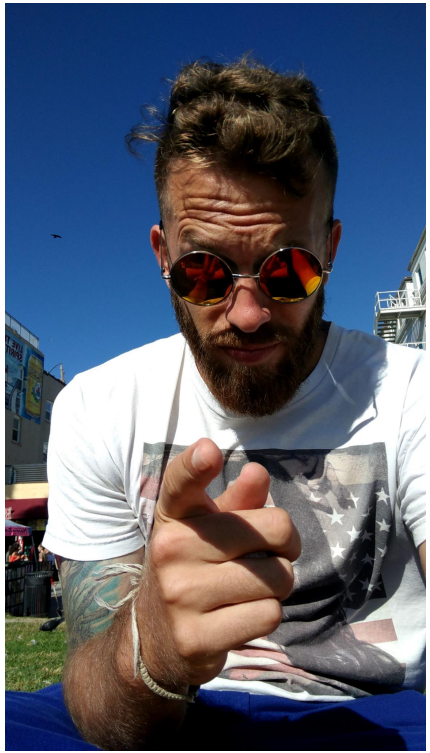
IPv666
Address of the **Beast**

Chris Grayson
@_lavalamp

Marc Newlin
@marcnewlin

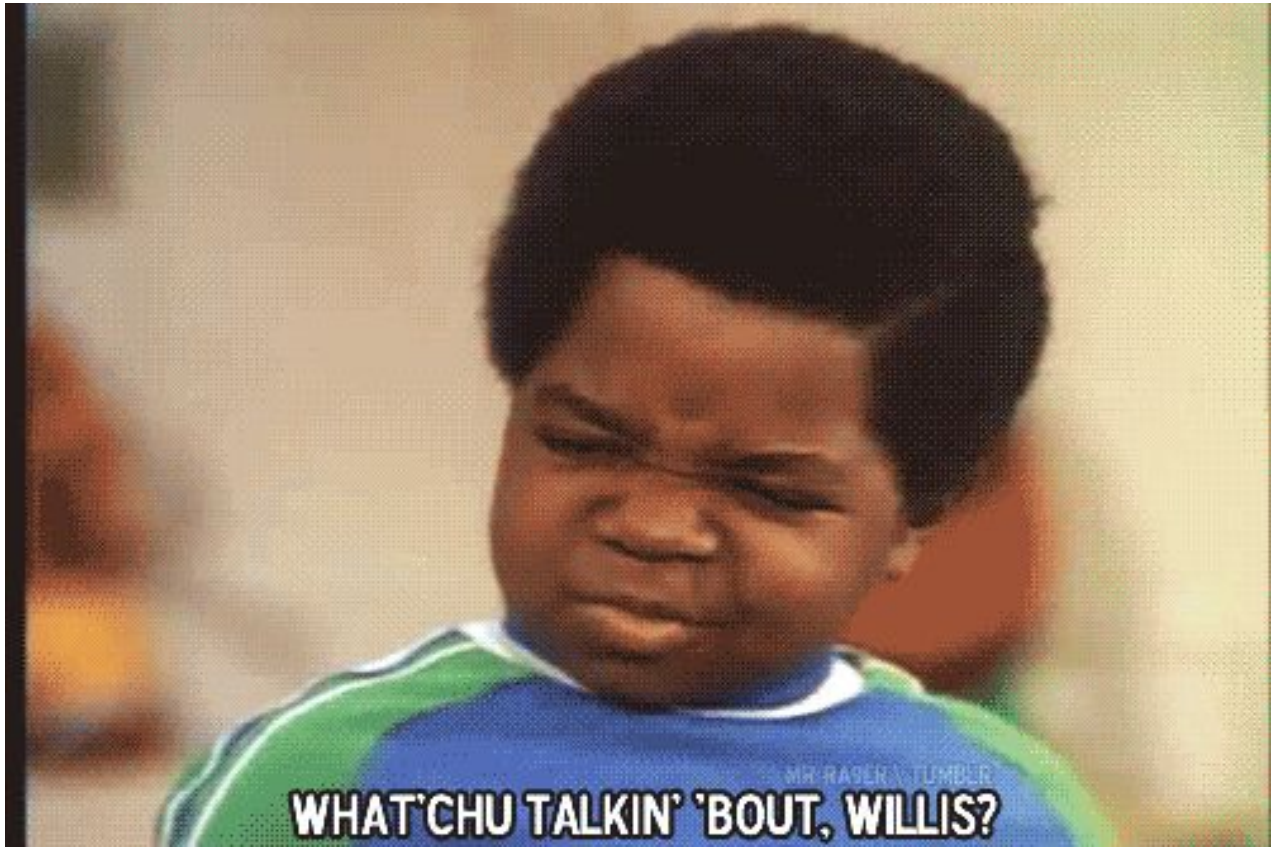

ALLOW ME TO
REINTRODUCE
MYSELF

MY NAME IS



HO (OH)

H TO THE OH VEE


WHAT'CHU TALKIN' 'BOUT, WILLIS?

IT’S THE FUTURE BAYBEEEEEE

The Future is a Scary Place...

- IPv6 is growing increasingly prevalent
- It’s great from a number of angles
- As far as security is concerned, IPv6 fails open in some interesting ways
- We’re p sure there are organizations out there with their whole corporate network on the open Internet

...and IPv6 is No Exception

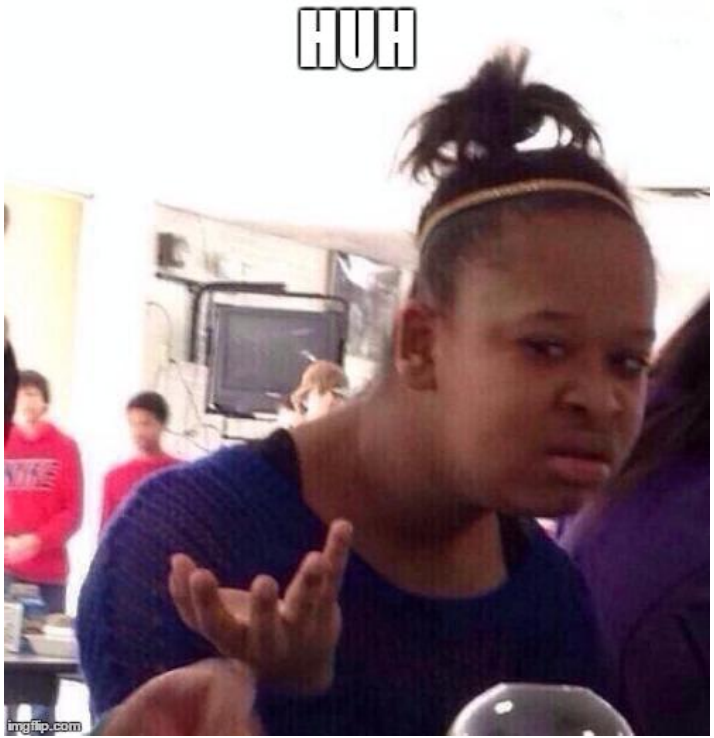
- Started learning about IPv6 security
- Things seemed like they might be bad
- Went to validate some hypotheses…
- Turns out IPv6 address discovery is a hard problem

Where to From Here Cap’n?

