When the Dike Breaks: Dissecting DNS Defenses During DDoS

Giovane C. M. Moura[1,2], John Heidemann[3],
Moritz Müller[1,4], Ricardo de O. Schmidt[5], Marco Davids[1]
giovane.moura@sidn.nl

[1]SIDN Labs, [2]TU Delft, [3]USC/ISI,
[4]University of Twente, [5]University of Passo Fundo

IETF 102 - MAPRG - Montreal, Canada

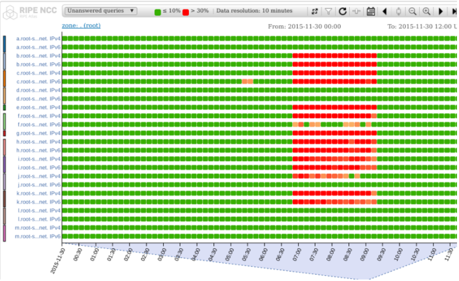
paper: https://www.isi.edu/~johnh/PAPERS/Moura18a.pdf

DDoS Attacks

- DDoS attacks are on the rise
- Getting bigger, more frequent, cheaper, and easier
  - Arbor: 1.7 Tb/s [2] (2018)
  - Github DDoS: 1.35 Tb/s [1] (2018)
  - Dyn DDoS: 1.2 Tb/s (Mirai IoT) [5] (2017)
  - DDoS as a service: few dollars with booters [7].
- The DNS is a juicy target
- Many DNS services have been victim of DDOS attacks
- You can do filtering, scrubbing, etc
  - But DNS has many built-in features to operate under stress
  - Tons of work at the IETF
  - We investigate those in the paper: the built-in robustness of DNS

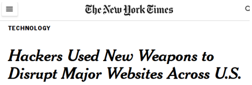
DDoS and DNS: two examples

Root DNS DDoS Nov 2015



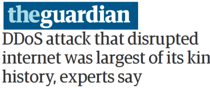
no known reports of errors seen by users [3]

Dyn Oct 2016

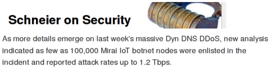

The New York Times

TECHNOLOGY

Hackers Used New Weapons to Disrupt Major Websites Across U.S.


theguardian

DDoS attack that disrupted internet was largest of its kin history, experts say


Schneier on Security

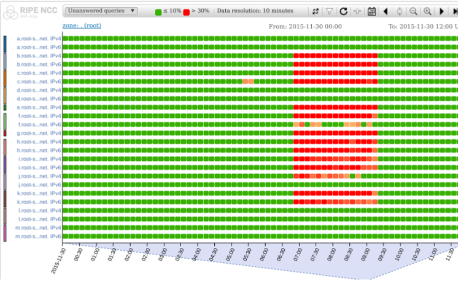
As more details emerge on last week's massive Dyn DNS DDoS, new analysis indicated as few as 100,000 Mirai IoT botnet nodes were enlisted in the incident and reported attack rates up to 1.2 Tbps.

some users could not reach popular sites [5]

Two large DDoSes, very different outcomes. Why?

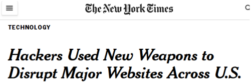
DDoS and DNS: two examples

Root DNS DDoS Nov 2015



no known reports of errors seen by users [3]

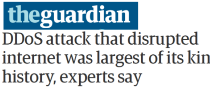
Dyn Oct 2016



The New York Times

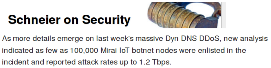
TECHNOLOGY

Hackers Used New Weapons to Disrupt Major Websites Across U.S.



theguardian

DDoS attack that disrupted internet was largest of its kin
history, experts say



Schneier on Security

As more details emerge on last week's massive Dyn DNS DDoS, new analysis indicated as few as 100,000 Mirai IoT botnet nodes were enlisted in the incident and reported attack rates up to 1.2 Tbps.

some users could not reach popular sites [5]

Two large DDoSes, very different outcomes. Why?

What Accounts for Different Outcomes?

- What factors affect the DNS **user experience**?
- When does DDoS cause “no change” vs. “sporadic problems”?
- Common knowledge: recursives **caching** and **retries** help?
- Can we quantify how much and when?
- Can DNS operators and purchasers of their services improve?

