TsuNAME vulnerability
Public disclosure

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

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

OARC 35

Virtual Meeting

2021-05-06

Responsible Disclosure

- We followed responsible disclosure guidelines

Date	Type	Group
2021-02-05	Private Disclosure	OARC34
2021-02-22	Private Disclosure	APTLD
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

Results obtained since the notifications

1. **Two large** public resolver services have repaired their code
   - Google Public DNS and Cisco Public DNS (kudos!)
2. Several contributors to `CycleHunter`
   - Shane Kern, Hugo Salgado, and several others users
   - https://github.com/SIDN/CycleHunter/graphs/contributors
   - 9 forks, 27 issues (5 open), 4 stars. Great response from the community
3. We know far more about the problem now
4. We are public releasing two documents:
   - Security Advisory: https://tsuname.io/advisory.pdf
   - Tech Report: https://tsuname.io/tech-report.pdf

TL;DR

- We disclose TsuNAME, 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
  3. (no way to prevent triggering queries)

tsuNAME

- We call it **tsuNAME**
- Website: `https://tsuname.io`

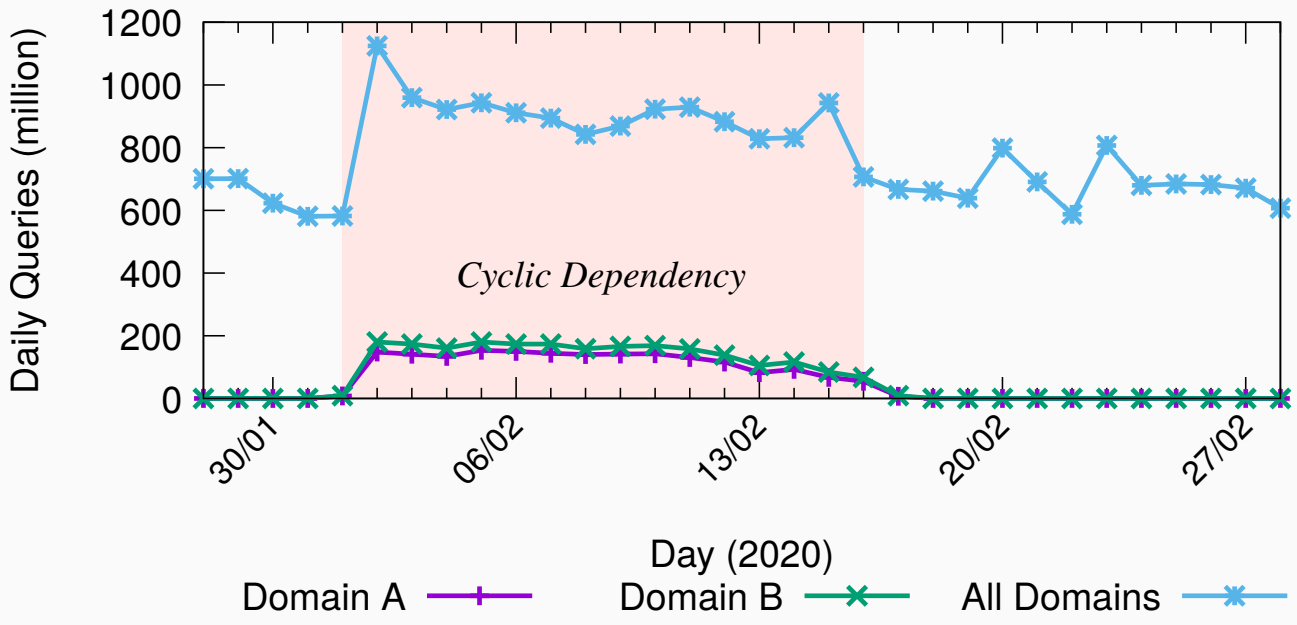
Private Disclosure Recap

- Cyclic Dependencies were first described in Pappas2009 [1]
- Simplest cyclic dependency: two domains, one nameservers each
  - `cat.nl NS ns1.dog.nz`
  - `dog.nz NS ns1.cat.nl`
- Observed in the wild: the `.nz` event
  - Two domains, two nameservers each. Event lasting 16 days starting Feb 1 2020. **50% traffic surge**
  - Mostly A and AAAA queries for the nameservers
  - Mostly coming from Google Public DNS

[1] Vasileios Pappas, Zhiguo Xu, Songwu Lu, Daniel Massey, Andreas Terzis, and Lixia Zhang. **Impact of configuration errors on DNS robustness**. SIGCOMM Comput. Commun. Rev., August 2004.

