dnstap: introduction and status update

Robert Edmonds (edmonds@fsi.io)
Farsight Security, Inc.

URL

- **http://dnstap.info**
  - Documentation
  - Presentations
  - Tutorials
  - Mailing list
  - Downloads
  - Code repositories

Introduction

- It's Protocol Buffers logging for DNS software.
- Schema file located here:
  - https://github.com/dnstap/dnstap.pb/blob/master/dnstap.proto

Protocol Buffers

- Natural fit for DNS data.
  - Binary clean.
  - Efficient encoding.
  - Extendable.
- Implementations available for many programming languages.
  - C, C++, Java, Python, Go, etc.

Schema

- Top-level **Dnstap** container message with fields:
  - **identity**: “NSID” analog.
  - **version**: “version.bind” analog.
  - **extra**: arbitrary annotation.
  - **type**: type of the contained message.
  - One of the following:
    - **message**: wire-format DNS message + metadata.
    - More possibilities to come.

Schema

- **Message** type encapsulates DNS wire-format messages.
  - **type**: AUTH_QUERY, AUTH_RESPONSE, RESOLVER_QUERY, RESOLVER_RESPONSE, ..., TOOL_QUERY, TOOL_RESPONSE
  - **socket_family**: INET, INET6
  - **socket_protocol**: UDP, TCP
  - **query_address, query_port**
  - **response_address, response_port**
  - **query_time_sec, query_time_nsec**
  - **query_message**
  - **query_zone**
  - **response_time_sec, response_time_nsec**
  - **response_message**

Framing

- Protobuf packs one payload at a time.
- How to pack a stream of many payloads?
- Solution: “Frame Streams”.
  - Write the payload length (32-bit integer).
  - Write the actual payload (variable length).
  - Repeat.

“Frame Streams”

- Lightweight protocol for streaming data frames.
  - Stream over a socket.
  - Or, read/write a file.
- Doesn't need to know how the data frames are encoded.
- Reference libfstrm implementation in C.
- Easy to parse. Python decoder is ~50 lines, no external dependencies.

Use cases

- These can all be accomplished with the **dnstap/Message** schema:
  - Interchange format for tools.
  - Passive DNS replication.
  - Query logging.

Interchange format

- Many tools send/receive DNS messages.
  - dig/delv(e), drill, kdig
  - looking glasses
- Immediately converted from DNS wire format to some other format.
  - Traditional “dig style”
  - JSON
  - ???

Interchange format

- Save a copy of the original DNS messages.
  - Display the message trace now **or** later.
  - Be able to refer to the original verbatim wire message, instead of whatever the tool printed to stdout.
- Looking glasses can communicate the exact response as received, rather than transcoding into, e.g. JSON.

Passive DNS replication

- Usually done by logging of authoritative responses to resolver initiated queries.
- Actually, instead of capturing the **responses**, the **packets containing the responses** are captured.
  - UDP responses may be spoofed.
  - IP fragments, TCP segments, UDP checksums...

