Passive DNS Collection and Analysis The 'dnstap' (& fstrm) Approach

Farsight Security, Inc.

December 2014

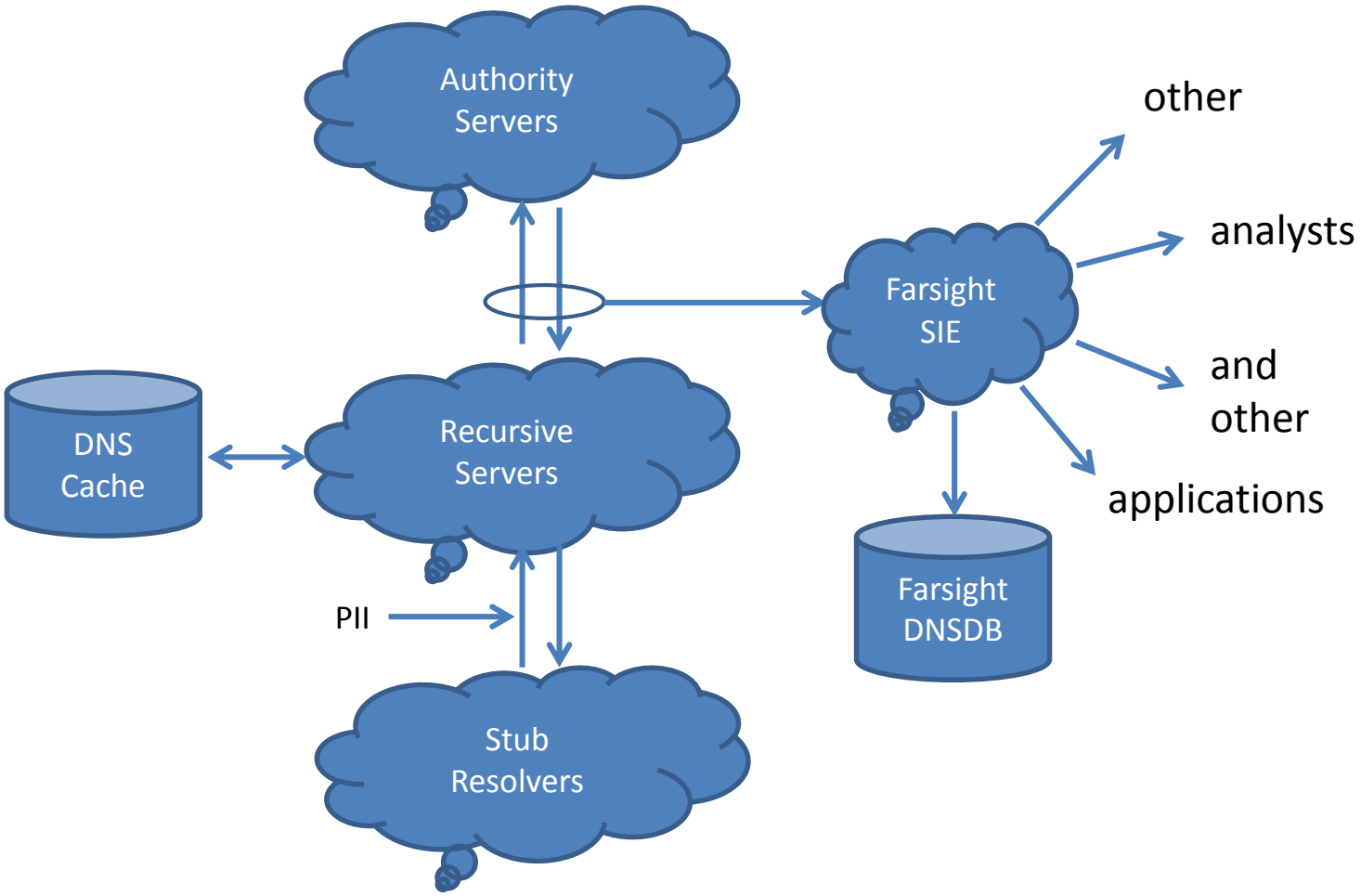
Importance of Measuring DNS

- High volume low latency datagram protocol
  - `Channel 202; 40 sources; 1,657,398,226,932 bytes/day; 153.463 Mbit/sec average rate.`
    `#1: 1,003,803,989,532 bytes,  97 sources (60%)`
    `#2:   272,389,753,714 bytes, 152 sources (16%)`
    `#3:   227,311,932,135 bytes,  54 sources (13%)`
- Enables almost all other network flows
  - A, AAAA, MX, NS, SRV records
- Traffic analysis: NetFlow vs. DNS
  - NetFlow tells you “what”
  - DNS tells you “why” (and “how”)

Challenges of Measuring DNS

- Historically, turning on logging in a DNS server slows it down to the speed of the file system
  - Operationally, measurement loss is always better
