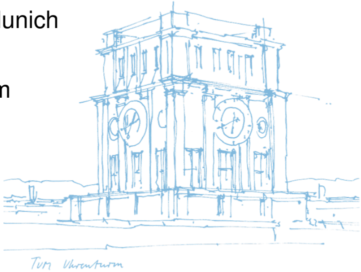
Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists

Oliver Gasser

Technical University of Munich

RIPE 77, Amsterdam



Joint work








Internet measurements

Active Internet measurements

- Important tool to understand specific networks
  - Which IP addresses run an HTTPS web server in the Internet?
  - How securely configured are IoT devices in a company network?
  - Are my DNS servers vulnerable to amplification attacks?
- Used by researchers, security companies,...

Internet measurements

Active Internet measurements

- Important tool to understand specific networks
  - Which IP addresses run an HTTPS web server in the Internet?
  - How securely configured are IoT devices in a company network?
  - Are my DNS servers vulnerable to amplification attacks?
- Used by researchers, security companies,... but also bad actors

Internet measurements

Active Internet measurements

- Important tool to understand specific networks
  - Which IP addresses run an HTTPS web server in the Internet?
  - How securely configured are IoT devices in a company network?
  - Are my DNS servers vulnerable to amplification attacks?
- Used by researchers, security companies,... but also bad actors

Why is this research relevant for operators?

- Learn measurements techniques used in IPv6 vs. in IPv4
- Understand how devices can be discovered in your network
- Take action by conducting measurements yourself

IPv6 measurements

Differences in IPv4 and IPv6 measurement approaches

- IPv4
  - Brute-force scan complete Internet in a few hours (e.g. ZMap)
- IPv6
  - Address space too expansive for brute force scanning
  - Assemble target list of IPv6 addresses for scanning $\rightarrow$ IPv6 hitlist

IPv6 hitlist

Assembling an IPv6 hitlist

- Leverage DNS to gather IPv6 addresses
- Exploit structural properties to learn new addresses
- Use crowdsourcing to get client addresses

IPv6 hitlist

Assembling an IPv6 hitlist

- Leverage DNS to gather IPv6 addresses
- Exploit structural properties to learn new addresses
- Use crowdsourcing to get client addresses

Challenges

1. Clusters in hitlist sources
2. Aliased prefixes
3. Finding reachable addresses

Hitlist sources

Where can we learn potential IPv6 addresses?

Hitlist sources

Where can we learn potential IPv6 addresses?

- Domain lists: zonefiles, toplists, blacklists
- Rapid7 ANY DNS
- Domains extracted from Certificate Transparency
- Bitcoin node addresses
- RIPE Atlas: traceroutes, ipmap
- Scamper: traceroute to all assembled addresses

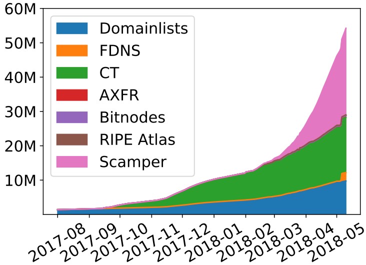


Figure 1: Cumulative runup of IPv6 addresses.

Hitlist sources

Where can we learn potential IPv6 addresses?

- Domain lists: zonefiles, toplists, blacklists
- Rapid7 ANY DNS
- Domains extracted from Certificate Transparency
- Bitcoin node addresses
- RIPE Atlas: traceroutes, ipmap
- Scamper: traceroute to all assembled addresses

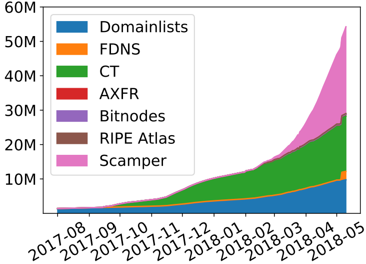


Figure 1: Cumulative runup of IPv6 addresses.

Observation

- Many addresses from domain lists, CT, and scamper

Hitlist sources

How diverse are the addresses from different sources?

Hitlist sources

How diverse are the addresses from different sources?

