# Target Generation for Internet-wide IPv6 Scanning

Austin Murdock, Frank Li, Paul Bramsen, Zakir Durumeric, Vern Paxson





	Towards Detecting Target Link Flooding Attack	A Measurement Study on	Co-residence Threat inside the	<b>p</b>		
	Lei 7 The Matter of Heartbleed	i	Imperfect Forw How Diffie-Hellman		Backed by Internet-	
	he Ho e.cxxli A Messy State of Taming the Composite Stat	the Union:	A Large-Scale Analys	is of the Security of Embedded	Firmwares Michael Ba	
Background	Ab rget lini connec Benjamin Beurdouche", Karthikeyan Bhargavan", Ant Elliptic Cu		Andrei Cost Using Frankencerts for rve Cryptography in Practice Testing of Certi		cate Validation	
•	aus nki LPA efri On the Practical Exploitability of Dual EC in TLS Implementations <sup>Halder</sup>		$^2,$ Nadia Heninger $^3,$ Jonathan Moore, Michael Naehri and Eric Wustrow $^2$	in SSL/TLS Implem	entations evel TP:	
	par ys Stephen Checkoway,* Matthew Fredr eas Matthew Green,* Tanja Lange, Daniel J. Bernstein <sup>‡</sup> § Jake Maskiew	U	ment and Malicious	n	ew of Internet-Wide Scanning	
IPv4 scanning - Zmap	the * token Hanking Haingareity <sup>4</sup> Haingareity of Wiegon Cloudy with a Chance of Breach: Forecasting Cyl		Networks A	Neither Snow Nor Rain n Empirical Analysis of Ema		
	When Governments Hack Opponents: A Look at Actors and Technol			ficate Reissues and Rev Vake of Heartbleed	rocations rsztein: alderman' gn	
	Attacking the Network Time Protocol		Liang Zhang Tap Northeastern University	Dance: End-to-Middle Anticensors	hip without Flow Blocking	
	Illuminating the Security Issues Surrounding Lights-Out Server Management		An Internet-wide analysis for electronic communic			
	For Tour See what I See?         Ferential Treatment of Anonymous Users         How I         A Systematic Analysis of the Juniper Dual EC Incident		Exercising as a Sometica		ed Certificates From Trus	
					WN: Breaking TLS using SSLv2	
					Schinzel <sup>2</sup> Jurai Somorovsky <sup>3</sup> Nadia Henir	
	Shaanan Cohney <sup>§</sup> Matthew C Ralf-Philipp Weinmann <sup>¶</sup> Eric R *University of Illinois at Chicago, <sup>†</sup> University of Calif		vare Analysis at Scale: Ided Web Interfaces	A Characterization of I	Lock the Back Door! Pv6 Network Security Po	
	Mapping Peering Interconnections to a F You've Got Vulnerability: Exploring Effective Vulner	Amp	onitoring and Defending A Dification DDoS Attacks	Analyzing the Grea	t Firewall of China	
	suring and Applying Invalid SSL Certificates: The Silent Majority	(arami° uxson <sup>†</sup> *	Killed by Proxy: Client-end TLS Interce	Over Space and Ti	me o Need for Black	
	Taejoong Chung: Yabing Liu* D	Malicious Activity w	vith DNS Backscatter		E-mail	

### Background

- $2^{128}$  addresses =>  $10^{30}$  years to scan
- 32 nybbles (hex characters), 8 groups
- 2001:0db8:0000:0001:0000:0000:22:33333
- *n* bit prefix + <u>m</u> bit subnet + 64 bit host ID
- Before 2001:0db8:0000:0001:0000:22:33333
- After 2001:db8:0:1::22:3333



# Current Strategy 1 – Use Known Patterns

Decouple where to scan from how to scan\*

• Target Generation Algorithm (TGA)

Previous Work:

Check Simple Patterns (2::1:0:0:0:1 ... 2::f:0:0:0:f) *Czyz et al.* 

Known Patterns eg. "wordy" (2001::cafe:face) RFC7707

# Current Strategy 2 – Discover Patterns

Extract patterns from "Seeds"

Seeds:

- Network Taps
- Traceroutes
- DNS
  - $\circ$  Reverse
  - Passive
  - $\circ$  Forward

Previous Work: Recursive Algorithms *Ullrich et al.* Machine Learning *Pawel et al.* 

# New Strategy – Exploit Locality

Goal: maximize number of hosts found\*

- Find address ranges local to seeds with high seed density
- Expand ranges to discover new addresses

