DNSFlow

Duane Wessels
The Measurement Factory/ISC/CAIDA
wessels@measurement-factory.com

July 25, 2005

Goals

- "NetFlow for DNS"
- Allow administrators to monitor DNS traffic in real-time
- Also useful for long-term archival storage of DNS traffic
- Use less bandwidth than just sending DNS packets
- Encryption
- Authentication
- Customizable (one size does not fit all)

Server-side

- Initially pcap-based
- Later built in to nameserver software?
- Some configuration (ACLs, keys)
- Support multiple clients
- Lightweight as possible

Client-side

- Real-time displays (both graphical and textual)
- Logging, archiving
- Configuration options
- Client specifies what, and how often, to report

Protocol

- Binary
- Encrypted (probably)
- Compressed (maybe)
- TCP
- Bi-directional hellos

DetailSpec

Client should be able to specify how much detail they require. For example, one client may only care about QNAMEs, while another may need to know the QNAME and QTYPE for each query. Clients will be able to specify the DNS message elements and attributes that they care about:

- QNAME
- QCLASS
- QTYPE
- Opcode
- Rcode
- Query ID
- AA, TC, RD, RA, etc bits
- QDCOUNT, ANCOUNT, NSCOUNT, ARCOUNT
- Question RR set
- Answer RR set
- Authority RR set
- Additional RR set
- Source IP address
- Destination IP address
- Source UDP/TCP port
- Destination UDP/TCP port
- Message size

DetailSpec

- Clients can give the server multiple DetailSpecs
- Clients can add and remove DetailSpecs to/from an active session
- Maybe one DetailSpec per TCP connection?

Filters

Clients should be able to specify filters. DNS messages that don't match the filter are not reported. For example, a client may choose to receive only messages where QCLASS != IN, or where QNAME ends with .ORG.

Mappers?

May also be useful to have "mappers." The purpose of a mapper is to turn large sets into smaller ones. For example, a mapper can turn QNAMEs into TLDs, or the set of all QTYPEs into a set of common QTYPEs.

However, implementing mappers on the server-side may add too much complexity and, therefore, conflict with the goal for a lightweight implementation.

Aggregation Delay

Clients should be able to specify an "aggregation delay." Aggregation allows the protocol to represent multiple occurrences of the same element set with a frequency counter instead of transmitting the same element set more than once. Longer aggregation delays require less bandwidth. Minimum aggregation delay should be 1 second.

For example, the server may say: I found 20 DNS messages with QTYPE = AAAA and QNAME $\sim$ /[a-z].root-servers.net/ in the last aggregation interval.

Unresolved

- Should queries and replies be combined into a single "flow" message?
  - Requires a timeout for servers that don't reply.
  - Replies include most of the question anyway.
- Message format details

The End