- So, success in DNS measurement has come from an asynchronous approach – BPF/pcap
  - NCAP (2006) – looked for authoritative responses, reassembling UDP datagrams as necessary (EDNS)
  - NMSG (2010) – like NCAP but has to see requests also, and then logs complete DNS transactions

Passive DNS Data Flow



Problems with NCAP/NMSG

- Blind to off-wire events like cache expiry due to DNS TTL, cache purge due to LRU
- Meaning is not tagged – NMSG receiver has to impute (“guess”) stub vs. cache miss query type, as well as transaction bailiwick
- Currently blind to TCP/53 – noting that there can be many transactions per TCP/53 session

Overload Handling Still Matters

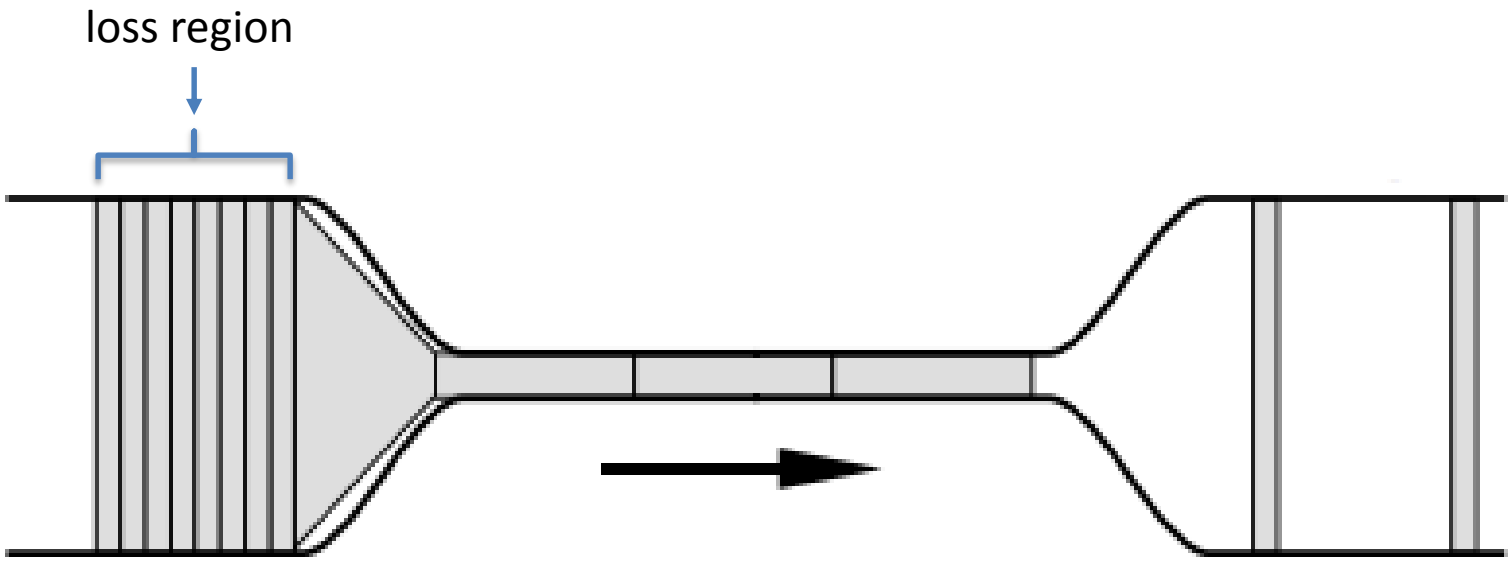


Diagram courtesy of Van Jacobson, 1995

Enter ‘dnstap’ (DNS Tap) & ‘fstrm’ (Frame Streams)

