



**ORACLE** + Dyn

# Dyn, DDoS, and DNS

---

Andrew Sullivan  
ICANN 58 Tech Day

13 March 2017  
Copenhagen DK

Many thanks to Chris Baker

# Some attacks

- Attacks against Dyn DNS infrastructure
- Noted because of dependent systems
- AKA “customers”
- I did no direct work on this
- Our NOC and ops did — as NOCs and ops do everywhere — great work under hard conditions
- Leads me to revisit some questions about Internet architecture

# Significant infrastructure

- Anycast-based DNS system
  - Mostly transit rather than peering based
  - At least two transit providers in all sites
  - Transit carefully arranged among sites for resilience
  - Transit diversity to avoid peering-based failures
- Large global installation in 18 sites

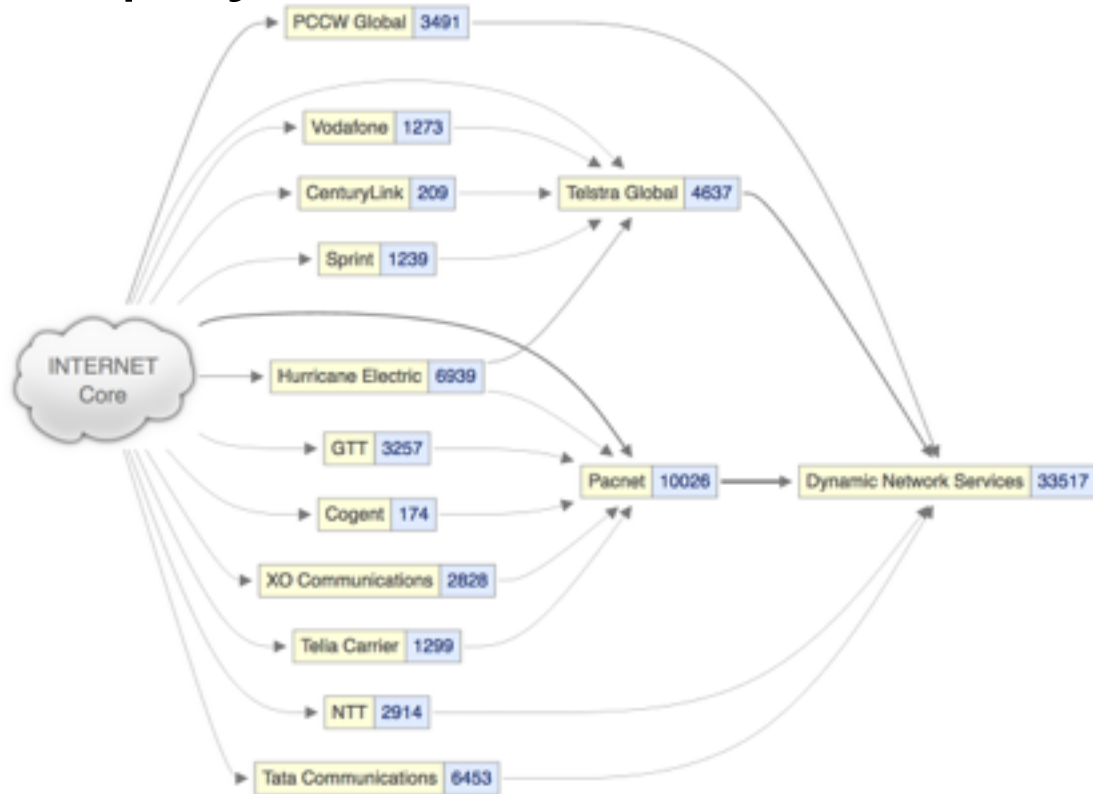
# Network Deployment: Anycast



# Network Deployment: Transit Based

- Right now this might not seem all that important to the larger narrative but it will later:
- Fundamentally shapes the path traffic takes to reach our edge
- We purchase transit from tier 1 providers around the globe
- The relationship between other ISPs and our transit providers affects the way traffic enters our network

# Network Deployment: Transit Based



# Economics of the endpoint

It is true that botnets aren't new and specifically that DDoS botnets aren't new

- Smokejumper / Tsunami
- Perl / IRC bots

It is also true that there are a number of other ways to monetize an infected system

- Click fraud
- SOCKS as a Service
- RDP as a Service
- Ransomware

# Hosting and Bandwidth

Client operating system: LINUX

Client IP (v4):

Client AS (v4):

IPv4 Probes: 87

Private IPv4 address spoofing status	received
Routable IPv4 address spoofing status	received
Largest neighbor prefix that can be spoofed	/8

**Servers: Vetted bulletproof – minimize down time**

- On Shore \$150 - 190
- Off Shore \$85 - \$125
- 1 GBps port
- 10 TB ( ~ \$30 )

*The CAIDA Spoofing report serves as a “CarFax” for the provider*

- Proof of Spoofability



# Booters/Stressers



Buy from  
provider  
→  
Just check CAIDA  
to verify spoofing  
allowed



Attacker Goal: Orchestrate on demand volumetric denial of service attacks ( NTP, SSDP, DNS, TFTP, Valve, TeamSpeak ... etc )

Exploits fundamental open design of Internet protocols

# Booter/Stresser Interface

## Stresser

☰ Stress Test Tool

Host:

Port:

Time:

Method:

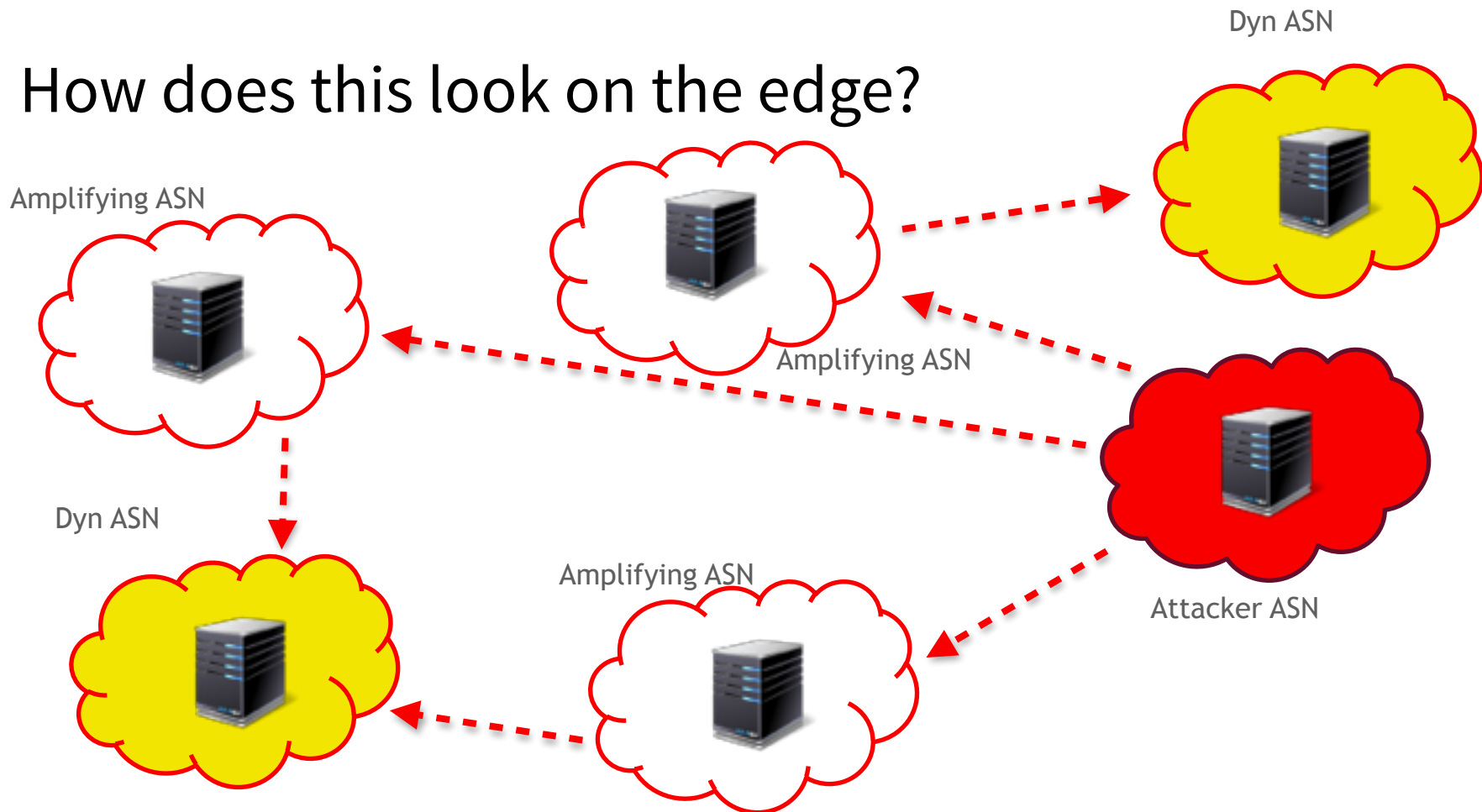
- Layer4*
- SSDP
- NTP
- UDP
- UDP LAG
- CHARGEN
- SSYN
- ESSYN
- Layer-7*
- XML-RPC
- JOOMLA

OPENCLOSE ENEMIES

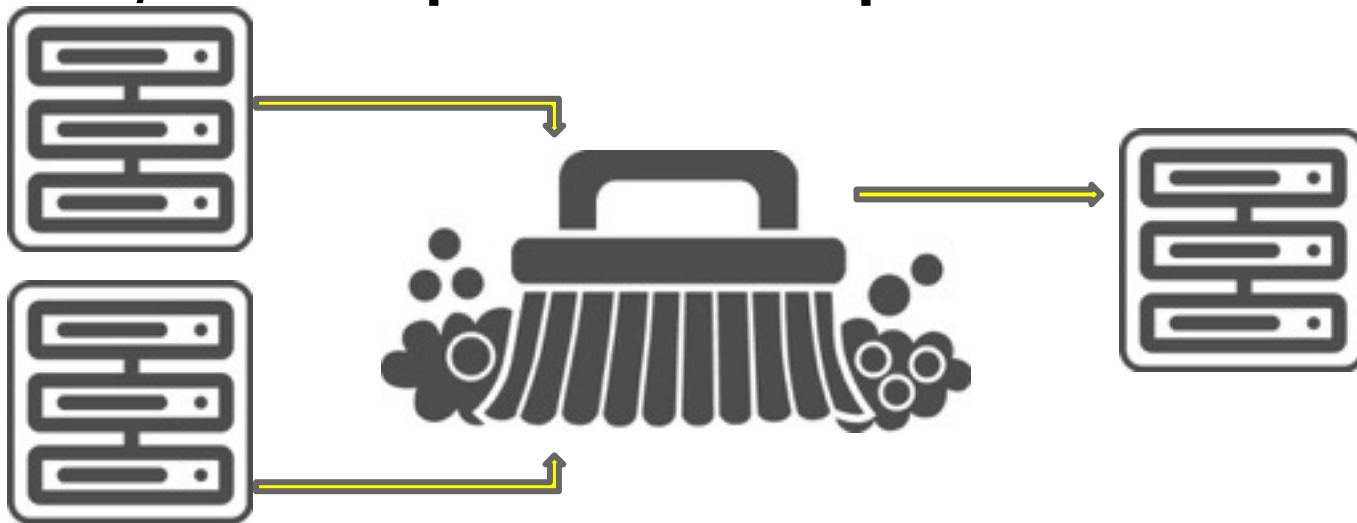
SEND ATTACK

Copyright © 2015 Rage Booter

# How does this look on the edge?



# Vendors / transit providers improve scrubbing tech



Service providers increase connectivity to soak up and scrub attacks

“Average bandwidth to cause downtime was only 4.3 GB/s” - Security Compass  
(Numbers make us all feel more comfortable!)