Background: the many parts of DNS

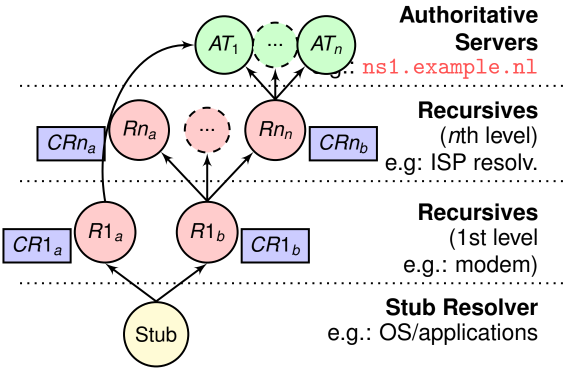


Figure: Relationship between stub resolver (yellow), recursive resolvers (red) with their caches (blue), and authoritative servers (green).

Important: Auth servers set TTL of DNS records → max value for recursives keep a record in cache

So, can we evaluate DNS built-in resilience?

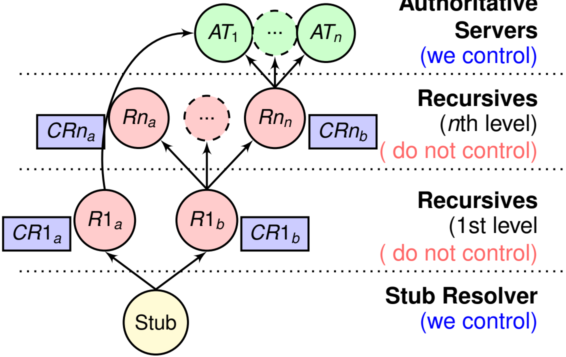
- **Part 1**: evaluate user experience under “normal” operations
  - learn about how much is cached/retried in a controlled env.
- **Part 2**: Verify results of part 1 in production zones (`.nl`)
- **Part 3**: Emulate DDoSes in the wild to evaluate caching/retrials under stress, **to observe user experience**

Part 1: measuring caching in the wild

Setup:

- register our new domain (`cachetest.nl`)
- run two unicast IPv4 authoritatives on EC2 Frankfurt
  - we do not analyze anycast auth in this work
- User Ripe Atlas and their resolvers as vantage points ($\sim$ 15k)
- Each VP sends a unique query, so no interference other
  - e.g.,: `500.cachetest.nl` for probeID=500
- Each DNS answer encodes a counter that allow us to tell if it was cache hit or miss (see paper)
- we probe every 20min (1200s), and run scenarios with different TTLs, for 2 to 3 hours
  - 60, 1800,3600, and 86400 seconds TTL for each answer

Part 1: measuring caching in the wild



Authoritative Servers
(we control)

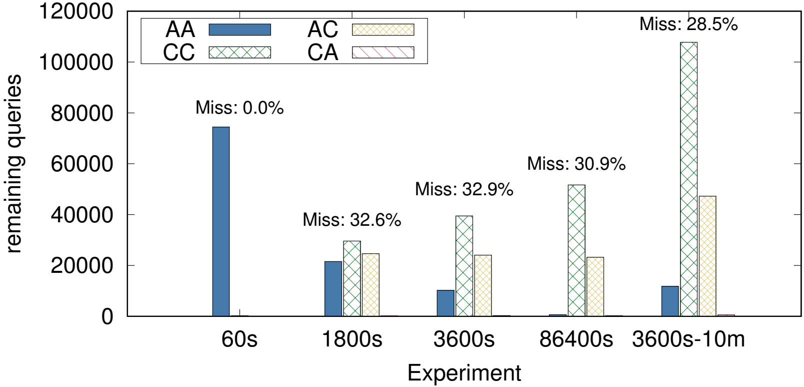
Recursives
(*n*th level)
(do not control)

Recursives
(1st level
(do not control)

Stub Resolver
(we control)

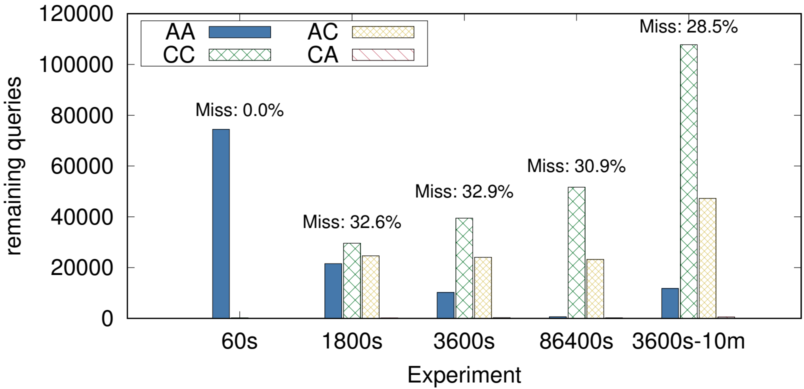
- We control auth severs and clients (stub resolver)
- How efficient is caching in the wild?

