

D*

(DNS, and DNSSEC and DDOS)

Geoff Huston
APNIC

How to be bad

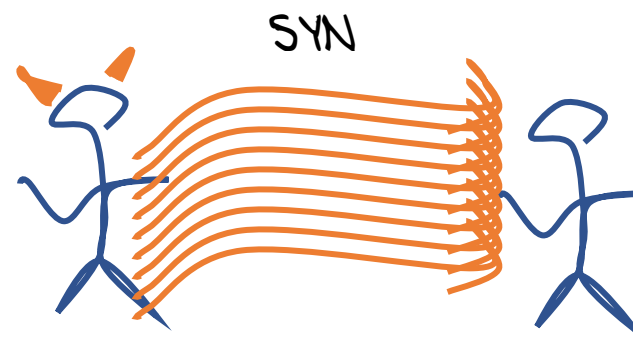


How to be bad

- Host and application-based exploits abound
And are not going away anytime soon!
- And there are attacks on the Internet infrastructure itself
 - These attacks don't compromise a service, but are intended to totally overwhelm the service or the local network such that nothing works!



How to be bad



TCP-based DDOS attacks:

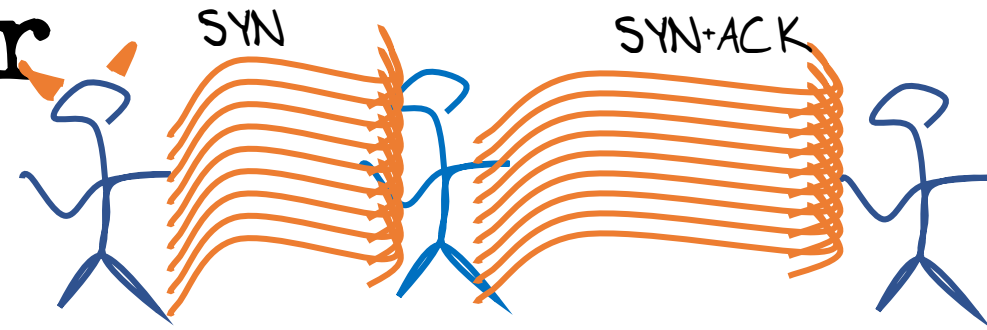
TCP SYN flooding attacks

- Try and exhaust the server's resources by saturating the server with TCP SYN packets
- Can be circumvented at the server with the use of SYN cookies

This attack was effective when servers allocated memory for a TCP process control block upon receipt of the first TCP SYN packet.

SYN cookies change the server behaviour to pass the initial server state to connecting client within the initial TCP sequence number, and no state is retained on the server.

How to be badder



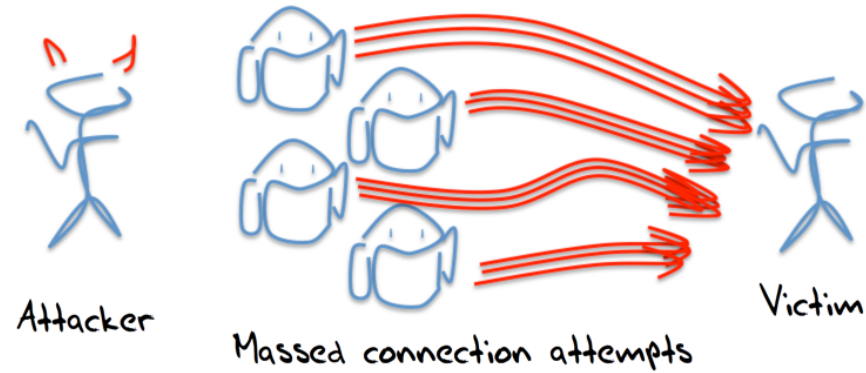
TCP-based DDOS attacks:

TCP SYN/ACK reflection attacks

- Use a spoofed source address in the initial SYN packet
- The server's SYN/ACK response will be directed to the victim's address
- This has limited attack leverage because the SYN and SYN/ACK packets are the same length
- SYN/ACK packets do not reserve state at the victim— they normally just generate a RST, if anything at all
- Widespread use of BCP38 filters would limit the extent to which coarse address spoofing is possible

Reflection attacks without amplification are of limited use!

How to be more bad



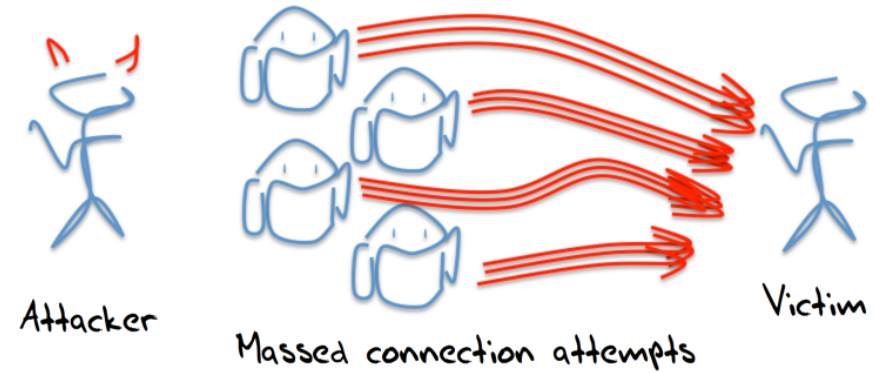
UDP-based DDOS attacks:

UDP is far easier to use for a DDOS attack

- Use a bot army to send UDP packets directly to a UDP-based server
- Or use spoofed sources to generate a reflection / amplification attack
- There are a number of cases in UDP applications where the response can be far larger than the query:
 - SNMP, NTP, chargen, finger, DNS
- Reflection/Amplification attacks can transform a small query UDP stream into a massive response stream

This has far more potential than TCP-based attacks!