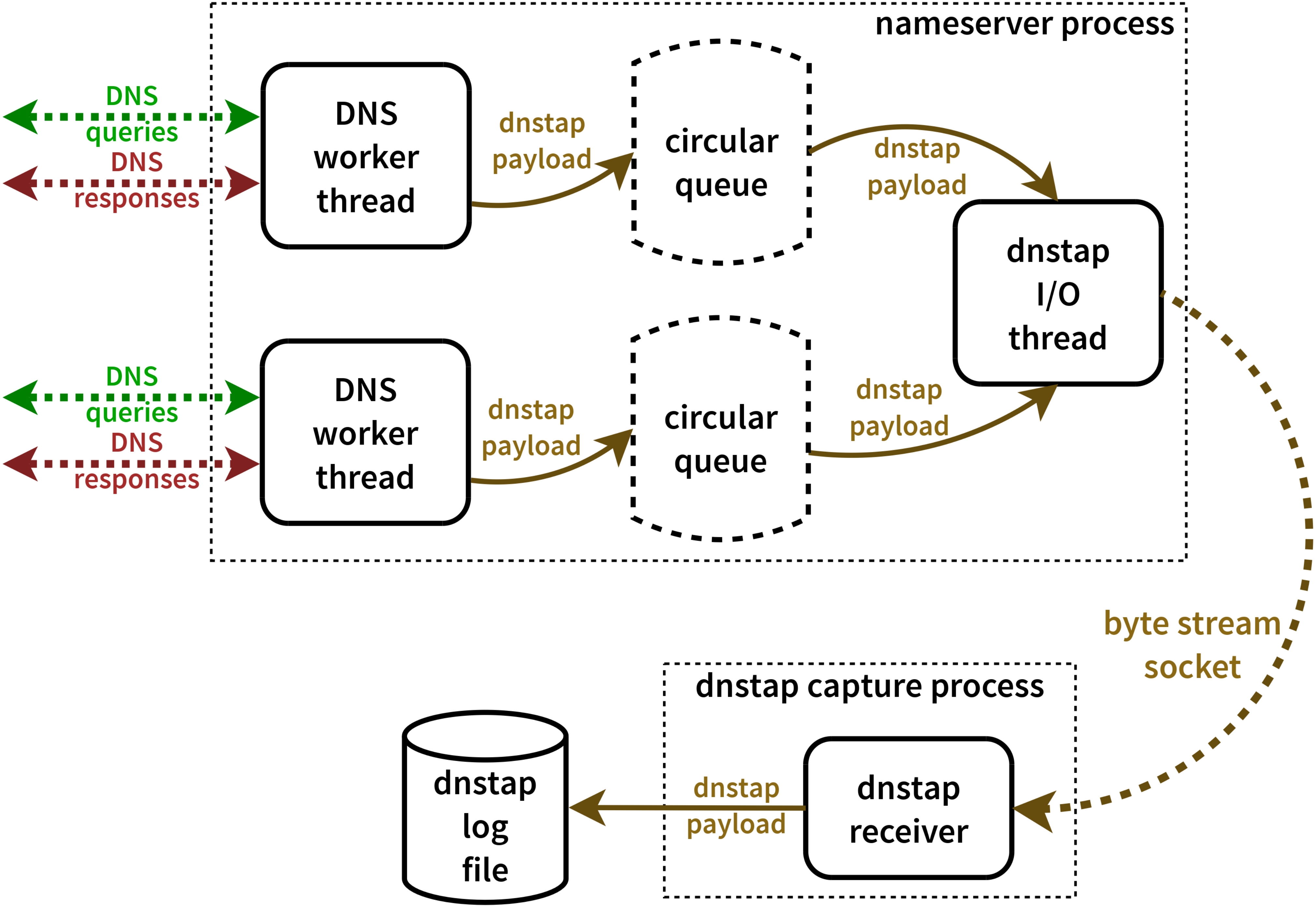
Passive DNS replication

- Because packet capture occurs outside of the DNS server, a critical piece of information is missing: the **bailiwick** of the transaction.
  - Must be laboriously reconstructed in order to avoid poisoning: “passive DNS bailiwick algorithm”.
- dnstap alternative: the DNS server can just log the needed information.

Query logging

- Log the queries the server receives.
- Metadata that would be nice to have:
  - Recursive case: whether the query hit a cache.
  - Authoritative case: which zone a query was served from.

dnstap-enabled DNS server



dnstap components

- Flexible, structured ***log format*** for DNS software.
  - dnstap.pb
- ***Helper libraries*** for adding support to DNS software.
  - libfstrm, libprotobuf-c
- Patch sets that ***integrate*** dnstap support into existing DNS software.
  - Unbound, Knot
- ***Capture tools*** for receiving dnstap messages from dnstap-enabled software.

Status update

- fstrm library under heavy development
- protobuf-c 1.0.0 release candidate
- Unbound patchset rebased against 1.4.22, almost complete
- Work on Knot/kdig patchset begun

URL

- **http://dnstap.info**
  - Documentation
  - Presentations
  - Tutorials
  - Mailing list
  - Downloads
  - Code repositories