TsuNAME: exploiting misconfiguration and vulnerability to DDoS DNS

Giovane C. M. Moura[1], Sebastian Castro[2],
John Heidemann[3], Wes Hardaker[3]

1: SIDN Labs, 2: InternetNZ, 3: USC/ISI

NCSC One conference

2021-09-29

Introduction

- The DNS is one of the **core** services on the Internet
- People notice it when it **breaks**:
    - 2016 DDoS against Dyn DNS 2016 [1, 4]
    - 2019 DDoS against Amazon AWS [6]

DDos against Dyn (2016): affected Netflix, Spotify, Airbnb, Reddit, and others. Source: [4]

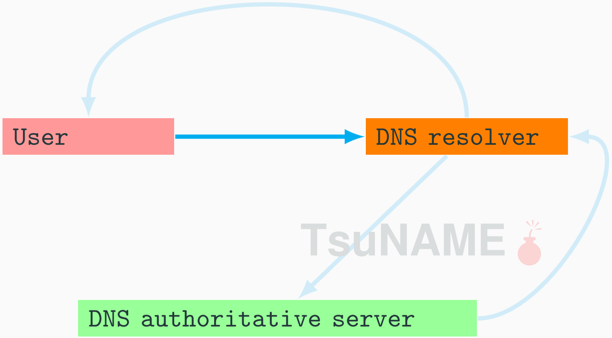

The New York Times

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

A map of the areas experiencing problems, as of Friday afternoon, according to

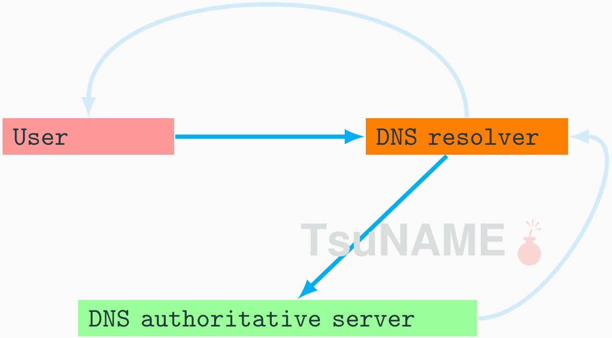
How DNS works

- People are bad in remember IP addresses
- So the DNS was first developed to map human-friendly names (domains) to IP addresses
    - http://www.wikipedia.org



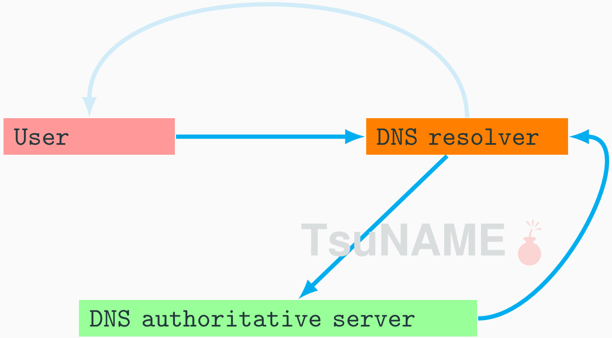
How DNS works

- People are bad in remember IP addresses
- So the DNS was first developed to map human-friendly names (domains) to IP addresses
  - http://www.wikipedia.org



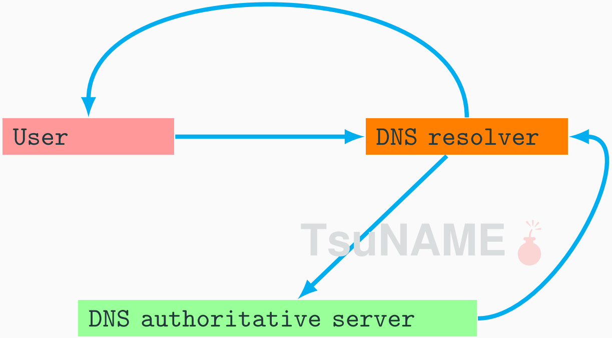
How DNS works

- People are bad in remember IP addresses
- So the DNS was first developed to map human-friendly names (domains) to IP addresses
  - http://www.wikipedia.org



