Update on experimental BIND features to rate-limit recursive queries

OARC Spring 2015 – Cathy Almond, ISC
What is this talk about?

- Random DNS query attacks against specific domains – a (very) quick recap
- Mitigation approaches
- Results from production environments
- Future thoughts/ideas/plans
The attack

- Attack is directed at DDOSing DNS authoritative provider, but incidentally degrades ISP resolvers in the path
- Higher query loads than usual
- Non-responding authoritative servers (directly filtering the resolvers, or simply overwhelmed)
- Increased network traffic levels
Identifying an attack

high volume of queries for non-existant sub-domains

<randomstring>.www.example.com
<anotherstring>.www.example.com

does not exist

exists
The source

- Open resolvers
  - your servers
  - your clients (CPE devices/proxies and forwarders)
- Compromised clients (botnets)
- Compromised devices
Symptoms

- Increased inbound client query traffic
- Increased outbound NXDOMAIN and SERVFAIL responses
- Resolution delays to clients
- Dropped responses
- Increased memory consumption
- Firewall connection table overflows
Evidence

- Backlog of recursive client queries
  – which queries are in the backlog?
  – is there a pattern?
  – originating from few or many clients?
- Open outbound sockets
  – to which servers; is there a pattern?
- Query logging / query-errors logging
- Network packet traces
First steps

- Eliminate open resolvers
  - is yours an open resolver?
  - open client CPE devices?
  - open resolvers forwarding to yours?

- Investigate compromised/infected clients
  - potentially several device types
  - source addresses may be spoofed
  - block spoofed addresses (internal)
What doesn’t help?

- Increasing server resources (e.g. recursive client contexts, sockets, network buffers etc..)
- Blocking clients (without investigating them properly first)
  - Some exceptions to this
Not enough...
MITIGATION TECHNIQUES

What can we do?

What has been tried in production?

What do we want to achieve?
Stage 1: Lie!

- Make recursive server temporarily authoritative for the target domain
  - Local zone
  - DNS-RPZ (*qname-wait-recurse no;)
- Manual configuration change
- Need to undo the mitigation afterwards
- Responds NXDOMAIN to all queries
Stage 2: Automate filtering

(Near) Real Time Block Lists

- Detect ‘bad’ domain names or just the problematic queries & filter them at ingress to the resolver
- Local auto-detection scripts
- Nominum Vantio
- BIND DNS-RPZ
- Costs associated with feeds
- Potential false-positives
Stage 3: Tune your resolver

Respond SERVFAIL without waiting to timeout
Fetches-per-server

Monitor responses vs timeouts

Adjust throttle

Throttle back queries

Monitor responses vs timeouts
fetches-per-server

- Per-server quota dynamically re-sizes itself based on the ratio of timeouts to successful responses.
- Completely non-responsive server eventually scales down to fetches quota of 2% of configured limit.
- Similar (loosely) in principle to what NLnet Labs is doing in Unbound.
fetches-per-server

- default tuning:
  *fetch-quota-params* 100 0.1 0.3 0.7;
  - Recalculate fetch quota every 100 queries
  - 10% or below timeout – raise threshold
  - 30% of above timeouts – reduce threshold
  - 70% weighting given to recent counting period when computing timeout ratio
**fetches-per-zone**

- Works with unique clients
- Default 0 (no limit enforced)
- Tune larger/smaller depending on normal QPS to avoid impact on popular domains
fetches-per-zone at Jazztel

Spanish triple-play ADSL carrier & ISP
Roberto Rodriguez Navio, Jazztel Networking Engineering
used with permission
More on fetches per zone

Spanish triple-play ADSL carrier & ISP
Roberto Rodriguez Navio, Jazztel Networking Engineering
used with permission
fetches-per-server

DNS Recursive Queries vs Servfail Last 12 Days

- SERVFAIL Responses: 138.4 (cur) : 314.0 (max) : 1.4 (min) : 146.3 (avg)
- Recursive Clients: 18.5 (cur) : 2073.7 (max) : 8.0 (min) : 295.9 (avg)

Updated: 12-Feb-2015 08:59:19
per-zone v. per-server

DNS Recursive Queries vs Servfail Last 48 Hours

- SERVFAIL Responses: 2270.9 (cur), 4876.3 (max), 246.2 (min), 1183.8 (avg)
- Recursive Clients: 524.3 (cur), 1169.6 (max), 241.7 (min), 603.1 (avg)

Updated: 04-Mar-2015 12:47:15

UDP Statistics Last 48 Hours

- UDP In Datagrams: 21.2k (cur), 22.6k (max), 6.9k (min), 14.5k (avg)
- UDP Out Datagrams: 21.0k (cur), 21.8k (max), 6.9k (min), 14.1k (avg)
- UDP In Errors: 27.7m (cur), 963.6m (max), 9.4m (min), 76.3m (avg)

Updated: 04-Mar-2015 12:47:29

© 2015 ISC
What will the user see?

- Situation normal – no change to their usual experience (for most)
- (Some) SERVFAIL responses to names in zones that are also served by under-attack authoritative servers (collateral damage)
- NXDOMAIN responses for names in legitimate zones for which we ‘lie’
But not yet perfect...
But not yet perfect…
But not yet perfect...
More ideas…

- SERVFAIL or drop (or NXDOMAIN)?
- Whitelists may be needed
- Per-server/zone override settings
- SERVFAIL cache (for client retries)
- Improved reporting & statistics
- Built-in ‘auto-DNS-RPZ’
- Persistent (non-expiring) RRsets (for ‘good’ answers)
Summary of techniques

1) Clean up your network
   eliminate open resolvers & compromised clients; look at BCP 38

2) Configure your resolver to lie
   answer authoritatively yourself; potentially automate your blacklist or subscribe to a feed for this.

3) Consider adaptive quotas
   per server; per zone
   (Good feedback on these from many sources)
QUESTIONS?

bind-suggest@isc.org, cathya@isc.org