How to be most bad



UDP-based DDOS attacks:

If you are going to use UDP then the DNS is the obvious weapon of choice:

- Highly suitable for amplification attacks
- Universally supported, and often permitted through firewalls
- Promiscuous servers that will attempt to respond to every query
- Individual responses are readily discarded by the victim, so the attack is useful only effective in very high volume – the attack is a form of resource starvation of the network in the region around the victim

in this case the victim is usually a DNS name server

Don't overthink it!

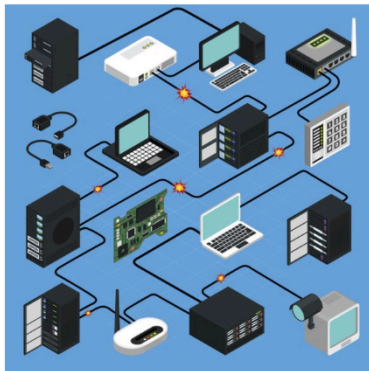
- Attacks don't need to extent any more effort than necessary
- Simple attack forms are more effective than complex ones
 - Code injection is complex – so even source address spoofing is harder to install on enlisted bots than simple scripted commands
 - Attacks will stick to simple scripted commands where possible – it's a whole lot easier than pulling in an extended hack library for a bunch of potential platforms
- Do what 'normal' traffic does
 - That way there is no clear signal of an attack traffic profile

Simple Works!



LILY HAY NEWMAN SECURITY 12.09.16 07:00 AM

THE BOTNET THAT BROKE THE INTERNET ISN'T GOING AWAY



© THEN ONE/WIRED

WHEN THE BOTNET named Mirai first appeared in September, it announced its existence with dramatic flair. After flooding a prominent security journalist's website with traffic from zombie Internet of Things devices, it managed to make much of the internet unavailable for millions of people by overwhelming Dyn, a company that provides a significant portion of



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

Dyn, the victim of last week's denial of service attack, said it was orchestrated using a weapon called the Mirai botnet as the 'primary source of malicious attack'

● Major cyber attack disrupts internet service across Europe and US



HOME ABOUT US CAREERS PUBLICATIONS ALERTS AND TIPS RELATED RESOURCES C' VP

Alert (TA16-288A)

Heightened DDoS Threat Posed by Mirai and Other Botnets

Original release date: October 14, 2016 | Last revised: October 17, 2017

Print Tweet Send Share

Systems Affected

Internet of Things (IoT)—an emerging network of devices (e.g., printers, routers, video cameras, smart TVs) that connect to one another via the Internet automatically sending and receiving data

Overview

Recently, IoT devices have been used to create large-scale botnets—networks of devices infected with self-propagating malware—that can execute or distributed denial-of-service (DDoS) attacks. IoT devices are particularly susceptible to malware, so protecting these devices and connected hardware protect systems and networks.

Description

On September 20, 2016, Brian Krebs' security blog (krebsonsecurity.com) was targeted by a massive DDoS attack, one of the largest on record, exceeding gigabits per second (Gbps). [1] An IoT botnet powered by Mirai malware created the DDoS attack. The Mirai malware continuously scans the Internet

The DNS

- If you want to cause maximal impact then attacking the DNS is a logical choice
- Every application uses the DNS
- If you want to disrupt users and the apps that they run then you need to turn to the DNS and try and disrupt the DNS
- If you want to be maximally ambitious, then attack the root itself
- Why would one attack the root of the DNS?

Resolving a DNS Name

Your resolver needs need to ask a DNS server for the zone that contains the terminal label for the associated information (resource record) associated with the DNS name

But...

Where exactly is the zone cut?

Who are the servers?

Resolvers *discover* this information by performing a top-down iterative search...