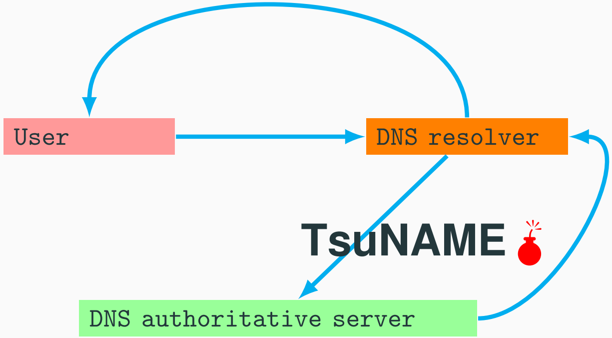
How DNS works

- People are bad in remember IP addresses
- So the DNS was first developed to map human-friendly names (domains) to IP addresses
  - http://www.wikipedia.org



How DNS works

- People are bad in remember IP addresses
- So the DNS was first developed to map human-friendly names (domains) to IP addresses
  - `http://www.wikipedia.org`



TL;DR slide

- TsuNAME is a vulnerability that can be used to DoS authoritative servers
- It requires three things:
  1. **Cyclic dependent** NS records
  2. **Vulnerable** resolvers
  3. User **queries** only to start/drive the process
- Problem: we've seen servers getting significant traffic for days
  - That's enough for going from 10qps to 5600qps (and more)
- To mitigate it:
  1. **Auth Ops**: detect cyclic records: use `CycleHunter`
     - BUT: difficult to prevent quick NS changes
  2. **Resolver Ops/Dev**: change resolvers
     - Google and Cisco fixed it
  3. (no way to prevent triggering queries)

What did we do?

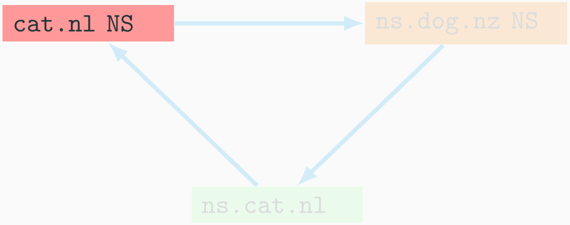
- We followed responsible disclosure guidelines

Date	Type	Group
2020-12-10	Private Disclosure	Google Notification
2020-12-10	Private Disclosure	SIDN DNSOPs
2021-02-05	Private Disclosure	OARC34
2021-02-22	Private Disclosure	APTLD
2021-02-22	Private Disclosure	NCSC-NL
2021-02-23	Private Disclosure	CENTR
2021-03-04	Private Disclosure	LACTLD
2021-02-18–2021-05-05	Private Disclosure	Private
2021-05-06	Public Disclosure	OARC35
2021-05-06	Public Disclosure	`https://tsuname.io`

Table 1: TsuNAME disclosure timeline

Cyclic Dependency is a loop; an error

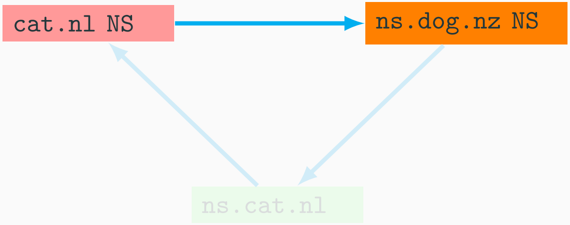
- First described in RFC1536, and later in Pappas2004 [3]



- Resolvers should return **SERVFAIL** , but some seem to loop a lot (huge amplification)

Cyclic Dependency is a loop; an error

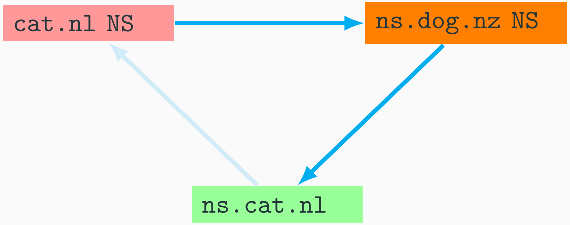
- First described in RFC1536, and later in Pappas2004 [3]



- Resolvers should return **SERVFAIL** , but some seem to loop a lot (huge amplification)