- Background
- IPv6 Security Implications
- The Scanning Problem
- Honeypotting for PSLAAC
- Modeling for non-PSLAAC
- IPv666
- Conclusion

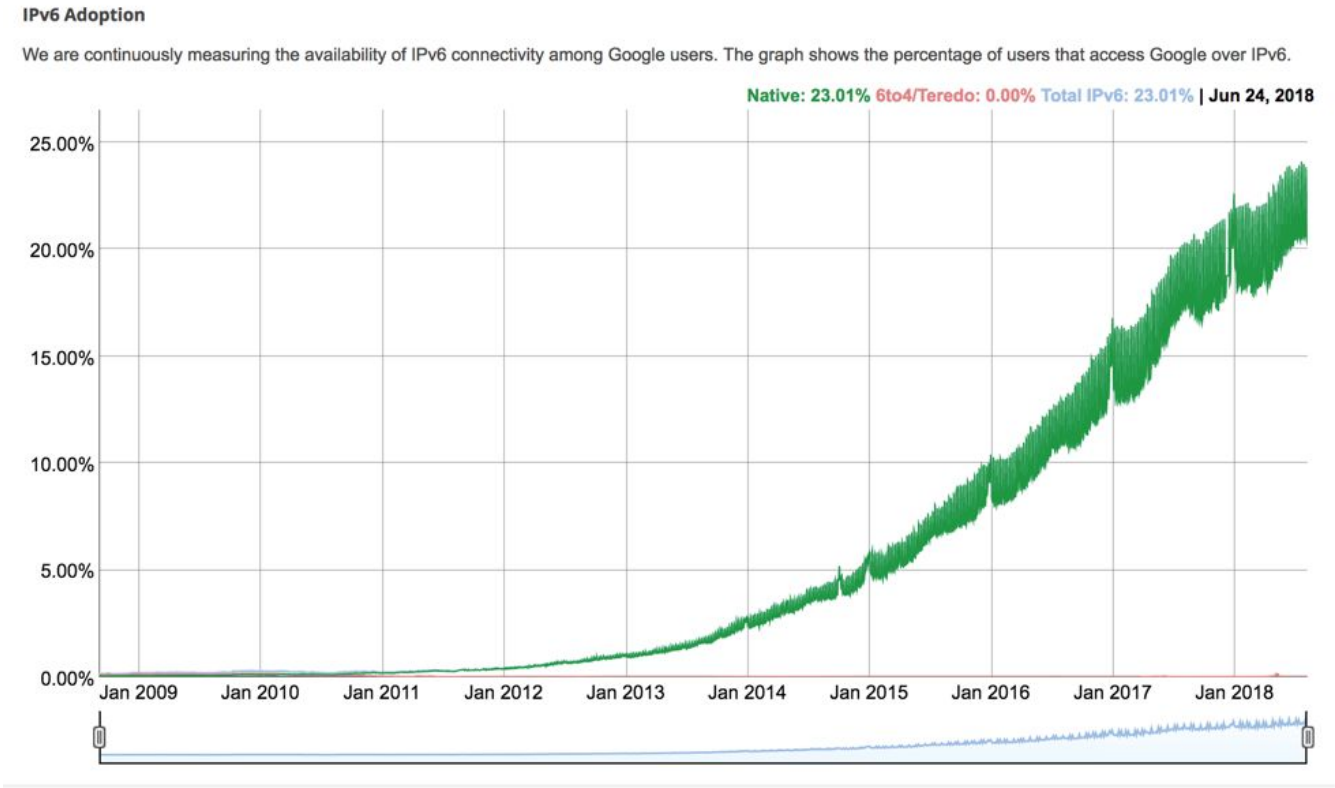
A Bit o’ Background

What is IPv6?



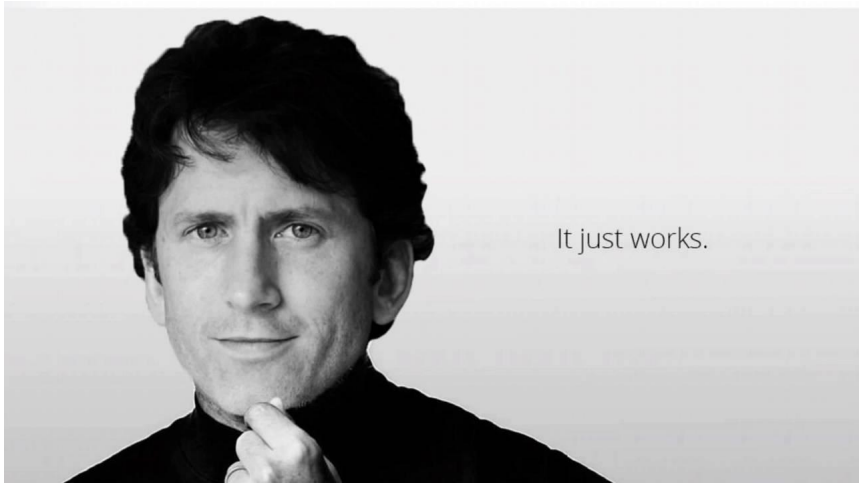
- ...it’s the future
- We ran out of IPv4 addresses (hence NAT, 10.*, 192.168.*, 172.16.*)
- IPv4 == 32-bit addresses, IPv6 == 128-bit addresses (3.4028237e+38)
- Lots of engineering improvements

How Common is IPv6?



IPv6 is a Great Engineering Feat!

- Better routing
- More efficient packet processing
- Multicast is the new broadcast
- Automatic configuration
- IT JUST WORKS!!

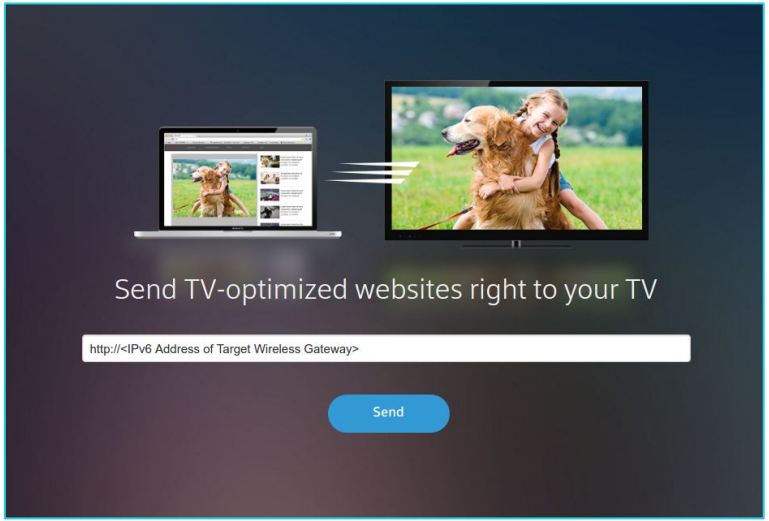


A Tale of Gateways and Set Top Boxes

- DEFCON 25
- 26 CVEs
- All in consumer premise equipment (CPE)
- Ie: gateways and set top boxes
- IPv6 was a big part
- https://github.com/BastilleResearch/CableTap

