A question of protocol

Geoff Huston

APNIC 36

Originally there was RFC791:

“All hosts must be prepared to accept datagrams of up to 576 octets (whether they arrive whole or in fragments). It is recommended that hosts only send datagrams larger than 576 octets if they have assurance that the destination is prepared to accept the larger datagrams.”

For:

- 20 bytes of IP header,
- <= 40 bytes of IP options and
- 8 bytes of UDP header,

that leaves a maximum of 512 bytes in a packet that will be accepted by any IP host.

Then RFC1123:

> ... it is also clear that some new DNS record types defined in the future will contain information exceeding the 512 byte limit that applies to UDP, and hence will require TCP. Thus, resolvers and name servers should implement TCP services as a backup to UDP today, with the knowledge that they will require the TCP service in the future.

The original DNS model

If the reply is <= 512 bytes, send a response over UDP

If the reply is > 512 bytes, send a response over UDP, but set the TRUNCATED bit

- Which should trigger a re-query using TCP

EDNS0

RFC2671:

```
4.5. The sender's UDP payload size (which OPT stores in the RR CLASS
     field) is the number of octets of the largest UDP payload that can
     be reassembled and delivered in the sender's network stack.  Note
     that path MTU, with or without fragmentation, may be smaller than
     this.
```

The sender can say to the resolver: “Its ok to send me DNS responses using UDP up to size <xxx>. I can handle it.”

So we started using DNS over UDP for larger queries, and stopped performing failback to TCP so readily. Commonly, we see a 4096 packet reassembly buffer being announced in EDNS0 queries.

Which leads to…

DNS Attacks

Send a small query in UDP to a DNS resolver

- Set EDNS0 to a large value
- Use the IP address of the intended victim as the source address
- Use a query that generates a large response in UDP
  - ISC.ORG IN ANY, for example
- 10x – 100x gain

Mix and repeat with a combination of a bot army and the published set of open recursive resolvers

Possible Mitigation…?

1) Get everyone to use BCP38

2) Use a smaller EDNS0 max size

So lets look at 2):

- This would then force the query into TCP
- And the TCP handshake does not admit source address spoofing

Could this work?

- How many customers use DNS resolvers that support TCP?
  - Lets find out…
    - Nurgle a DNS server with the EDNS0 max size set to 275
    - Set up an ad with a short (<275) and a long (>275) DNS name response
    - And see who can resolve the short and fails on the long

Numbers

Clients

Experiments:	2,033,535
Truncated UDP Responses:	2,032,176
TCP Queries:	1,978,396
Drop Off:	53,780

That's 2.6%

Numbers, Numbers

Resolvers

- Total Seen: 80,505
- UDP only: 13,483

17% of resolvers cannot ask a query in TCP following receipt of a truncated UDP response

6.4% of clients uses these resolvers

- 3.8% of them failover to use a resolver that can ask a TCP query
- 2.6% do not

If we really want to use DNS over TCP

Then maybe its port 53 that’s the problem for these 17% of resolvers

Why not go **all** the way?

- How about DNS over XML over HTTP over port 80 over TCP? 😈