Cyclic Dependency is a loop; an error

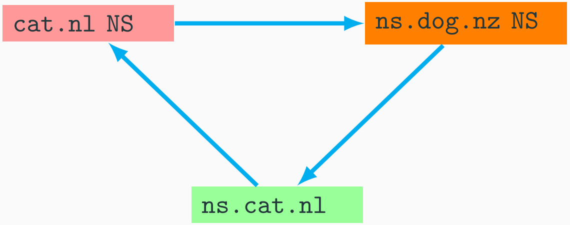
- First described in RFC1536, and later in Pappas2004 [3]



- Resolvers should return **SERVFAIL** , but some seem to loop a lot (huge amplification)

Cyclic Dependency is a loop; an error

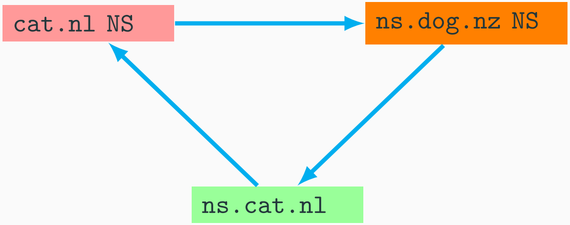
- First described in RFC1536, and later in Pappas2004 [3]



- Resolvers should return **SERVFAIL** , but some seem to loop a lot (huge amplification)

Cyclic Dependency is a loop; an error

- First described in RFC1536, and later in Pappas2004 [3]



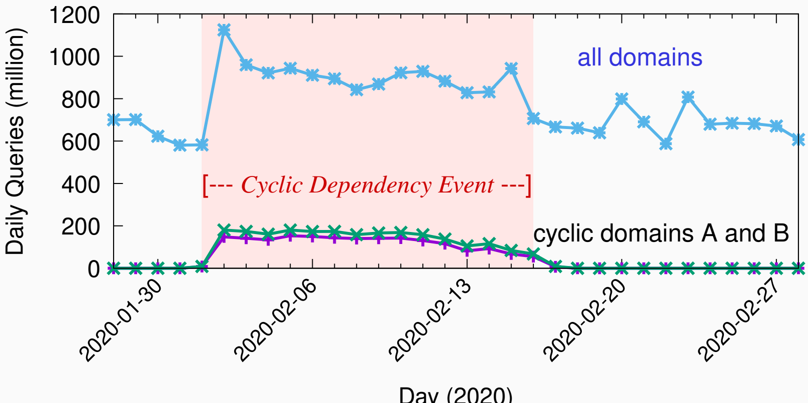
- Resolvers should return **SERVFAIL** , but some seem to loop a lot (huge amplification)

Cyclic Dependency is a loop; an error

- RFC1536 (1993)! mentioned the existence of such loops
  - We, however, show how it can be used for DDoS
- RFC1536 says that resolvers must “ bound the amount of work so a request can’t get into an infinite loop"
- We add that resolvers **must implement negative caching**, so subsequent queries don’t trigger extra queries

TsuNAME `.nz` event: traffic surged

- On 2020-02-01, two `.nz` domains (A and B) were misconfigured with cyclic dependency
- Total traffic **surged 50%**



Where these resolvers come from?

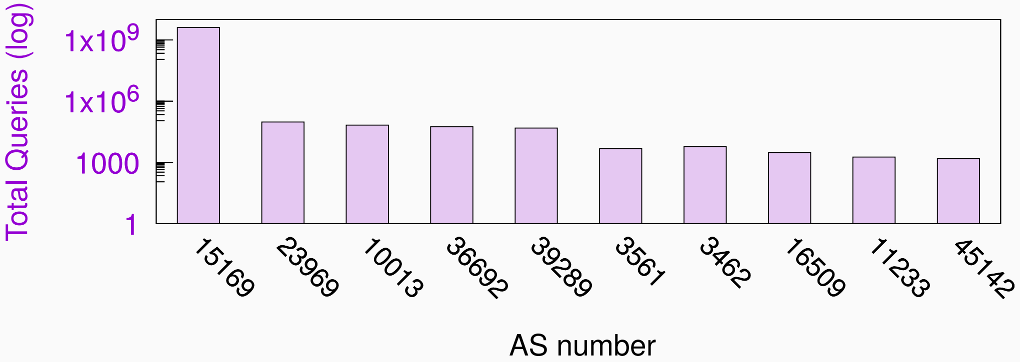


Figure 1: Queries for cyclic domains: 99% from Google (AS15169)

Where these resolvers come from?

