Some thoughts about

A Speculation on DNS for DDOS

Geoff Huston
APNIC

What we know about the October DYN attack…

Well – guess - from the snippets that have been released…

It was a Mirai attack
It used a compromised device collection
It used a range of attack vectors
    TCP SYN, TCP ACK, GRE, …
    One of these was DNS

DDOS Attacks

Are nothing new − unfortunately

And our response is often responding like for like

- Build thicker and thicker bunkers of bandwidth and processing capacity that can absorb the attacks
- And leave the undefended open space as toxic wasteland!

But using the DNS for attacks opens some new possibilities…

What we understand about direct DNS DDOS attacks

- These are **not** reflection/ amplification attacks
- They are directed at the authoritative name servers / root servers
- It loads the authoritative servers with query traffic
  - It can saturate the carriage / switching infrastructure of the server
  - It can exhaust the server itself of resources so it drops “legitimate” queries
- The attack queries look exactly like other queries that are seen at these servers
  - So front end pattern matching and filtering may not work
  - The qname is likely to be *<random>.target* so as to defeat caches and simple filters
  - These queries look just like Chrome’s behaviour!

The intended outcome of the attack

- Because the <*random*>.*target* qname form will defeat the recursive resolver caching function, the query is passed to the auth name server to resolve

- With an adequately high query volume, the authoritative server will start to discard queries due to resource starvation

- The result is that the *target* name will fade away on the Internet as recursive resolvers' cache entries expire, and they cannot refresh their cache from the authoritative servers

Possible Mitigations – 1

”A Bigger Bunker”

Add more *Foo*

- More authoritative name servers
- More bandwidth to authoritative name servers
- More CPU and memory to authoritative name servers

- i.e. deploy more ”foo” and try and absorb the attack at the authoritative name server infrastructure while still answering “legitimate” queries

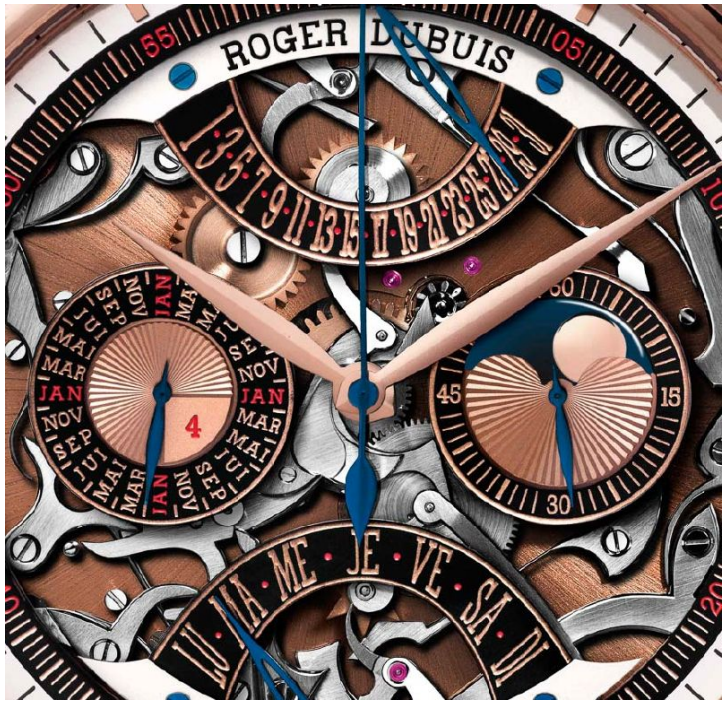
Possible Mitigations - 2

Longer TTLs:

- Low TTL's make the auth servers more vulnerable because recursives need to refer to authoritatives more frequently
- With a longer TTL, the attack will still happen, but the legit recursives may not get a cache expiry so quickly
- The recursive resolvers will still serve cached names from their cache even when the authoritative name server is offline
- Attackers will need to attack for longer intervals to cause widespread visible damage

But..