TsuNAME.nz event: traffic surged

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



Domains A and B: from 30k queries to 334M tops ($x10^4$)

Where these resolvers come from?

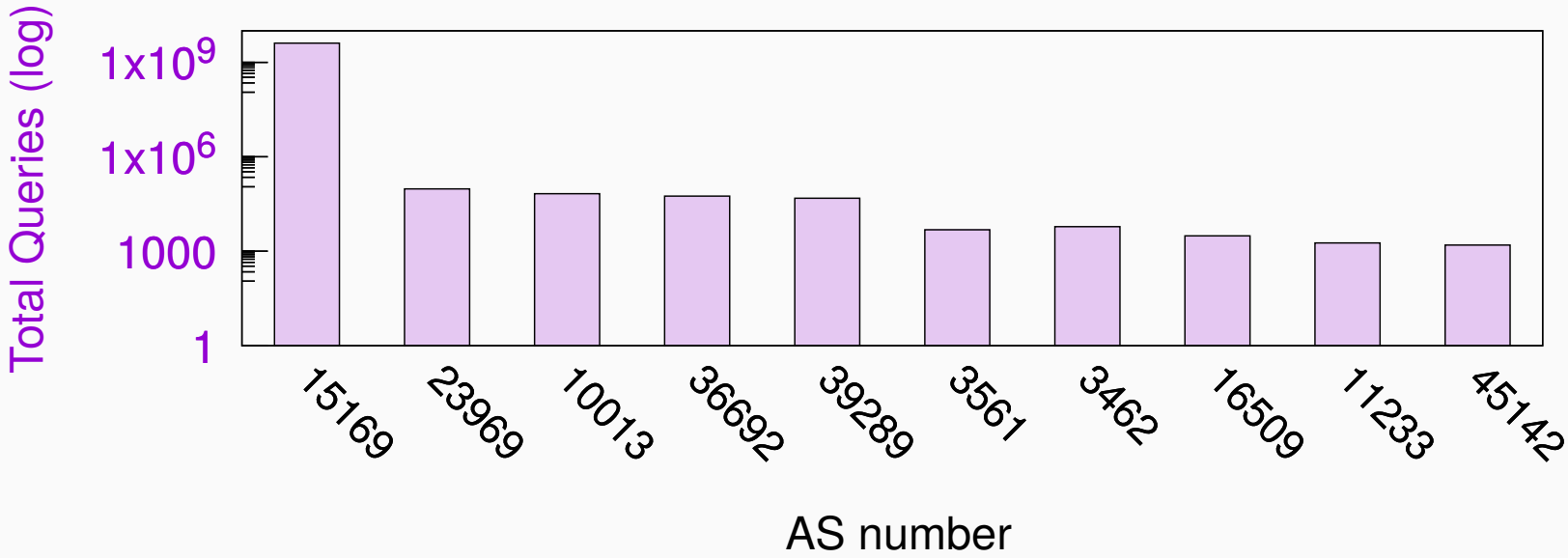


Figure 1: Queries for cyclic domains: 99% from Google (AS15169)

Threat profile

- Observed events caused by accident. Other ccTLDs have seen them too.
- An attacker could:
  - Hold multiple domains (register or already has)
  - Intentionally create cycles by changing NS records
  - Inject queries by using, for example, a botnet
- Easy and straigforward to setup, it can be weaponized

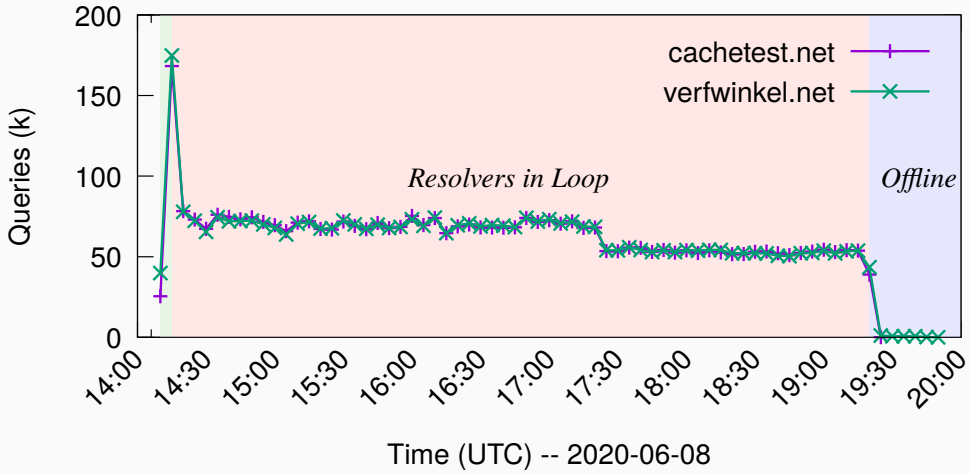
Was this an isolated event?