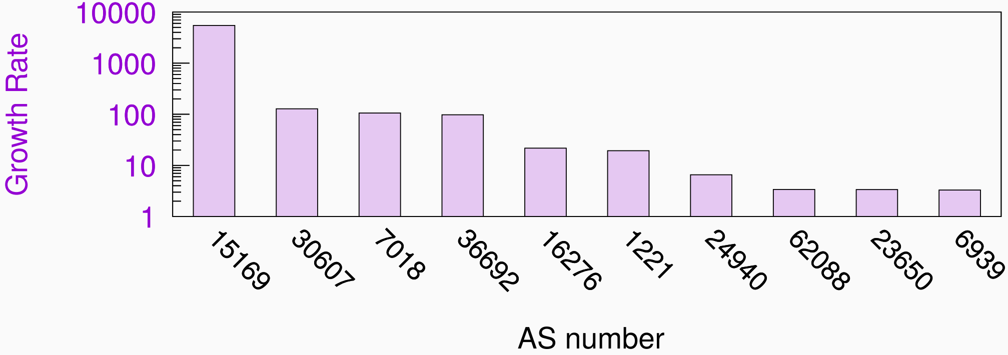


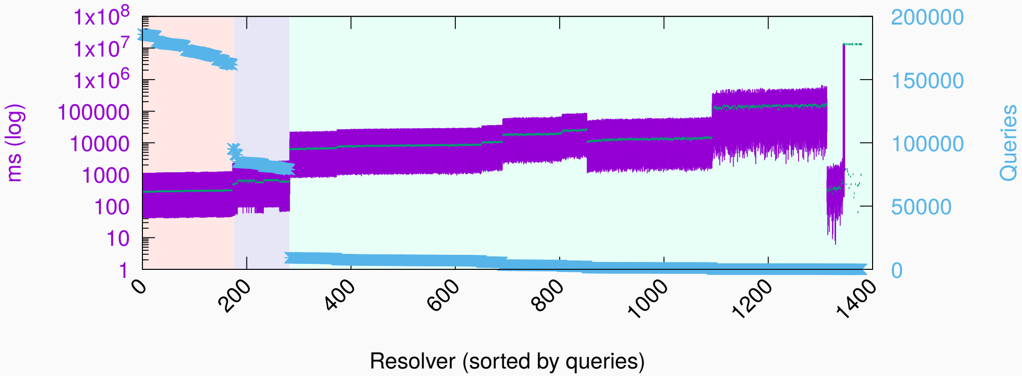
Figure 2: Traffic increase

- Traffic increase: queries during event / queries during “normal” period

AS list of .nz TsuNAME event

AS Number	AS name	Country
15169	Google	US
23969	TOT Public Company Limited	Thailand
10013	FreeBit	Japan
36692	Cisco OpenDNS	US
39289	MediaSeti	Russia
3561	CENTURYLINK-LEGACY-SAVVIS	US
3452	University of Alabama at Birmingham	US
16509	Amazon, Inc	US
11233	Gorge Networks	US
45142	Loxley Wireless	Thailand
200050	ITSVision	France
30844	Liquid Telecom	UK

How often Google sent queries to `.nz`?



Three groups of resolvers

- Heavy hitters: every 300ms
- Modetare hitters: every 600ms

The Real Threat

- `.nz` saw a 50% traffic surge due to 2 misconfigured domains
- **The threat:**
  - Adversary holds multple domains (register or already has)
  - then change their NS records (create cycles)
  - then query from a botnet (inject queries)

  That got us very **concerned**.
- How many anycast providers could withstand that?
- How many TLDs would remain up?
- That's why we are disclosing this here

The Real Threat

- `.nz` saw a 50% traffic surge due to 2 misconfigured domains
- **The threat:**
  - Adversary holds multple domains (register or already has)
  - then change their NS records (create cycles)
  - then query from a botnet (inject queries)

That got us very **concerned**.