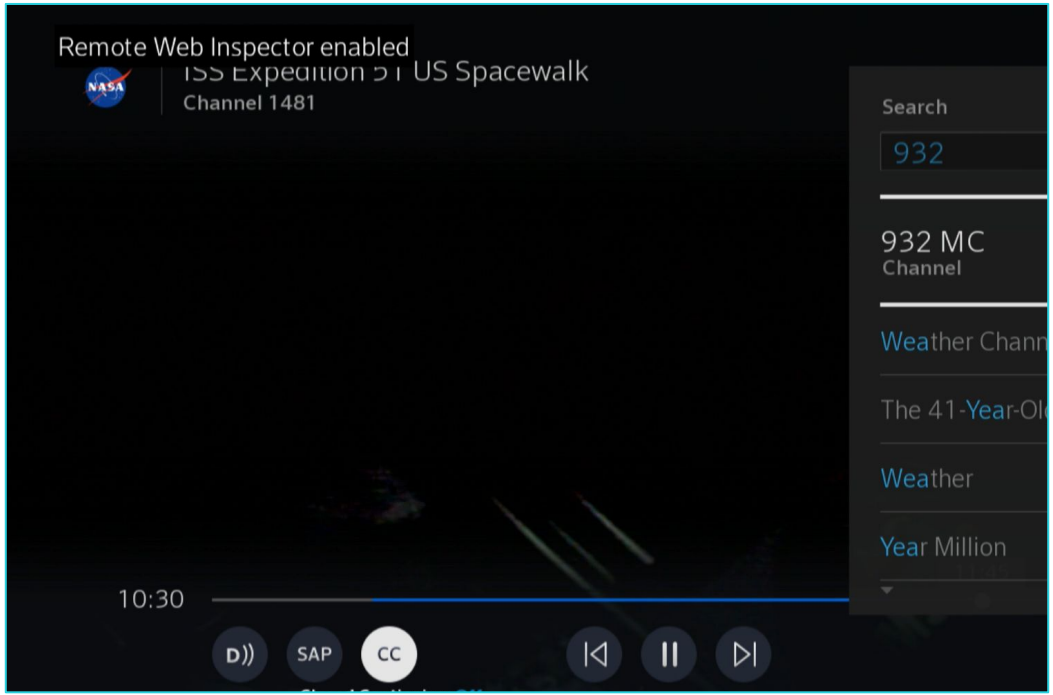
Send-to-TV / Remote Web UI

- Gateway web UI accepts remote requests from ISP infrastructure
- IPv6 address of target gateway provides remote web UI access via set-top box



Remote Web Inspector

Comparable to FireFox and Chrome DevTools, accessible from over the internet



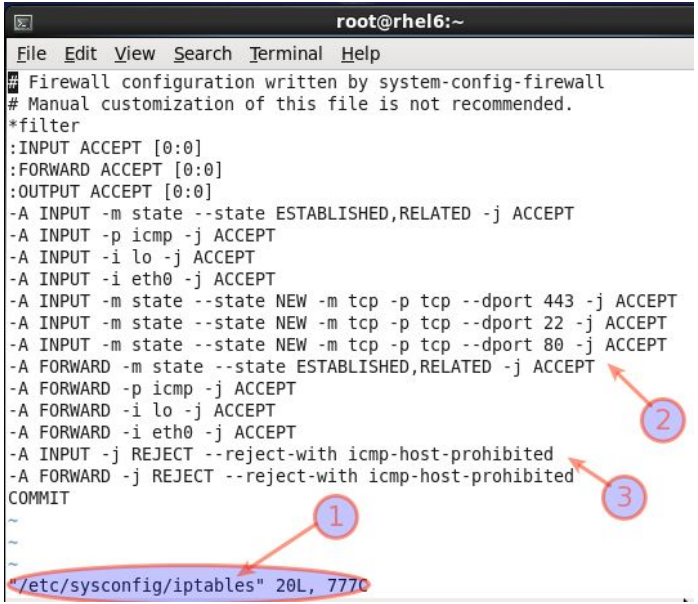
IPv6 Security Implications

No More NAT (Kinda)

- RFC 2663 - Network Address Translation
- Allocated 10.*, 192.168.*, 172.16.* for private, repeated usage
- Intended to help with address space exhaustion
- Ended up being a slammin’ security control

Iptables? More Like Ip6tables!

- If you take a look at iptables -L and you see all input is rejected, you're good right??
- Maybe you didn't hear about ip6tables
- Iptables rules do not apply to IPv6, and lots of people don't know this
- Configure your firewalls separately



SLAAC

- Originally, statically configure your IP addresses
- Then comes DHCP servers to automatically allocate IP addresses
- Need to have a DHCP server for most networks
- Nah bruh - check out Stateless Address Autoconfiguration (SLAAC)
- ...and PSLAAC

IPv6 is the Preferred Communications Protocol

- All modern operating systems prefer to talk over IPv6
- All modern networking equipment supports IPv6 by default
- If your machine can talk to a remote host over IPv6, it will do so over IPv6 in most cases

Many-to-one Addresses to Interfaces

- Ever configured a server to listen on an interface?
- You might be listening for more than you bargained for

We Need ICMPv6

- In IPv4 address resolution protocol (ARP) is used to tie layer two addresses (MAC) to layer three addresses (IP)
- This is required for routing traffic - without it no talky
- Built into ICMPv6 is Neighbor Discovery Protocol
- Ergo blocking or disabling ICMP entirely means your networks don’t work

Multicast is the New Broadcast

Address(s)	Description
FF0X:0:0:0:0:0:0:0	Reserved Multicast Address
FF0X:0:0:0:0:0:0:C	SSDP
FF0X:0:0:0:0:0:0:FB	mDNSv6
FF0X:0:0:0:0:0:0:FC	ALL_MPL_FORWARDERS
FF0X:0:0:0:0:0:0:FD	All CoAP Nodes
FF0X:0:0:0:0:0:0:FE-FF0X:0:0:0:0:0:0:FF	Unassigned
FF0X:0:0:0:0:0:0:100	VMTP Managers Group
FF0X:0:0:0:0:0:0:101	Network Time Protocol (NTP)
FF0X:0:0:0:0:0:0:102	SGI-Dogfight
FF0X:0:0:0:0:0:0:103	Rwhod
FF0X:0:0:0:0:0:0:104	VNP
FF0X:0:0:0:0:0:0:105	Artificial Horizons - Aviator

- Broadcasting is cool, but in many cases it’s wasteful
- What if you could specify a distance for propagation and the type of devices to receive the data?
- You’re talkin’ about multicast brothaaaa

In Summary

- IPv6 works out of the box without any configuration
- All your devices and networking equipment prefers it
- There’s no such thing as private address space (for the most part)
- Your existing firewall rules don’t apply
- You can’t easily prevent ping scans
- Single packets can be relayed to lots and lots of hosts

Let’s Go Hunting

The Problem of Scale

- IPv4
  32 bit addresses
  2^32 possible addresses

  4,294,967,296 addresses

- IPv6
  128 bit
  2^128 possible addresses

  340,282,366,920,938,463,463,374,607,431,768,211,456 addresses

PSLAAC Makes Things Harder

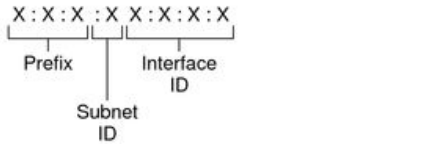
Request for Comments: 4941 — IBM Corporation
Obsoletes: 3041 — R. Draves
Category: Standards Track — Microsoft Research
S. Krishnan
Ericsson Research
September 2007

Privacy Extensions for Stateless Address Autoconfiguration in IPv6

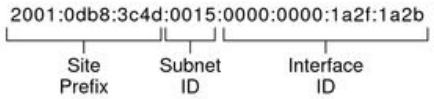
Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Figure 3-2 Basic IPv6 Address Format



Example:



- Addresses have host and network bits
- Network is site prefix and subnet ID
- Host is interface ID
- PSLAAC means interface ID is pseudorandom
- Even “small” networks (/96) have 32 bits of randomness