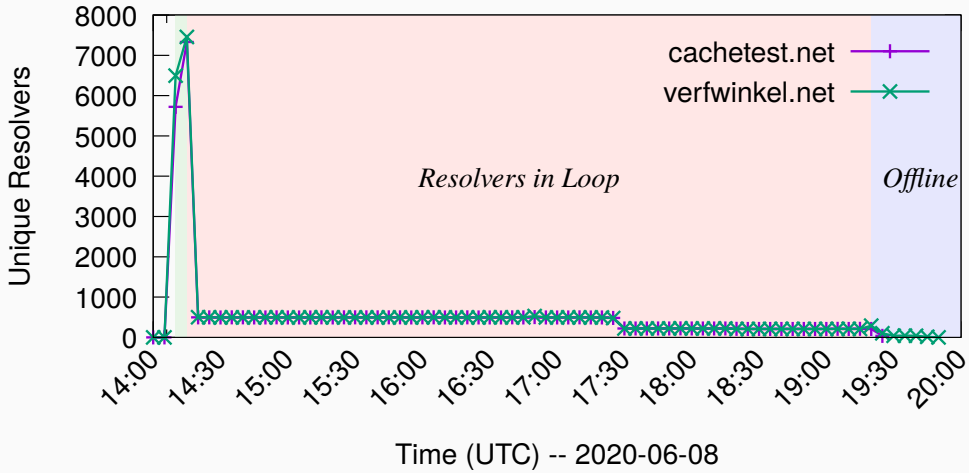
No: we managed to reproduce it multiple times

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

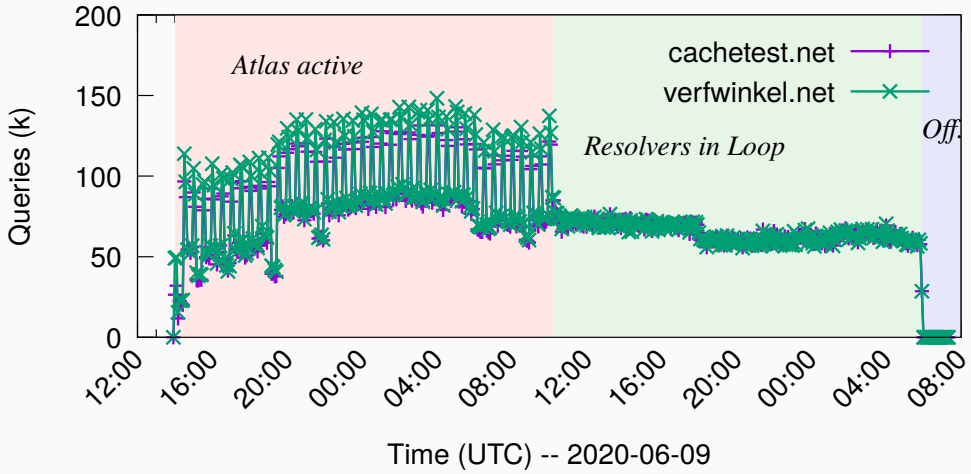
- 10k Ripe Atlas : 1 query to their local resolvers
- View from Auth Servers