- How many anycast providers could withstand that?
- How many TLDs would remain up?
- That's why we are disclosing this here

The Real Threat

- `.nz` saw a 50% traffic surge due to 2 misconfigured domains
- **The threat:**
    - Adversary holds multple domains (register or already has)
    - then change their NS records (create cycles)
    - then query from a botnet (inject queries)

  That got us very **concerned**.
- How many anycast providers could withstand that?
- How many TLDs would remain up?
- That's why we are disclosing this here

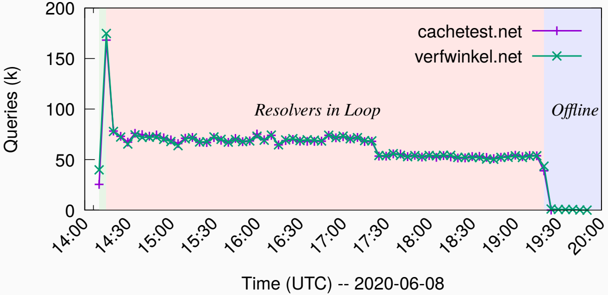
Was this an isolated event? Reproducing TsuNAME

No: we managed to reproduce it multiple times

1. Lower bound with 1 query/resolver from Ripe Atlas
2. Influence of recurrent queries with Ripe Atlas
3. Domain without Atlas queries

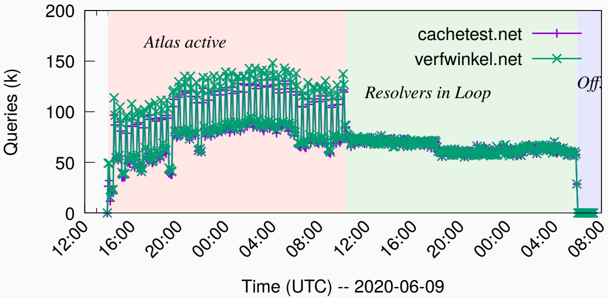
Some resolvers will loop without user queries

- 10k Ripe Atlas : 1 query to their local resolvers
- View from Auth Servers



Recurrent Queries Amplify the Problem

- 10k Ripe Atlas : 1 query every 10min to local resolvers
- View from Auth Servers



What can we do prevent this?

- We don't know how **big** a DDoS can get with this
  - We did not measure this: that'd be vandalism

1. Fix Resolvers: (**notification**)
   - We notified Google and Cisco OpenDNS; **they both fixed it**
   - Notified top 10 ASes, only 3 responded.
     - Two were running old DNS software: 2008 (MS) and 2015 (PowerDNS) versions
2. Auth OPs: **prevention**:
   - remove cyclic dependencies from zone files with `CycleHunter`, our open-source tool

What can we do prevent this?

- We don’t know how **big** a DDoS can get with this
  - We did not measure this: that’d be vandalism

1. Fix Resolvers: (**notification**)
   - We notified Google and Cisco OpenDNS; **they both fixed it**
   - Notified top 10 ASes, only 3 responded.
     - Two were running old DNS software: 2008 (MS) and 2015 (PowerDNS) versions
2. Auth OPs: **prevention**:
   - remove cyclic dependencies from zone files with `CycleHunter`, our open-source tool

What can we do prevent this?

- We don’t know how **big** a DDoS can get with this
    - We did not measure this: that’d be vandalism

1. Fix Resolvers: (**notification**)
    - We notified Google and Cisco OpenDNS; **they both fixed it**
    - Notified top 10 ASes, only 3 responded.
        - Two were running old DNS software: 2008 (MS) and 2015 (PowerDNS) versions
2. Auth OPs: **prevention**:
    - remove cyclic dependencies from zone files with `CycleHunter`, our open-source tool

