Target Acquired? Evaluating Target Generation Algorithms for IPv6

Lion Steger, Liming Kuang, Johannes Zirngibl
Georg Carle, Oliver Gasser

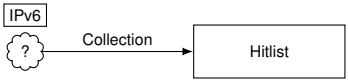
Monday 12th June, 2023

Chair of Network Architectures and Services
School of Computation, Information, and Technology
Technical University of Munich

Motivation

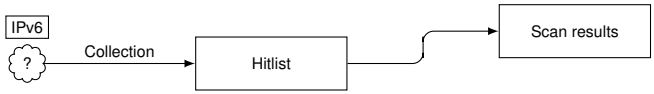
- The IPv6 address space is too large and sparse be scanned exhaustively.

Motivation



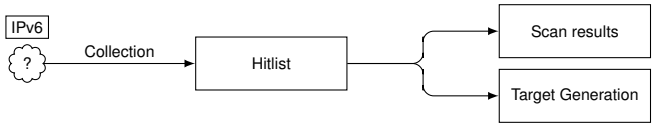
- The IPv6 address space is too large and sparse be scanned exhaustively.
- Internet measurements rely on collections of active IPv6 addresses called hitlists.

Motivation



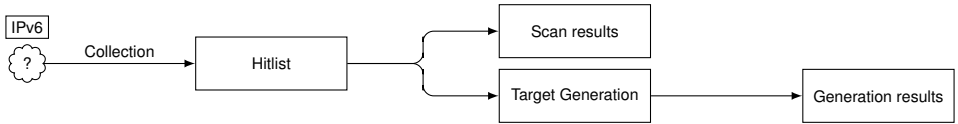
- The IPv6 address space is too large and sparse be scanned exhaustively.
- Internet measurements rely on collections of active IPv6 addresses called hitlists.

Motivation



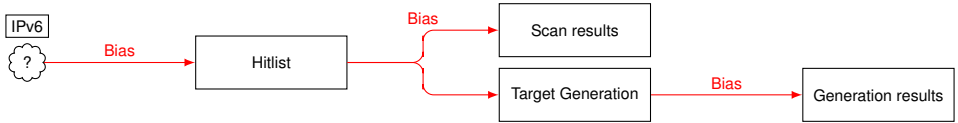
- The IPv6 address space is too large and sparse be scanned exhaustively.
- Internet measurements rely on collections of active IPv6 addresses called hitlists.
- Often used by Target Generation Algorithms (TGAs) to generate more addresses.

Motivation



- The IPv6 address space is too large and sparse be scanned exhaustively.
- Internet measurements rely on collections of active IPv6 addresses called hitlists.
- Often used by Target Generation Algorithms (TGAs) to generate more addresses.

Motivation



- The IPv6 address space is too large and sparse be scanned exhaustively.
- Internet measurements rely on collections of active IPv6 addresses called hitlists.
- Often used by Target Generation Algorithms (TGAs) to generate more addresses.
- Can they represent the IPv6 Internet or are they biased?

Motivation

Research Questions

Client devices, web servers, Internet infrastructure are all seen as part of a homogenous set.

Motivation

Research Questions

Client devices, web servers, Internet infrastructure are all seen as part of a homogenous set.

- Are popular hitlists biased towards certain address types? How do different address types behave?

Motivation

Research Questions

Client devices, web servers, Internet infrastructure are all seen as part of a homogenous set.

- Are popular hitlists biased towards certain address types? How do different address types behave?
  → We analyze the IPv6 Hitlist Service.

Motivation

Research Questions

Client devices, web servers, Internet infrastructure are all seen as part of a homogenous set.

- Are popular hitlists biased towards certain address types? How do different address types behave?
  → We analyze the IPv6 Hitlist Service.
- How do TGAs behave with biased input?

Motivation

Research Questions

Client devices, web servers, Internet infrastructure are all seen as part of a homogenous set.

- Are popular hitlists biased towards certain address types? How do different address types behave?
  → We analyze the IPv6 Hitlist Service.
- How do TGAs behave with biased input?
  → We evaluate ten different TGAs.

Motivation

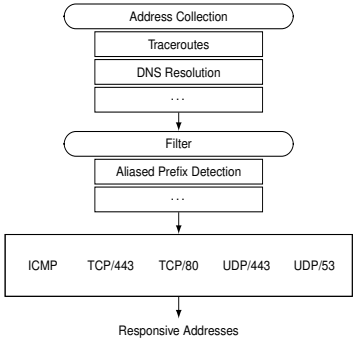
Research Questions

Client devices, web servers, Internet infrastructure are all seen as part of a homogenous set.

- Are popular hitlists biased towards certain address types? How do different address types behave?
  → We analyze the IPv6 Hitlist Service.
- How do TGAs behave with biased input?
  → We evaluate ten different TGAs.
- What are the benefits of categorizing the hitlist contents?