Results: how good caching is in the wild?



1. Good news: caching works fine for 70% of all 15,000 VPs
   - With our not popular domain
2. Not so good news: ∼ 30% of cache misses (AC)

Results: how good caching is in the wild?



1. Good news: caching works fine for 70% of all 15,000 VPs
    - With our not popular domain
2. Not so good news: ∼ 30% of cache misses (AC)

Why cache misses (Why AC?)

Possible: capacity limits, cache flushes, complex caches
Mostly: complex caches

- cache fragmentation with multiple servers
- (previous work on Google DNS [8])

TTL	**60**	**1800**	**3600**	**86400**	**3600-10m**
AC Answers	37	24645	24091	23202	47,262
Public R_1	0	12000	11359	10869	21955
Google Public R_1	0	9693	9026	8585	17325
other Public R_1	0	2307	2333	2284	4630
Non-Public R_1	37	12645	12732	12333	25307
Google Public R_n	0	1196	1091	248	1708
other R_n	37	11449	11641	12085	23599

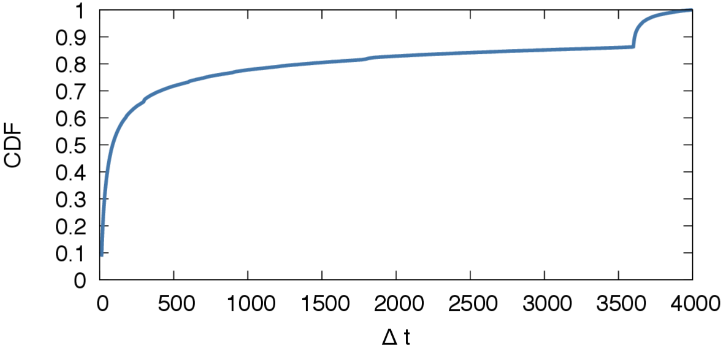
Table: AC answers public resolver classification.

Part 2: caching in production zones

- OK, in our controlled environment, we show that caching works 70% as expected
- Are these experiments representative?
- We look at `.nl` data
  - we compute Δt (time since last query)
  - Compare to TTL of 3600s
  - 485k queries from 7,779 recursives

Part 2: caching in production zones

- Two main peaks: start and 3600 (TTL of the record)
- First: happy eye ball (not related to cache) , second yes
- **Yes, experiments are like real zone**
- (we also look into the roots , see paper)

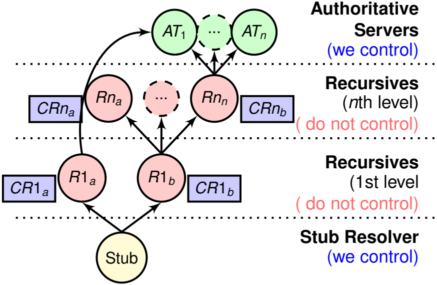


OK , so far, what do we have?

- **We know how caching works in the wild**
- Time to move Part 3: emulate DDoS to evaluate DNS built-in resilience
- Goal: understand client experience under DDoS

Part 3: Emulating DDoS

- Similar setup as other experiments:
  - Two NSes on EC2 (Frankfurt)
  - 15,000 Vantage Points (Ripe Atlas)
  - Emulate DDoS: drop incoming queries at certain rates at Authoritative servers, with `iptables`
- Question: (when) do caches protect clients?
- Or why some DDoS attacks seem to have more impact?



Complete DDoS: TTL: 60min, 100% failure

- This is **doomsday** for DNS ops: all auth DNS down
- How much cache can protect? For how long?