CycleHunter

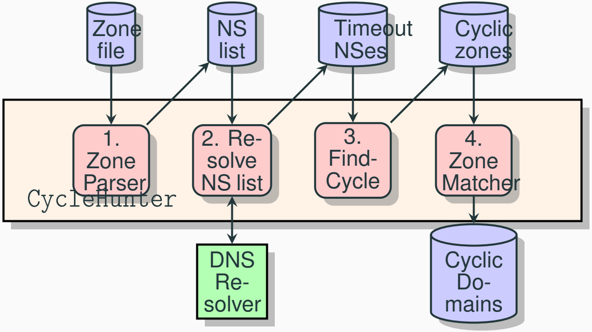


Figure 3: CycleHunter workflow

- We release it at: https://tsuname.io

Not many cyclic dependencies in the wild, ATM

zone	Size	NSSet	Cyclic	Affec.	Date
.com	151445463	2199652	21	1233	2020-12-05
.net	13444518	708837	6	17	2020-12-10
.org	10797217	540819	13	121	2020-12-10
.nl	6072961	79619	4	64	2020-12-03
.se	1655434	27540	0	0	2020-12-10
.nz	718254	35738	0	0	2021-01-11
.nu	274018	10519	0	0	2020-12-10
Root	1506	115	0	0	2020-12-04
Total	184409371	3602839	44	1435	