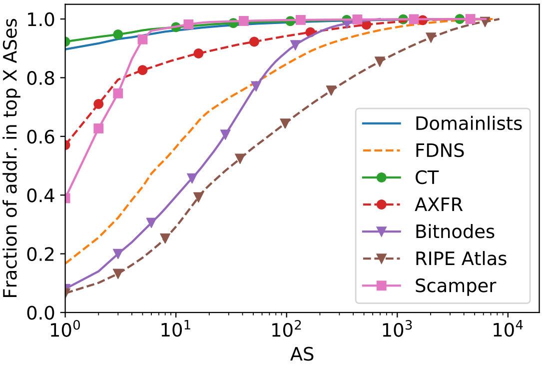


Figure 2: AS distribution for hitlist sources.

Hitlist sources

How diverse are the addresses from different sources?

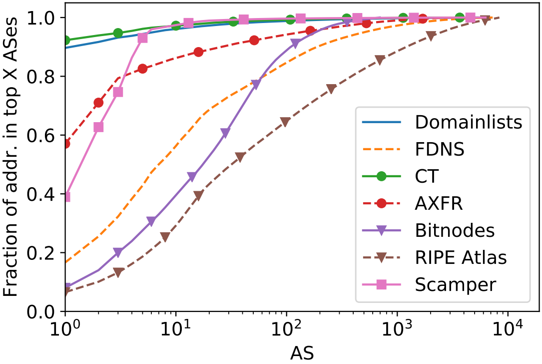


Figure 2: AS distribution for hitlist sources.

Autonomous System distribution

- Unbalanced (CT, domain lists) vs. balanced (RIPE Atlas)

Hitlist sources

How much of the announced address space do we cover?

Hitlist sources

How much of the announced address space do we cover?

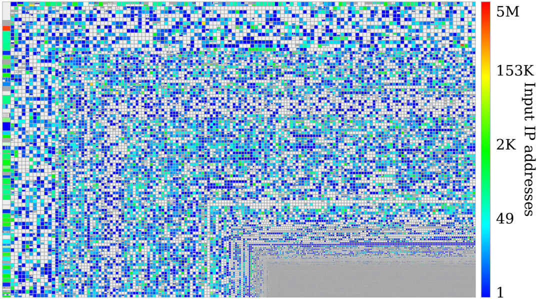


Figure 3: IPv6 prefixes with number of hitlist addresses per prefix.

Hitlist sources

How much of the announced address space do we cover?

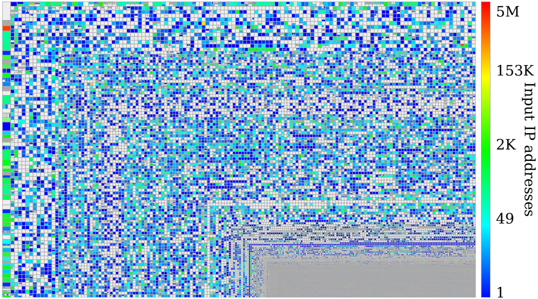


Figure 3: IPv6 prefixes with number of hitlist addresses per prefix.

BGP prefix distribution

- Good coverage of BGP prefixes: 25.5 k of 51.2 k
- Some prefixes with many addresses

Hitlist sources

Key take-aways for network operations

1. IPv6 address space too vast to conduct brute-force measurements
2. Your addresses can be gathered from many different publicly available sources (e.g. DNS, CT)
3. About 50 % of announced prefixes are covered in our IPv6 hitlist

Address entropy clustering

Addressing schemes

- Question: How similar are addressing schemes in our hitlist?
- Approach: Group addresses to find similar address schemes

Address entropy clustering

Addressing schemes

- Question: How similar are addressing schemes in our hitlist?
- Approach: Group addresses to find similar address schemes

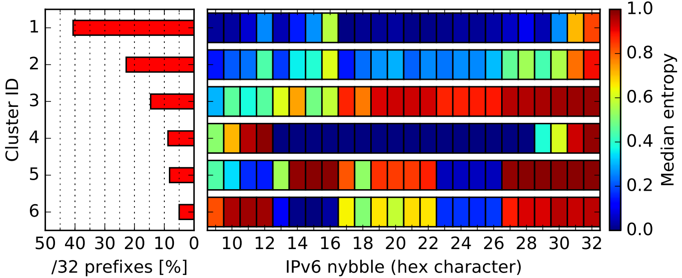


Figure 4: Addressing schemes.

