Passive vs Active Measurements in the DNS

Geoff Huston, Joao Damas

APNIC

This is work-in-progress

- So its not clear what the conclusions might be
- But we have noticed a set of not-well-explained behaviours in the DNS, and we are wondering if the observational behaviour has an influence on the observed DNS behaviour

Some anomalies

- In previous studies associated with the KSK roll we noticed that the profile of resolvers and their query volumes seen at root servers had a very different profile to the resolvers who ask authoritative servers for “terminal” DNS names
- We see repeat queries at servers that are inconsistent with our perceptions of how caching by recursive resolver systems should mitigate queries

Some questions we’d like to ask

- If we actively “plant” a sequence of DNS resolution queries into the edge, and record the queries we see at the authoritative name server for the DNS name being queried, then what can this tell us about the general behaviour of the DNS?
- What proportion of queries are the result of stub resolution questions and what proportion are the result of the DNS talking to itself (such as self-triggered cache refresh)
- Why are there query “storms”?

An active observation platform

We might understand the *effect* better if we controlled the *cause*

i.e. generate queries in a known context and look at their effect within in the DNS resolution environment

From the Inside looking Out

Instrument a DNS client

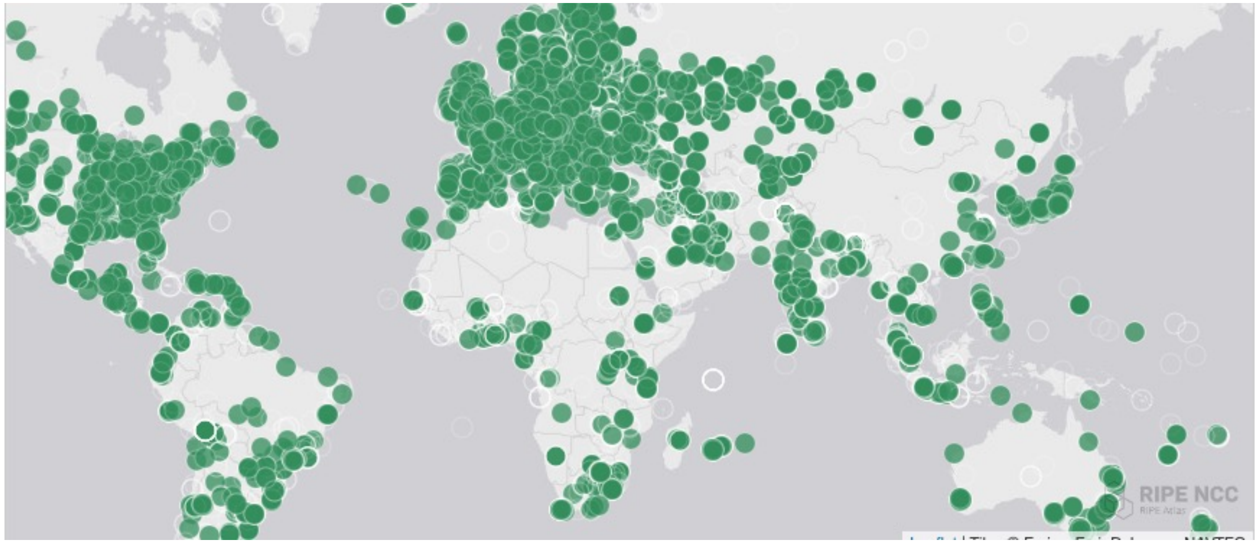
- Use the client to generate various DNS queries
- Measure the absolute outcomes and the variance

This needs the ability to either coopt or manufacture a collection of willing clients

From the Inside looking Out

RIPE Atlas

- Many thousands of end points installed in end user networks
- Programmable DNS queries
- Report back



From the Outside looking In

Set up authoritative server(s)

- Enroll end users to send queries to it
- Measure the outcomes from the perspective of the server, not the end client

In the Middle looking both ways

- Instrument recursive resolvers and observe both stub behaviours and authoritative server behaviours for queries
- There are obvious privacy issues that lurk very close to the surface here

How to measure using millions of end devices?

APNIC Lab’s approach

- we originally wanted to measure IPv6 deployment as seen by end users
- We wanted to say something about ALL users
- So we were looking at a way to sample end users in a random but statistically significant fashion across the entire network
- We stumbled across the advertising networks...

