Improving the Accuracy of Blocklists by Aggregation and Address Reuse Detection

Sivaram Ramanathan

University of Southern California

In collaboration with: Minlan Yu, Jelena Mirkovic, Anushah Hossain and Sadia Afroz

IP Blocklists

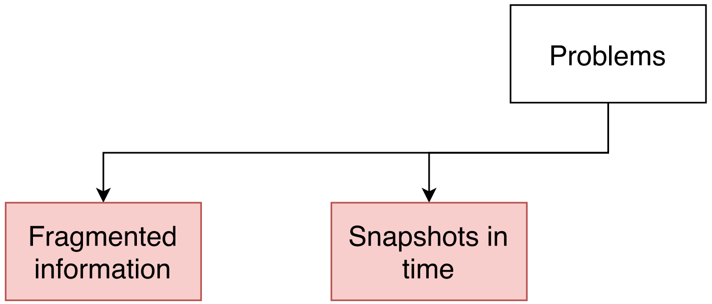
- IP Blocklists contain a list of known malicious IP addresses.
- IP Blocklists are commonly used to aid more sophisticated defenses such as spam filters, IDS, etc.
- IP blocklists can be used as an emergency response under a novel or large volumetric attack.
  - Easy to implement as only IP addresses are checked and can be done at line rate.

1. 198.38.89.61	2. 175.230.213.33	3. 182.74.165.174	4. 178.137.90.85
5. 111.40.73.83	6. 61.132.233.195	7. 193.150.72.50	8. 221.4.205.30
9. 60.172.69.66	10. 61.163.36.24	11. 60.166.48.158	12. 117.214.17.72
13. 180.121.141.117	14. 114.232.216.5	15. 183.159.83.71	16. 121.239.86.33
17. 92.73.213.217	18. 162.248.74.123	19. 183.159.95.87	20. 14.207.215.126
21. 222.191.179.90	22. 217.110.92.194	23. 156.216.145.235	24. 81.17.22.206
25. 41.251.33.175	26. 114.223.61.210	27. 114.232.193.38	28. 114.231.141.136
29. 170.51.62.241	30. 49.67.83.155	31. 180.121.141.119	32. 39.40.30.104
33. 209.54.53.185	34. 167.114.84.153	35. 223.240.208.236	36. 183.150.34.181
37. 95.37.125.239	38. 171.14.238.42	39. 1.55.199.83	40. 222.191.177.40
41. 45.234.101.139	42. 117.85.56.142	43. 123.54.107.199	44. 45.119.81.235
45. 186.47.173.213	46. 49.67.67.141	47. 95.211.149.134	48. 113.128.132.9
49. 49.67.67.140	50. 119.180.198.174	51. 103.69.46.81	52. 128.199.35.34
53. 159.255.167.131	54. 181.215.89.206	55. 192.210.201.168	56. 128.199.44.20
57. 218.72.108.217	58. 113.120.60.120	59. 111.125.140.155	60. 60.50.145.121

Problems with IP Blocklists

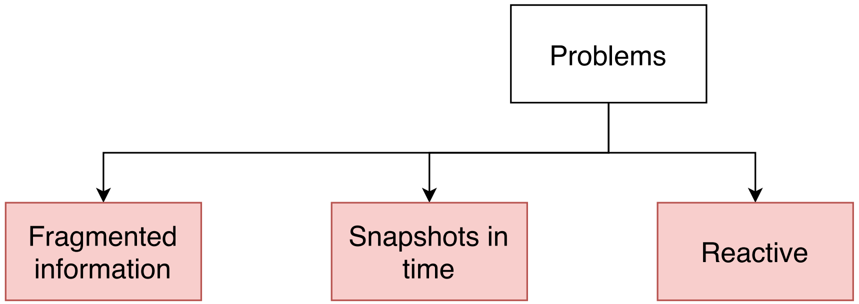
- Focus only on specific attack types with limited vantage points.