Breaking Down the Problem

- Modeling for cryptographic entropy is no bueno
- Two independent problems instead
  - Identifying PSLAAC hosts
  - Identifying non-PSLAAC hosts

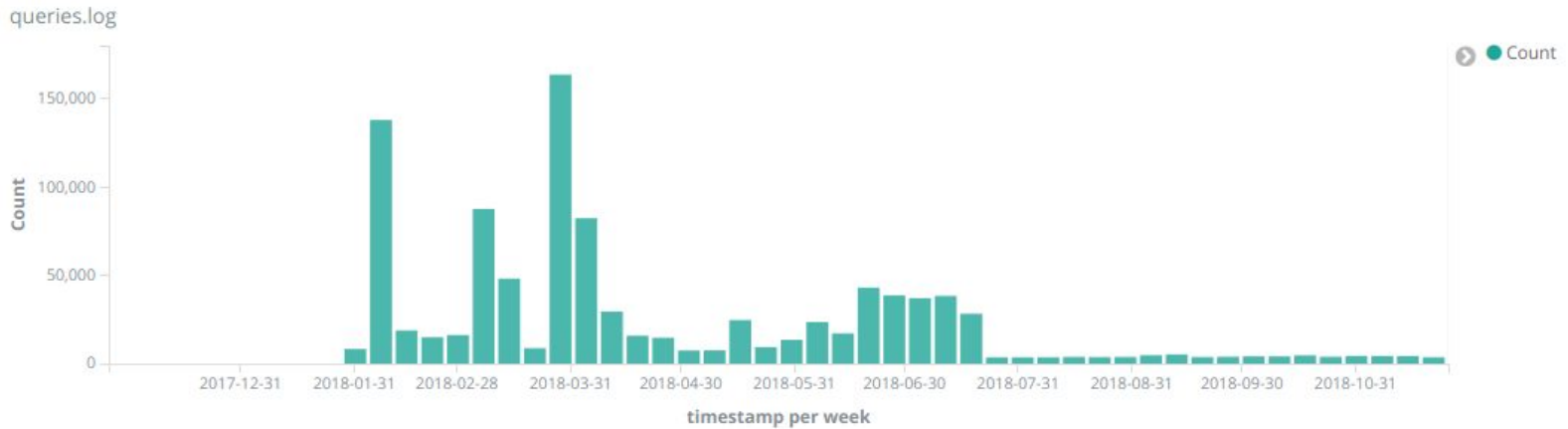
Honeypotting for PSLAAC

Why Honeypotting?

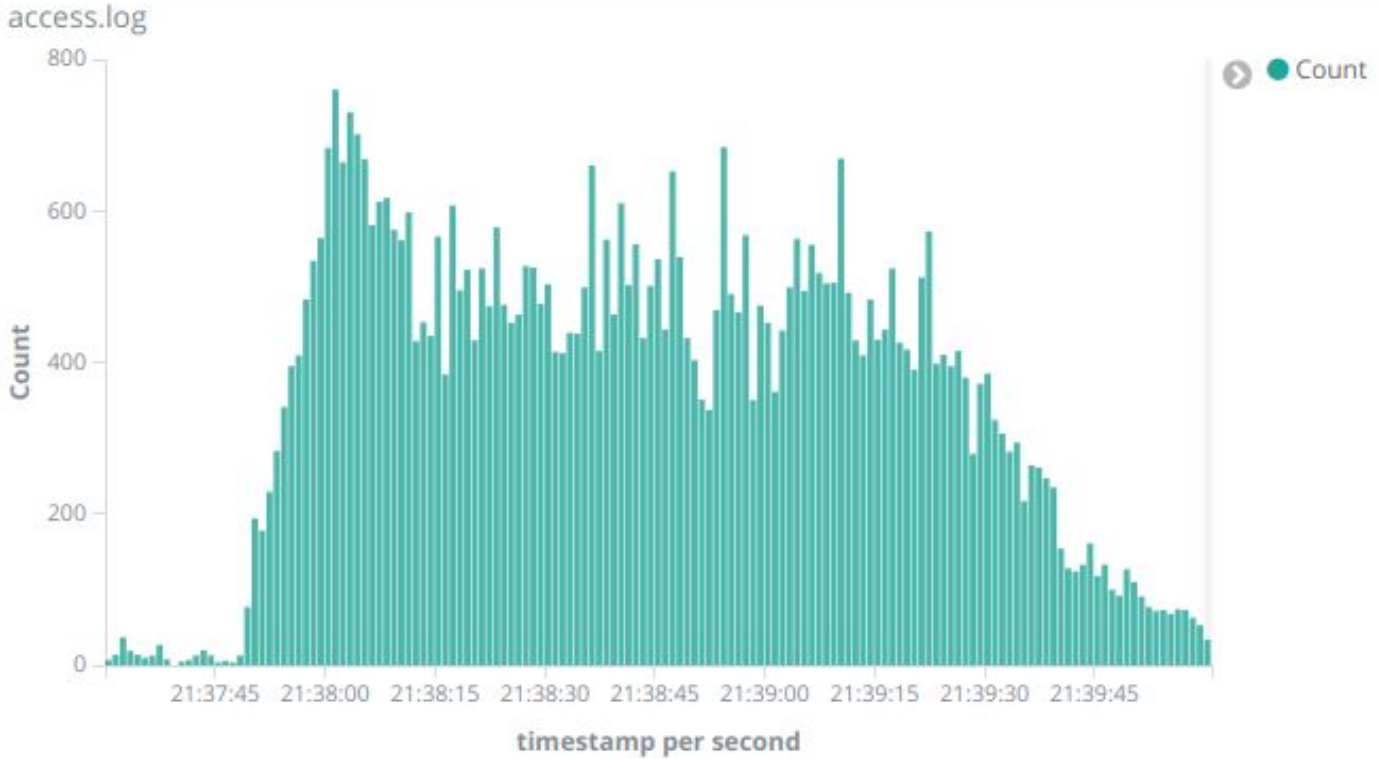
- Search space too massive
- Instead of finding them, have them find us
- Took multiple approaches
  - DNS Server
  - SMTP Server
  - Web Server
- Popads!

Honey DNS Server

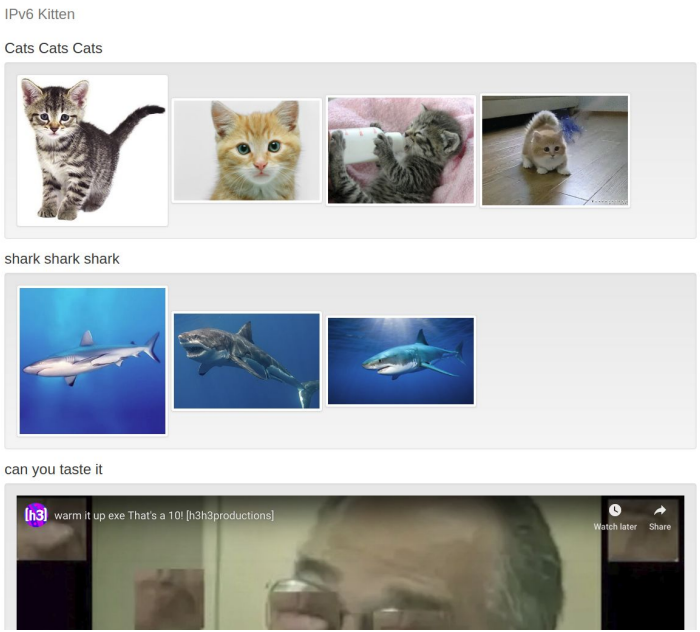
- Set up Bind server
- Glue records point to IPv6
- Zones delegated to IPv4 then IPv6
- Post links all over social media
- Popads!