The IPv6 Hitlist service

- Service was introduced by Gasser et al. in 2018. [1]

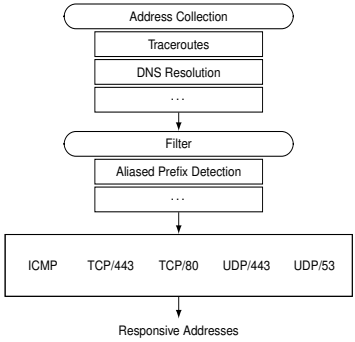


[1] O. Gasser, Q. Scheitle, P. Foremski, et al., "Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists," in Proc. ACM Int. Measurement Conference (IMC), Boston, MA, USA, 2018. DOI: 10.1145/3278532.3278564

[2] J. Zirngibl, L. Steger, P. Sattler, et al., "Rusty clusters? dusting an IPv6 research foundation," in Proc. ACM Int. Measurement Conference (IMC), Nice, France, 2022. DOI: 10.1145/3517745.3561440

The IPv6 Hitlist service

- Service was introduced by Gasser et al. in 2018. [1]
- Collects more than 1.1B addresses from various sources.

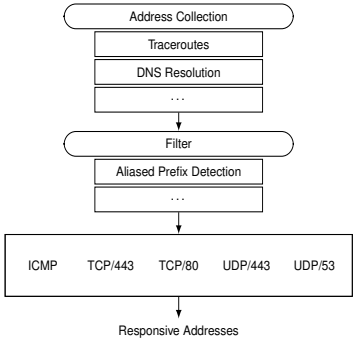


[1] O. Gasser, Q. Scheitle, P. Foremski, et al., "Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists," in Proc. ACM Int. Measurement Conference (IMC), Boston, MA, USA, 2018. DOI: 10.1145/3278532.3278564

[2] J. Zirngibl, L. Steger, P. Sattler, et al., "Rusty clusters? dusting an IPv6 research foundation," in Proc. ACM Int. Measurement Conference (IMC), Nice, France, 2022. DOI: 10.1145/3517745.3561440

The IPv6 Hitlist service

- Service was introduced by Gasser et al. in 2018. [1]
- Collects more than 1.1B addresses from various sources.
- Runs addresses through filters and scans them on different ports.
  - TCP/80, TCP/443, UDP/53, UDP/443, ICMP.

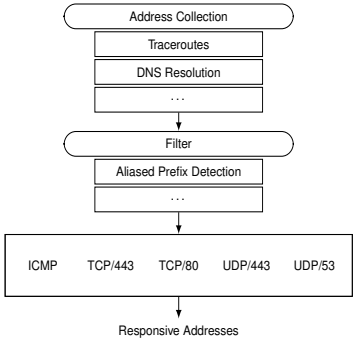


[1] O. Gasser, Q. Scheitle, P. Foremski, et al., "Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists," in Proc. ACM Int. Measurement Conference (IMC), Boston, MA, USA, 2018. DOI: 10.1145/3278532.3278564

[2] J. Zirngibl, L. Steger, P. Sattler, et al., "Rusty clusters? dusting an IPv6 research foundation," in Proc. ACM Int. Measurement Conference (IMC), Nice, France, 2022. DOI: 10.1145/3517745.3561440

The IPv6 Hitlist service

- Service was introduced by Gasser et al. in 2018. [1]
- Collects more than 1.1B addresses from various sources.
- Runs addresses through filters and scans them on different ports.
  - TCP/80, TCP/443, UDP/53, UDP/443, ICMP.
- Contains 6.7 M addresses responsive on at least one port.

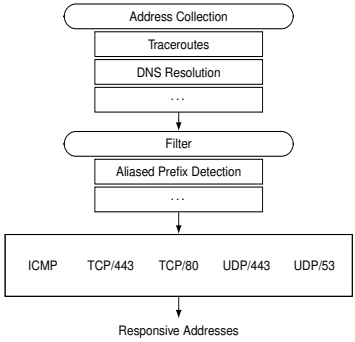


[1] O. Gasser, Q. Scheitle, P. Foremski, et al., "Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists," in Proc. ACM Int. Measurement Conference (IMC), Boston, MA, USA, 2018. DOI: 10.1145/3278532.3278564

[2] J. Zirngibl, L. Steger, P. Sattler, et al., "Rusty clusters? dusting an IPv6 research foundation," in Proc. ACM Int. Measurement Conference (IMC), Nice, France, 2022. DOI: 10.1145/3517745.3561440

The IPv6 Hitlist service

- Service was introduced by Gasser et al. in 2018. [1]
- Collects more than 1.1B addresses from various sources.
- Runs addresses through filters and scans them on different ports.
  - TCP/80, TCP/443, UDP/53, UDP/443, ICMP.
- Contains 6.7 M addresses responsive on at least one port.
- TGAs were employed by Zirngibl et al. in 2022. [2]
  - Generate new addresses from Hitlist addresses.
  - Used to increase coverage of the IPv6 address space by 168%.