# Narrowing the population

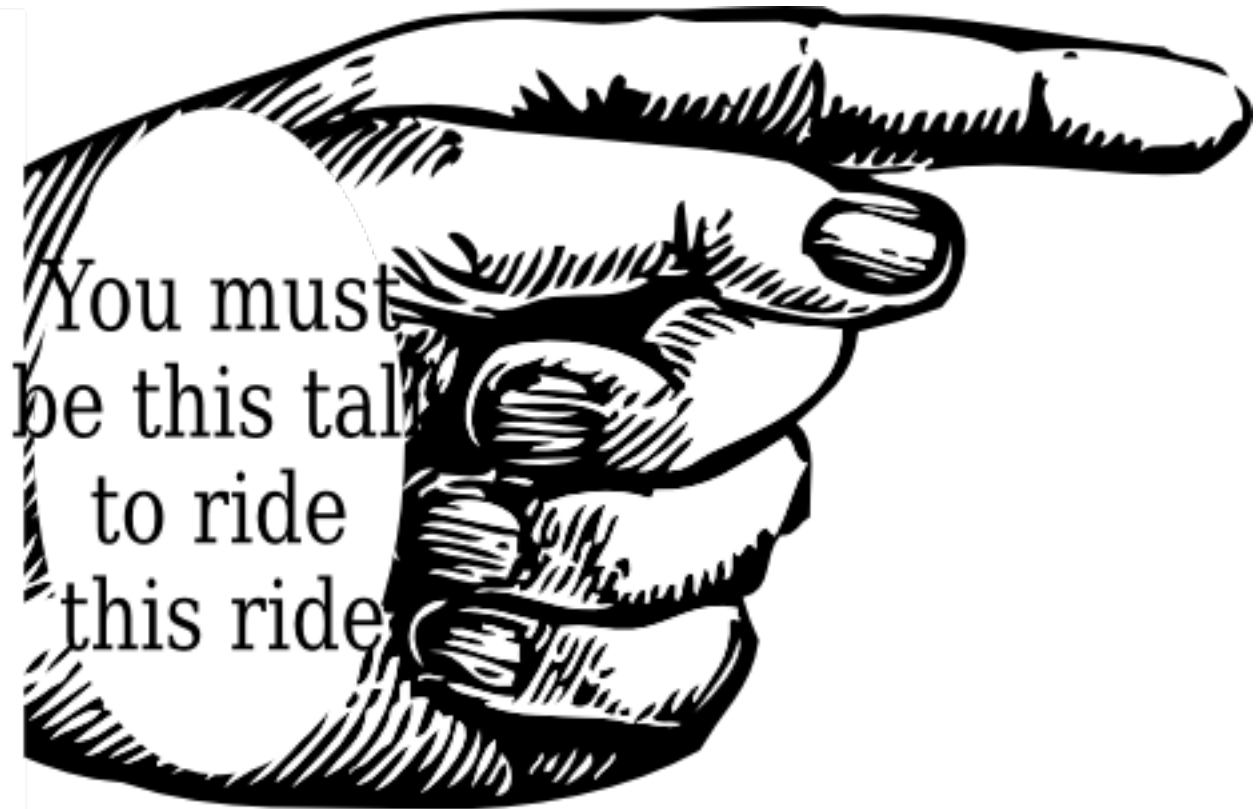
ALL Inbound Internet Traffic

Traffic to approved ports and services

\*Legitimate Traffic\*

Services





You must  
be this tall  
to ride  
this ride

# Other things to consider

Preceding about 80% of the whole of “attack traffic”. Remaining 20% of “attack traffic”:

- Fundamentally broken devices
  - Equipment which doesn't understand EDNS0 and TCP retry semantics
- DNS Edge cases
  - *Lots* of lame delegations
- Botnets
  - Authoritative Exhaustion DNS Attacks
  - Distributed non-spoofed TCP attacks

A blurred background image showing a person's hands typing on a laptop keyboard. A white coffee cup sits on the desk in front of the laptop. The scene is lit with warm, natural light, possibly from a window on the left.

**ORACLE** + Dyn

# Authoritative exhaustion

---

August 2016



# August – New Patterns

- In early August we started to see an uptick in DDoS attacks
  - Targeting our customers
  - Observed in our recursive DNS traffic
- The attacks were out of the norm when contrasted with traditional booter / stresser attacks
  - Not DNS amplification using cpssc.gov, SSDP, NTP, Chargen ... etc
  - The queries were being sent to known recursive resolvers
  - The queries had some common patterns

# Authoritative Exhaustion aka DNS Water Torture

The attacks fit the profile of authoritative DNS exhaustion

- A large volume of properly formed in protocol queries
- Targeting domains that were delegated to Dyn's nameservers
- Pseudo-random sub domain prepended to cause a cache miss
  - Example: lq18v2V3N2lQ.<sub domain>.<domain>.<tld>
- Consistently 12 character pseudorandom string attached to the valid domain
  - Example: Gk4d95kg9qrl.example.com
  - “Random” seems to exclude certain values ('xyz' for example)