Bottom up, expand from seeds to ranges

• Allocation patterns can be tricky to leverage

1K seeds matching a random pattern
prefix:subnet:<16 random nybbles>
16^16 possible targets

100 seeds matching a wordy pattern
prefix:subnet::<word>
1,296 possible targets

• 2/3 of routed prefixes had less than 10 seeds

• There may be different patterns in one subnet

2403:d000:0004:0100:0000:0000:0000:0001 Sequential 2403:d000:0004:0100:0000:0000:0002 2403:d000:0004:0100:0225:90ff:fe37:358b Embedded MAC 2403:d000:0004:0100:0225:90ff:fe37:760f 2403:d000:0004:0100:0230:48ff:fe34:fe96 2403:d000:0004:0100:0000:0000:café Wordy (Actual Seeds)

• Often networks do not allocate addresses using least significant nibbles

2a02:04e8:00de:1000:5b6d:0a03:0000:0001 2a02:04e8:00de:1000:5b6d:0a07:0000:0001 2a02:04e8:00de:1000:5b6d:0a08:0000:0001 2a02:04e8:00de:1000:5b6d:0a09:0000:0001 2a02:04e8:00de:1000:5b6d:0a0a:0000:0001 2a02:04e8:00de:1000:5b6d:0a0b:0000:0001 (Actual Seeds)

Find dense ranges *not* dense prefixes

• Whats going on here?

### What don't we do

- Rely on known patterns or strategies
- Reverse engineer allocation patterns
- Set algorithmic parameters
  - E.g. No notion /64 is significant,
  - no "arbitrary" thresholds





### Strategy

- Select ranges of addresses local to the seeds
- Target the most promising ranges first (high density)
- Expand these ranges to encourage discovery
- Sole parameter: "probe budget"

### **Generating Ranges**

#### Create a range

#### Grow a Range

```
2::1:?
2::2:b →2::[0-f]:[0-f]→2::?:?
```

# "Tight" vs "Loose" Ranges

- 2::3
- 2::5
- 2::9

#### 2::[3-9] Discovery space of 4

2::? -> 2::[0-f] Discovery space of 13

Uses more probes, but increases opportunity

# **Growing Ranges**

Grow ranges incrementally to support granular budget levels

Compute change in size with Hamming distance

- 2::<u>a</u> Hamming distance 1 2::<u>b</u> (2::? is 16<sup>1</sup> times larger than 2::a)
- 2::<u>1</u>:? Hamming distance 1 2::<u>2</u>:b

# Example

2::1

2::2

2::3

2::1:1

2::1:2

2::a0

2::b1

2::c3

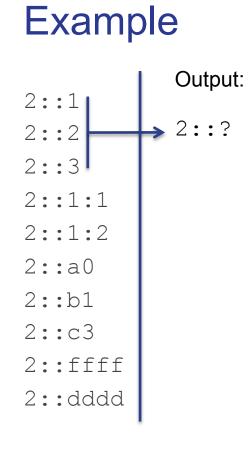
- - - -

2::ffff

2::dddd

Seed	Closest	Dist	Range	Density
2::1	2::2	1	2::?	3/161
		<u> </u>		8/164

Cost: 16



Seed	Closest	Dist	Range	Density	
2::1	2::2 2::ffff	1 4	2::? 2::????	3/16	5 <sup>1</sup> -
2 <b>::</b> 1 <b>:</b> 1	2 <b>::</b> 1 <b>:</b> 2	1	2 <b>::</b> 1 <b>:</b> ?	2/16	5 <sup>1</sup>
2 <b>::</b> a0	2 <b>::</b> b1	2	2::??	3/10	5 <sup>2</sup>
2 <b>::</b> ffff	2::dddd	4	2::????	2/16	5 <sup>4</sup>
•••					

Example	Seed	Closest	Dist	Range	Density
Output: 2::1 2::? 2::2	2::?	2::a0 2::1:1	1	2::?? 2::?:?	6/16 <sup>2</sup> 5/16 <sup>2</sup>
2::3 2::1:1 2::1:?					
2::1:2 2::a0					
2::b1 2::c3					
2::ffff 2::dddd					