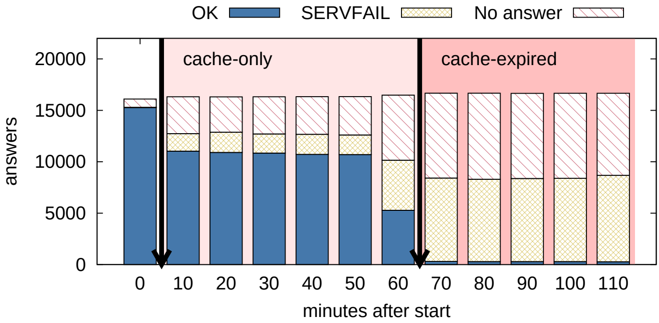


Figure: Scenario A: 100% failure after 10min, TTL: 60min

- DDoS starts after 1st query (fresh cache)
- During DDoS: **35%-70% of clients are served**,from cache
- After cache expires: only 0.2% clients served (serve state)
  - `draft-ietf-dnsop-serve-stale-00`

Complete DDoS: TTL: 60min, 100% failure

- This is **doomsday** for DNS ops: all auth DNS down
- How much cache can protect? For how long?

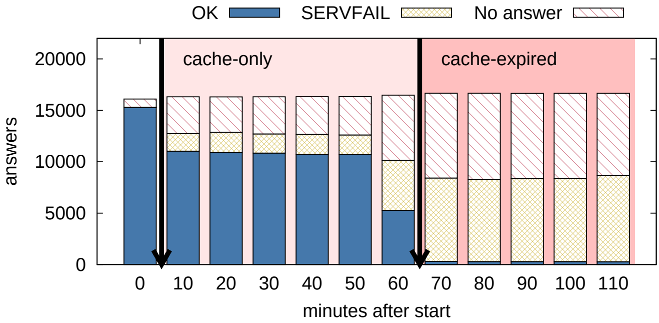


Figure: Scenario A: 100% failure after 10min, TTL: 60min

- DDoS starts after 1st query (fresh cache)
- During DDoS: **35%-70% of clients are served**,from cache
- After cache expires: only 0.2% clients served (serve state)
  - `draft-ietf-dnsop-serve-stale-00`

Complete DDoS: changing cache freshness

- Carrying on with more Doomsday
- Scenario B: Cache freshness: about to expire
- How clients will experience DDoS?

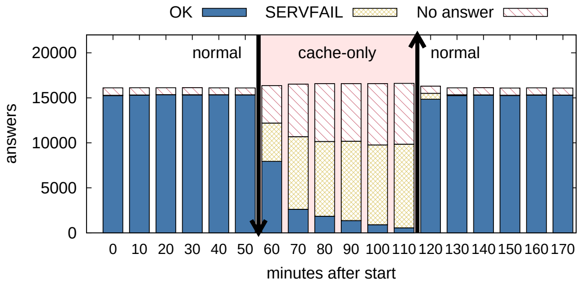


Figure: Scenario B: 100% failure after 60min, TTL: 60min

- Cache much less effective (as time out near attack)
- Fragmented cached helps some (by filling later)

Complete DDoS: changing cache freshness

- Carrying on with more Doomsday
- Scenario B: Cache freshness: about to expire
- How clients will experience DDoS?

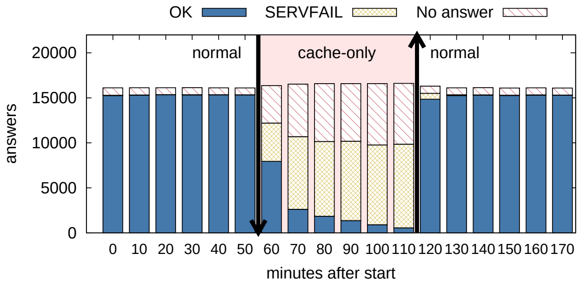


Figure: Scenario B: 100% failure after 60min, TTL: 60min

- Cache much less effective (as time out near attack)
- Fragmented cached helps some (by filling later)

Complete DDoS: TTL record influence

- Influence of TTL: reducing from 60min to 30min
- How clients will experience DDoS?

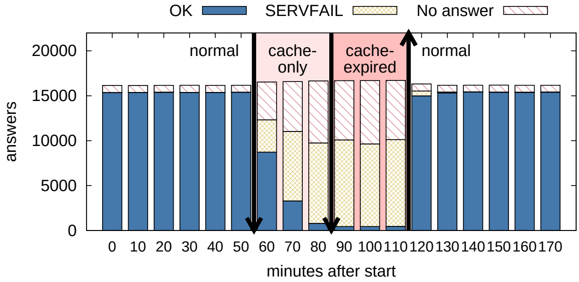


Figure: Scenario C: 100% failure after 60min, TTL: 30min

- Users experience worsens a lot with shorter TTL
- OPs: choose wisely the TTL of your records when engineering for DDoS