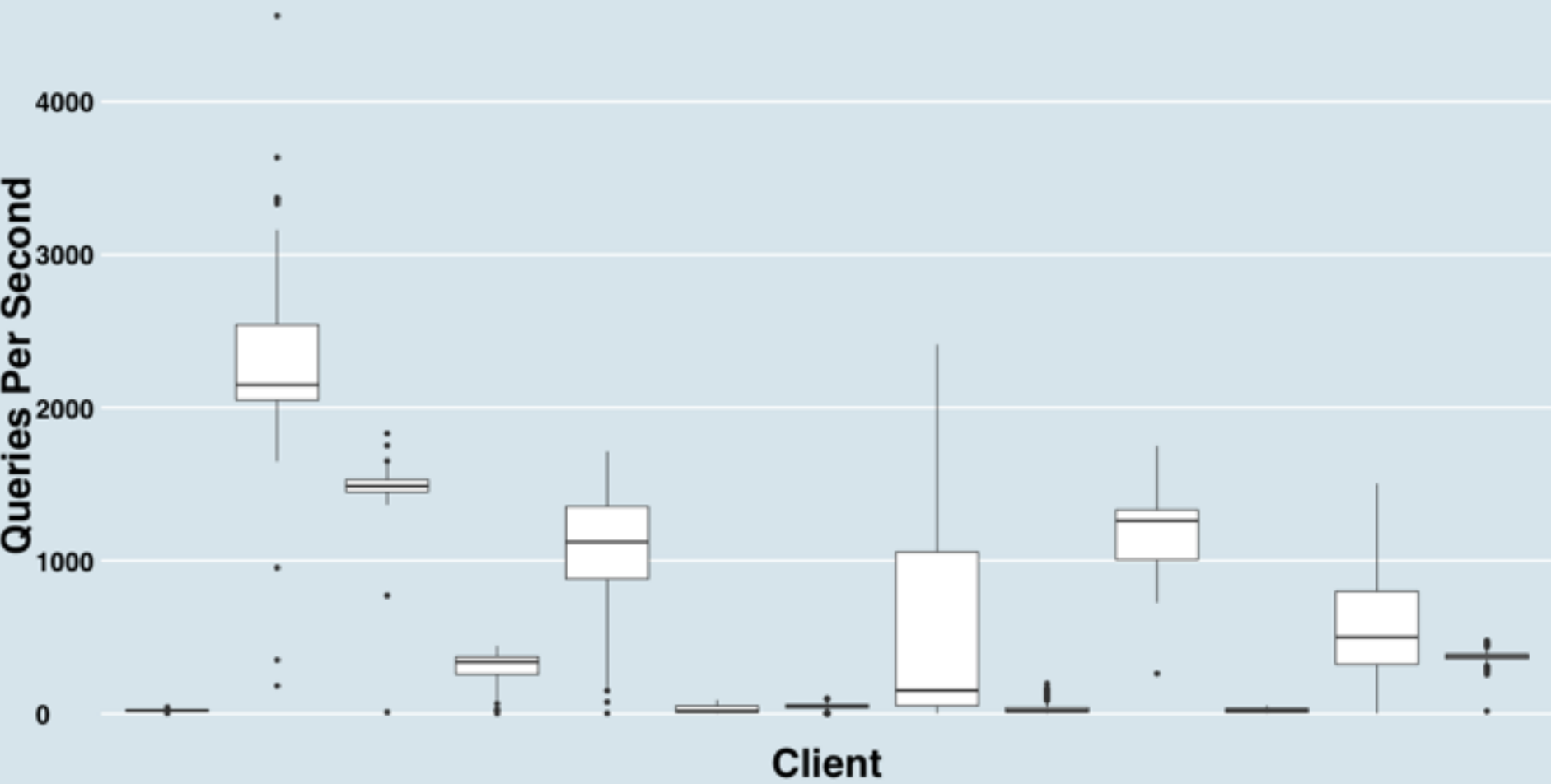
# Contextualizing Traffic

- 0x20 bit character randomization provided strong signal that the traffic was not spoofed
  - Example: ExAmPle.com, eXaMPLe.cOm
  - <https://tools.ietf.org/html/draft-vixie-dnsexp-dns0x20-00>
- Not only did it stand out but, it matched known patterns
  - Traffic from resolvers known to have implemented 0x20 bit randomization were consistently randomized
  - TTLs were inline with legitimate traffic

