 00 0e 10 00 10 20 01 15 28 01 81 00 00
0170   00 00 00 00 02 33 02 36 c0 37 00 01 00 01 00 00
0180   0e 10 00 04 59 b9 fd 03 c0 37 00 1c 00 01 00 00
0190   0e 10 00 10 2a 01 04 30 00 36 00 01 00 00 00 00
01a0   02 53 00 03 00 00 29 04 d0 00 00 80 00 00 00
```

← typical DNS reponse

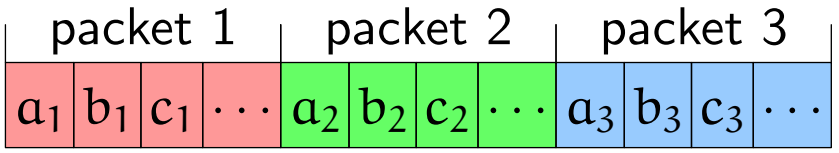
- Only octets in bold are useful for feeding Entrada (**53**/431).
- Query data were already in the matching query packet.

Apache Parquet

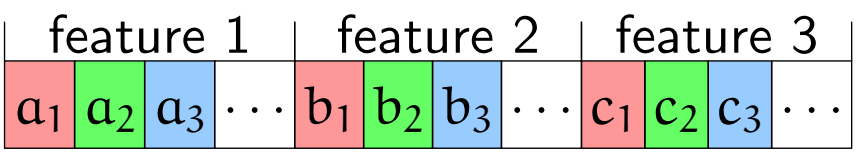
Generic column-oriented data storage format of the Hadoop ecosystem.

Specification: https://parquet.apache.org/documentation/latest/

PCAP: record/row-oriented



Parquet: column-oriented



Advantages:

- better data compression – feature values: same type, repetitive
- Hadoop users need no further conversion

Drawbacks:

- all data have to be in memory in order to assemble columns
- Parquet is complicated – unnecessarily for this use case

Compacted-DNS (C-DNS)

Data format specially designed for efficient storage and transmission of large captures of DNS traffic.

RFC 8618: https://tools.ietf.org/html/rfc8618

Open-source implementation: https://github.com/dns-stats/compactor

High-level structure of C-DNS file:

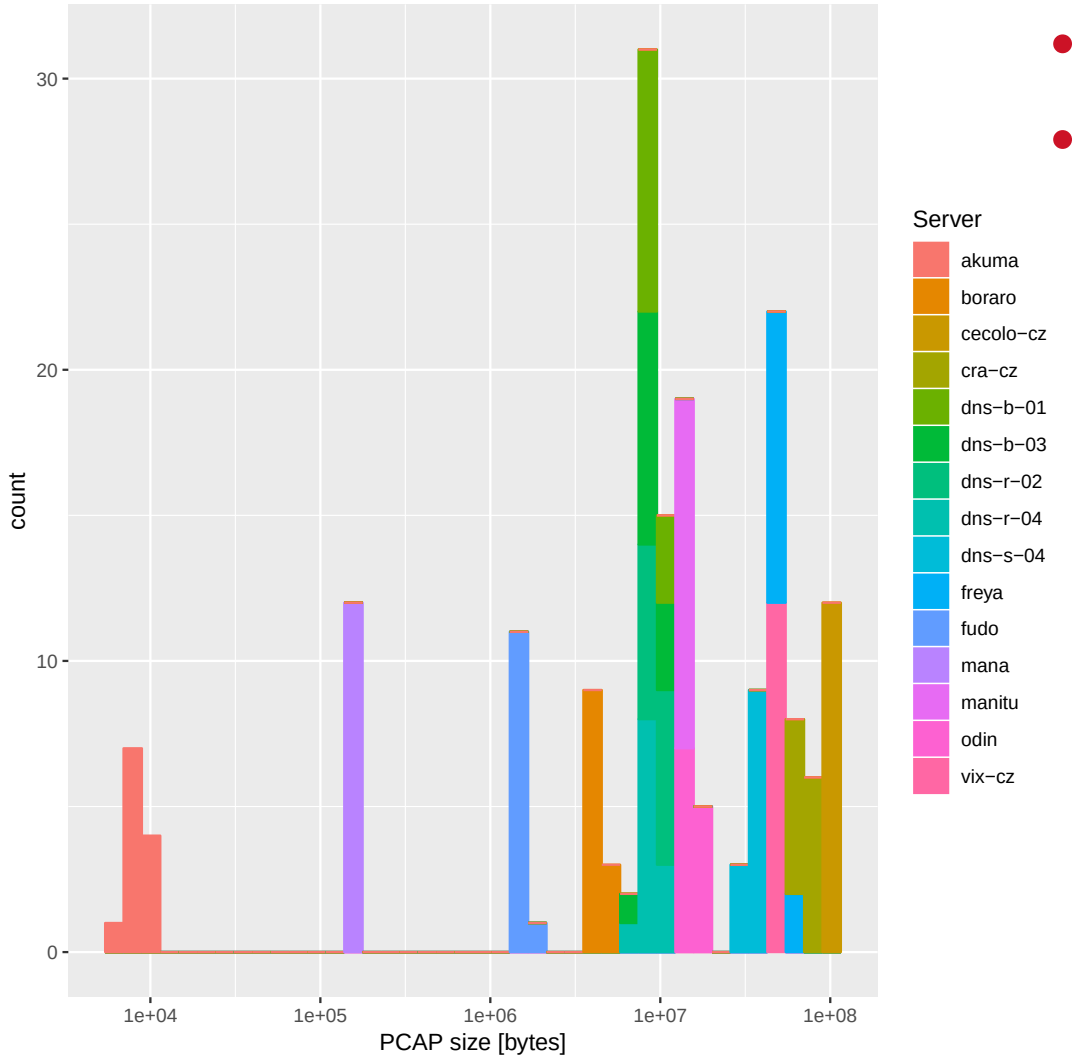
Each block contains:

- DNS query/response items
- address/event counts – per-client counts of IP events
- malformed message items
- block statistics
- block tables – arrays of common data