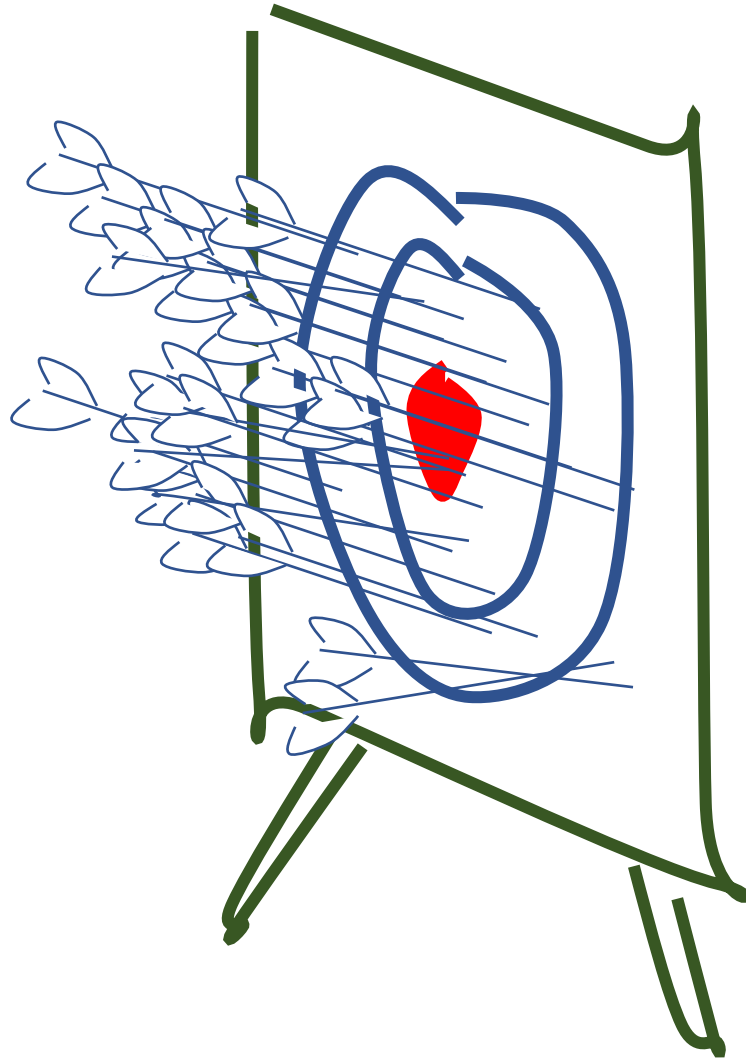
How to be bad

Every DNS
resolution procedure
starts with a query
to the root!

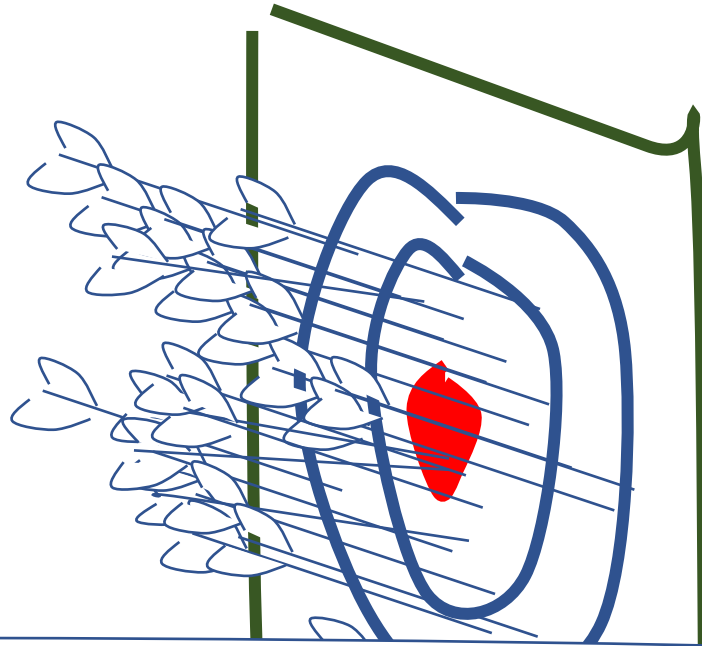


if an attacker could prevent the root servers
from answering DNS queries then the entire
internet will suffer!





it's no surprise that the
DNS Root Servers are a
highly visible attack target



it's no surprise that the
DNS Root Servers are a
highly visible attack target

if you can prevent resolvers from getting
answers from the root then the resolvers will
stop answering all queries as their local cache
expires



1 March 2007

Factsheet

Root server attack on 6 February 2007

Executive summary

- The Internet sustained a significant distributed denial of service attack, originating from the Asia-Pacific region, but withstood it.
- Six of the 13 root servers that form the foundation of the Internet were affected; two badly. The two worst affected were those that do not have new Anycast technology installed.
- The attacks highlighted the effectiveness of Anycast load balancing technology.
- More analysis is needed before a full report on what happened can be drawn up. The reasons behind the attack are unclear.

On 6 February 2007, starting at 12:00 PM UTC (4:00 AM PST), for approximately two-and-a-half hours, the system that underpins the Internet came under attack. Three-and-a-half hours after the attack stopped, a second attack, this time lasting five hours, began.

Fortunately, thanks to the determined efforts of engineers across the globe and a new technology developed and implemented after the last DNS attack of this size, on 21 October 2002, the attack had a very limited impact on actual Internet users.

This factsheet provides the most important details of the attack and briefly explains how the domain name system works and the systems in place to protect it. It also outlines how such attacks are possible and discusses possible solutions to future attacks.

What happened?