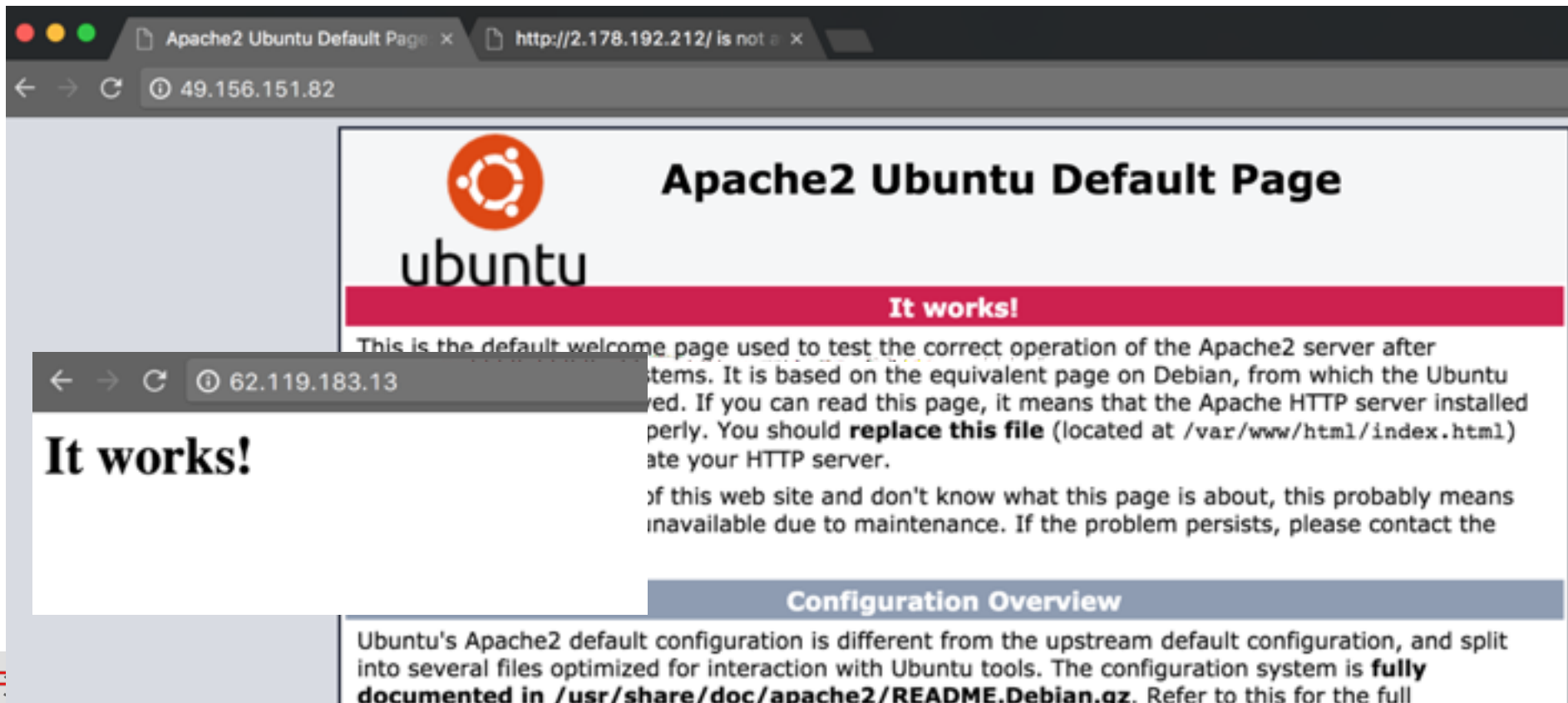
# To the Recursive Layer!

- Dyn operates a recursive resolver platform
- A handful of infected devices were configured to use Dyn's recursive resolvers from a number of different autonomous systems
- This provides us some data which can be used to measure the number of queries per second individual endpoints were issuing
- The variability in the rate of queries per second and the IPs / ASNs which were involved in one attack and not other is also of interest

# Queries Per Second by End Point




# What are they?



Apache2 Ubuntu Default Page x http://2.178.192.212/ is not a x

49.156.151.82

  
ubuntu

## Apache2 Ubuntu Default Page

**It works!**

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu version is derived. If you can read this page, it means that the Apache HTTP server installed on your system is working properly. You should **replace this file** (located at `/var/www/html/index.html`) with your own HTML page to serve as your HTTP server.

If you are unable to access this web site and don't know what this page is about, this probably means that the server is unavailable due to maintenance. If the problem persists, please contact the system administrator.

### Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full details.

62.119.183.13

## It works!

Country Code	Number of Prefixes	Count of ASNs	% of Traffic
US	226	99	31.54
CN	269	42	15.12
BR	580	264	8.33
RU	522	373	5.58
TW	22	12	3.88
UA	181	139	3.03
KR	26	14	2.95
BG	128	86	2.71
CL	18	9	1.95
ID	35	21	1.94

# Attack Source Code Leaked on HackForums

```
void attack_udp_dns(uint8_t targs_len, struct attack_target *targs, uint8_t opts_len, struct attack_option *opts)
{
    int i, fd;
    char **pkts = calloc(targs_len, sizeof(char *));
    uint8_t ip_tos = attack_get_opt_int(opts_len, opts, ATK_OPT_IP_TOS, 0);
    uint16_t ip_ident = attack_get_opt_int(opts_len, opts, ATK_OPT_IP_IDENT, 0xffff);
    uint8_t ip_ttl = attack_get_opt_int(opts_len, opts, ATK_OPT_IP_TTL, 64);
    BOOL dont_frag = attack_get_opt_int(opts_len, opts, ATK_OPT_IP_DF, FALSE);
    port_t sport = attack_get_opt_int(opts_len, opts, ATK_OPT_SPORT, 0xffff);
    port_t dport = attack_get_opt_int(opts_len, opts, ATK_OPT_DPORT, 53);
    uint16_t dns_hdr_id = attack_get_opt_int(opts_len, opts, ATK_OPT_DNS_HDR_ID, 0xffff);
    uint8_t data_len = attack_get_opt_int(opts_len, opts, ATK_OPT_PAYLOAD_SIZE, 12);
    char *domain = attack_get_opt_str(opts_len, opts, ATK_OPT_DOMAIN, NULL);
    int domain_len;
    ipv4_t dns_resolver = get_dns_resolver();

    if (domain == NULL)
    {
#ifdef DEBUG
        printf("Cannot send DNS flood without a domain\n");
#endif
    }
}
```



# If Local Resolver Not Found ...

```
switch (rand_next() % 4)
{
case 0:
    return INET_ADDR(8,8,8,8);
case 1:
    return INET_ADDR(74,82,42,42);
case 2:
    return INET_ADDR(64,6,64,6);
case 3:
    return INET_ADDR(4,2,2,2);
}
```

*This explains why 34% of the traffic is being attributed to the US*

Google  
Hurricane Electric  
Verisign  
Level 3

The defaults make up 24% of aggregate traffic.