Address entropy clustering

Addressing schemes

- Question: How similar are addressing schemes in our hitlist?
- Approach: Group addresses to find similar address schemes

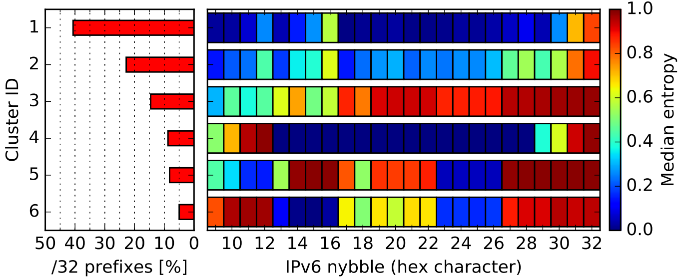


Figure 4: Addressing schemes.

- Only few addressing schemes
- Low-bit addresses (e.g. `::1`), privacy extensions, and EUI-64 mapped MAC addresses clearly visible

Address entropy clustering

Key take-aways for network operations

1. Most networks use one of a handful of addressing schemes
2. Good: Industry best practices are followed
3. Bad: Addressing schemes might uncover “hidden” hosts

Detecting aliased prefixes

Aliases

- Alias: Multiple addresses belonging to the same host
- Aliased prefix: Complete prefix bound to the same host
- Bias: As some hosts are overrepresented, aliased prefixes introduce bias in the hitlist

Detecting aliased prefixes

Aliases

- Alias: Multiple addresses belonging to the same host
- Aliased prefix: Complete prefix bound to the same host
- Bias: As some hosts are overrepresented, aliased prefixes introduce bias in the hitlist

Detecting aliased prefixes using pseudo-random probing

2001:0db8:0407:8000::/64
2001:0db8:0407:8000:0151:2900:77e9:03a8
⋮
2001:0db8:0407:8000:f693:2443:915e:1d2e

Table 1: IPv6 fan-out for multi-level aliased prefix detection.

Detecting aliased prefixes

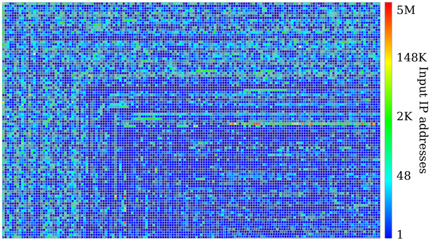


Figure 5: All prefixes covered by hitlist.

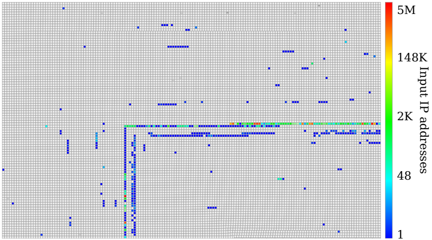


Figure 6: Aliased prefixes.

- 55.1 M raw IPv6 addresses in hitlist
- Few prefixes are aliased (e.g. Amazon, see right figure)
- 25.7 M IPv6 addresses in aliased prefixes (46.6 %)
- Validation using fingerprinting (iTTL, TCP opts, TCP TS)

Detecting aliased prefixes

Key take-aways for network operations

1. Aliased prefixes can introduce bias in IPv6 measurements
2. Can be detected with pseudo-random probing
3. Using aliasing to hide your prefixes and hosts is not very effective

Address responsiveness

Cross protocol responsiveness

- If address responds on protocol X, how likely is it to respond on protocol Y?
- Goal: Identify relevant addresses for specific measurements

Address responsiveness

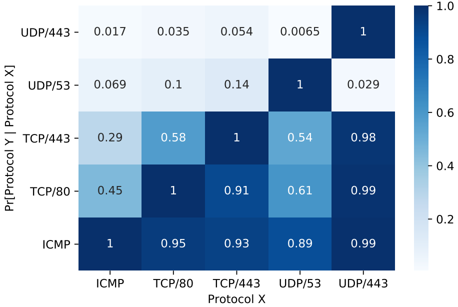


Figure 7: Likeliness to respond on protocol Y, if responding to protocol X.

Address responsiveness

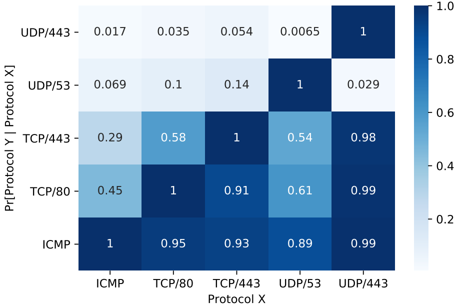


Figure 7: Likeliness to respond on protocol Y, if responding to protocol X.