- ‘dnstap’ is server-embedded
- ‘fstrm’ has reliable front-loss
- Implementation has begun
- Deployment is commencing

‘dnstap’ – Server-Embedded

- ‘dnstap’ messages are generated from within DNS implementations, via instrumentation
  - No UDP fragment or TCP stream reassembly
  - No guessing the transaction bailiwick
  - No matching of on-wire queries with responses
  - No imputing stub vs. cache-miss query
- Encoded using Google Protocol Buffers
  - Fast, lean, open, high quality, de-facto standard

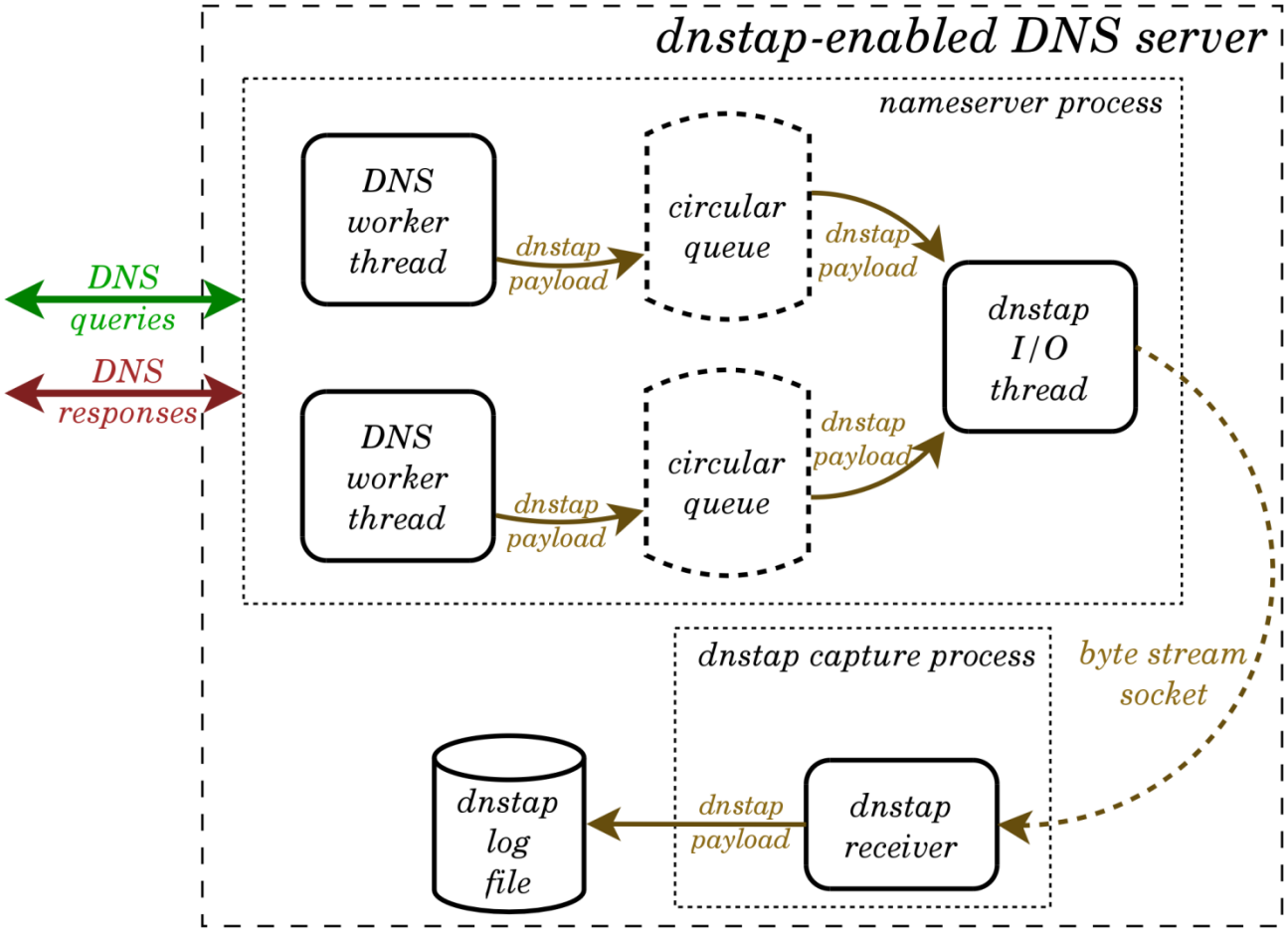
'dnstap' – Perspectives

- Messages can be annotated with off-wire information, e.g.:
  - Identity of the server, similar to NSID (for anycast)