# Duration

- The DNS attacks from Mirai targeting our customers in early August lasted between 7 - 10 mins.
- Some researchers have attributed this to the limitations of the command and control infrastructure relative to the size of the infrastructure being managed
- Other researchers attribute this to issues with the stability of the devices the attack is being launched from.
- Anna-Senpai, the handle which dumped the source code, mentioned the botnet was ~380K devices however post Krebs DDoS the number was down to 300K and dropping due to ISP reactions

A person is sitting at a desk, working on a laptop. A white coffee cup is on the desk in front of them. The background is slightly blurred, showing what appears to be an office or meeting room setting. The overall tone is professional and focused.

ORACLE® + Dyn

# Mirai and others attack

---

## October 21st

# #HugOps

Let us Pause for Ops

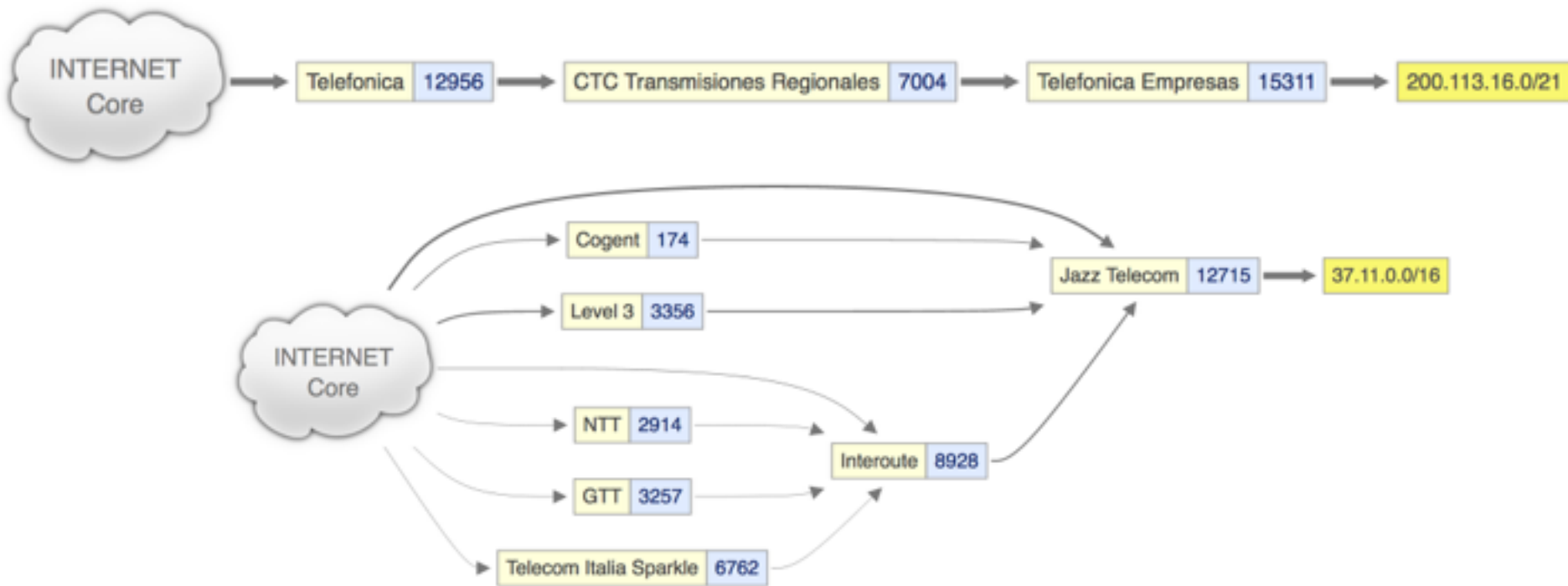
# October 21st – Morning into Wave 1

- October 21st a little after 10:00 UTC two authoritative exhaustion attacks were launched with a small space in between targeting \$CUSTOMER\_1
  - These attacks matched the 12 character subdomain pattern mentioned earlier
    - 11:08 UTC \$Attack\_Type\_1 came in targeting \$INFASTRUCTURE\_1
    - 11:10 UTC \$Attack\_Type\_2 came in targeting \$INFASTRUCTURE\_2
    - 11:12 UTC \$Attack\_Type\_3 came in targeting \$INFASTRUCTURE\_3
    - 11:13 UTC \$Attack\_Type\_4 came in targeting \$INFASTRUCTURE\_2
    - 11:15 UTC Size of \$Attack\_Type\_4 targeting \$INFASTRUCTURE\_2 became much larger

# Nuances of Layer 7 Attack Traffic & Anycast

- Around 11:19 UTC one attacker expanded to include a larger portion of our infrastructure
- When the layer 7 traffic started rolling in en masse there one question: “Is this spoofed?”
- Peering and Transit interact!
  - ISP in Hong Kong went different places:
    - Southern California
    - Hong Kong
    - Northern California
- Makes mitigation harder

# Looking at end point connectivity



# Fingerprinting

- Before the attacks we'd been working with a tool developed by the team at Flashpoint and was ready to fingerprinting target population devices
  - <https://github.com/trylinux/lift>
    - For each site take the newly added anomalous top talkers identified via netflow and run lift on them to get details about the devices
    - As time allows process network traffic captures to identify the attack traffic sources and take the IP addresses and fingerprint as many endpoints as possible as fast as possible
    - Goal: Verify as many IP device combos as possible



# Timing is everything

- IP addresses which are assigned by DHCP ( Dynamic Host Configuration Protocol ) by ISPs ( Internet Service Providers ) change for one of four main reasons:
- A client is disconnected from the network or loses power for long enough causing it to fail to renew its DHCP lease
- A client is rebooted or reconnected causing a PPP, point to point protocol over ethernet or ATM, to change address
- Changes are made by the provider either restarting the DHCP server for a subnet or other administrative change
- The provider limits the duration that a lease can be held so addressed periodically change day to day or week to week.