What can be scripted in an Ad

Not much:

- http.FetchImg()

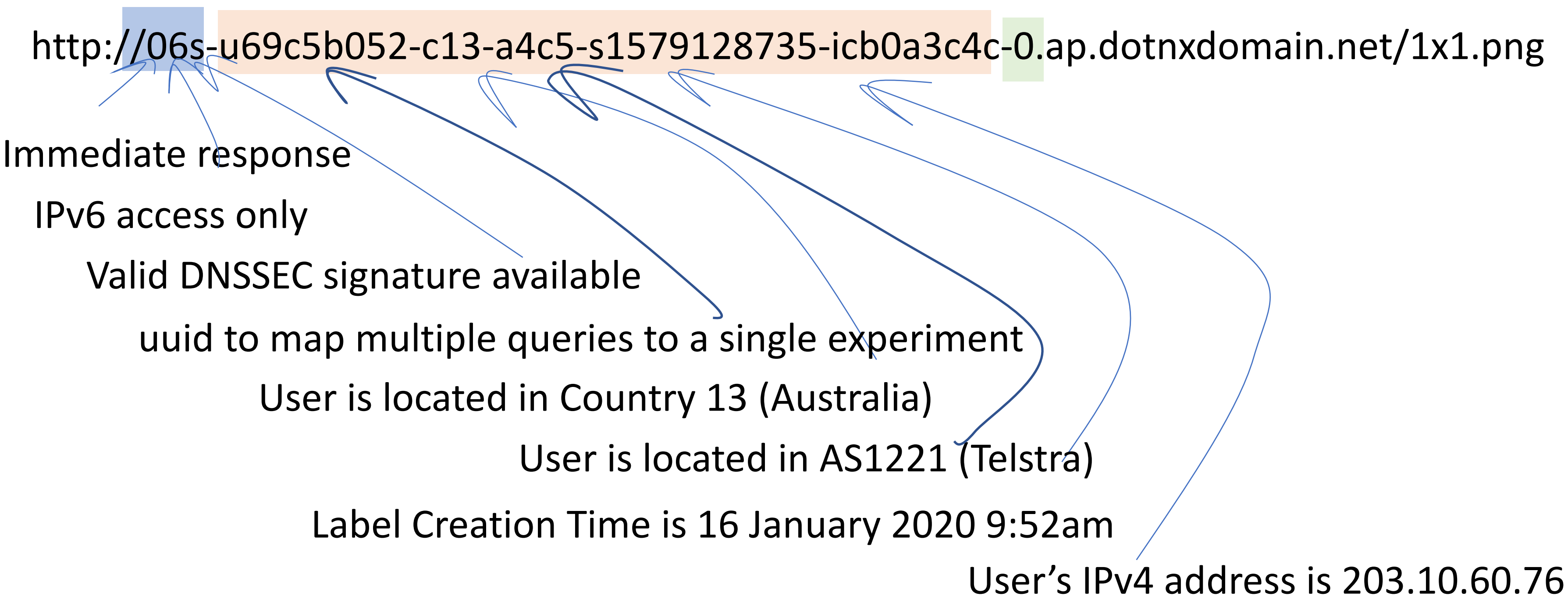
  i.e. attempt to retrieve a URL

But that's enough!

- It's EXACTLY what users do!
- A URL consists of a DNS question and an HTML question
- What if we point both the DNS and the HTML to servers we run?
- As long as each Ad execution uses unique names we can push the user query back to our servers

DNS Label Encoding