[1] O. Gasser, Q. Scheitle, P. Foremski, et al., "Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists," in Proc. ACM Int. Measurement Conference (IMC), Boston, MA, USA, 2018. DOI: 10.1145/3278532.3278564

[2] J. Zirngibl, L. Steger, P. Sattler, et al., "Rusty clusters? dusting an IPv6 research foundation," in Proc. ACM Int. Measurement Conference (IMC), Nice, France, 2022. DOI: 10.1145/3517745.3561440

Analyzing the IPv6 Hitlist

Category distribution

- Device type can only be estimated.

[3] https://www.peeringdb.com/

Analyzing the IPv6 Hitlist

Category distribution

- Device type can only be estimated.
- Analysis of origin network.

[3] https://www.peeringdb.com/

Analyzing the IPv6 Hitlist

Category distribution

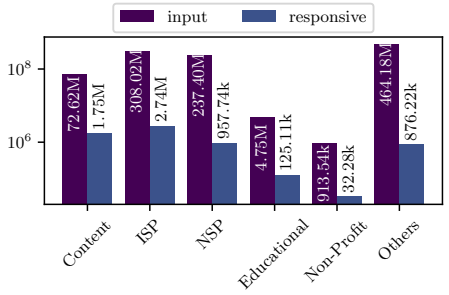
- Device type can only be estimated.
- Analysis of origin network.
- Categorization via PeeringDB. [3]
  - Community-maintained database.
  - Offers categorization on AS-level.
  - Includes 11 categories, we chose 5.
  - Remaining categories combined to *Others*.

[3] https://www.peeringdb.com/

Analyzing the IPv6 Hitlist

Category distribution

- Device type can only be estimated.
- Analysis of origin network.
- Categorization via PeeringDB. [3]
  - Community-maintained database.
  - Offers categorization on AS-level.
  - Includes 11 categories, we chose 5.
  - Remaining categories combined to *Others*.
- Category representation in Hitlist is not uniform.

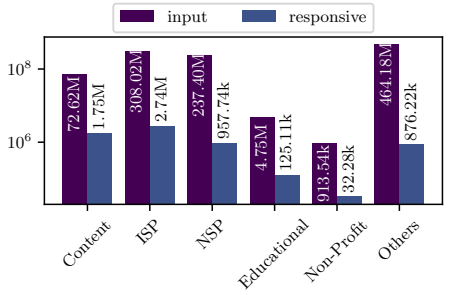


[3] https://www.peeringdb.com/

Analyzing the IPv6 Hitlist

Category distribution

- Device type can only be estimated.
- Analysis of origin network.
- Categorization via PeeringDB. [3]
  - Community-maintained database.
  - Offers categorization on AS-level.
  - Includes 11 categories, we chose 5.
  - Remaining categories combined to *Others*.
- Category representation in Hitlist is not uniform.
- Most frequent categories are ISP, CDN and NSP.



[3] https://www.peeringdb.com/

Analyzing the IPv6 Hitlist

Category Behavior

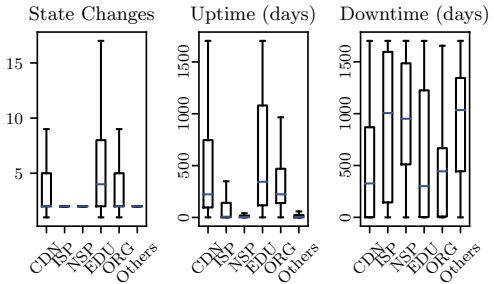
Difference in temporal stability:

Analyzing the IPv6 Hitlist

Category Behavior

Difference in temporal stability:

- State changes denote a change in responsiveness.

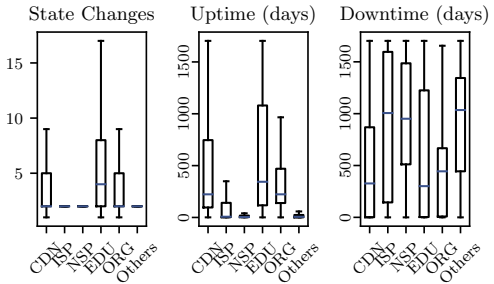


Analyzing the IPv6 Hitlist

Category Behavior

Difference in temporal stability:

- State changes denote a change in responsiveness.
- Sum of down- and uptimes since inclusion in Hitlist.

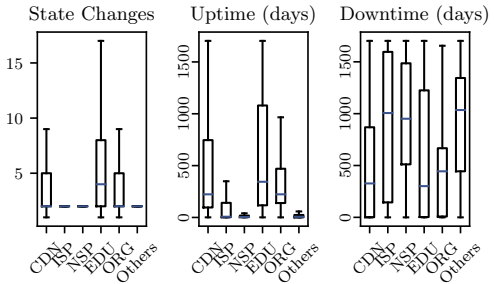


Analyzing the IPv6 Hitlist

Category Behavior

Difference in temporal stability:

- State changes denote a change in responsiveness.
- Sum of down- and uptimes since inclusion in Hitlist.
- ISP addresses have one uptime around seven days.