Table 3: `CycleHunter`: evaluated DNS Zones

- Human error plays a role

We found a parked `.nl` domain: it lasted for months

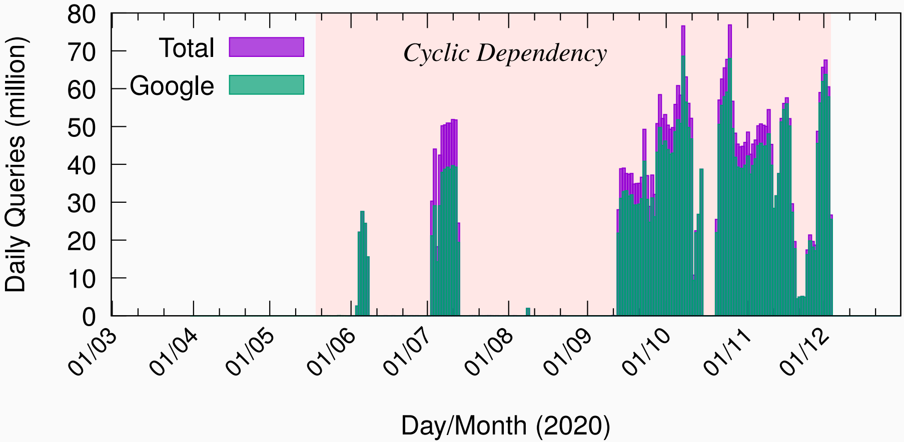


Figure 4: Timeseries of queries – it started on 2020-05-19

We found a parked `.nl` domain: it lasted for months

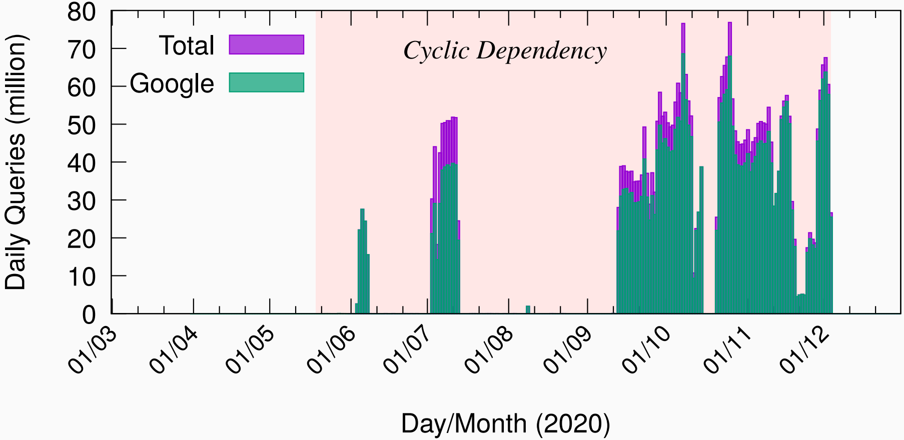


Figure 4: Timeseries of queries – it started on 2020-05-19

We evaluated other resolver software too

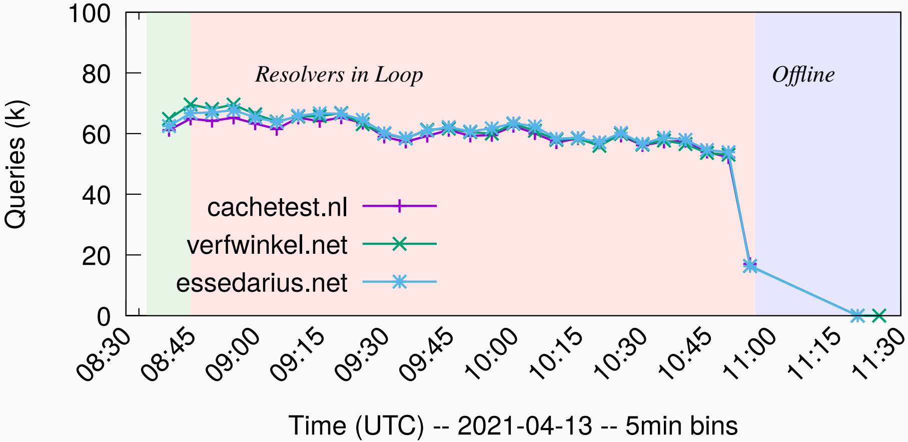
- No recurring cycles with these (they stop):
    - Unbound
    - BIND
    - PowerDNS
    - Public DNS: Quad1,Quad9
- But we don’t know what other other ASes are running
- Whatever they are running, expect a long time to be fixed
- Looping old resolvers:
    - PowerDNS 3.6.2-2, from 2014 [5]
    - Windows 2008R2.

Shared materials on `https://tsuname.io`

- Technical Report
  - Paper will appear on the forthcoming ACM IMC 2021 conference
- Security Advisory
- `CycleHunter`

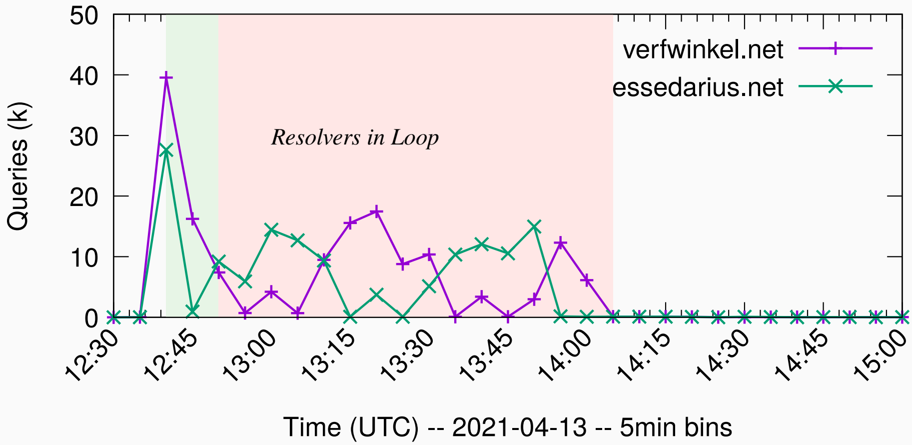
What have we learned since the private disclosure?

1. Longer cycles (triple) cause even more problems



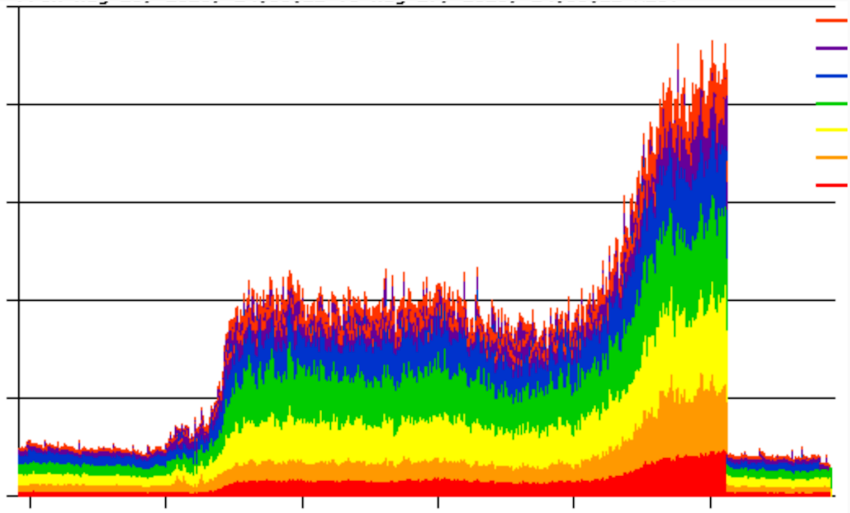
What have we learned since the private disclosure?