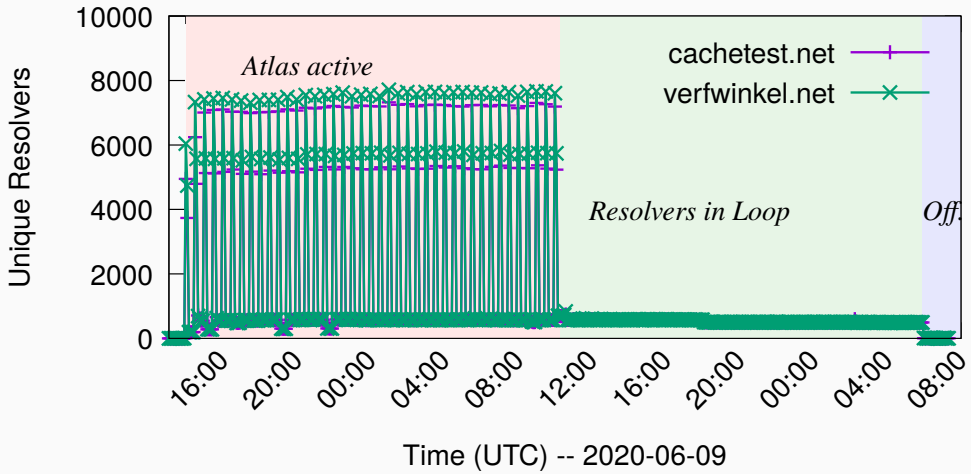




Recurrent Queries Amplify the Problem

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





What can be done to prevent this?

1. Fix Resolvers: (**notification**)
   - We notified Google and OpenDNS; **both fixed their software**
   - Other DNS implementations confirmed as unaffected
2. Auth OPs: **prevention**:
   - remove cyclic dependencies from zone files with `CycleHunter`, our open-source tool
   - Give plently of leeway to be aware and prepare to solve cycles affecting their zones

What can be done to prevent this?

1. Fix Resolvers: (**notification**)
    - We notified Google and OpenDNS; **both fixed their software**
    - Other DNS implementations confirmed as unaffected
2. Auth OPs: **prevention**:
    - remove cyclic dependencies from zone files with `CycleHunter`, our open-source tool
    - Give plently of leeway to be aware and prepare to solve cycles affecting their zones

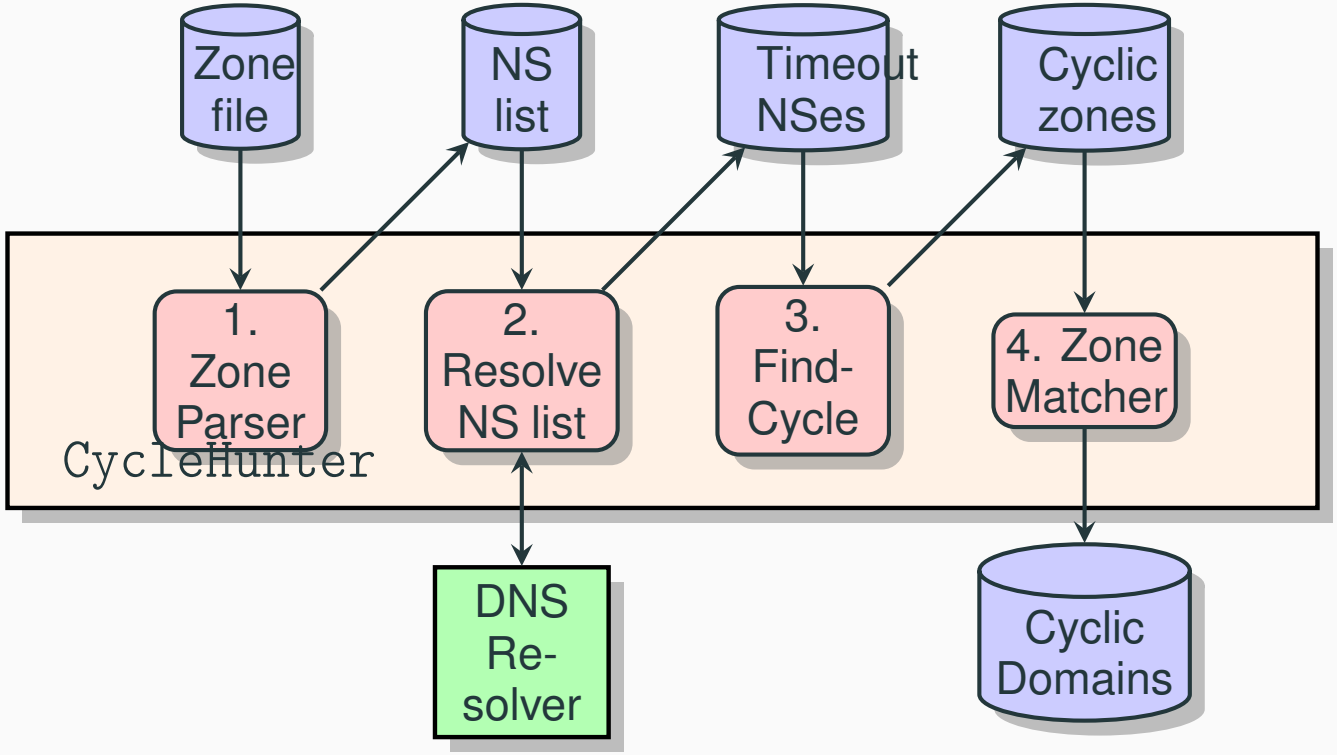


Figure 2: CycleHunter workflow

- We release it at: https://tsuname.io
- Also in GitHub at: https://github.com/SIDN/CycleHunter

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 2: `CycleHunter`: evaluated DNS Zones

- Human error plays a role

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 [1]
  - Windows 2008R2.