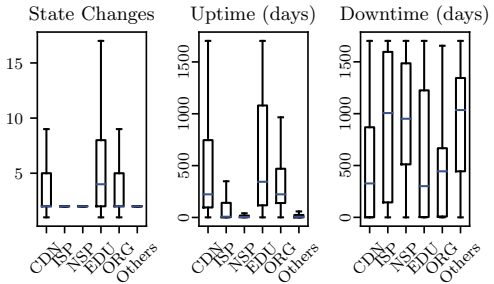


Analyzing the IPv6 Hitlist

Category Behavior

Difference in temporal stability:

- State changes denote a change in responsiveness.
- Sum of down- and uptimes since inclusion in Hitlist.
- ISP addresses have one uptime around seven days.
- CDN addresses have longer average uptimes and a low number of downtimes.

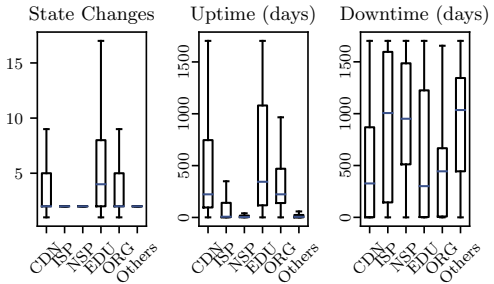


Analyzing the IPv6 Hitlist

Category Behavior

Difference in temporal stability:

- State changes denote a change in responsiveness.
- Sum of down- and uptimes since inclusion in Hitlist.
- ISP addresses have one uptime around seven days.
- CDN addresses have longer average uptimes and a low number of downtimes.
- Should be considered in longitudinal measurements.



Analyzing the IPv6 Hitlist

Category Behavior

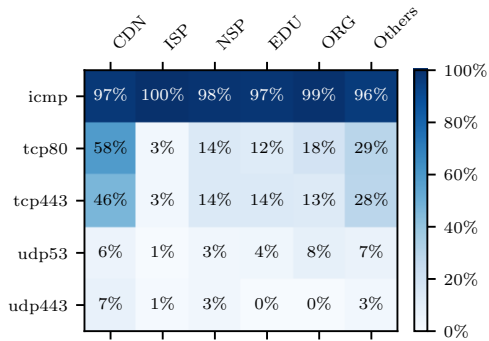
Difference in port responsiveness:

Analyzing the IPv6 Hitlist

Category Behavior

Difference in port responsiveness:

- Response rate to each port probe per category.

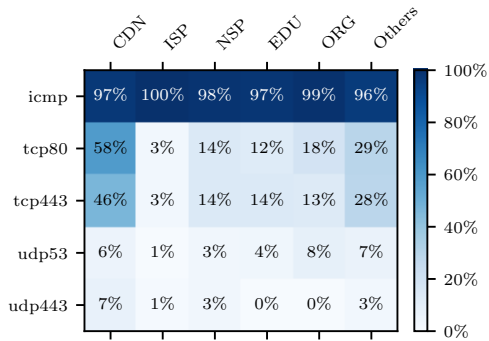


Analyzing the IPv6 Hitlist

Category Behavior

Difference in port responsiveness:

- Response rate to each port probe per category.
- Categories share high response rates to ICMP probes.

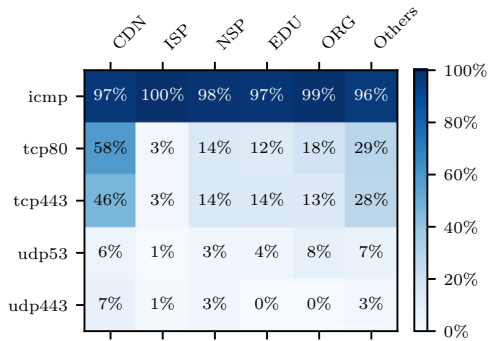


Analyzing the IPv6 Hitlist

Category Behavior

Difference in port responsiveness:

- Response rate to each port probe per category.
- Categories share high response rates to ICMP probes.
- ISP addresses only have high response rates to ICMP.

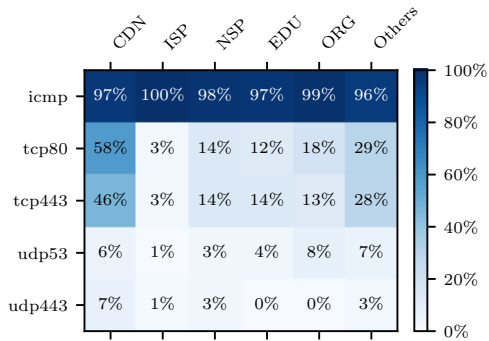


Analyzing the IPv6 Hitlist

Category Behavior

Difference in port responsiveness:

- Response rate to each port probe per category.
- Categories share high response rates to ICMP probes.
- ISP addresses only have high response rates to ICMP.
- CDN addresses have the highest response rates to TCP/80, TCP/443 and UDP/443.