Think of a URL name as a microcoded instruction set directed to programmable DNS and HTTP servers …



http://06s-u69c5b052-c13-a4c5-s1579128735-icb0a3c4c-0.ap.dotnxdomain.net/1x1.png

Immediate response

IPv6 access only

Valid DNSSEC signature available

uuid to map multiple queries to a single experiment

User is located in Country 13 (Australia)

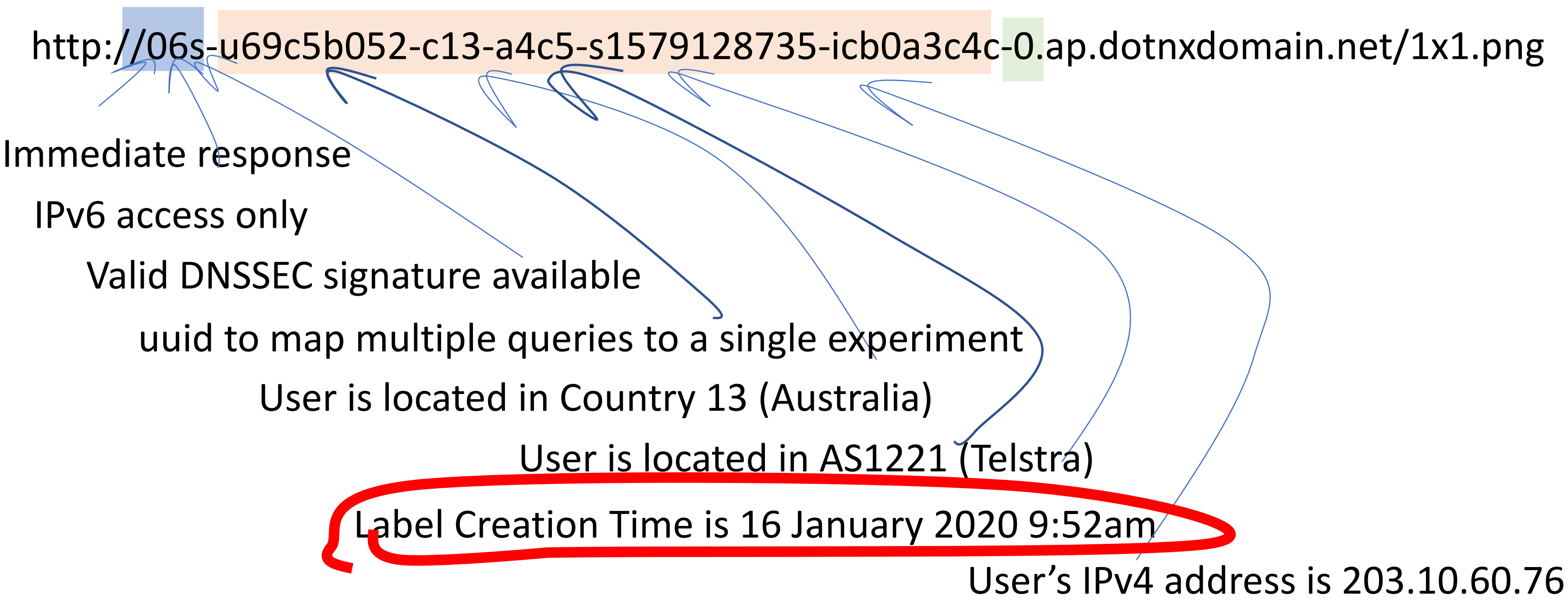
User is located in AS1221 (Telstra)