The core DNS servers of the Internet were hit with a significant distributed denial of service attack, or DDoS. In such an attack, billions of worthless data packets are sent from thousands of different points on the Internet to specific computer servers in order to overwhelm them with requests and so disrupt the smooth running of the Internet.

surprise that the
Loot Servers are a
visible attack target

if you can
answers fr
stop answe
expires



Events of 2015-11-30

1 March 2007

Abstract

On November 30, 2015 and December 1, 2015, over two separate intervals, several of the Internet Domain Name System's root name servers received a high rate of queries. This report explains the nature and impact of the incident.

While it's common for the root name servers to see anomalous traffic, including high query loads for varying periods of time, this event was large, noticeable via external monitoring systems, and fairly unique in nature, so this report is offered in the interests of transparency.

1. Nature of Traffic

On November 30, 2015 at 06:50 UTC DNS root name servers began receiving a high rate of queries. The queries were well-formed, valid DNS messages for a single domain name. The elevated traffic levels continued until approximately 09:30 UTC.

On December 1, 2015 at 05:10 UTC DNS root name servers again received a similar rate of queries, this time for a different domain name. The event traffic continued until 06:10 UTC.

Most, but not all, DNS root name server letters received this query load. DNS root name servers that use IP anycast observed this traffic at a significant number of anycast sites.

The source addresses of these particular queries appear to be randomized and distributed throughout the IPv4 address space. The observed traffic volume due to this event was up to approximately 5 million queries per second, per DNS root name server letter receiving the traffic.

2. Impact of Traffic

The incident traffic saturated network connections near some DNS root name server instances. This resulted in timeouts for valid, normal queries to some DNS root name servers from some locations.

the attack are unclear. Internet.



ig at 12:00 PM UTC (4:00 AM PST), for hours, the system that underpins the free-and-a-half hours after the attack time lasting five hours, began.

etermined efforts of engineers across y developed and implemented after the 21 October 2002, the attack had a very net users.

most important details of the attack and in name system works and the systems tlines how such attacks are possible and future attacks.

met were hit with a significant distributed denial n attack, billions of worthless data packets are nts on the Internet to specific computer servers quests and so disrupt the smooth running of the

surprise that the
Loot Servers are a
visible attack target

if
ansi
stop
exp

}}
?}

Caching in the DNS

The main role of the DNS server system is to answer queries that are not cached in local name caches

If it wasn't for caching the DNS would not be here today!

How to be bad

To attack a name's servers you need to get past DNS resolvers' caches.

This means you need to have every query in the DNS attack flow ask for a different non-existent name

So how can we defend ourselves from attack?



How should we defend the DNS?

- Larger Server platforms?
- More Authoritative Servers?
- More Anycast Instances?
- Change Server response behaviours?
- Or...

How should we defend the DNS?

- Larger Set *Can't scale * THIS?*
- More Authoritative Servers?
- More Anycast Instances?
- Change Server response behaviours?
- Or...

** Distributed parallel attacks can scale up in intensity more effectively than a single point of service can scale its defence mechanisms*

How should we defend the DNS?

- Larger Set *Can't scale **
- More Mirrors *Err, umm - well no **
- More Anycast Instances?
- Change Server response behaviours?
- Or...

- Longer lists of servers for a name make name resolution slower, not faster. So its probably a bad idea

How should we defend the DNS?

- Larger Se *Can't scale **
- M *Err, umm - well no **
- More Anycast Instances?
- Change Server response behaviours?
- Or...

Today's practice for many hosted DNS servers

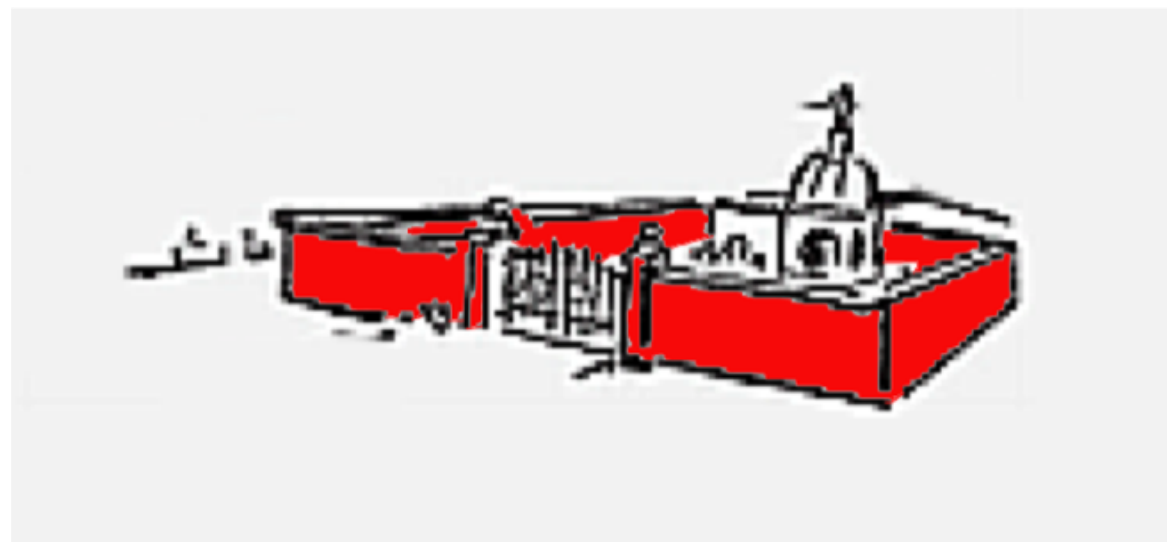
How do we defend the DNS today?