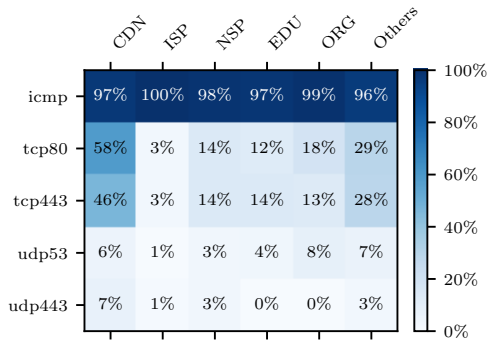


Analyzing the IPv6 Hitlist

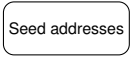
Category Behavior

Difference in port responsiveness:

- Response rate to each port probe per category.
- Categories share high response rates to ICMP probes.
- ISP addresses only have high response rates to ICMP.
- CDN addresses have the highest response rates to TCP/80, TCP/443 and UDP/443.
- Port responses are important depending on use case.



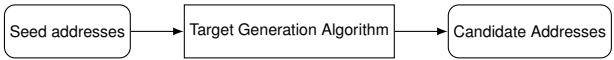
Target Generation



Target Generation

- Target Generation Algorithms (TGAs) discover patterns in known active addresses (seed data set).

Target Generation



- Target Generation Algorithms (TGAs) discover patterns in known active addresses (seed data set).
- Generate new potentially active addresses (candidate data set).

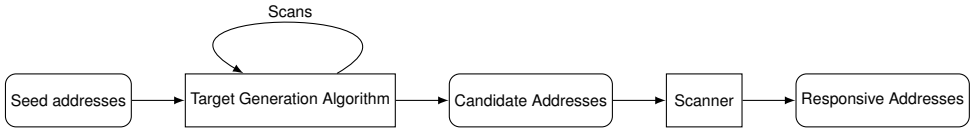
Target Generation

- Target Generation Algorithms (TGAs) discover patterns in known active addresses (seed data set).
- Generate new potentially active addresses (candidate data set).
- Candidates are scanned to check responsiveness.

Target Generation

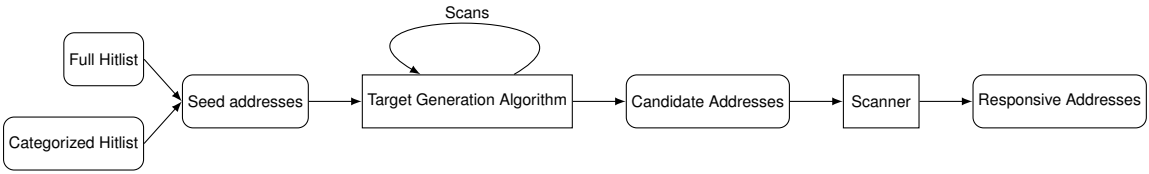
- Target Generation Algorithms (TGAs) discover patterns in known active addresses (seed data set).
- Generate new potentially active addresses (candidate data set).
- Candidates are scanned to check responsiveness.

Target Generation



- Target Generation Algorithms (TGAs) discover patterns in known active addresses (seed data set).
- Generate new potentially active addresses (candidate data set).
- Candidates are scanned to check responsiveness.
- Some algorithms implement custom scanning to dynamically adapt generation.

Target Generation



- Target Generation Algorithms (TGAs) discover patterns in known active addresses (seed data set).
- Generate new potentially active addresses (candidate data set).
- Candidates are scanned to check responsiveness.
- Some algorithms implement custom scanning to dynamically adapt generation.
- We use the full Hitlist (default input) as well as the categorized Hitlist (specific input).

Target Generation

Target Generation Algorithms

- We choose 10 open source algorithms from peer-reviewed publications.
- Methods include, language models, machine learning, graph theory.

Year	Authors	Name	Scanning	Ref
2016	Foremski et al.	Entropy/IP	Static	[3]
2019	Liu et al.	6Tree	Dynamic	[4]
2020	Song et al.	DET	Dynamic	[5]
2020	Cui et al.	6GCVAE	Static	[6]
2021	Cui et al.	6VecLM	Static	[7]
2021	Cui et al.	6GAN	Static	[8]
2021	Hou et al.	6Hit	Dynamic	[9]
2022	Yang et al.	6Graph	Static	[10]
2022	Yang et al.	6Forest	Static	[11]
2023	Hou et al.	6Scan	Dynamic	[12]

Target Generation

Generation results

	6Graph		6Scan		6VecLM		...
	cand.	resp.	cand.	resp.	cand.	resp.	...
ISP	25M	3M	8M	4M	18k	2k	...
EDU	2M	22k	10M	38k	84k	1k	...
Non-Profit	296k	15k	10M	946k	0	0	...
...	...	...	...	...	...	...	...
Full	106M	5M	6M	2M	49k	4k	...