Label Creation Time is 16 January 2020 9:52am

User’s IPv4 address is 203.10.60.76

DNS Label Encoding

Think of a URL name as a microcoded instruction set directed to programmable DNS and HTTP servers …



http://06s-u69c5b052-c13-a4c5-s1579128735-icb0a3c4c-0.ap.dotnxdomain.net/1x1.png

Immediate response

IPv6 access only

Valid DNSSEC signature available

uuid to map multiple queries to a single experiment

User is located in Country 13 (Australia)

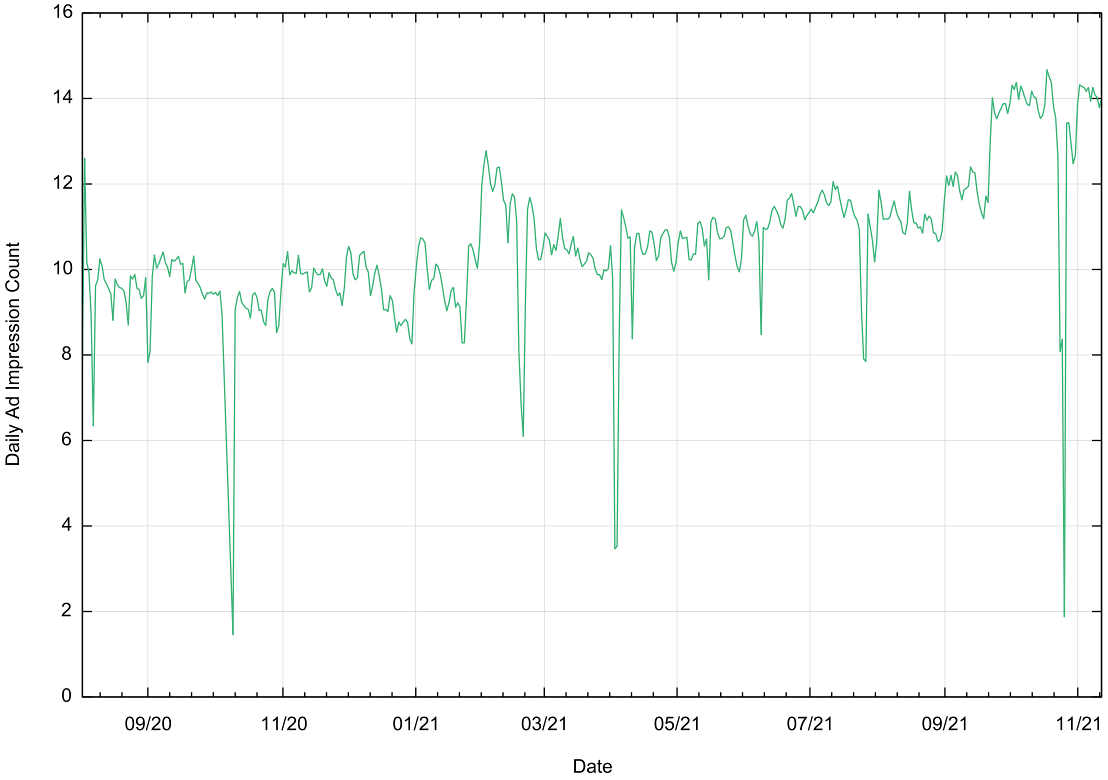
User is located in AS1221 (Telstra)