As the traffic levels to DNS servers increases both as steady state query levels and instances of attacks, we keep on adding more instances to the existing anycast clouds and spend more money on deploying larger and more distributed servers

The attacks get bigger



Our defence is to build bigger walls!



But the attackers are our own recursive resolvers!

We are scaling the DNS root server infrastructure in order to be resilient against floods of queries about non-existent random names coming from the existing DNS resolvers, who are scaling their own capabilities to survive the very same query attacks that are being directed against them!



How do we defend the DNS today?

As the traffic levels to DNS servers increases both as steady state query levels and instances of attacks, we keep on adding more instances to the existing anycast clouds and spend more money on deploying larger and more distributed servers

What we are in effect doing is building ever bigger and larger trash processors to handle ever larger amounts of garbage queries to cope with ever larger attacks via the DNS!

How do we defend the DNS today?

Can we jump out of this vicious cycle?

Can we change the behaviour of the DNS system to improve both its service and its resilience?

DNSSEC changes Everything

Before DNSSEC we relied on the assumption that if we asked an IP address of a root server, then the response was genuine

With DNSSEC we can ask anyone, and then use DNSSEC validation to assure ourselves that the answer is genuine

How can we use this?

Caching NXDOMAIN

If we could answer NXDOMAIN queries from recursive resolvers we could reduce the load on the DNS servers

For the root zone we've measured this to be close to 70%

NXDOMAIN would be a very significant win:

- reducing root query traffic
- providing faster response to these queries
- reduces the local cache load on recursive resolvers

NSEC caching - RFC 8198

- A DNSSEC-signed NXDOMAIN response actually describes a *range* of labels that do not exist, and it's the *range* that is signed, not the actual query name
- If resolvers cached this range and the signed response, then they can use the same signed response to locally answer a query for any name that falls within the same label range

NSEC caching

For example, if you were to query the root server for the non-existent name `www.example.` the returned response from the root says that there are NO TLDs between `everbank.` and `exchange.`

The same response can be used to respond to queries for every TLD between these labels.

So we can cache this range response and use it to respond to subsequent queries that fall into the same range

```
[gih@rongrong ~]$ dig +dnssec @f.root-servers.net www.example.

; <<> DiG 9.11.0-P3 <<> +dnssec @f.root-servers.net www.example.
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 59536
;; flags: qr aa rd; QUERY: 1, ANSWER: 0, AUTHORITY: 6, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
; COOKIE: e8aee4619b3dd9cb37c892d65994b66428d99e23452b3c80 (good)
;; QUESTION SECTION:
;www.example.                IN      A

;; AUTHORITY SECTION:
.                86400    IN      SOA     a.root-servers.net. ns
.                86400    IN      RRSIG   SOA 8 0 86400 20170829
CBUwMAzLH0P LYwBwfGwQrpZhBiHeWcqLhC8d8MiDcq6KzffL5mjo5kgJyg6d0MzrPL B
b9DhXMrgMFKICxxj3ePN7Ebrb0iw6lWnlms+w THQFHfXvE7HBZyYkOv9DNQxNNNM0hEuV
vxENYm VL2Iew==
.                86400    IN      NSEC    aaa. NS SOA RRSIG NSEC
.                86400    IN      RRSIG   NSEC 8 0 86400 2017082
j6FIp0yK0+yb MQqjiLwymEqURbVc+Lm1lCu5HZ6p/6s1iagYoAZBSBZWubmq4bGQBGwD
tJ0yX8Xhi3ga5+gT93wyEZTwGsH3tWqiHeGc3N vp2Qsc4Crf9cZ2Np9bUJqfKozpLNMHC
kf/lnYR VJZzDA==
everbank.        86400    IN      NSEC    exchange. NS DS RRSIG
everbank.        86400    IN      RRSIG   NSEC 8 1 86400 2017082
W/CDza/huRXL 2125SgCXY2wYLba0z4ohFqIdC9gLwVuqi5gKNA2Dvr09oy0f+Mp3/kP9
AiYhd1Apg0nw6Aa0FKlj0PKSTQpJYQfPc19B5q z41q47lXu0vNW2u4L2IjiQE0IoqSx7E
Gix2cN3 JHI/XQ==

;; Query time: 1 msec
;; SERVER: 2001:500:2f::f#53(2001:500:2f::f)
;; WHEN: Wed Aug 16 21:17:24 UTC 2017
;; MSG SIZE rcvd: 1065
```


Architecturally speaking...