[https://labs.ripe.net/Members/ramakrishna\\_padmanabhan/reasons-dynamic-addresses-change](https://labs.ripe.net/Members/ramakrishna_padmanabhan/reasons-dynamic-addresses-change)

# But wait ... there's more

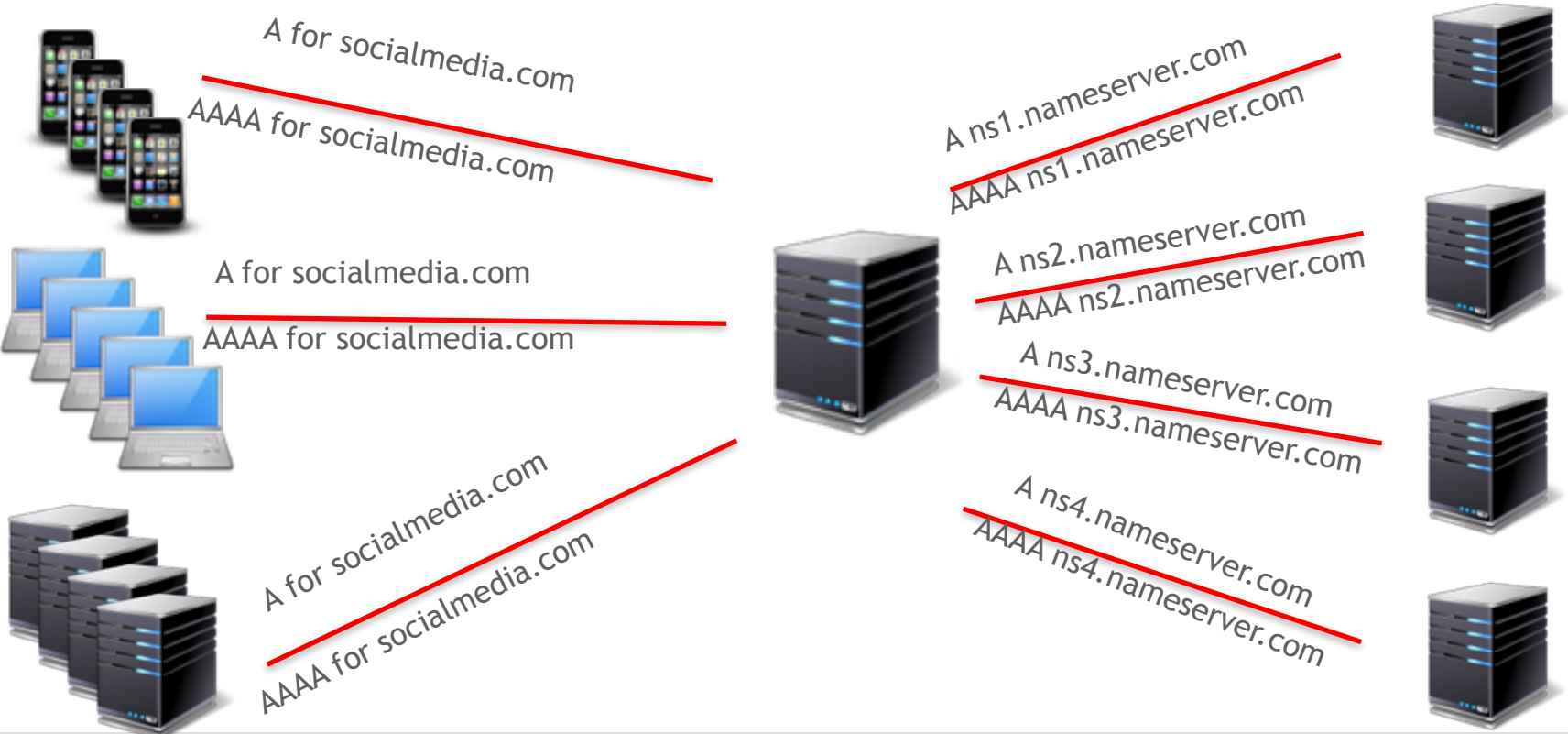
- 11:20 UTC ~ 12 mins after the first traffic of wave 1 hit the edge non-Mirai traffic joined the mix targeting \$CUSTOMER\_2
- Unlike the Mirai fixed 12 character subdomain authoritative exhaustion this traffic was different
- The length of the subdomain and its contents were drastically different
- The corpus was vast including everything from Microsoft support phone numbers, github project names, O'Reilly books ... etc
  - The size of the corpus hints that this attacking infrastructure is most likely not IoT based ( this of course is a post incident note )
  - In the heat of the moment this added to the collection of abnormal traffic signals



# Nimble vs. Durable

- Some companies use the DNS to steering traffic to specific locations based on requestor or end point / path performance.
- This steering relies on setting short TTLs because in the TTL drives your ability to make changes
- Short TTLs mean your records quickly disappear from caches
- Bad news in some situations