A Quick Note on PopAds

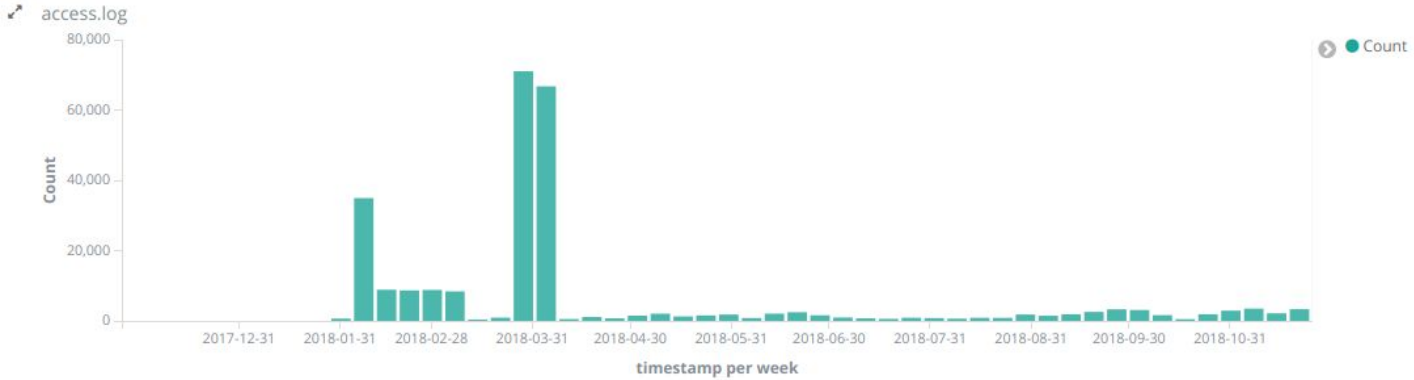


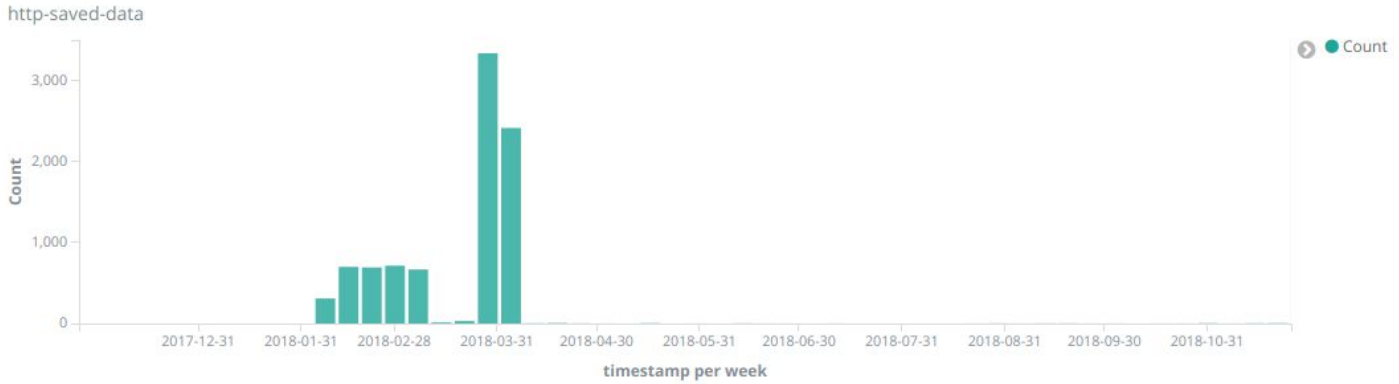
Honey Web Server



- Set up best site ever at http://ipv6.exposed/
- Available over IPv6, some other shenanigans for forcing IPv6
- WebRTC for IP address enumeration
- Post all over social media
- Popads!

Honey Web Server (cont'd)





Honey SMTP Server

- Set up SMTP server
- Have our DNS honeypot point to it
- Sign up for ALL THE THINGS
- Use MailBait to sign up for spam
- Mega womp womp

Sweet(?) Honey Results

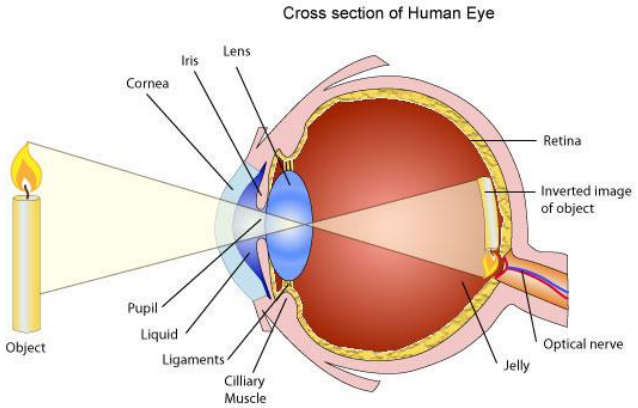
- 92,609 unique IP addresses over ~10 months
- Cost $500+
- Lost focus, but still suboptimal

Modeling for Non-PSLAAC

Structure in IPv6 Addresses

2001:1284:f01c:2c0a:8238:bcff:fed3:4e03
2001:1498:1::32:48
2001:16b8:101:3175:a96:d7ff:fe7e:c4aa
2001:16b8:6100:0:3631:c4ff:fe14:3d72
2001:1890:1f8:46::1:6
2001:1bc8:1004::2:0:99
2001:260:450:7b::4
2001:41d0:2:275b::182
2001:41d0:2:3d77::
2001:4998:44:6027::2005
2001:4b78:2::f06b
2001:558:370:ba::1
2001:579:6f03:600:c0b3:b230:5c3:df35
2001:610:1a0:30::2
2001:638:708:30da:ea11:32ff:fe70:8ae6
2001:8d8:921:c500::70:18e1
2001:980:6972::1
2001:bc8:2800:36dd:87f9:1ef0:8a7a:c21f
2001:e42:102:1819:160:16:234:111
2001:ee0:4041:37a5:8acf:98ff:fee7:fec
2001:ee0:4140::1230:2502:7004
2001:ee0:4501:5062:1894:5a03:9cc3:216f
2001:ee0:5500:8b5a:585d:c41a:7a09:f73a
2003:0:2e02:1050::1
2003:5a:4049::1
2400:6180:100:d0::34:8001
2400:8500:1302:803:133:130:127:180
2400:b800:1:1::18
2400:cb00:2048:1::6810:7166
2400:cb00:2048:1::6818:d3a
2404:6800:4007:801::2003
2406:da00:ff00::1715:7892
2406:e00:120:391c:0:30:ab4:4f40
2407:500::2:5a9:b7bf
2600:0:2:1239:144:232:2:85
2600:3c00::f03c:91ff:feae:ee1c
2600:3c00::f03c:91ff:fedf:426
2600:3c03::f03c:91ff:fe79:e1a4
2600:3c03::f03c:91ff:fea1:4761
2600:3c03::f03c:91ff:fea2:42c7
2603:3006:103c:b000::17f6
2604:2d80:4030:0:91c2:c016:2329:7219
2605:de00:1:1:4a:32:0:23
2606:b400:8808:f000::a022:fa6d
2620:11:0:c2b4:749a:46bf:291:38cf
2620:8d:0:7f47::d827:7f47
2800:370:2:418a:b8d3:ddc8:13a8:8768
2800:370:2:972d:dd4a:a203:9d92:3353
2800:370:44:256d:5950:afb3:7627:10f4
2800:370:55:3c60:404a:d212:49dd:4477
2800:370:55:b9c6:70b1:9c61:69e:2b02
2800:370:61:11dc:e57f:1804:a456:eddc
2800:370:84:0:d0bd:6968:d1ae:d26f
2800:370:84:bba1:857c:f02d:cd6c:cd51
2800:370:a:ae43:79b9:348b:3c0:8c7b
2800:4f0:1:ecd5:dd66:8006:1203:be04
2800:4f0:62:850c:92d:85a1:3431:a7b8
2803:c300::2
2804:14d:1a87:0:7815:7414:e01:dd02
2804:14d:8e8c:1000:b42f:9577:7fbb:778b
2804:292c:1::5
2804:a8:2:c8::12a
2806:102e:9:5055:272:63ff:fe83:e620
2806:108e:c:2e3:7279:90ff:fe9c:2a07
2a00:d0c0:200:0:b9:1a:9c36:20c
2a01:488:42:1000:50ed:8479:33:339a
2a01:488:42:1000:50ed:84f5:1c:ff1f
2a01:5a60:3::92
2a01:7c8:d002:1c::1
2a01:a8:dc0:330:1::1c5d
2a02:2028:80c:e900::1
2a02:26f0:df:202:e3cc:80db:ebaa:3e93
2a02:6b8:0:161b:ec4:7aff:fe18:c48
2a02:6b8:b000:63a:96de:80ff:fe81:1258
2a02:6b8:b000:6509:215:b2ff:fea9:66fa
2a02:8108:0:12:587b:b48e:e629:436b
2a02:8108:8000:21:2864:5009:d4d2:36f0
2a02:810d:8000:29:e8b0:6937:7ddd:e597