- Nobody likes to cement their DNS with long TTLs
- And current recursive resolvers don't seem to honour longer TTLs anyway!



Possible Mitigations – 3

Filter queries:

- Try to get a fix on the *<random>* name component in the queries
- Set of a front end query filter and block these queries

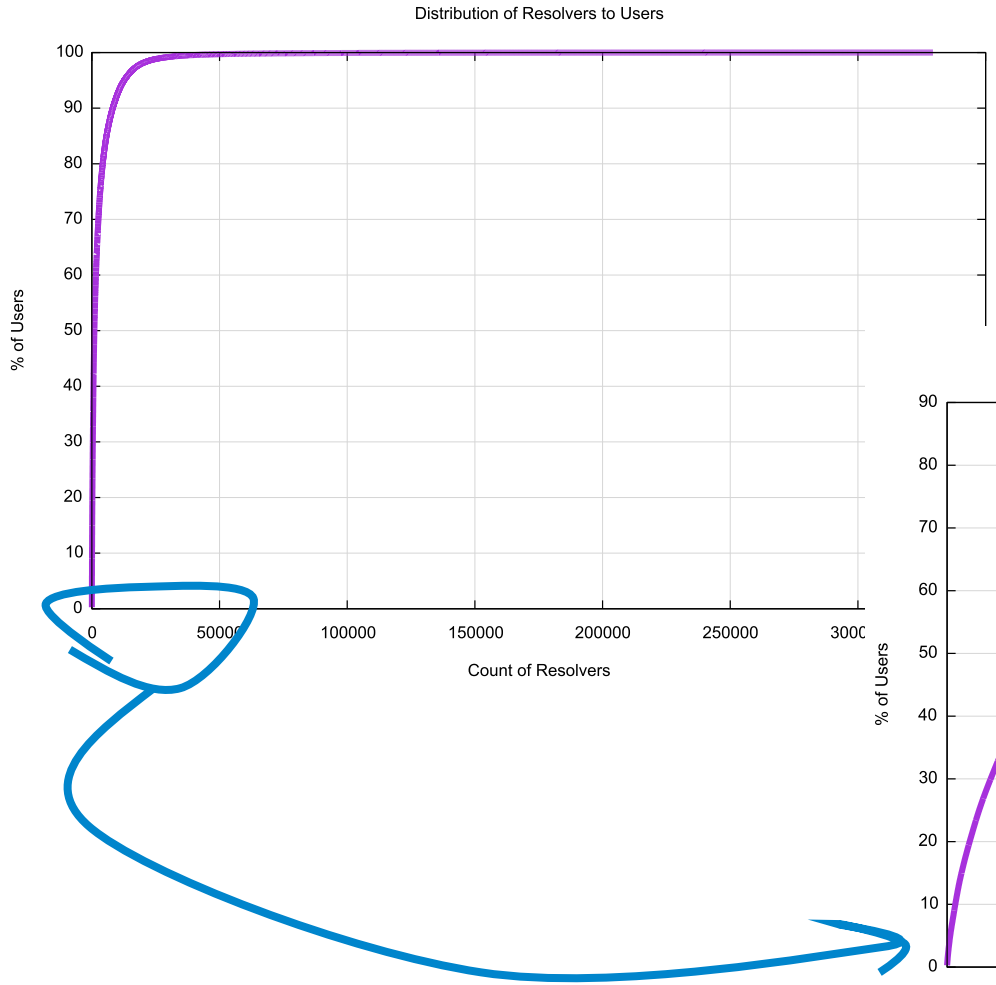
- But
  - This is just tail chasing!