- Size of candidates (cand.) varies greatly from 18 k (or zero for 6VecLM) to 106 M.
- Size of candidate set depends on algorithm as well as input.

Target Generation

Default behavior

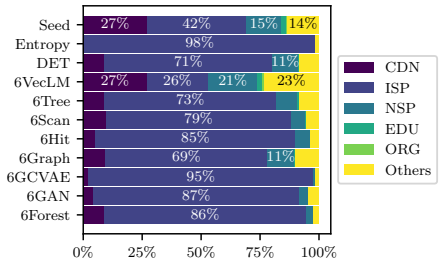
Target generation on default input:

Target Generation

Default behavior

Target generation on default input:

- Biased towards discovering more ISP addresses.

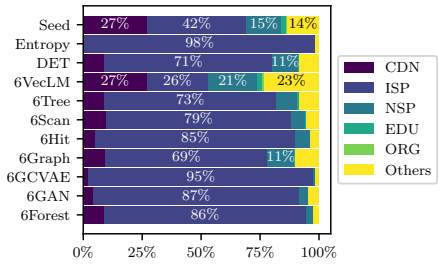


Target Generation

Default behavior

Target generation on default input:

- Biased towards discovering more ISP addresses.
- ISP addresses cointain more recognizable patterns.

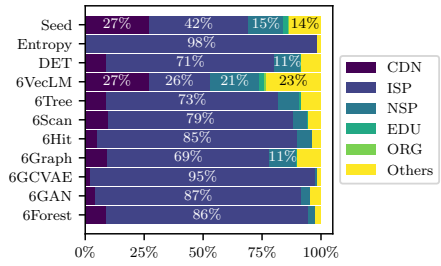


Target Generation

Default behavior

Target generation on default input:

- Biased towards discovering more ISP addresses.
- ISP addresses cointain more recognizable patterns.
- All algorithms except 6VecLM generate a higher percentage of ISP addresses than contained in the seeds.



Target Generation

Default behavior

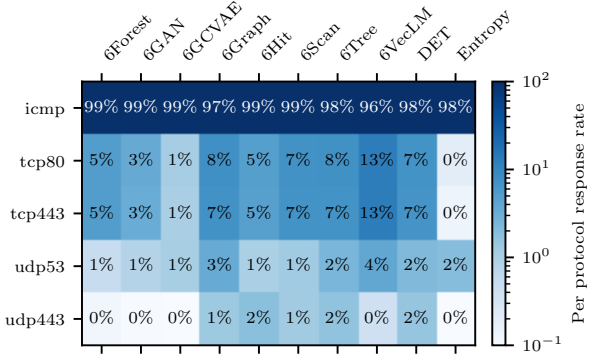
Port responses on default input:

Target Generation

Default behavior

Port responses on default input:

- Low port responses except for ICMP.

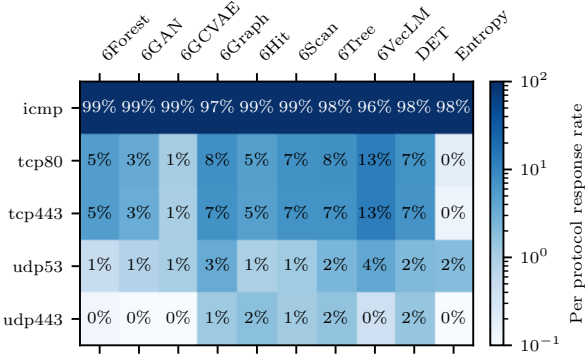


Target Generation

Default behavior

Port responses on default input:

- Low port responses except for ICMP.
- Default input introduces large scan overhead for some use cases.



Target Generation

Input-dependent behavior

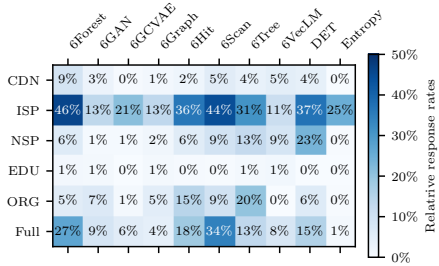
Category-dependent response rates:

Target Generation

Input-dependent behavior

Category-dependent response rates:

- Percentage of generated addresses per input responsive on at least one port.

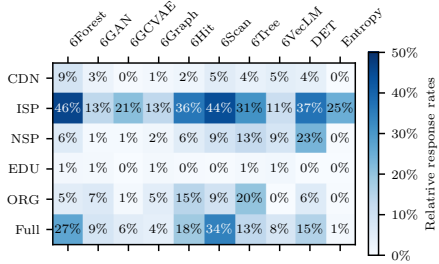


Target Generation

Input-dependent behavior

Category-dependent response rates:

- Percentage of generated addresses per input responsive on at least one port.
- TGAs have vastly different response rates.