Label Creation Time is 16 January 2020 9:52am

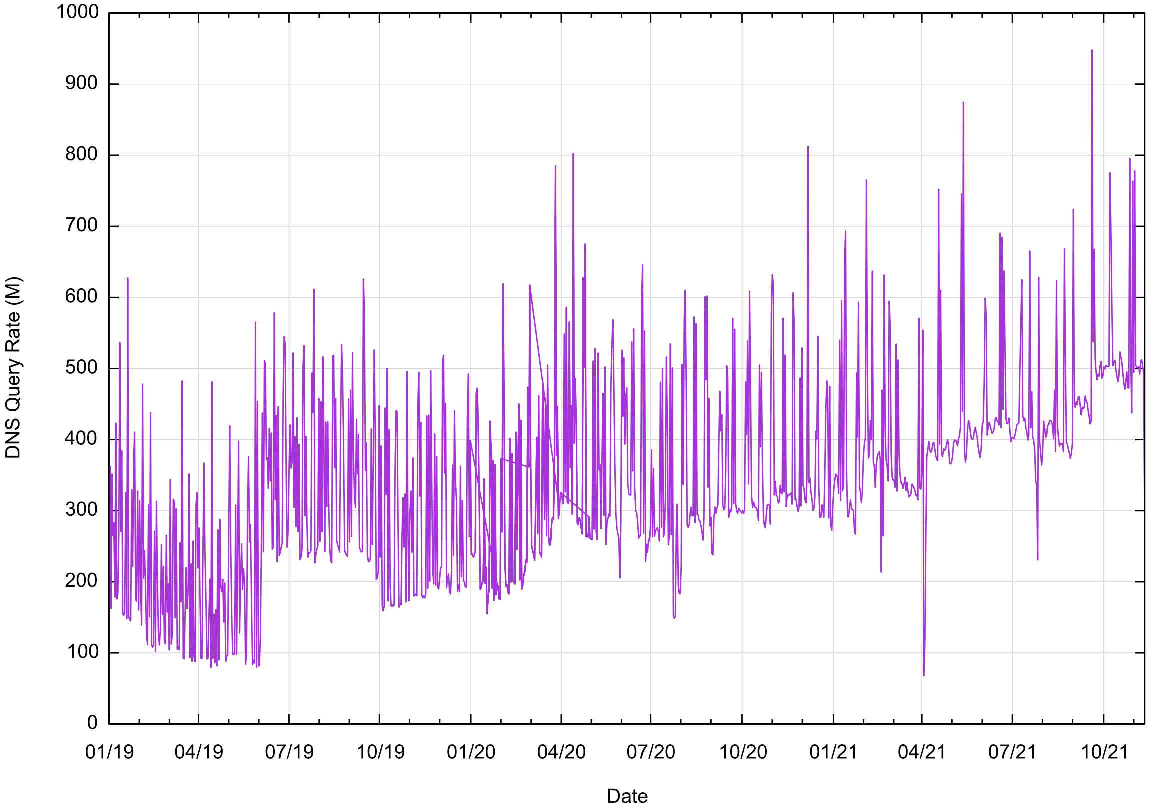
User’s IPv4 address is 203.10.60.76

Experiment Profile

Daily Ad Impression Count