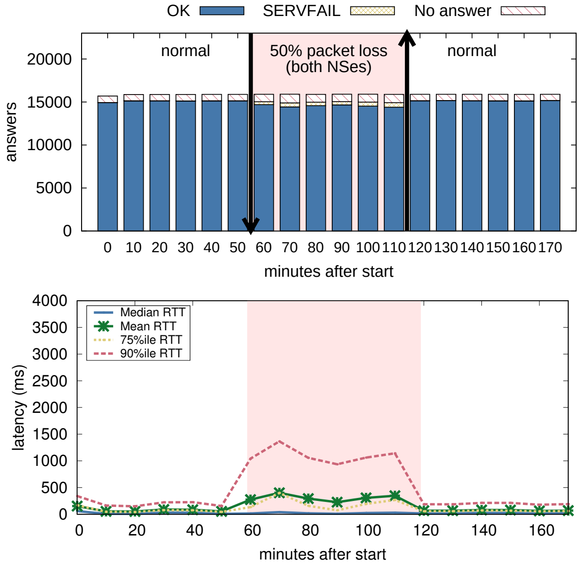
Discussion complete DDoS and user experience

- Caching is partially successful during complete DDoS
- OPs: don’t expect protection for clients as long as your TTL; depends on their cache state (even pop domains)
- Serve stale provides the last resort for Doomsday scenario
  - some ops (Google, OpenDNS) seem to do it, but it is not widespread yet
- TTL of records: the shorter you set them, the less you protect users during a complete DDoS

Partial DDoS

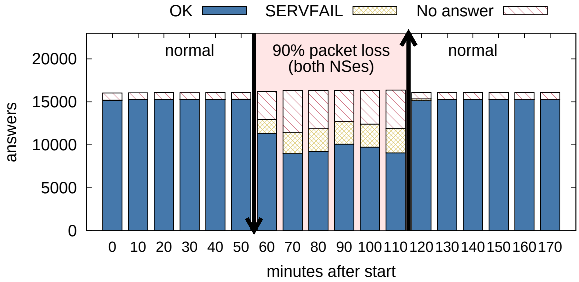
- Not all DDoS are a complete success;
- Some lead to partial failure (Root DNS Nov 2015 [3])
  - Partial failure: some of the available authoritative fail to answer all queries, or take longer to answer; then users experience longer latencies
- In this case, how would users experience the attack?

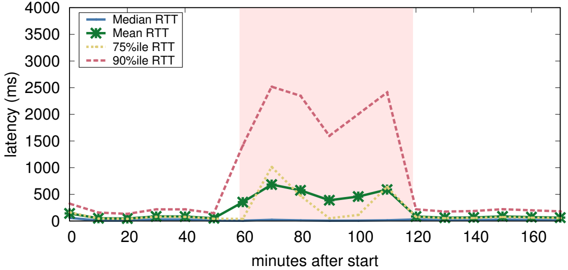
Experiment E: 50% success DDoS, TTL: 30min



Good! Most clients are happy, as they retry (but takes longer)

Experiment H: 90% success DDoS, TTL: 30min

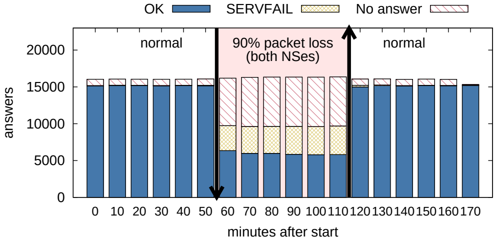


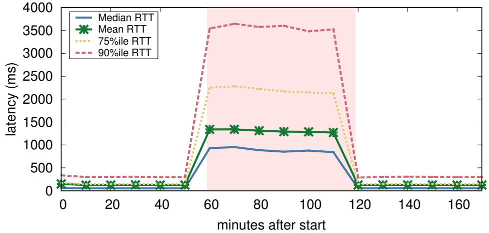


Good! Even at 90% packet loss with TTL 30min, most clients (60%) get an answer!! **Thanks IETFers! Good Engineering!**

Experiment I: 90% success DDoS, TTL: 1min

- What's TTL influence in partial DDoS?