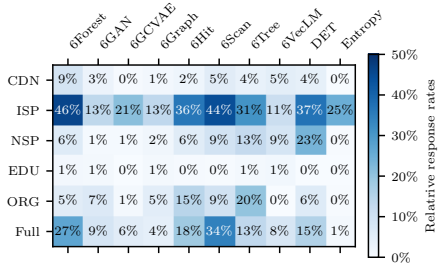


Target Generation

Input-dependent behavior

Category-dependent response rates:

- Percentage of generated addresses per input responsive on at least one port.
- TGAs have vastly different response rates.
- Dynamic algorithms have higher response rates.

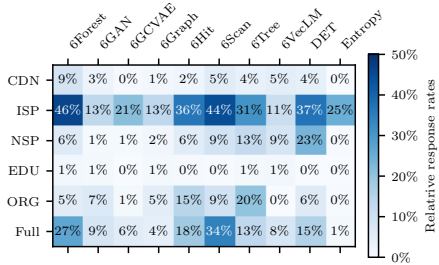


Target Generation

Input-dependent behavior

Category-dependent response rates:

- Percentage of generated addresses per input responsive on at least one port.
- TGAs have vastly different response rates.
- Dynamic algorithms have higher response rates.
- ISP input yields more responsive addresses.



Conclusion

- Network categories are not evenly distributed in the Hitlist service.

https://ipv6hitlist.github.io

Conclusion

- Network categories are not evenly distributed in the Hitlist service.
  - Bias towards ISP addresses.

https://ipv6hitlist.github.io

Conclusion

- Network categories are not evenly distributed in the Hitlist service.
  - Bias towards ISP addresses.
  - Categories show different behavior in port responses and temporal stability.

https://ipv6hitlist.github.io

Conclusion

- Network categories are not evenly distributed in the Hitlist service.
  - Bias towards ISP addresses.
  - Categories show different behavior in port responses and temporal stability.
  - ISP addresses are less stable than CDN addresses.

https://ipv6hitlist.github.io

Conclusion

- Network categories are not evenly distributed in the Hitlist service.
  - Bias towards ISP addresses.
  - Categories show different behavior in port responses and temporal stability.
  - ISP addresses are less stable than CDN addresses.
  - ISP responds only to ICMP, CDN best to TCP/80,443, UDP/443.

https://ipv6hitlist.github.io

Conclusion

- Network categories are not evenly distributed in the Hitlist service.
  - Bias towards ISP addresses.
  - Categories show different behavior in port responses and temporal stability.
  - ISP addresses are less stable than CDN addresses.
  - ISP responds only to ICMP, CDN best to TCP/80,443, UDP/443.
    → **Filtering input can avoid scanning overhead**.

https://ipv6hitlist.github.io

Conclusion

- Network categories are not evenly distributed in the Hitlist service.
  - Bias towards ISP addresses.
  - Categories show different behavior in port responses and temporal stability.
  - ISP addresses are less stable than CDN addresses.
  - ISP responds only to ICMP, CDN best to TCP/80,443, UDP/443.
    → **Filtering input can avoid scanning overhead**.
- TGAs by default are biased towards ISP addresses.

https://ipv6hitlist.github.io

Conclusion

- Network categories are not evenly distributed in the Hitlist service.
  - Bias towards ISP addresses.
  - Categories show different behavior in port responses and temporal stability.
  - ISP addresses are less stable than CDN addresses.
  - ISP responds only to ICMP, CDN best to TCP/80,443, UDP/443.
    → **Filtering input can avoid scanning overhead**.
- TGAs by default are biased towards ISP addresses.
  - Default input leads to ICMP-biased responsiveness.

https://ipv6hitlist.github.io

Conclusion

- Network categories are not evenly distributed in the Hitlist service.
  - Bias towards ISP addresses.
  - Categories show different behavior in port responses and temporal stability.
  - ISP addresses are less stable than CDN addresses.
  - ISP responds only to ICMP, CDN best to TCP/80,443, UDP/443.
    → **Filtering input can avoid scanning overhead**.
- TGAs by default are biased towards ISP addresses.
  - Default input leads to ICMP-biased responsiveness.
  - Response rates vary depending on input.