- Messages are tightly bound to the role of the protocol agent who generates them
  - RESOLVER_QUERY and AUTH_QUERY are distinct in ‘dnstap’ but identical in BPF/pcap
- ‘dnstap’ is for *observation* not *eavesdropping*
  - Its use proves that an endpoint is cooperating

‘fstrm’ – Reliable Front-Loss

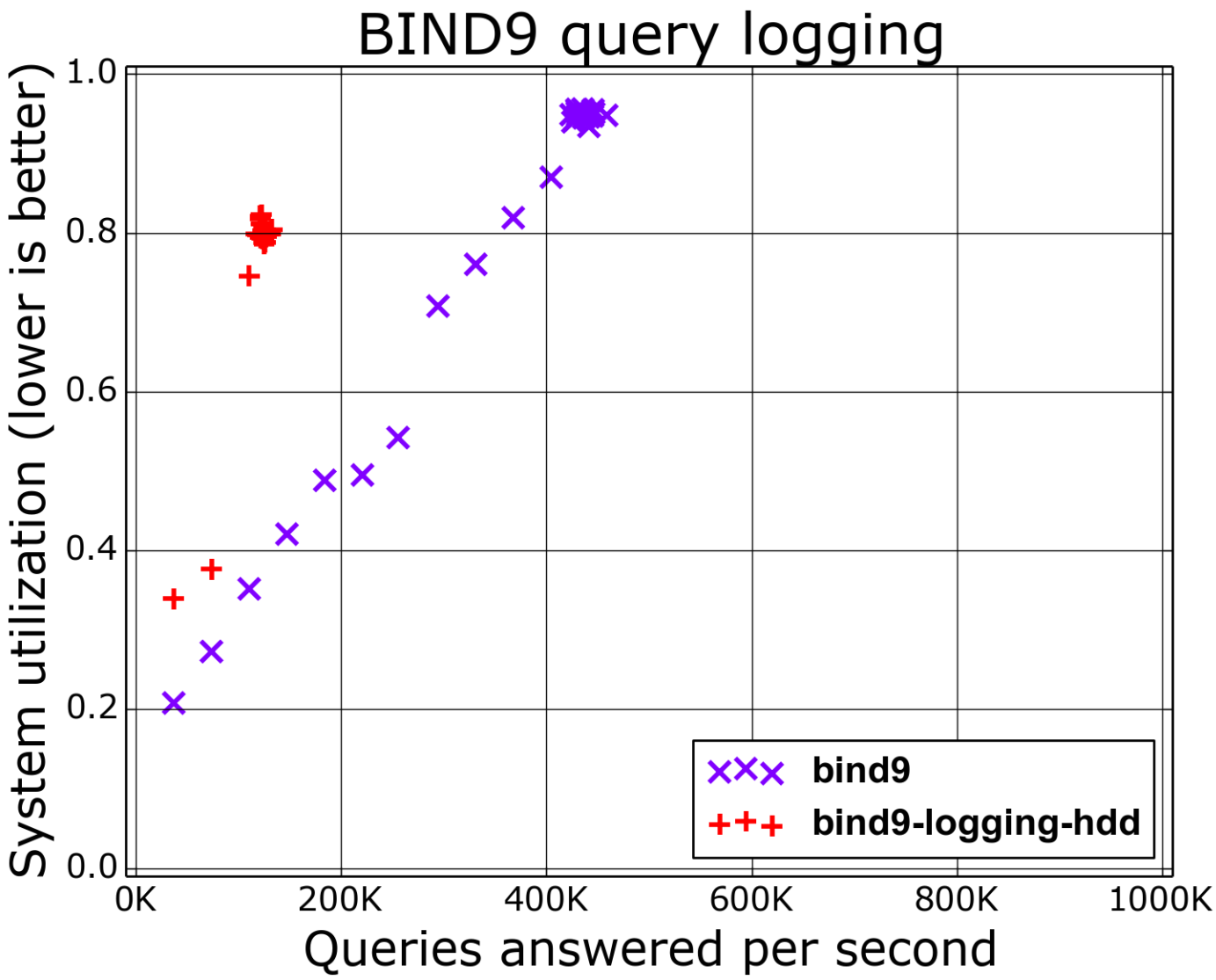
- TCP protocol vs. “BSD Sockets API”
  - Nonblocking UDP socket rejects full datagrams
  - Nonblocking TCP socket rejects overflow octets
    - Which breaks framing unless sender keeps state
- Solution: ‘fstrm’ writer thread
  - Lockless SP/SC ring buffer
  - ‘fstrm’ socket is blocking, so, thread can block
  - Reliable front-loss occurs when a ring buffer fills