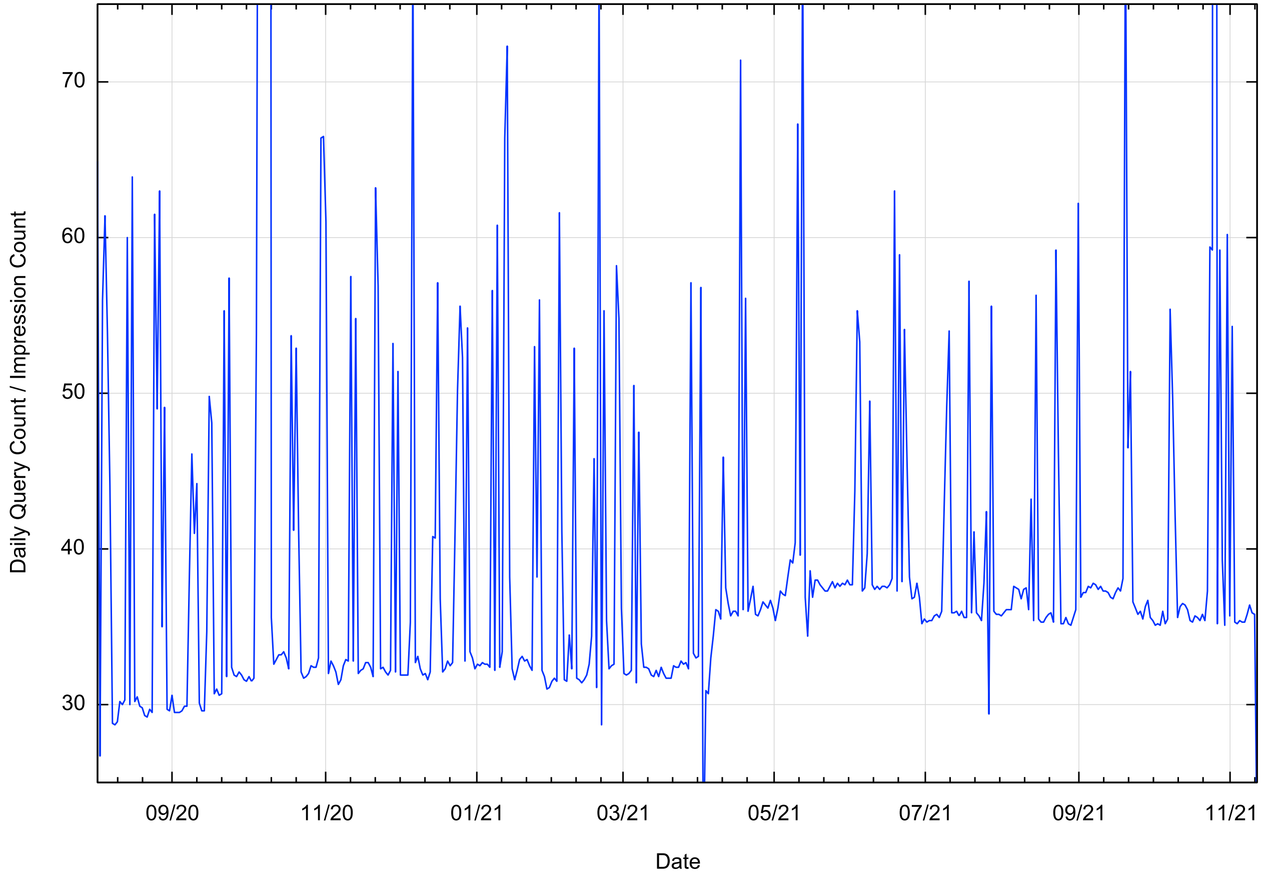


Daily DNS Query Count



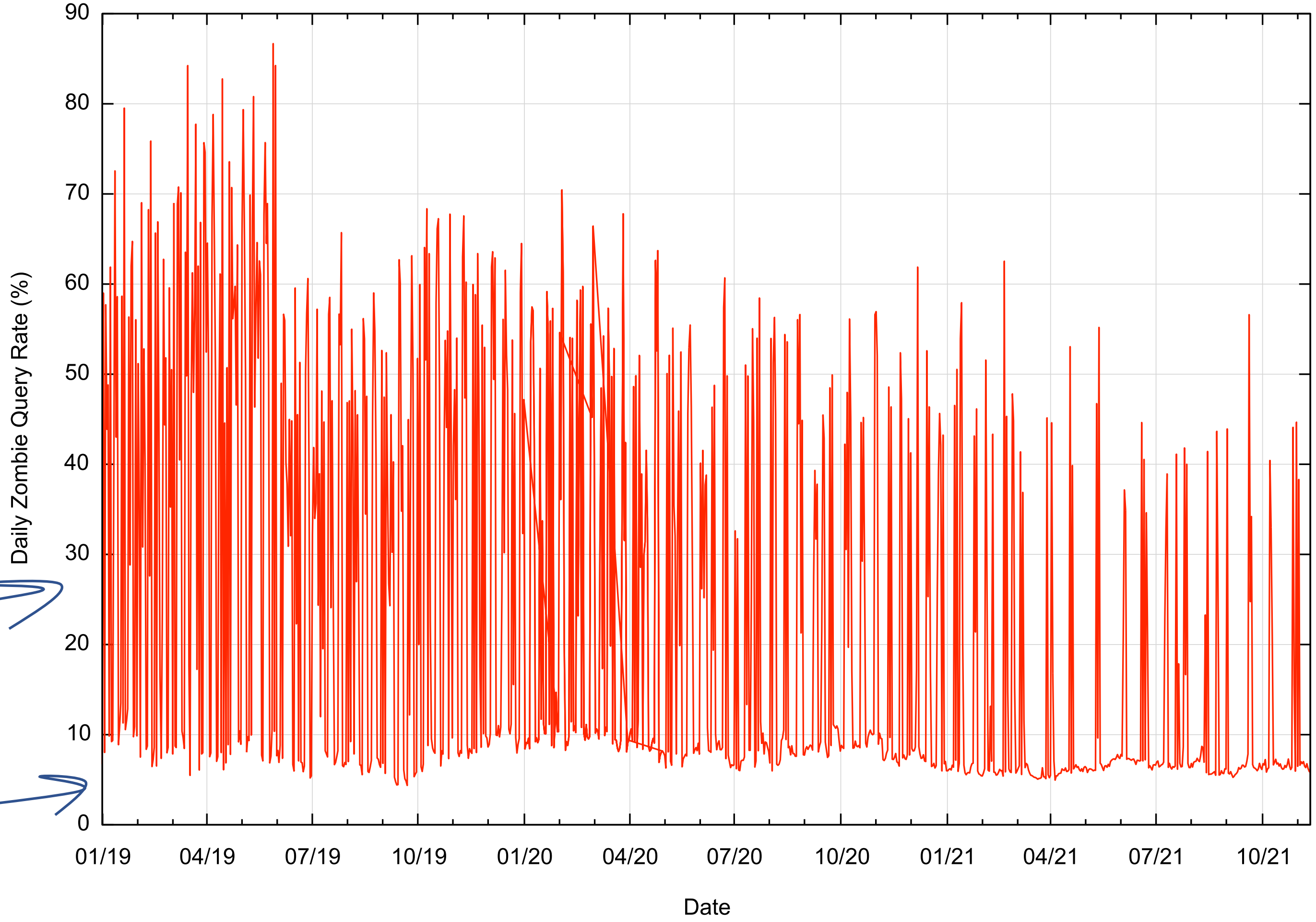
DNS Amplification

This compares the daily DNS query count against the daily new label "injection" count