MACHINE LEARNING BAYBEEEE



- Model is a compact representation of data set
- Projection through model creates new data set with error %
- Errors are representative of structure in IPv6 addresses
- Hopefully find new addresses

lol jk

- All attempts resulted in over-fitting
- Projected addresses were the same as our input addresses
- We’re not ML experts sooooo….

The Entropy/IP Paper

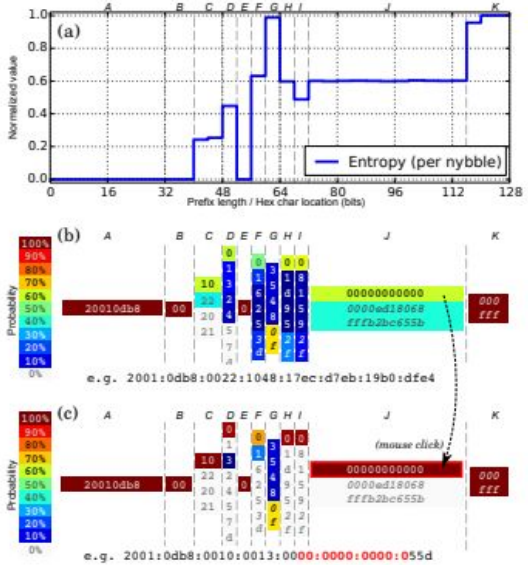
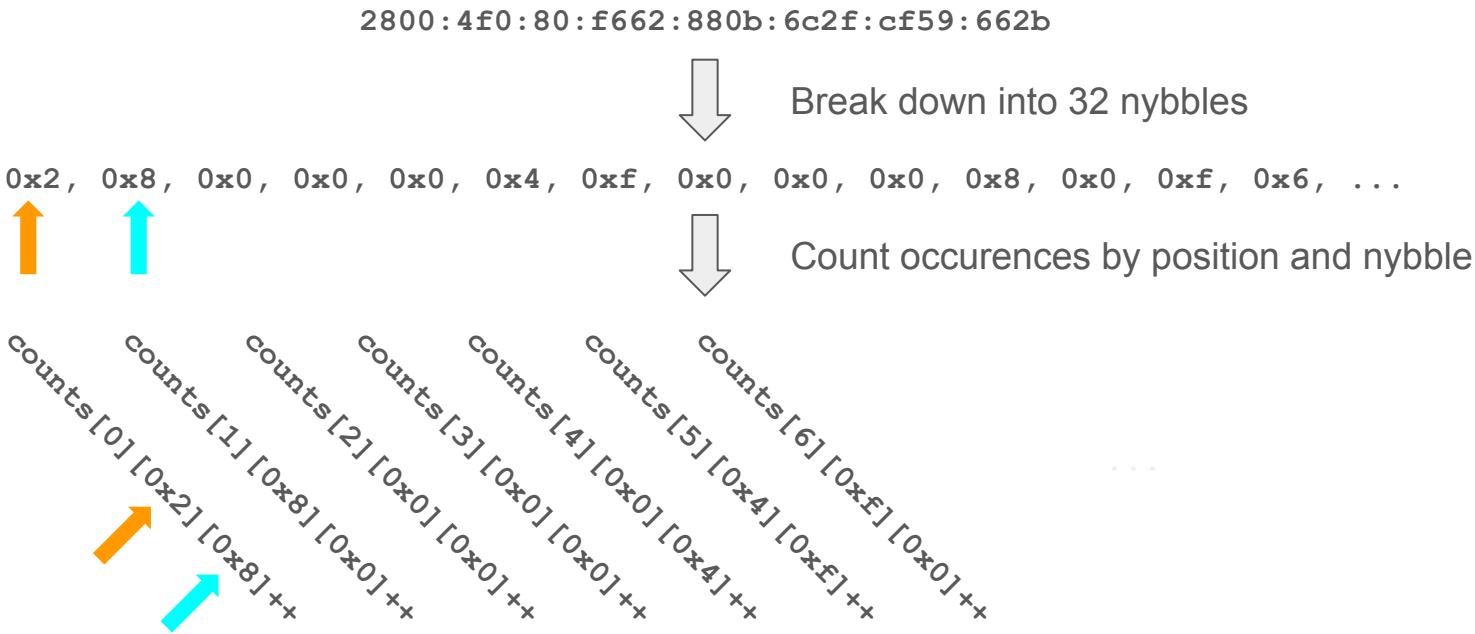


Figure 1: Entropy/IP's user interface displaying an analysis of a Japanese telco prefix with 24K active client IPs. Entropy by nybble plotted in (a). In (b), we select the 00000... value (60%) for segment J by mouse click, resulting in updated probabilities in (c) (*e.g.*, 100%).