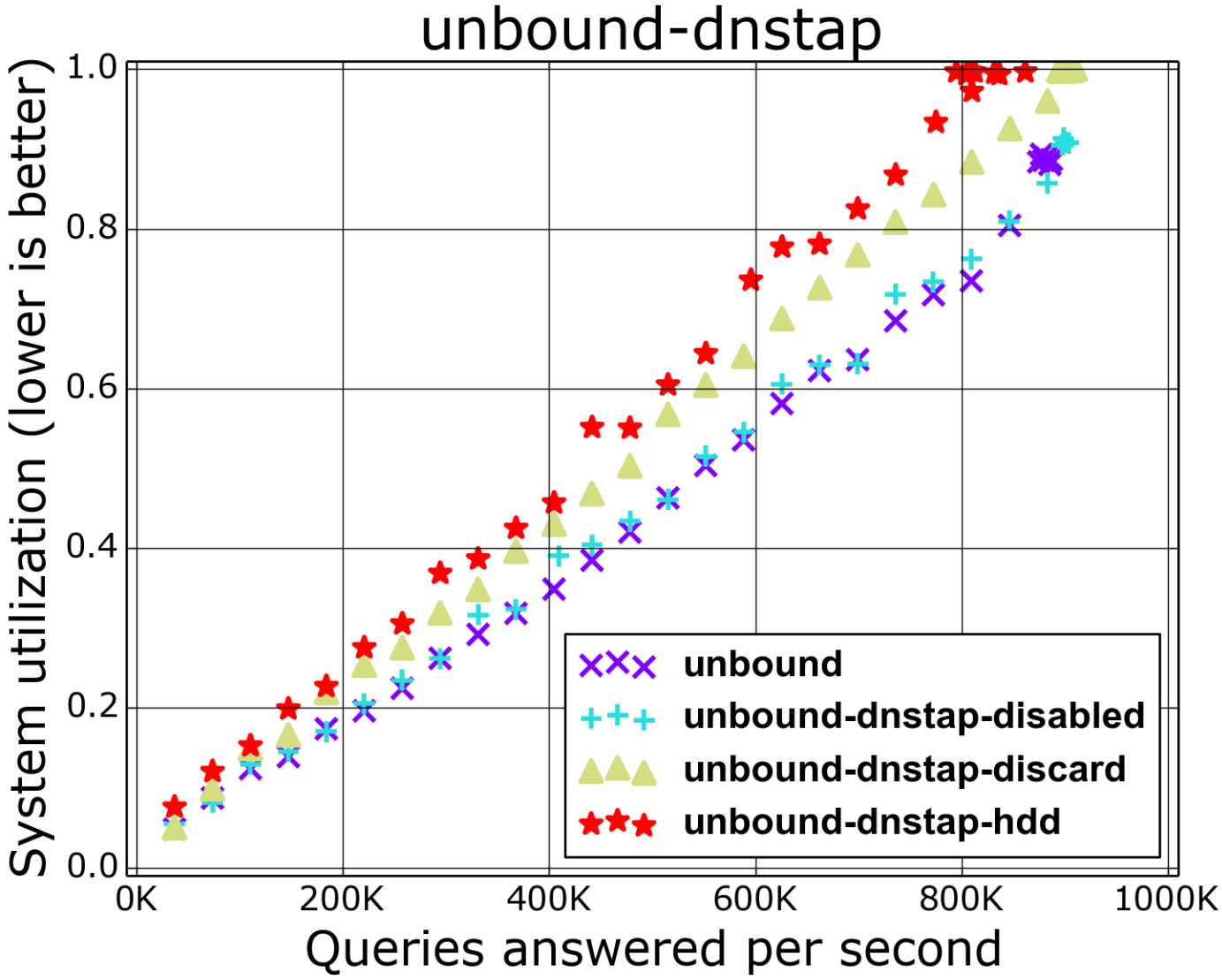
‘dnstap’ / ‘fstrm’ Architecture



‘dnstap’ – Message Types

- Present:
  - Stub {Query, Response}
  - Authoritative {Q, R}
  - Resolver {Q, R}
  - Client {Q, R}
  - Forwarder {Q, R}
- Prospective:
  - RRL bucket {Start, End}
  - Zone transfer in {S, E}
  - Zone transfer out {S, E}
  - Cache purge (LRU)
  - Cache expiry (TTL)


BIND9 query logging
System utilization (lower is better)
1.0
0.8
0.6
0.4
0.2
0.0
0K
200K
400K
600K
800K
1000K
Queries answered per second
bind9
bind9-logging-hdd


unbound-dnstap
System utilization (lower is better)
1.0
0.8
0.6
0.4
0.2
0.0
0K
200K
400K
600K
800K
1000K
Queries answered per second
unbound
unbound-dnstap-disabled
unbound-dnstap-discard
unbound-dnstap-hdd