Exam	ple	Seed	Closest	Dist	Range	Density
2::1 2::2	<b>Output:</b> 2 <b>: :</b> ?	2::?	2::a0 2::1:1	1 1	2::?? 2::?:?	<b>6/16<sup>2</sup></b> 5/16 <sup>2</sup>
2::2		2::1:?	2::1	1	2::?:?	5/16 <sup>2</sup>
2::1:1	2::1:?	2 <b>::</b> a0	2 <b>::</b> b1	2	2::??	3/162
2::1:2 2::a0		2::ffff	2::dddd	4	2::????	2/164
2::b1		•••				
2::c3						
2::ffff 2::dddd						

Cost:	16 <sup>2</sup> +	16 =	272
-------	-------------------	------	-----

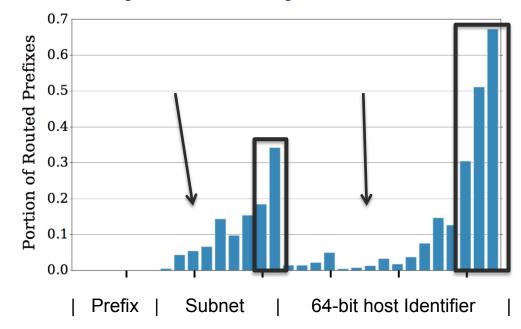
Examp	ble					
		Seed	Closest	Dist	Range	Density
2::1 2::2	Output: <del>2::?</del> 2::??	2::?	2::a0 2::1:1	1 1	2::?? 2::?:?	<b>6/16<sup>2</sup></b> 5/16 <sup>2</sup>
2::2		2::1:?	2::1	1	2::?:?	5/16 <sup>2</sup>
2::1:1	2::1:?	<del>2::a0</del>	<del>2::b1</del>	2	2::??	<del>3/16<sup>2</sup></del>
2::1:2		2::ffff	2::dddd	4	2::????	2/164
2::a0 2::b1						
2::c3						
2::ffff				1		
2::dddd						

### **Evaluation**

- 1. ~3M DNS AAAA seeds from Rapid7
  - ~ 8K routes prefixes\*
  - ~7K ASes
- 2. Run 6Gen on each routed prefix (1M budget per prefix)
- 3. Convert list of target ranges to addresses (~6B targets)\*\*
- 4. Probe addresses on tcp/80 (SYN scan)

Post talk note: \*In the talk I mentioned that this total is for prefixes with 2 or more seeds. In the paper we do not remove prefixes with one seed and report this number as 10,038. \*\*I mention in the talk that this total is less than 8B because 6Gen does not always generate 1M targets.

# Where are the dynamic nybbles?



### **Evaluation**

~55 million responses from ~6B probes

- ~30 Million from Akamai
- ~20 Million from Amazon

Encounter large blocks of responsive addresses

• E.g., Akamai has "active" /56s

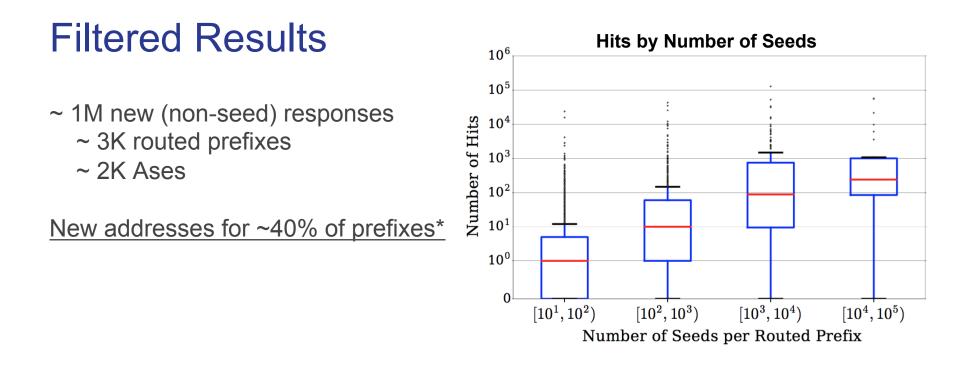
### How can we quickly detect large active regions?

Randomly probe each /96 -> 2<sup>32</sup> possible addresses

Filter removed { 10.0 M / 10.2 M } /96s from 138 ASes

Manually removed two additional ASes (after /96 filtering)

/96 filter + manual inspection removed 98% of hits



Post talk note: \*In the talk I mentioned that this percentage is for prefixes with 2 or more seeds. In the paper we do not exclude prefixes with one seed and report this metric as 28%.

### **Future Work**

Better detection of "active" blocks

Adaptive Scanning

- Density validation
- Pattern recognition for ranges

# Thank You

Austin Murdock austinmurdock@berkeley.edu @austinkarch