- http://www.entropy-ip.com/
- Really interesting paper from Akamai
- Maps entropy of different segments of IPv6 addresses
- Big takeaways:
  - Not THAT much entropy in non-PSLAAC IPs
  - Simpler modeling might work better

Dumbing Things Down (Modeling)



Dumbing Things Down (Prediction)

0x2
Position 0

probabilities[0][0x2]

```
p(0x0) => 0.05
p(0x1) => 0.05
p(0x2) => 0.01
p(0x3) => 0.09
p(0x4) => 0.00
p(0x5) => 0.15
p(0x6) => 0.05
p(0x7) => 0.09
p(0x8) => 0.00
p(0x9) => 0.06
p(0xa) => 0.15
p(0xb) => 0.09
p(0xc) => 0.06
p(0xd) => 0.10
p(0xe) => 0.03
p(0xf) => 0.02
```

0xa
Position 1

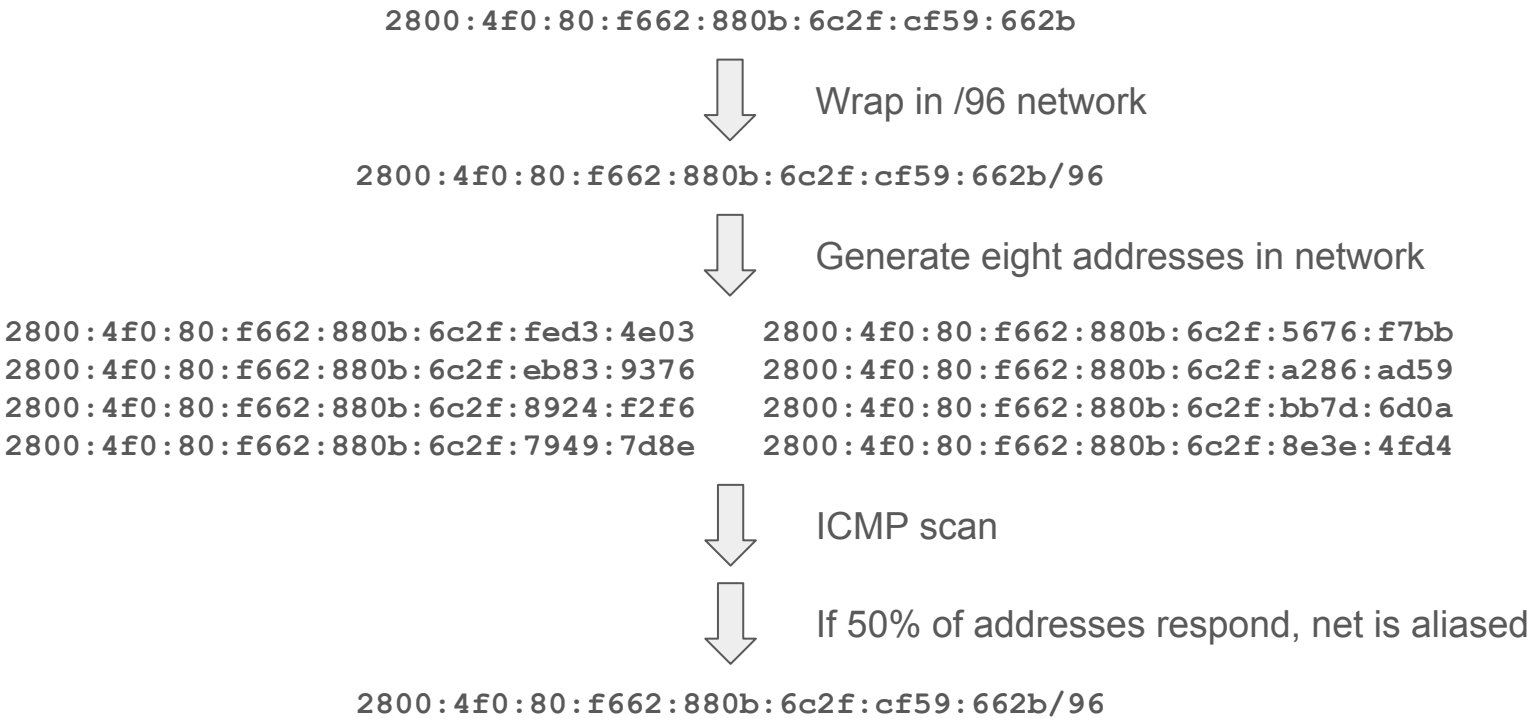
Looks Promising… TOO PROMISING

- Generated 10mm addresses
- After scanning, over 50k responded to ICMP probes
- WOW WE TOTES SOLVED THE PROBLEM!!! ZOMG CELEBRATION TIME!!!!

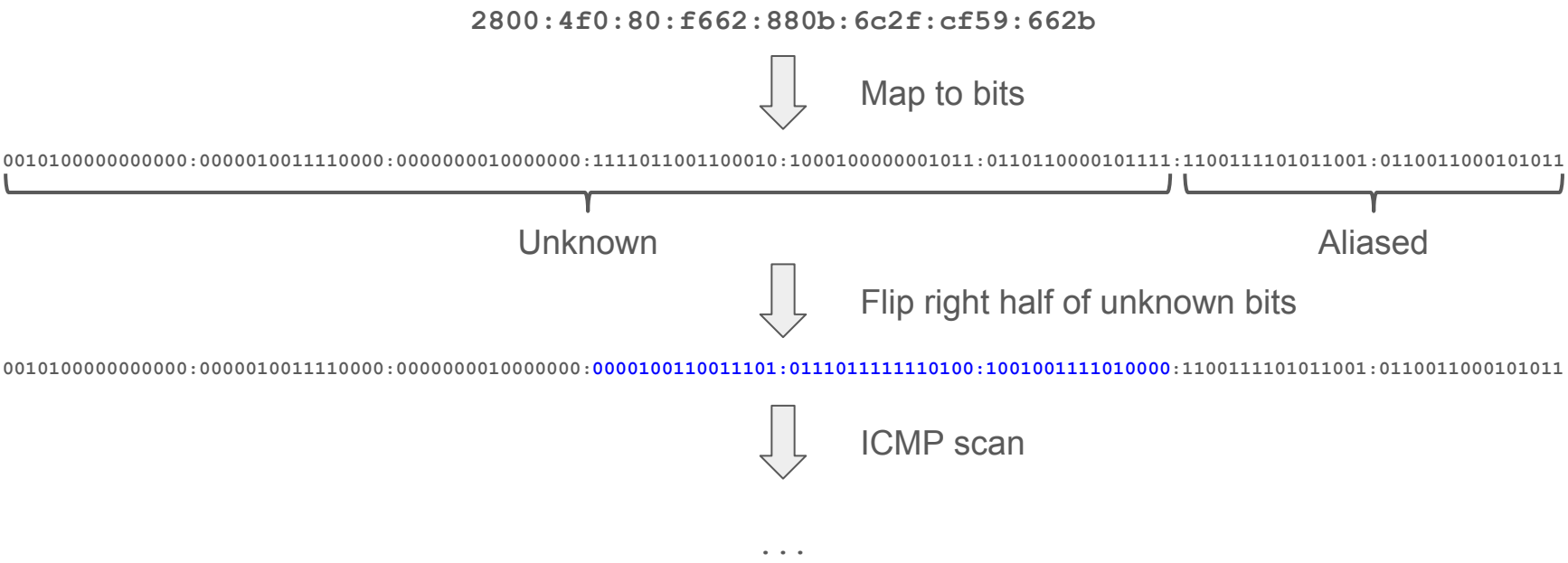
Enter Aliased (ie: Jerk) Networks

- Network ranges where every IP address responds to ICMP pings
- Why?? Because they’re jerks
- Not great for scanning
- Even worse for statistical modeling