Possible Mitigations - 4

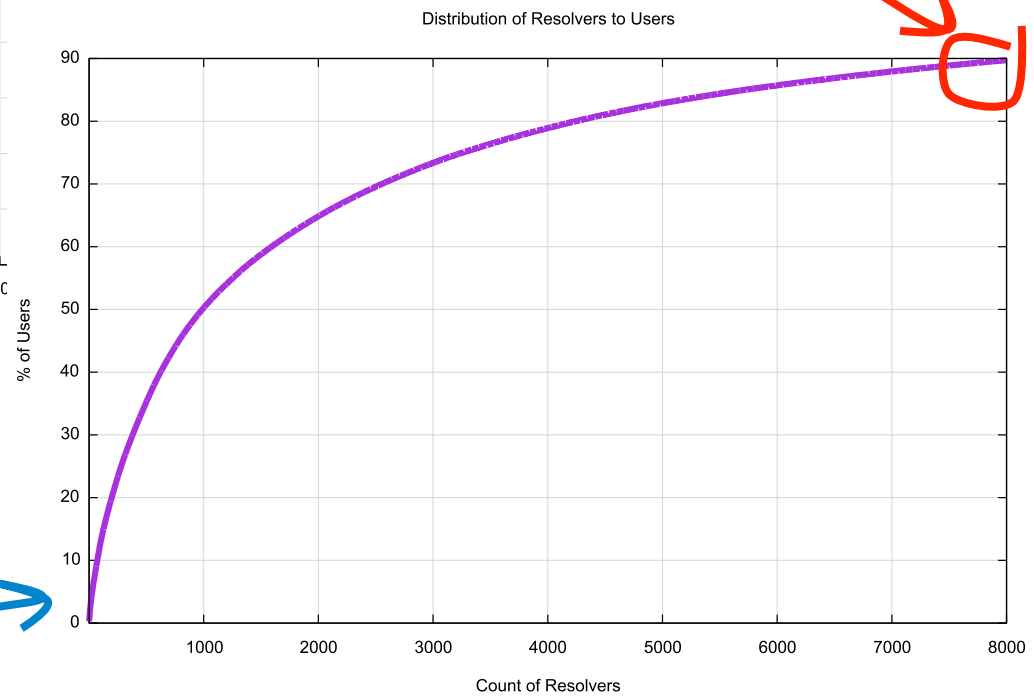
What if the attacking devices are passing the queries directly to the authoritative name servers?

Filter IP addresses!

All resolvers might be equal, but some resolvers are more equal than others!



8,000 distinct iP addresses (2.3÷ of all seen iP addrs) for resolvers serve 90÷ of all experiments



Possible Mitigations - 4

“Filter Filter Filter” (IP sources)

Only 8,000 discrete IP addresses account for more than 90% of the users’ DNS queries

These are the main recursive resolvers used by most of the internet − so its probably good to answer them!

Put all other source IP address queries on a lower priority resolution path within the authoritative name server

Divide queriers into separate queues of “Friends” and “Strangers”: Just like SMTP!

Possible Mitigations - 5

What if the devices are passing the queries via recursive resolvers?

Possible Mitigations - 5

Get assistance!

Use DNSSEC and apply *NSEC Aggressive caching* *

- The attack will generate NXDOMAIN answers
- So why not get the recursive resolvers closer to the individual devices to answer the NXDOMAIN query directly
- This can be done with the combination of DNSSEC and NSEC signing, using the NSEC span response to then respond to further queries within the span without reference to the authoritative servers
- This means that the recursive system absorbs the DNS query attack and does not refer the queries back to the auth servers

* draft-ietf-dnsop-nsec-aggressiveuse-05

If only…

- Piecemeal solutions deployed in a piecemeal fashion will see attackers pick off the vulnerable again and again
- And the long term answer is not bigger and thicker walls, as the IoT volumes will always be higher
- We need to think again how to leverage the existing DNS resolution infrastructure to be more resilient
- And for that we probably need to talk about this openly and constructively and see if we can be smarter and make a more resilient DNS infrastructure
- And for that we probably need to talk about the DNS and DNSSEC and how it works, and how it can work for us to defend these attacks