What have we learned since the private disclosure?

1. Longer cycles (triple) cause even more problems

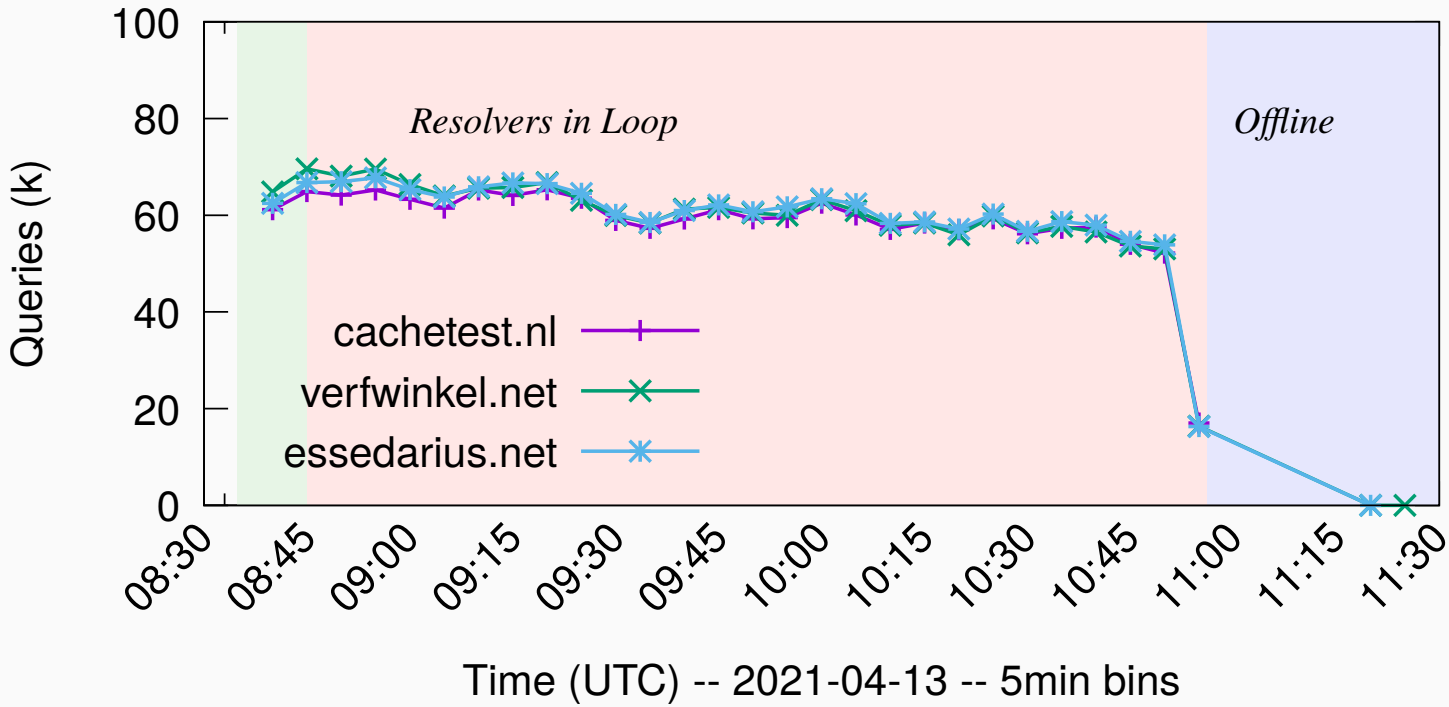


Figure 3: TripleDep measurement: Queries to authoritative servers (5min bins)

What have we learned since the private disclosure?