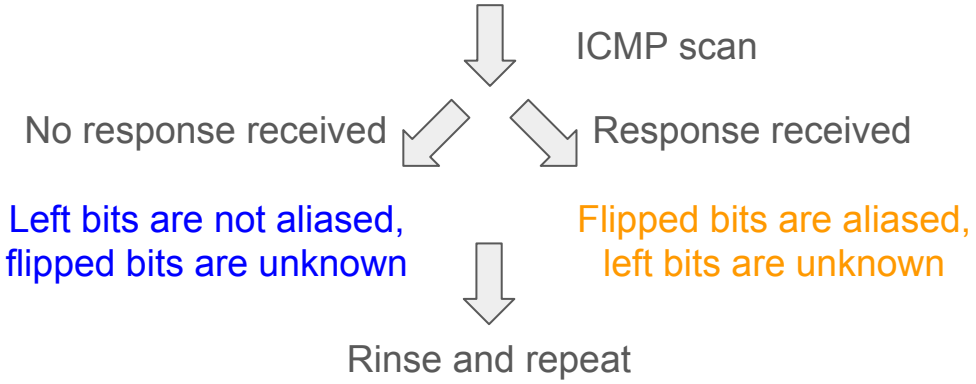
Identifying Aliased Networks (Initial)

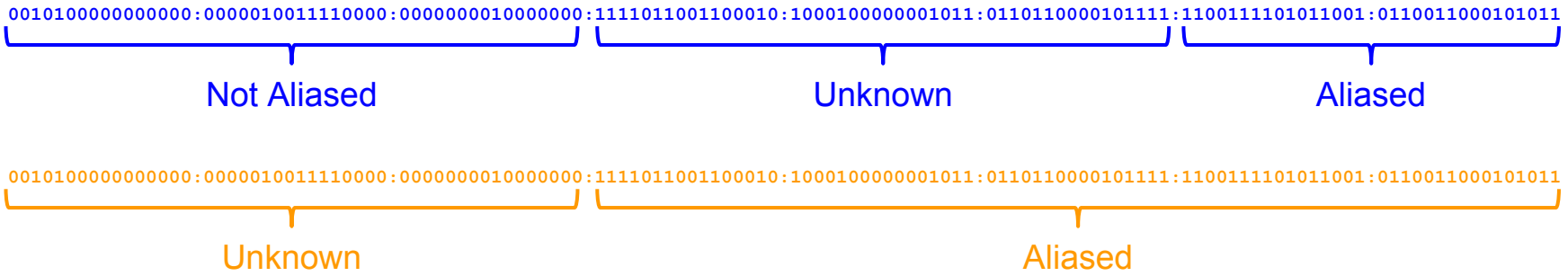


Identifying Aliased Networks (Network Size)



Identifying Aliased Networks (Network Size cont'd)





IPv666

Was Ist Das?

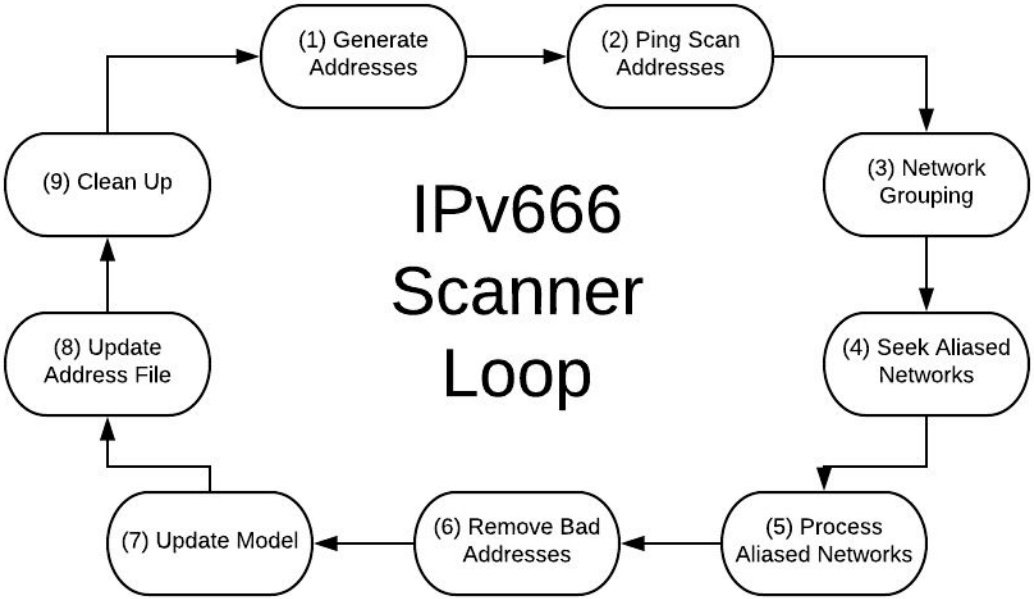
`ipv666` is a set of tools that enables the discovery of IPv6 addresses both in the global IPv6 address space and in more narrow IPv6 network ranges. These tools are designed to work out of the box with minimal knowledge of their workings.

If you're interested in how these tools work please refer to this blog post.

The tools included in this codebase are as follows:

- `666scan` - Locates live hosts over IPv6 using statistical modeling and ICMP ping scans
- `666alias` - Tests a single IPv6 network range to see if the network range is aliased
- `666blgen` - Adds the contents of a file containing IPv6 network ranges to the aliased network blacklist
- `666clean` - Cleans the contents of a file containing IPv6 addresses based on an aliased network blacklist

The Scanning Loop



666scan

```
Usage of ./build/666scan:
  -config string
    Local file path to the configuration file to use. (default "config.json")
  -force
    Whether or not to force accept all prompts (useful for daemonized scanning).
  -input string
    An input file containing IPv6 addresses to initiate scanning from.
  -input-type string
    The type of file pointed to by the 'input' argument (bin or txt). (default "txt")
  -network string
    The target IPv6 network range to scan in. If empty, defaults to 2000::/4
  -output string
    The path to the file where discovered addresses should be written.
  -output-type string
    The type of output to write to the output file (txt or bin). (default "txt")
```

666alias

```
Usage of ./build/666alias:
  -config string
    Local file path to the configuration file to use. (default "config.json")
  -net string
    An IPv6 CIDR range to test as an aliased network.
```

666blgen

```
Usage of ./build/666blgen:
  -config string
    Local file path to the configuration file to use. (default "config.json")
  -input string
    An input file containing IPv6 network ranges to build a blacklist from.
```

666clean

```
Usage of ./build/666clean:
  -blacklist string
    The local file path to the blacklist to use. If not specified, defaults to the most recent blacklist in
    the configured blacklist directory.
  -config string
    Local file path to the configuration file to use. (default "config.json")
  -input string
    An input file containing IPv6 addresses to clean via a blacklist.
  -out string
    The file path where the cleaned results should be written to.
```

Proof in the Pudding?

- Global address space, 20mbps, eight days
  - 58,388 unique addresses found
  - ~80% of them not in public datasets
- Targeted address space (ISP, /48), 20mbps, 30 minutes
  - 4,901 addresses
  - ...mostly consumer premise equipment
  - ...with web login portals
  - OGOD THE FLASHBACKS

Linky Links

- IPv666 Blog Post
  https://l.avala.mp/?p=285
- IPv666 GitHub Repository
  https://github.com/lavalamp-/ipv666

Conclusion

Recap

- Background
- IPv6 Security Implications
- The Scanning Problem
- Honeypotting for PSLAAC
- Modeling for non-PSLAAC
- IPv666
- Conclusion

Moar Links

- Entropy/IP
  http://www.entropy-ip.com/
- 6gen
  https://zakird.com/papers/imc17-6gen.pdf
- Clustering of IPv6 address structure
  https://arxiv.org/pdf/1806.01633.pdf
- IPv6 hitlist
  https://ipv6hitlist.github.io/

Q&A

THANKS!

<3

Chris Grayson

@_lavalamp

Marc Newlin

@marcnewlin