- If responsive to one of the probes → at least 89% chance it will answer to ICMPv6
- Web protocols: QUIC → HTTPS and HTTP, HTTPS → HTTP; but not the other way around

Address responsiveness

Key take-aways for network operations

1. Knowing responsiveness on one service might leak information about other services
2. Horizontal port scanning on all devices is not necessary
3. Attackers might pick one port (e.g. TCP/80) and then continue with only responsive hosts

Learning new addresses

Techniques to learn new addresses

- Entropy/IP: Generate new addresses by leveraging entropy of seed addresses
  - Similar approach to grouping addresses based on their structure as shown earlier
  - Presented at RIPE74 in Budapest by Paweł Foremski

Learning new addresses

Techniques to learn new addresses

- Entropy/IP: Generate new addresses by leveraging entropy of seed addresses
  - Similar approach to grouping addresses based on their structure as shown earlier
  - Presented at RIPE74 in Budapest by Paweł Foremski
- 6Gen: Generate new addresses in dense address regions
  - If we see addresses
    - `2001:0db8:0407:8000::`**3**
    - `2001:0db8:0407:8000::`**4**
    - `2001:0db8:0407:8000::`**5**
    - `2001:0db8:0407:8000::`**8**
    - `2001:0db8:0407:8000::`**9**
  - Likely other valid addresses
    - `2001:0db8:0407:8000::`**6**
    - `2001:0db8:0407:8000::`**7**

Learning new addresses

How well do Entropy/IP and 6Gen perform?

- Input: All previously found IPv6 addresses
- Generation: 118 M and 129 M, only 675 k overlapping
- Responsiveness: 278 k and 489 k
- Magnitude higher response rate for overlapping addresses

Learning new addresses

How well do Entropy/IP and 6Gen perform?

- Input: All previously found IPv6 addresses
- Generation: 118 M and 129 M, only 675 k overlapping
- Responsiveness: 278 k and 489 k
- Magnitude higher response rate for overlapping addresses

Table 2: Top 5 responsive protocol combinations for 6Gen and Entropy/IP.

ICMP	TCP/80	TCP/443	UDP/53	UDP/443	6Gen	Entropy/IP
✓	✗	✗	✗	✗	66.8 %	41.1 %
✓	✓	✓	✗	✗	9.2 %	12.3 %
✗	✗	✗	✓	✗	7.3 %	23.1 %
✓	✓	✗	✗	✗	4.9 %	3.4 %
✓	✓	✓	✗	✓	3.2 %	6.1 %

Learning new addresses

How well do Entropy/IP and 6Gen perform?

- Input: All previously found IPv6 addresses
- Generation: 118 M and 129 M, only 675 k overlapping
- Responsiveness: 278 k and 489 k
- Magnitude higher response rate for overlapping addresses

Table 2: Top 5 responsive protocol combinations for 6Gen and Entropy/IP.

ICMP	TCP/80	TCP/443	UDP/53	UDP/443	6Gen	Entropy/IP
✓	✗	✗	✗	✗	66.8 %	41.1 %
✓	✓	✓	✗	✗	9.2 %	12.3 %
✗	✗	✗	✓	✗	7.3 %	23.1 %
✓	✓	✗	✗	✗	4.9 %	3.4 %
✓	✓	✓	✗	✓	3.2 %	6.1 %

- Different host populations

Learning new addresses

Key take-aways for network operations

1. Address learning uncovers previously unknown addresses
2. Techniques provide complementary address sets
3. Hiding in the expansive IPv6 address space might be more difficult

Conclusion

- IPv6 Internet too vast to conduct brute-force measurements
- But you might be less “hidden” in IPv6 than you’d have thought
- Addressing schemes might uncover “hidden” hosts
- Responsiveness of one service might leak information about other services

Conclusion

- IPv6 Internet too vast to conduct brute-force measurements
- But you might be less “hidden” in IPv6 than you’d have thought
- Addressing schemes might uncover “hidden” hosts
- Responsiveness of one service might leak information about other services

ipv6hitlist.github.io

Conclusion

- IPv6 Internet too vast to conduct brute-force measurements
- But you might be less “hidden” in IPv6 than you’d have thought
- Addressing schemes might uncover “hidden” hosts
- Responsiveness of one service might leak information about other services

ipv6hitlist.github.io

Oliver Gasser <gasser@net.in.tum.de>

`https://www.net.in.tum.de/~gasser/`