2. CNAME cycles are not as problematic

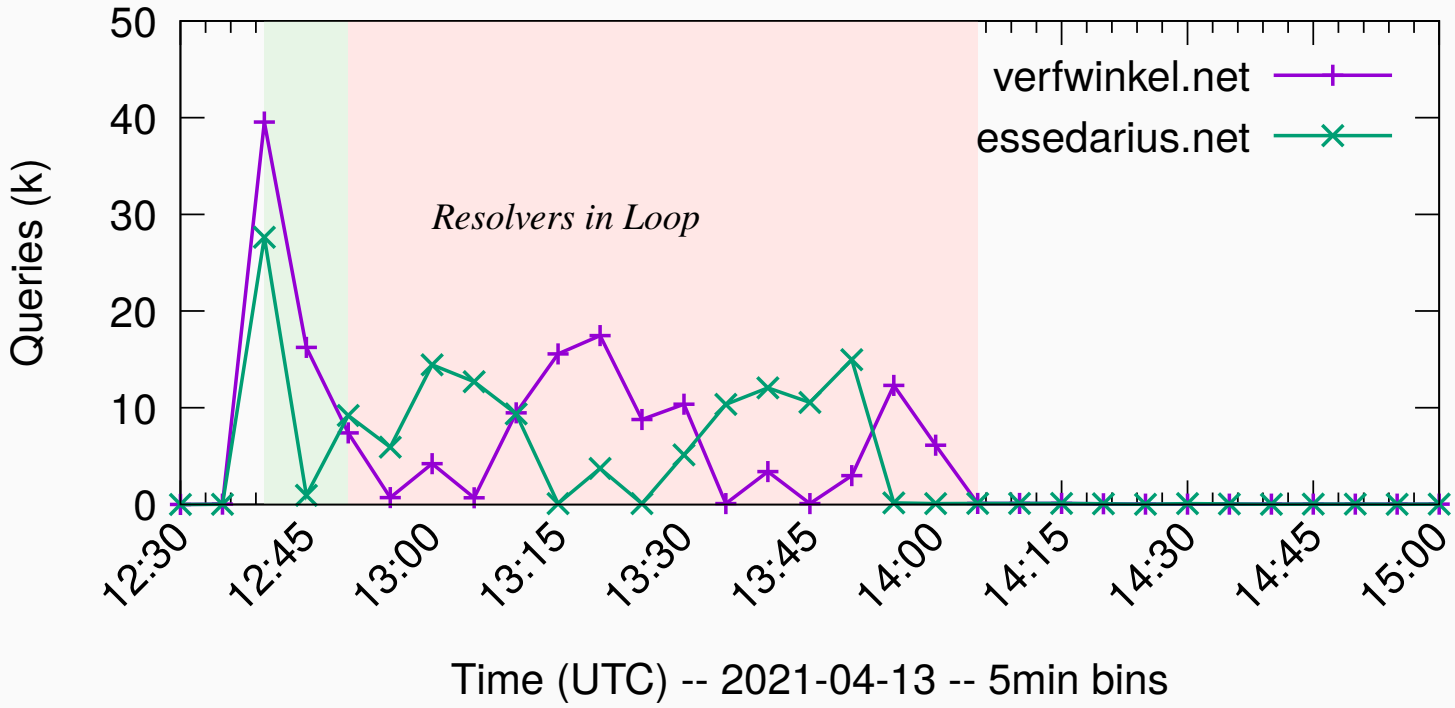


Figure 4: CNAME measurement: Querie to authoritative servers (5min bins)

What have we learned since the private disclosure?