- + other metadata

C-DNS properties

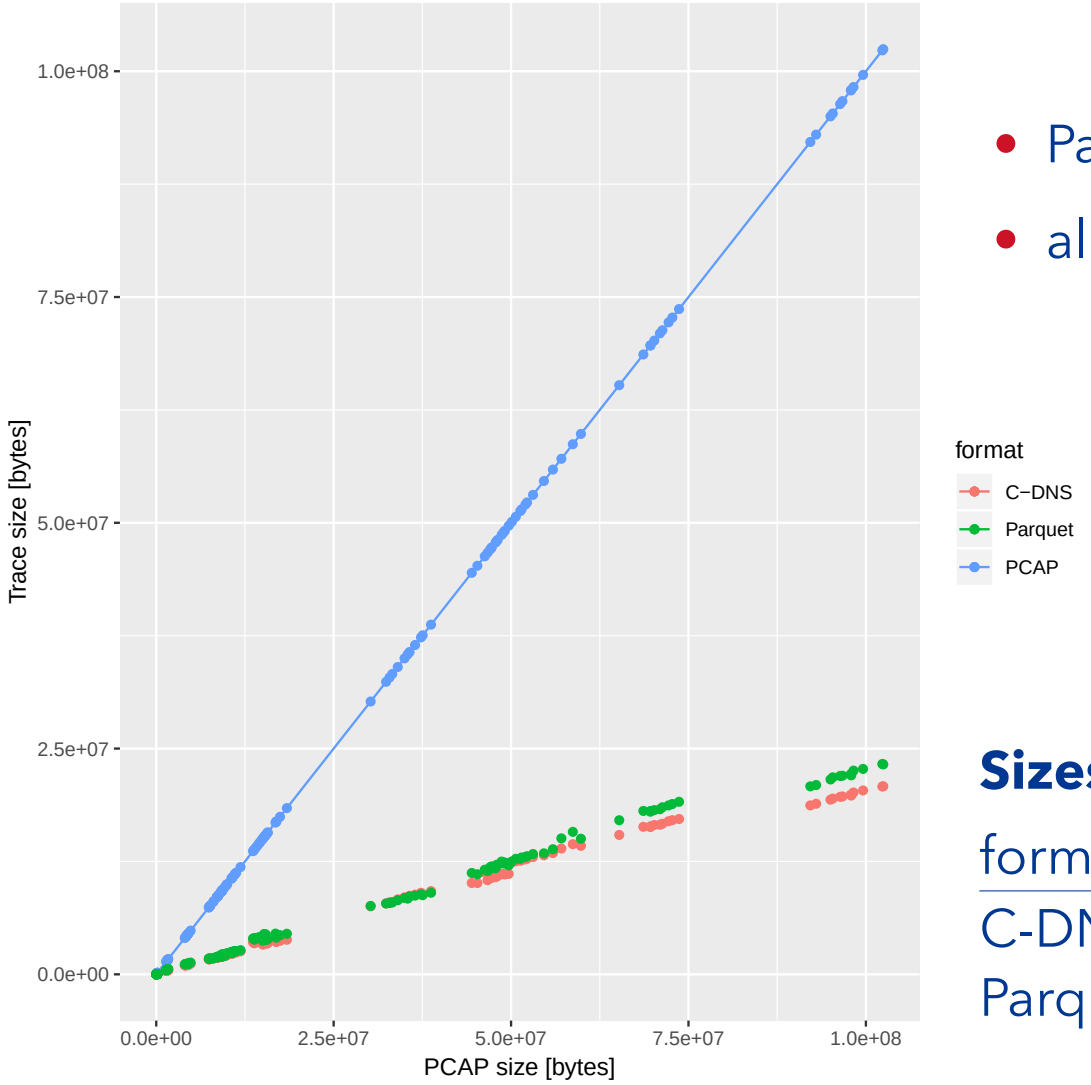
- data encoding: *Concise Binary Object Representation (CBOR)* [RFC 7049]
- variable block size; recommended: up to 10 thousand Q/R item
- variable resolution of time stamps
- properties stored in Q/R items can be selected (per block)
- built-in data compresion: common values are stored in block tables, Q/R items refer to them
- malformed messages may be included

PCAP samples



- DNS traffic from 15 .CZ servers
- 12 5-minute samples from each

C-DNS and Parquet Sizes



- Parquet and C-DNS: only Entrada
- all files xz-compressed (level 6)

Sizes relative to PCAP:

format	min	mean	max
C-DNS	20.16%	23.13%	27.90%
Parquet	12.32%	25.15%	44.13%

Conclusions

1. Both Parquet and C-DNS yield approx. 75% reduction in the effective size of traffic captures.
2. Compared to Parquet, C-DNS seems to achieve slightly better compression factors with less variance.
3. When choosing between Parquet and C-DNS, file size is probably not the most important factor.

CZ.NIC project: DNS Probe

- extract DNS transactions from network traffic or PCAP files (UDP&TCP)
- match DNS queries and responses
- generate C-DNS or Parquet
- store locally or send directly to a remote receiver
- configurable: selection of parameters to store
- optional DPDK support (https://www.dpdk.org/)
- side effect: C++ library for C-DNS

Questions?

Ladislav Lhotka • lhotka@nic.cz

CZ DOMAIN REGISTRY