- Rather than have recursive resolvers act as “amplifiers” for DNS queries for non-existent names, NSEC caching enlists these recursive resolvers to act on behalf of the zone’s authoritative servers, and provide the answers for them.
- This approach uses existing DNS functionality and existing queries – there is nothing new in this.
- The change here is to take advantage of the use of the NSEC response to define a range of names, allowing what is in effect semi-wildcard cache entries that can be used to respond to a range of query labels

Impacts...

- Rather than trying to expand the capabilities of the root zone servers, we can leverage the massive number of already deployed recursive resolvers to extend their cache to cover both defined and non-existent root labels
- We anticipate that this will have a major effect on the DNS by absorbing most of the current root query load at the edge, rather than passing these queries into the root system

Impacts...

- Its not just the Root Zone – its **all** signed zones
- This is a general approach that also provides the same level of projection to other servers in the DNS from the same form of random name query attack

Impacts...

NSEC caching can also help recursive resolvers:

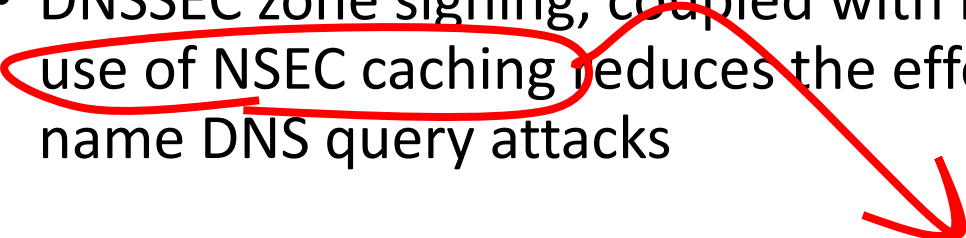
- Instead of caching non-existent individual names they can cache the NSEC-described range, and refresh the cached NSEC record instead of any individual name
- This will shrink the demands placed on the local cache, which can improve local cache performance in the recursive resolver

There is no silver bullet for DNS DDOS

- But we can take incremental steps to decrease the effectiveness of some of these DNS DDOS attacks
 - BCP38 source address filtering reduces the ability to mount DNS reflection / amplification attacks that leverage open DNS resolvers
 - Shutting down open DNS resolvers would be good too!
 - DNSSEC zone signing, coupled with resolver DNSSEC validation and resolver use of NSEC caching reduces the effectiveness of various forms of random name DNS query attacks

There is no silver bullet for DNS DDOS

- But we can take incremental steps to decrease the effectiveness of some of these DNS DDOS attacks
 - BCP38 source address filtering reduces the ability to mount DNS reflection / amplification attacks that leverage open DNS resolvers
 - Shutting down open DNS resolvers would be good too!
 - DNSSEC zone signing, coupled with resolver DNSSEC validation and resolver use of NSEC caching reduces the effectiveness of various forms of random name DNS query attacks



APNIC has sponsored the inclusion of this NSEC caching code in the forthcoming Bind 9.12 release. This function will be enabled by default in this release

So, we can improve this situation!

- But to do that, we all need to take some steps here

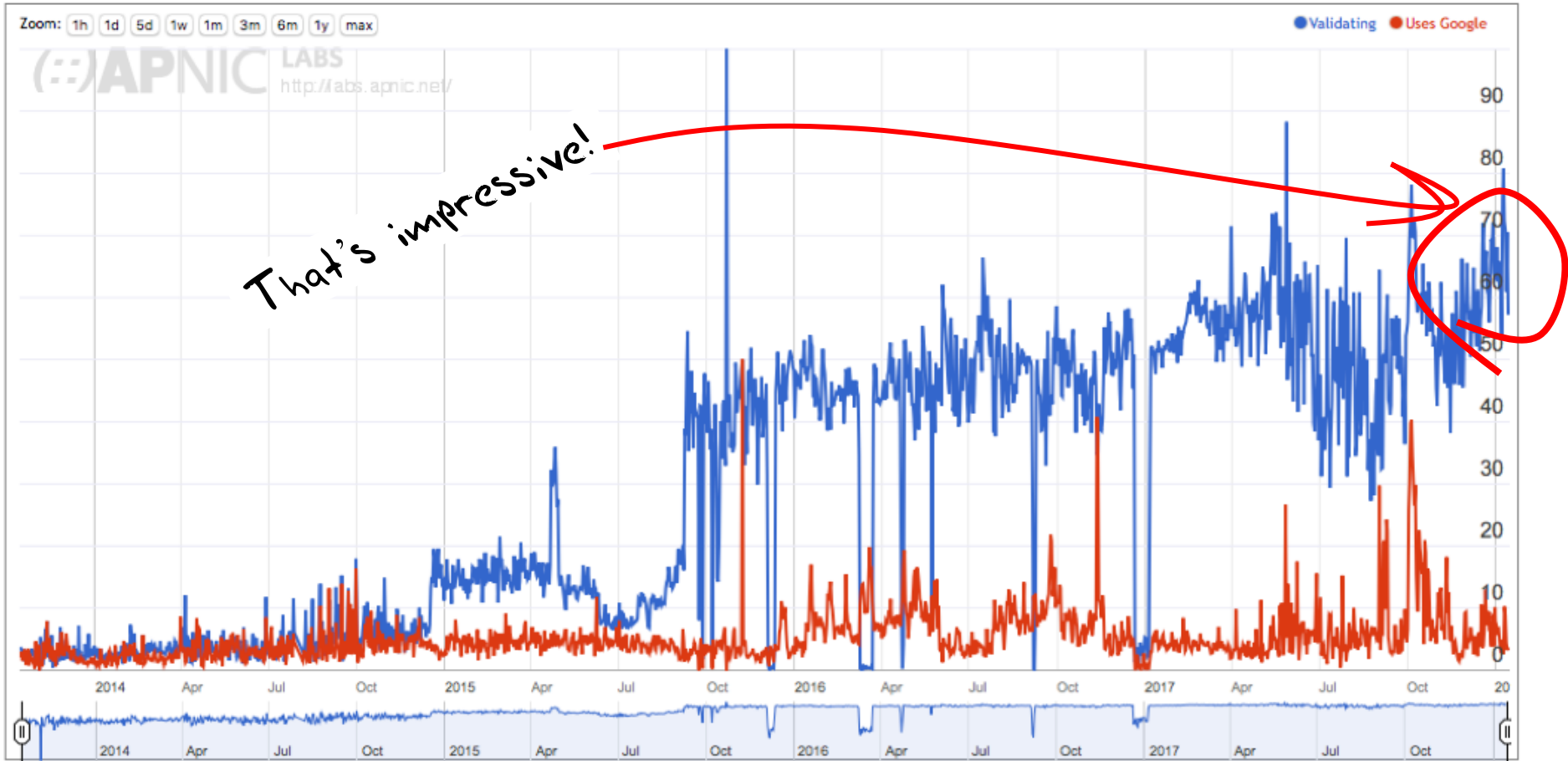
So, we can improve this situation!