3. Other ccTLDs have seen such events too

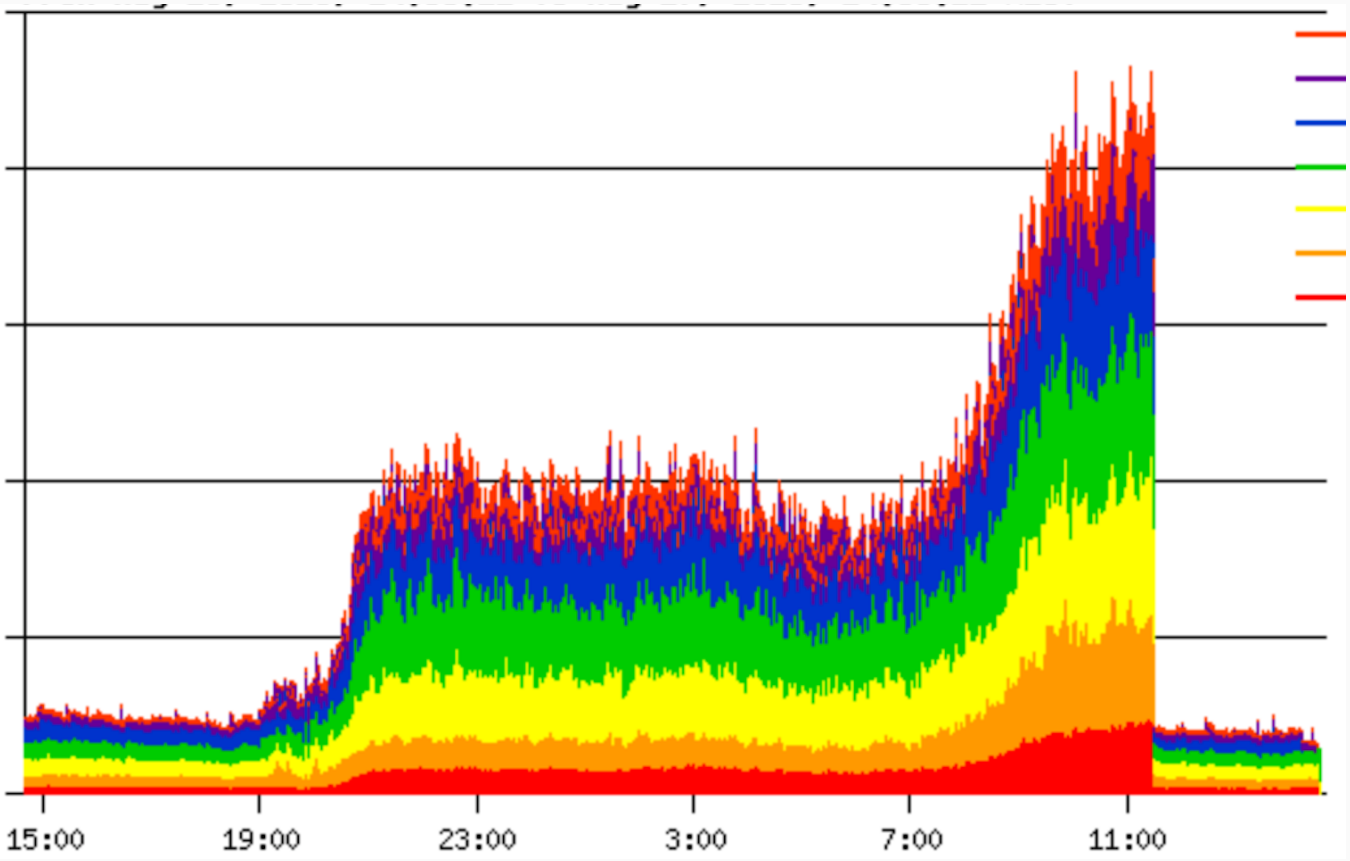


Figure 5: TsuNAME event at an Anonymous EU-based ccTLD operator.

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

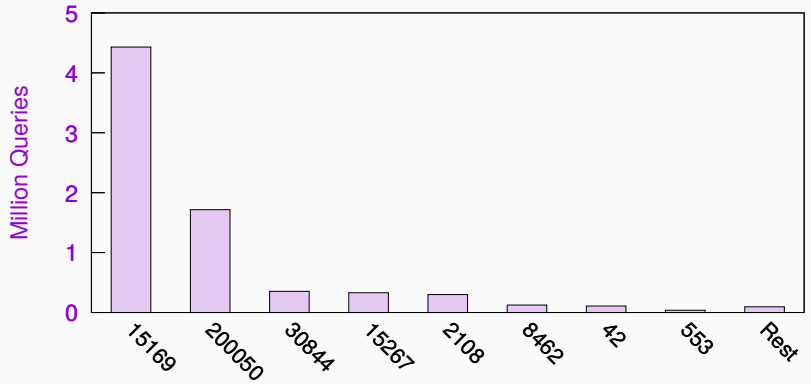


Figure 6: Measurement **BEFORE** Google fix

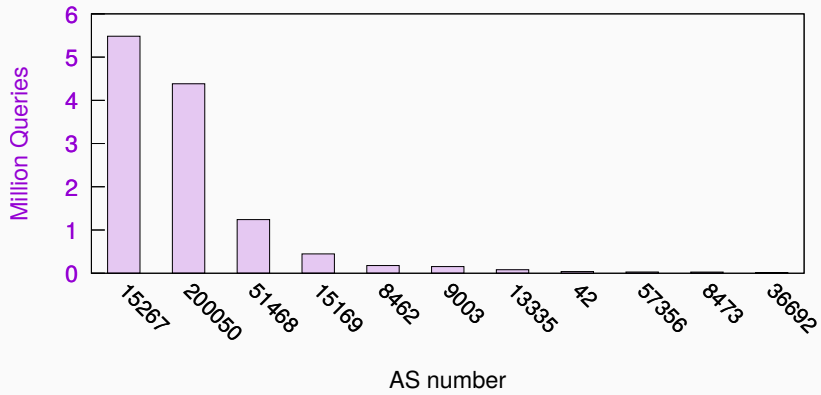


Figure 7: Measurement **AFTER** Google fix

Question: if I run `CycleHunter` once a day, will I be OK?