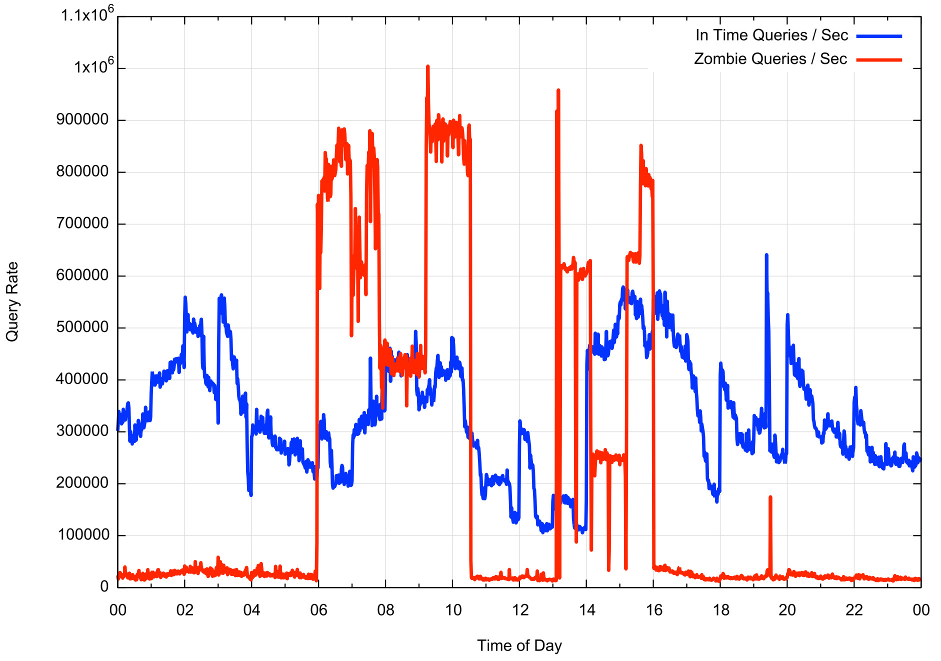


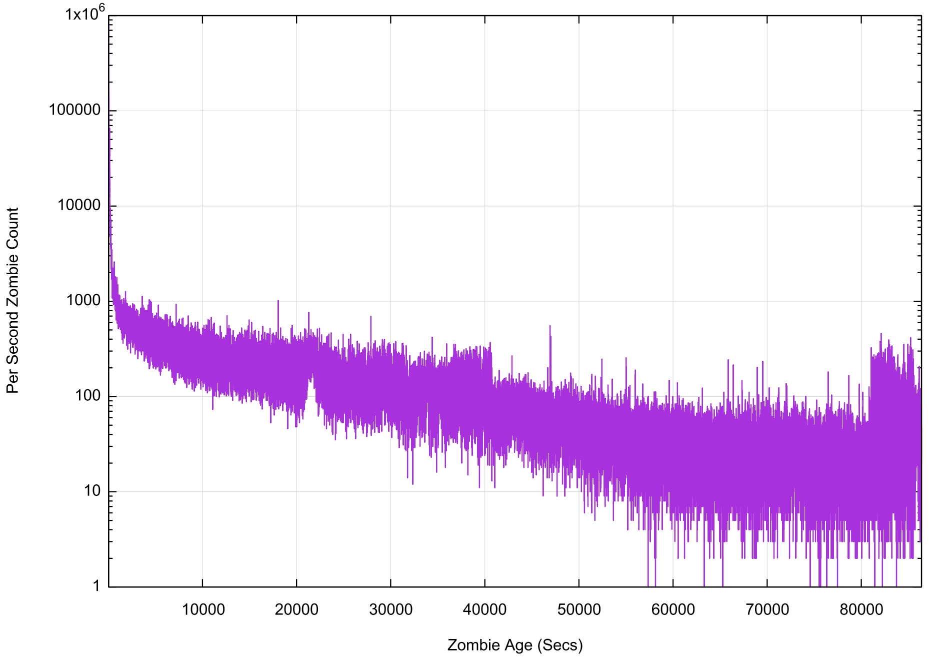
DNS Zombies per day

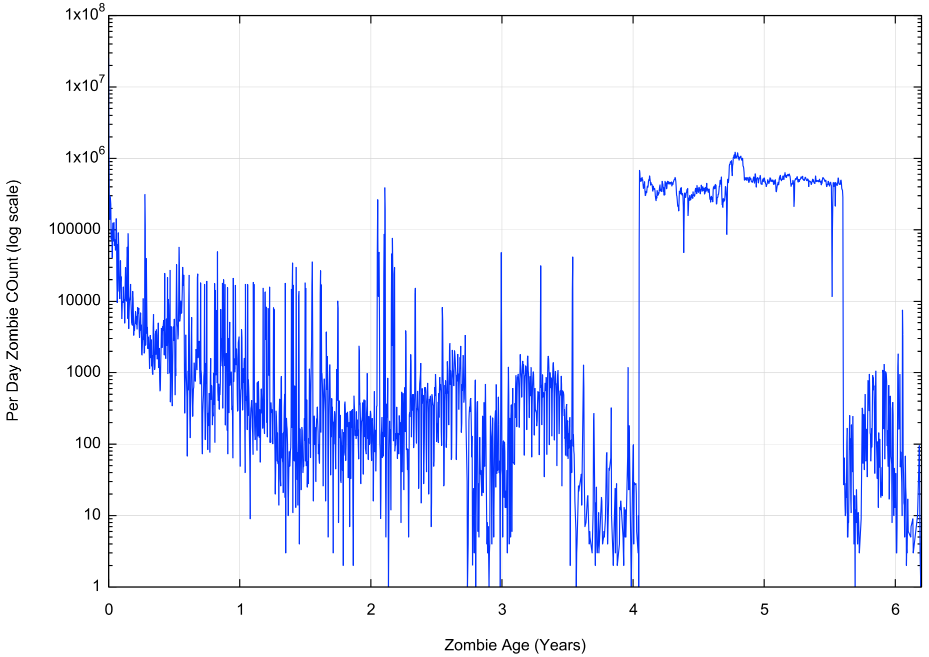
- These are single use DNS labels
- So let’s look at the queries where the time is more than 30 seconds older than the label creation time
- Some days have a high avg peak rate – some don’t – **why?**
- The zombie rate on non-peaking days is (slowly) falling – **why**?