- But to do that, we all need to take some steps here

How well is New Zealand DNS infrastructure doing with this?

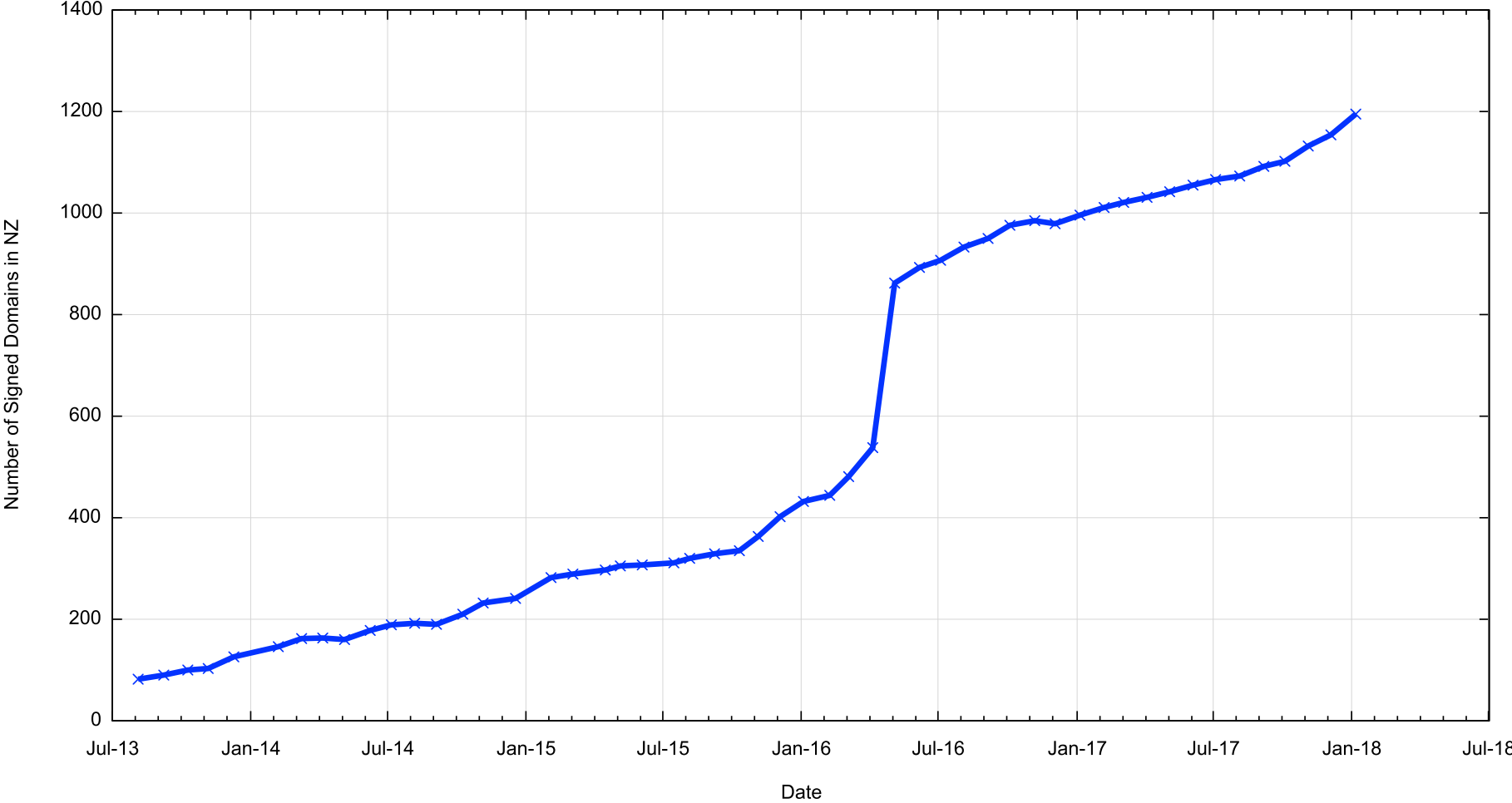
DNSSEC in New Zealand

Use of DNSSEC Validation for New Zealand (NZ)