https://ipv6hitlist.github.io

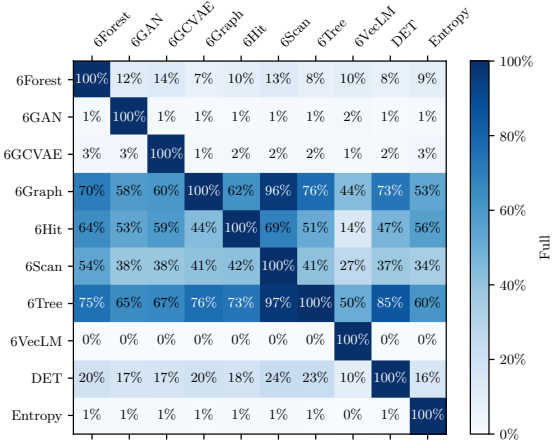
Conclusion

- Network categories are not evenly distributed in the Hitlist service.
  - Bias towards ISP addresses.
  - Categories show different behavior in port responses and temporal stability.
  - ISP addresses are less stable than CDN addresses.
  - ISP responds only to ICMP, CDN best to TCP/80,443, UDP/443.
    → **Filtering input can avoid scanning overhead**.
- TGAs by default are biased towards ISP addresses.
  - Default input leads to ICMP-biased responsiveness.
  - Response rates vary depending on input.
    → **Filtering input can avoid biased candidate addresses**.

https://ipv6hitlist.github.io

Backup

Cross-algorihm responsiveness



References

[1] O. Gasser, Q. Scheitle, P. Foremski, Q. Lone, M. Korczyński, S. D. Strowes, L. Hendriks, and G. Carle, "Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists," in Proc. ACM Int. Measurement Conference (IMC), Boston, MA, USA, 2018. DOI: 10.1145/3278532.3278564.

[2] J. Zirngibl, L. Steger, P. Sattler, O. Gasser, and G. Carle, "Rusty clusters? dusting an IPv6 research foundation," in Proc. ACM Int. Measurement Conference (IMC), Nice, France, 2022. DOI: 10.1145/3517745.3561440.

[3] P. Foremski, D. Plonka, and A. Berger, "Entropy/IP: Uncovering Structure in IPv6 Addresses," in Proc. ACM Int. Measurement Conference (IMC), Santa Monica, California, USA, 2016. DOI: 10.1145/2987443.2987445.

[4] Z. Liu, Y. Xiong, X. Liu, W. Xie, and P. Zhu, "6Tree: Efficient dynamic discovery of active addresses in the IPv6 address space," Computer Networks, vol. 155, May 2019. DOI: 10.1016/j.comnet.2019.03.010.

[5] G. Song, J. Yang, Z. Wang, L. He, J. Lin, L. Pan, C. Duan, and X. Quan, "DET: Enabling Efficient Probing of IPv6 Active Addresses," IEEE/ACM Transactions on Networking, vol. 30, no. 4, Aug. 2022. DOI: 10.1109/tnet.2022.3145040.

[6] T. Cui, G. Gou, and G. Xiong, "6GCVAE: Gated Convolutional Variational Autoencoder for IPv6 Target Generation," in Advances in Knowledge Discovery and Data Mining, Springer International Publishing, 2020. DOI: 10.1007/978-3-030-47426-3_47.

[7] T. Cui, G. Xiong, G. Gou, J. Shi, and W. Xia, "6VecLM: Language Modeling in Vector Space for IPv6 Target Generation," in Machine Learning and Knowledge Discovery in Databases: Applied Data Science Track, Springer International Publishing, 2021. DOI: 10.1007/978-3-030-67667-4_12.

References

[8] T. Cui, G. Gou, G. Xiong, C. Liu, P. Fu, and Z. Li, "6GAN: IPv6 Multi-Pattern Target Generation via Generative Adversarial Nets with Reinforcement Learning," in Proc. IEEE Int. Conference on Computer Communications (INFOCOM), IEEE, May 2021. DOI: 10.1109/infocom42981.2021.9488912.

[9] B. Hou, Z. Cai, K. Wu, J. Su, and Y. Xiong, "6Hit: A Reinforcement Learning-based Approach to Target Generation for Internet-wide IPv6 Scanning," in Proc. IEEE Int. Conference on Computer Communications (INFOCOM), IEEE, May 2021. DOI: 10.1109/infocom42981.2021.9488794.

[10] T. Yang, B. Hou, Z. Cai, K. Wu, T. Zhou, and C. Wang, "6Graph: A graph-theoretic approach to address pattern mining for Internet-wide IPv6 scanning," Computer Networks, vol. 203, Feb. 2022. DOI: 10.1016/j.comnet.2021.108666.

[11] T. Yang, Z. Cai, B. Hou, and T. Zhou, "6Forest: An Ensemble Learning-based Approach to Target Generation for Internet-wide IPv6 Scanning," in Proc. IEEE Int. Conference on Computer Communications (INFOCOM), IEEE, May 2022. DOI: 10.1109/infocom48880.2022.9796925.

[12] B. Hou, Z. Cai, K. Wu, T. Yang, and T. Zhou, "6Scan: A High-Efficiency Dynamic Internet-Wide IPv6 Scanner With Regional Encoding," IEEE/ACM Transactions on Networking, 2023. DOI: 10.1109/tnet.2023.3233953.