# Recursive Resolvers and Retries: Happy Eyeballs



# Recursive Resolvers and Retries

- Standard DNS resolution happens
- If the client resolver has implemented “Happy Eyeballs” they will ask the recursive for both A and AAAA ( iOS for example 😊 )
- If no answer, retry
- At this point we are now experiencing botnet attack traffic and what is best classified as a “retry storm”
  - Looking at certain large recursive platforms > 10x normal volume

# Wave 1

- Wave 1 continued with additional rounds of layer 7 attacks and a rolling in protocol DNS attack
- Mitigations were put in place to rate limit / drop / block abusive traffic
- Everything calmed down

# Wave 2 – Bigger and Longer

- 15:50 UTC – \$Attack\_Type\_4 came in targeting \$INFASTRUCTURE\_1 , \$INFASTRUCTURE\_2 , \$INFASTRUCTURE\_3 , \$INFASTRUCTURE\_4 with greater volume
- 15:52 UTC - \$Attack\_Type\_4 started to target a new infrastructure endpoint
  - Later analysis would show that this was most likely a second Mirai botnet
- 16:00 UTC – DNS “In Protocol Attacks”



# Community

This event showed the strength of the internet operator community

- Competitors reached out to offer support
- Threat Intel companies reached out to offer insight

# Response

How do we reach out to all the other potentially affected parties?

- ISPs and Entities with infected devices in their networks?
- Other Infrastructure as a service operators?

What format does the information need to be in to be useful?

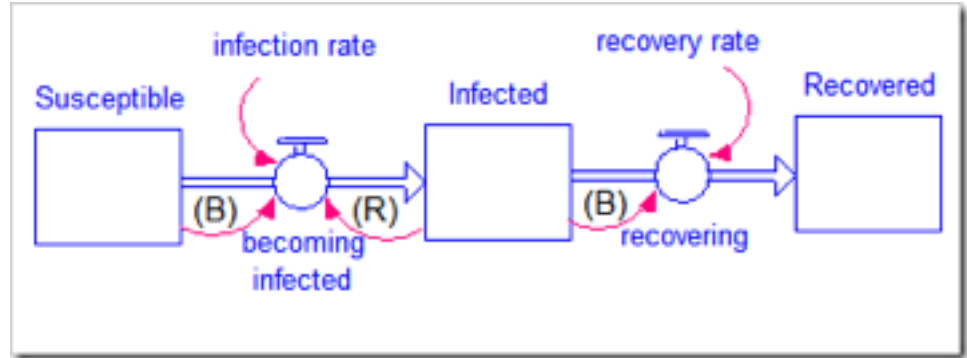
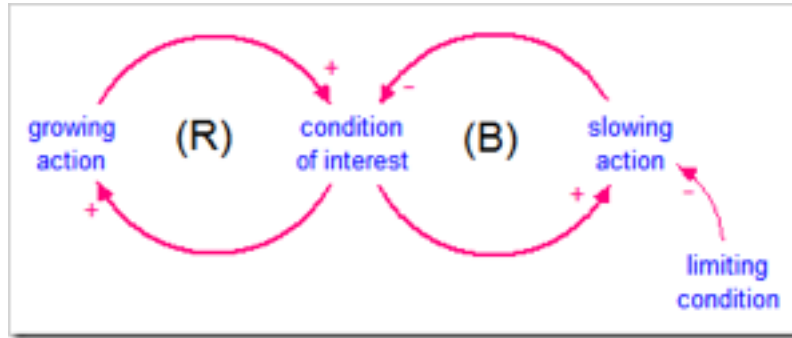
- Timestamp , IP address ( Minimum )

How do we and help networks to aid in infection remediation?

- Language barriers
- Legal barriers

# Limits to Growth

- Susceptible Population: Devices with telnet port 25 exposed to the internet
- Converted into Infected Population by the Infection / Malware load rate
- Infected Population: Devices which have had the malware loaded on to them



<http://blog.iseesystems.com/stella-ithink/limits-to-growth/>

# Remember that dumb network?

- Maximal intelligence at edge
  - Could still be a dumb device
- We want the network to avoid making a lot of decisions
  - Easier to upgrade endpoints
  - People who want the advantages have the incentive to upgrade

# This is what makes for the attack

- Really bad story when too many of the endpoints are bad guys
- IoT is just a means to this end
- Insecurity of devices is not the main problem
- None of this is DNS- or even UDP-specific
- Limited improvement from more providers
  - Which also create more barriers to entry

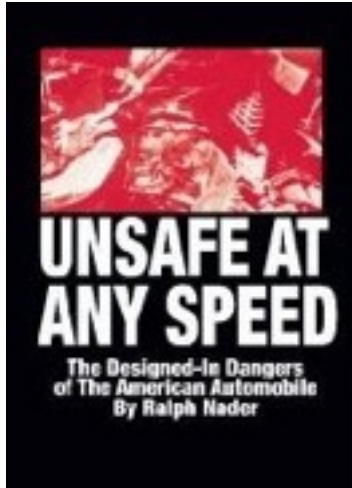
# Proposals I have personally heard

- Government-mandated BCP 38 implementation
  - Doesn't always help
- License to connect
  - Devices or people?
- Top-speakers preferred access for DNS
  - Submission for DNS?
- Your end-of-open-network answer here

# Are there things IETF could do?

Build on the tradition of the network of networks

- Why can't network devices or applications advertise what kinds of traffic they want to send?
- Why can't network devices or applications advertise what scope they want to be in?
- New feedback mechanisms? Manufacturer Usage Description?
- Would anything we do be an invitation to better or new attacks?



The roots of the unsafe vehicle problem are so entrenched that the situation can be improved only by the forging of new instruments of citizen action.

– Ralph Nader, in the Preface



**ORACLE<sup>®</sup>** + **Dyn**