Even with no caching (TTL 1min), 27% get an answer: stale + retries

Retries cost: hammering Auth servers

- Part of DNS resilience is that recursives keep on trying to resolve
- There's a cost to it however: 8.1x in case of no caching!
- Implications: OPS: be ready for friendly fire
  - usually not noticed during DDoS
  - If you overprovision level is 10x, imagine that 8.1x extra is only for friendly fire

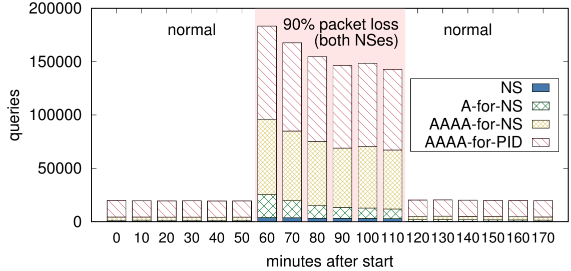


Figure: Queries received at Auth Servers .Experiment I: 90% success DDoS, TTL: 1min

Implications

- Caching and retries work *really well*
  - provided some authoritative stays partially up
  - and caches last longer than DDoS (as in TLS, not in CDNs)
  - For OPs: make one auth very strong? (careful with load distrubtion, see [4])
- Explains prior root DDoS outcomes
- There is a clear **trade-off** between TTL and DNS resilience
  - provided caches are filled and not about to expire
  - But enable quicker changes in the DNS (Amazon EC2 resolvers cap all answershorter TTLss to 60s [6])
- Many commercial websites have short TTLs
  - explains the pain of Dyn‘s customers and users perception
  - shorter TTLs given them quicker management options

Conclusions

- Caching and retries are important part of DNS resilience
  - Good engineering: thanks for all IETFers/devs who have build this
- Experiments show when they help and when they won't
  - No more "it will be in cache , no problem" assumption
- Consistent with recent outcomes
- DNS community:
  - There's a clear trade-off between TTL and DDoS robustness, choose wisely
  - Shall we advocate for serve-state deployment ?
  - `draft-ietf-dnsop-serve-stale-00`

Questions?

- Tech report: https://www.isi.edu/~johnh/PAPERS/Moura18a.pdf
- Contact: giovane.moura@sidn.nl
- Thanks RIPE NCC and reviewers of various drafts:
  - Wes Hardaker, Duanne Wessels, Warren Kumari, Stephane Bortzmeyer, and Maarten Aertsen

References I

[1] Sam Kottler.

February 28th DDoS Incident Report | Github Engineering, March 2018.

. https://githubengineering.com/ddos-incident-report/.

[2] Carlos Morales.

February 28th DDoS Incident Report | Github EngineeringNETSCOUT Arbor Confirms 1.7 Tbps DDoS Attack; The Terabit Attack Era Is Upon Us, March 2018.

https://www.arbornetworks.com/blog/asert/netscout-arbor-confirms-1-7-tbps-ddos-attack-terabit-attack-

[3] Giovane C. M. Moura, Ricardo de O. Schmidt, John Heidemann, Wouter B. de Vries, Moritz Müller, Lan Wei, and Christian Hesselman.

Anycast vs. DDoS: Evaluating the November 2015 root DNS event.

In *Proceedings of the ACM Internet Measurement Conference*, November 2016.

References II

[4] Moritz Müller, Giovane C. M. Moura, Ricardo de O. Schmidt, and John Heidemann.

Recursives in the wild: Engineering authoritative DNS servers.

In *Proceedings of the ACM Internet Measurement Conference*, pages 489–495, London, UK, 2017.

[5] Nicole Perlroth.

Hackers used new weapons to disrupt major websites across U.S.

New York Times, page A1, Oct. 22 2016.

[6] Alec Peterson.

Ec2 resolver changing ttl on dns answers?

Post on the DNS-OARC dns-operations mailing list, `https://lists.dns-oarc.net/pipermail/dns-operations/2017-November/017043.html`, November 2017.

References III

[7] José Jair Santanna, Roland van Rijswijk-Deij, Rick Hofstede, Anna Sperotto, Mark Wierbosch, Lisandro Zambenedetti Granville, and Aiko Pras.

Booters—an analysis of DDoS-as-a-Service attacks.

In *Proceedings of the 14th IFIP/IEEE Interatinoal Symposium on Integrated Network Management*, Ottowa, Canada, May 2015. IFIP.

[8] Kyle Schomp, Tom Callahan, Michael Rabinovich, and Mark Allman.

On measuring the client-side DNS infrastructure.

In *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*, pages 77–90. ACM, October 2013.