DNSSEC in New Zealand

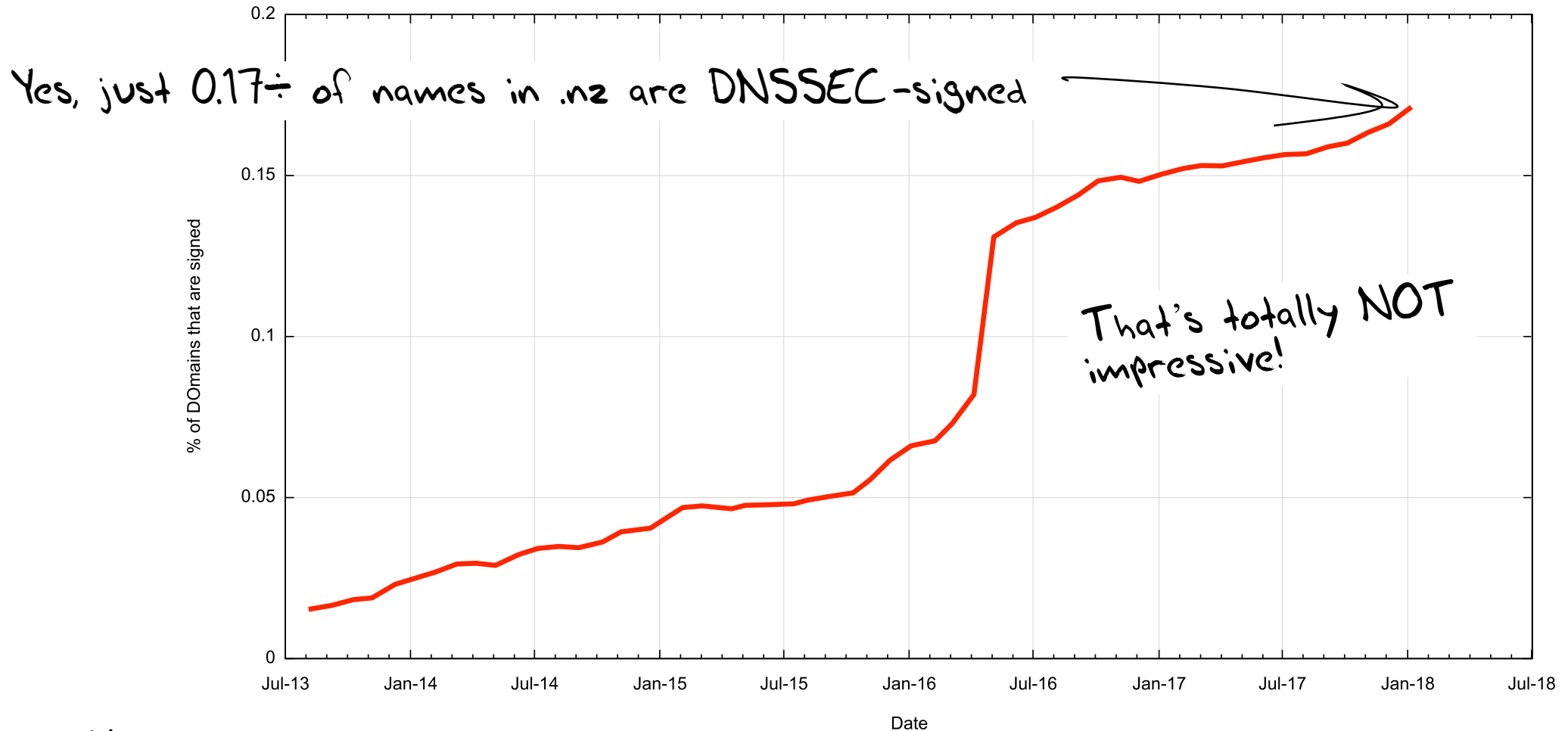
Count of DNSSEC signed zones in .NZ



Data Source: idp.nz

DNSSEC in New Zealand

Relative Count of DNSSEC signed zones in .NZ



Thanks!