Licensing/Packaging

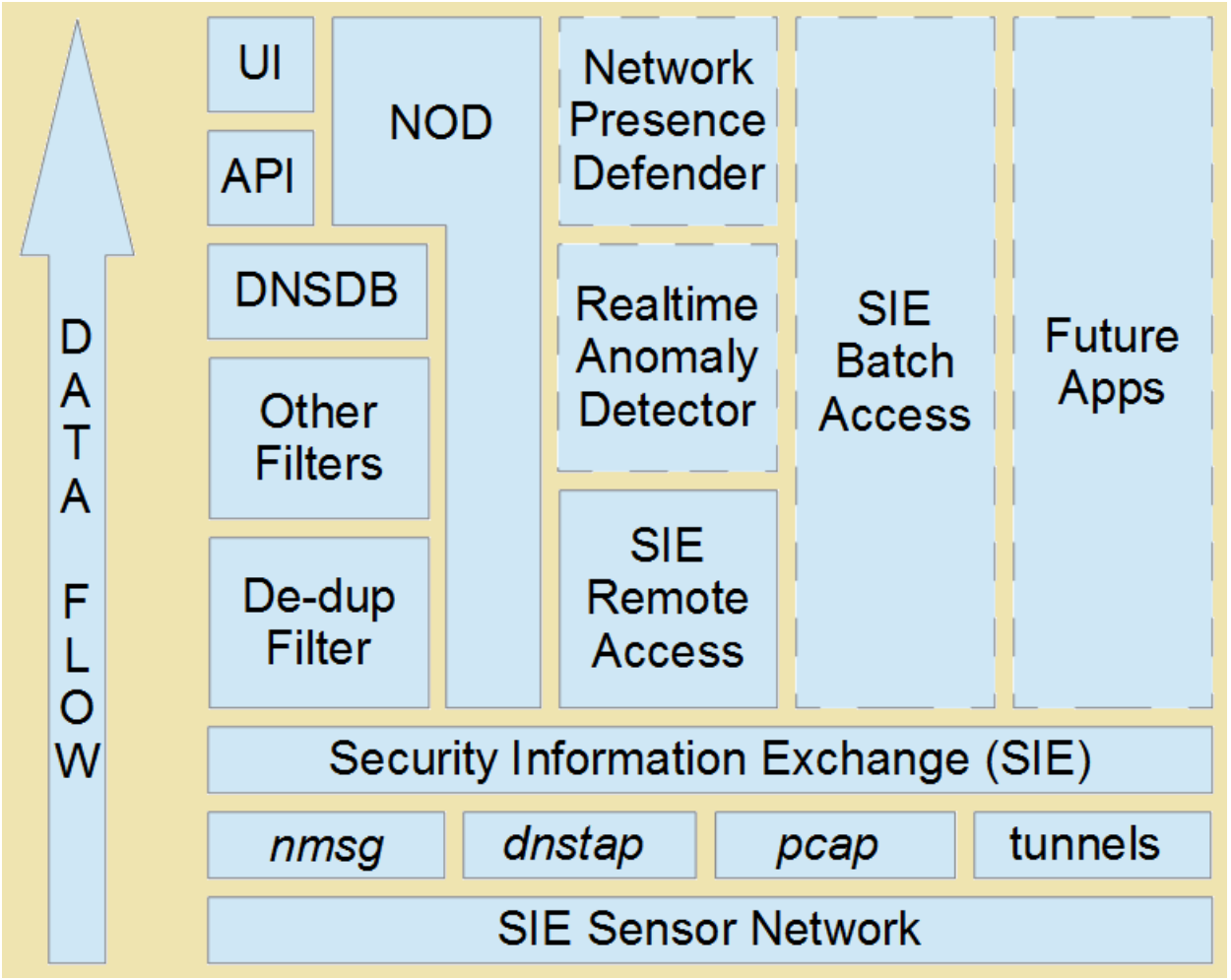
- Using Apache Open Source License V2.0
  - We loved BSD/ISC license, but AOSL2 is “better”
- Protocol, reference API, reference toolset
  - Working now in Unbound, Knot; BIND is next

- Our commercial interest is: wide adoption
  - So, it’s all on GitHub (see http://dnstap.info/)
- We intend to patch all F/L/OSS DNS servers
  - Eventually this should pressure Nominum and Microsoft to join the ‘dnstap’ ecosystem

Context of DNS Measurements

- Farsight SIE – Security Information Exchange
  - Commoditize security-relevant Internet telemetry
  - Channels for Passive DNS (raw, dedup'd, chaff, etc)
- Filtered output goes into DNSDB
  - Hierarchical MTBL (Google Sorted String Tables)
  - Contains all of SIE's DNS since June 2010
  - RESTful API with JSON output
- SIE and DNSDB are cash-free for nonprofit research/academia (pay us in data of like kind)

Passive DNS, SIE, DNSDB – Context



Demonstration

- DNSDB API
  - online `dnsdb_query` tool
- SRA
  - SIE Remote Access
- NOD
  - Newly Observed Domains

Summary

- Passive DNS collection (NCAP, NMSG, ‘dnstap’)
- Worked example: DNSDB, SRA, NOD
- More Information:
  - http://dnstap.info/
  - https://dnsdb.info/
  - https://api.dnsdb.info/
  - http://github.com/farsightsec
  - http://dnsrpz.info/