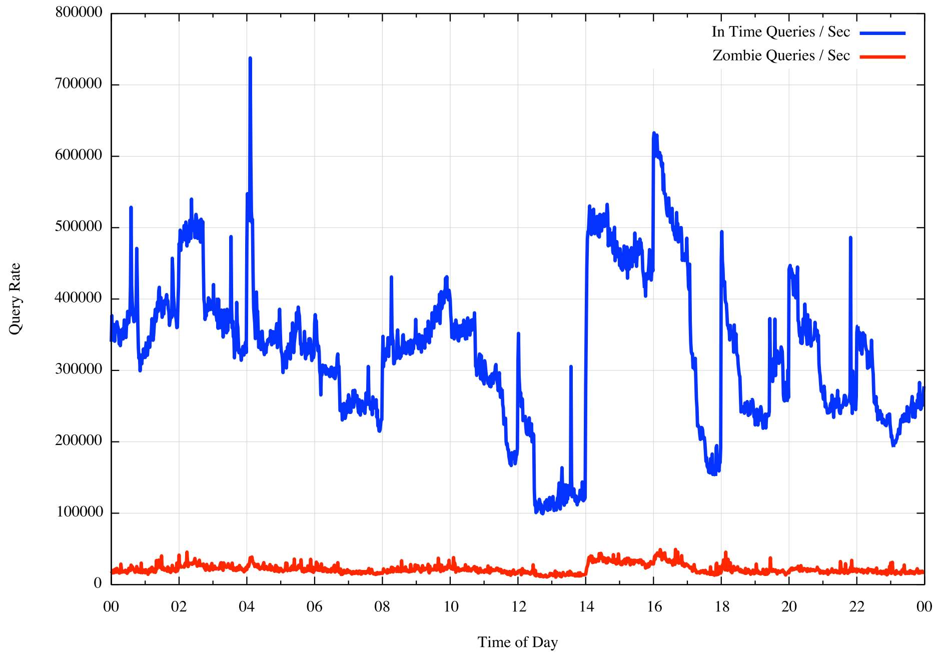
One ‘Intense’ Zombie Day

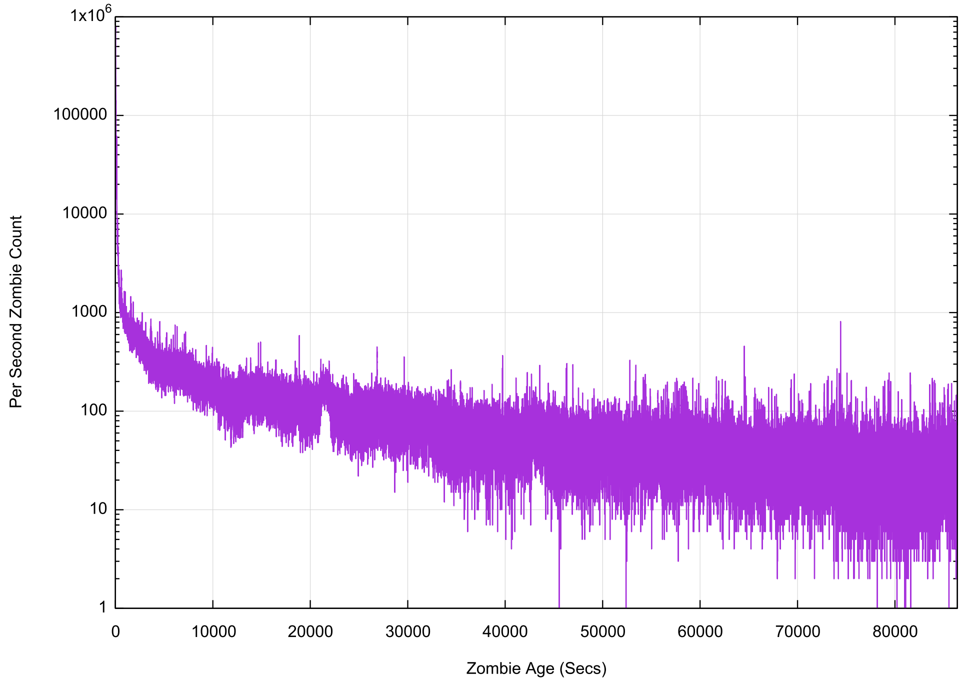


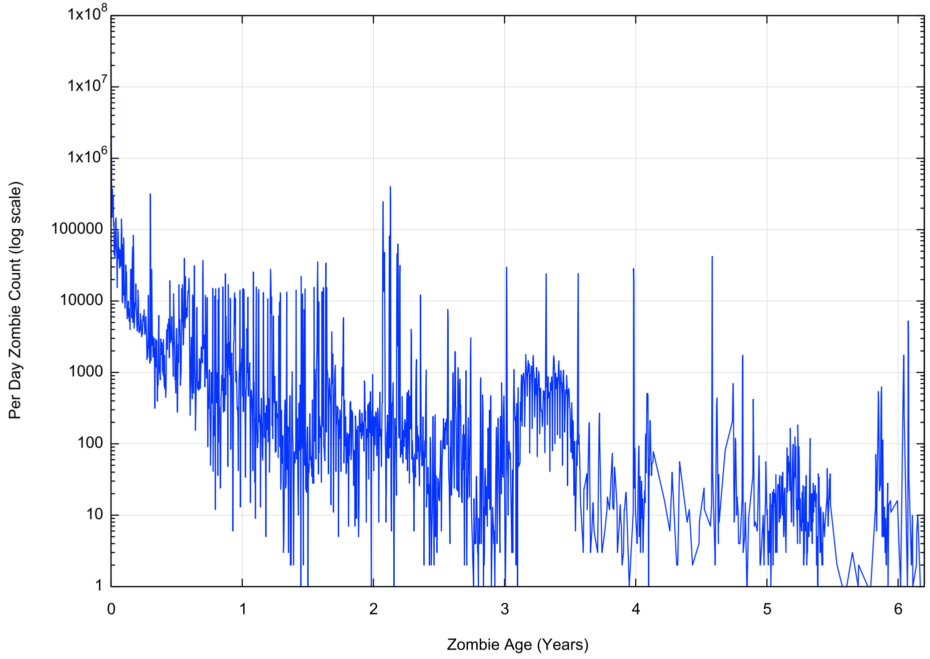




One ‘Other’ Zombie Day







Some Questions

- Some of the Zombie activity might be based on high speed query log replay
  - How prevalent is this behaviour across the entire DNS query landscape?
  - How would this impact on passive query observations
- What contributes to the background query profile?
  - Why does the background have such a long tail?

Further Studies?

- Correlate select query data from recursor(s) with query data from authoritatives
- Look for query teleportation (geo shift from original to zombie)
- How much DNS stalking is going on?
- Would changing the response code for zombie queries change the zombie query behaviour?

- How much of the query data is based on end user queries and how much on synthetic queries and DNS thrashing?