2. CNAME cycles are not as problematic



What have we learned since the private disclosure?

3. Other ccTLDs have seen such events too



What have we learned since the private disclosure?

What have we learned since the private disclosure?

5. **We identified the root causes of looping:**
   - Some resolvers will **loop** indefinitely (∞)
   - Others won't loop, but they **won't cache**: every new client query trigger new queries

The fix: **detect the loop, and cache it.**

What have we learned since the private disclosure?

6. We confirmed Google fixed its Public DNS

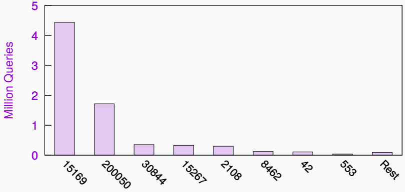
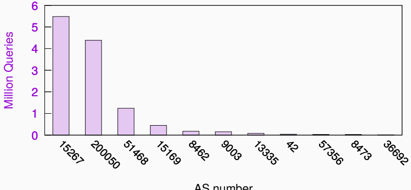


Figure 8: Measurement **BEFORE** Google fix



What have we learned since the private disclosure?

7. And Google awarded us (USD 1000.00)

We donate it all to Wikipedia

benevity | Google.org

NEW GIFTCARD

Hi Bughunter

Hello Google Bughunter,

Thank you for choosing to donate your reward to charity! See below for your gift card code, it can be redeemed (anonymously) at https://google.benevity.org/community/redeem-a-giftcard.

Gift card code will be valid for 1 year after being issued. After that date we will donate the reward to a charity of our choosing.

Note: There will be no confirmation email once the donation has been placed.

Discussion

- If you're an **auth operato**r, check your zone
  - You can use `CycleHunter`
  - Don't forget about **collateral damage**
- if you're a **resolver op/dev,**
  - Detect cyclic dependencies and return SERVFAIL
  - Cache the SERVFAIL for future clients (negative caching)

Discussion

- RFC1536 predict these loops, but that was 28 years ago
- They emphasize the role of the single recursive resolver without considering the interactions in today's DNS ecosystem.
- Which is far more concentrated and centralized:
    - 1/3 of the DNS traffic to `.nl` and `.nz` come from 5 companies only [2].
- We recommend negative caching of cyclic dependent domains
- Overall, we've manage to identify and help others to fix their sofware and protecting users

References i

[1] ANTONAKAKIS, M., APRIL, T., BAILEY, M., BERNHARD, M., BURSZTEIN, E., COCHRAN, J., DURUMERIC, Z., HALDERMAN, J. A., INVERNIZZI, L., KALLITSIS, M., KUMAR, D., LEVER, C., MA, Z., MASON, J., MENSCHER, D., SEAMAN, C., SULLIVAN, N., THOMAS, K., AND ZHOU, Y.

Understanding the Mirai botnet.

In *Proceedings of the 26th USENIX Security Symposium* (Vancouver, BC, Canada, Aug. 2017), USENIX, pp. 1093–1110.

References ii

[2] Moura, G. C. M., Castro, S., Hardaker, W., Wullink, M., and Hesselman, C.

Clouding up the Internet: How Centralized is DNS Traffic Becoming?

In *Proceedings of the ACM Internet Measurement Conference* (New York, NY, USA, 2020), IMC '20, Association for Computing Machinery, p. 42–49.

[3] Pappas, V., Xu, Z., Lu, S., Massey, D., Terzis, A., and Zhang, L.

Impact of configuration errors on DNS robustness.

SIGCOMM Comput. Commun. Rev. 34, 4 (Aug. 2004), 319–330.

References iii

[4] Perlroth, N.

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

New York Times (Oct. 22 2016), A1.

[5] PowerDNS.

Changelogs for all pre 4.0 releases.

https://doc.powerdns.com/recursor/changelog/pre-4.0.html, Jan. 2021.

[6] WILLIAMS, C.
Bezos DDoS'd: Amazon Web Services' DNS systems knackered by hours-long cyber-attack.
https://www.theregister.co.uk/2019/10/22/aws_dns_ddos/, 10 2019.