- **No**
- Changes may occur at any time:
  - `cat.nl NS ns1.dog.nz`
  - `ns1.dog.nz A 192.168.1.1`

  5 min later:
  - `cat.nl NS ns1.dog.nz`
  - `ns1.dog.nz NS ns1.dog.nl`
- This will find problems at point in time
- There is no continuous solution

Question: if I run `CycleHunter` once a day, will I be OK?

- **No**
- Changes may occur at any time:
  - `cat.nl NS ns1.dog.nz`
  - `ns1.dog.nz A 192.168.1.1`

  5 min later:
  - `cat.nl NS ns1.dog.nz`
  - `ns1.dog.nz NS ns1.dog.nl`
- This will find problems at point in time
- There is no continuous solution

Question: if I run `CycleHunter` once a day, will I be OK?

- **No**
- Changes may occur at any time:
  - `cat.nl NS ns1.dog.nz`
  - `ns1.dog.nz A 192.168.1.1`

  5 min later:
  - `cat.nl NS ns1.dog.nz`
  - `ns1.dog.nz NS ns1.dog.nl`
- This will find problems at point in time
- There is no continuous solution

Question: if I have cyclic dependencies, do I get DDoS'ed?

- **Maybe**
  - as in the `.nl` example, only having cyclic dependencies does not lead to DDoS per se
  - You'll need vulnerable resolvers to find you
  - We need someone to inject traffic
- An attacker can create these situations if they want

Question: if I have cyclic dependencies, do I get DDoS'ed?

- **Maybe**
  - as in the `.nl` example, only having cyclic dependencies does not lead to DDoS per se
  - You'll need vulnerable resolvers to find you
  - We need someone to inject traffic
- An attacker can create these situations if they want

Question: if I have cyclic dependencies, do I get DDoS'ed?

- **Maybe**
  - as in the `.nl` example, only having cyclic dependencies does not lead to DDoS per se
  - You'll need vulnerable resolvers to find you
  - We need someone to inject traffic
- An attacker can create these situations if they want

Question: I have RRL, so I'll be OK, right?

- **No**
  - RRL converts queries to TCP
  - Resolvers react to that by retrying heavily[2]
  - So they you have yet another amplification
- It may slow your attack, but it's not going to block it

[2] G. C. M. Moura, John Heidemann, Moritz Müller, Ricardo de O. Schmidt and Marco Davids. **When the Dike Breaks: Dissecting DNS Defenses During DDoS**. Proceedings of the 2018 ACM Internet Measurement Conference

Question: I have RRL, so I'll be OK, right?

- **No**
  - RRL converts queries to TCP
  - Resolvers react to that by retrying heavily[2]
  - So they you have yet another amplification
- It may slow your attack, but it's not going to block it

[2] G. C. M. Moura, John Heidemann, Moritz Müller, Ricardo de O. Schmidt and Marco Davids. **When the Dike Breaks: Dissecting DNS Defenses During DDoS**. Proceedings of the 2018 ACM Internet Measurement Conference

Question: I have RRL, so I'll be OK, right?

- **No**
  - RRL converts queries to TCP
  - Resolvers react to that by retrying heavily[2]
  - So they you have yet another amplification
- It may slow your attack, but it's not going to block it

[2] G. C. M. Moura, John Heidemann, Moritz Müller, Ricardo de O. Schmidt and Marco Davids. **When the Dike Breaks: Dissecting DNS Defenses During DDoS**. Proceedings of the 2018 ACM Internet Measurement Conference

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
  - Check your amplification factor

Slides and report :

- `https://tsuname.io/`

Many thanks to **OARC team** for supporting us!

Many thanks to the **commmunity** for their help

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
  - Check your amplification factor

Slides and report :

- https://tsuname.io/

Many thanks to **OARC team** for supporting us!

Many thanks to the **commmunity** for their help

References i

[1] PowerDNS.

Changelogs for all pre 4.0 releases.

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