Problems with IP Blocklists



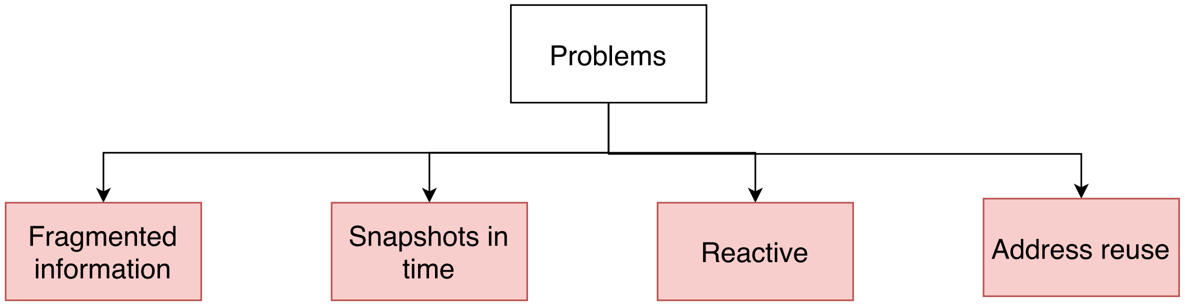
- Focus only on specific attack types with limited vantage points.
- Historical blocklist data can capture reoffending malicious addresses.

Problems with IP Blocklists



- Focus only on specific attack types with limited vantage points.
- Historical blocklist data can capture reoffending malicious addresses.
- Addresses are added only after a malicious event is observed.

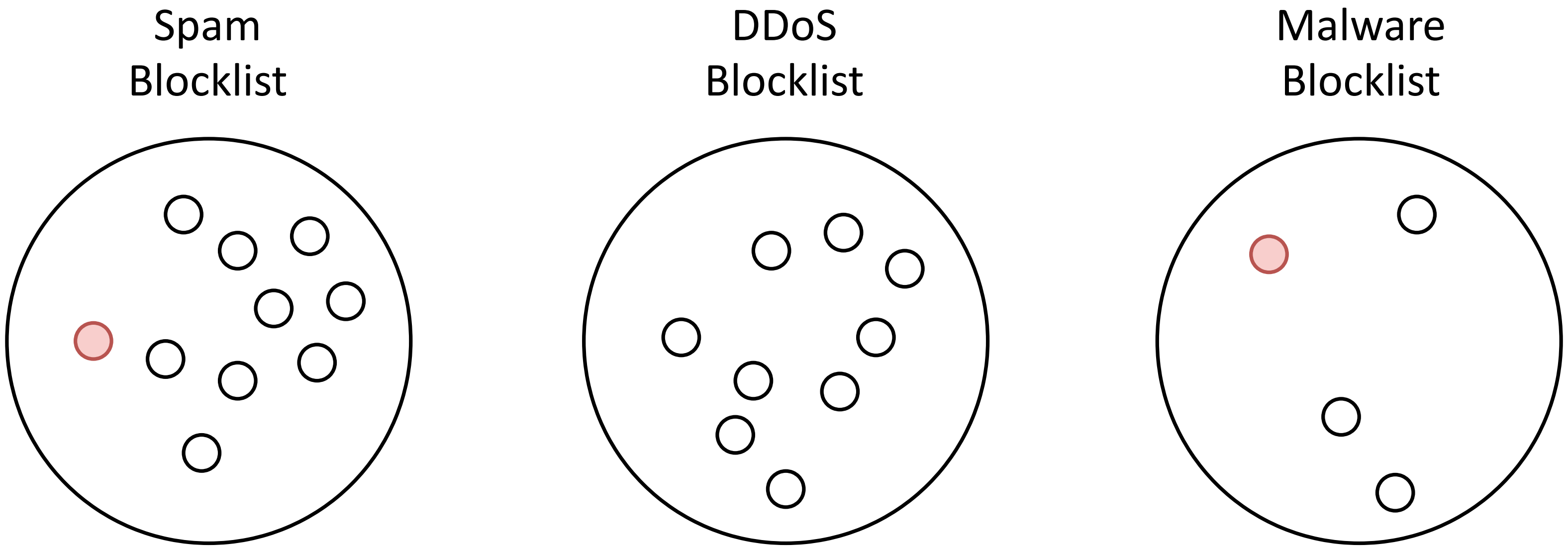
Problems with IP Blocklists



- Focus only on specific attack types with limited vantage points.
- Historical blocklist data can capture reoffending malicious addresses.
- Addresses are added only after a malicious event is observed.
- Blocking reused addresses can lead to unjust blocking of many more users.

P1: Fragmented Information

- offenders in one given attack

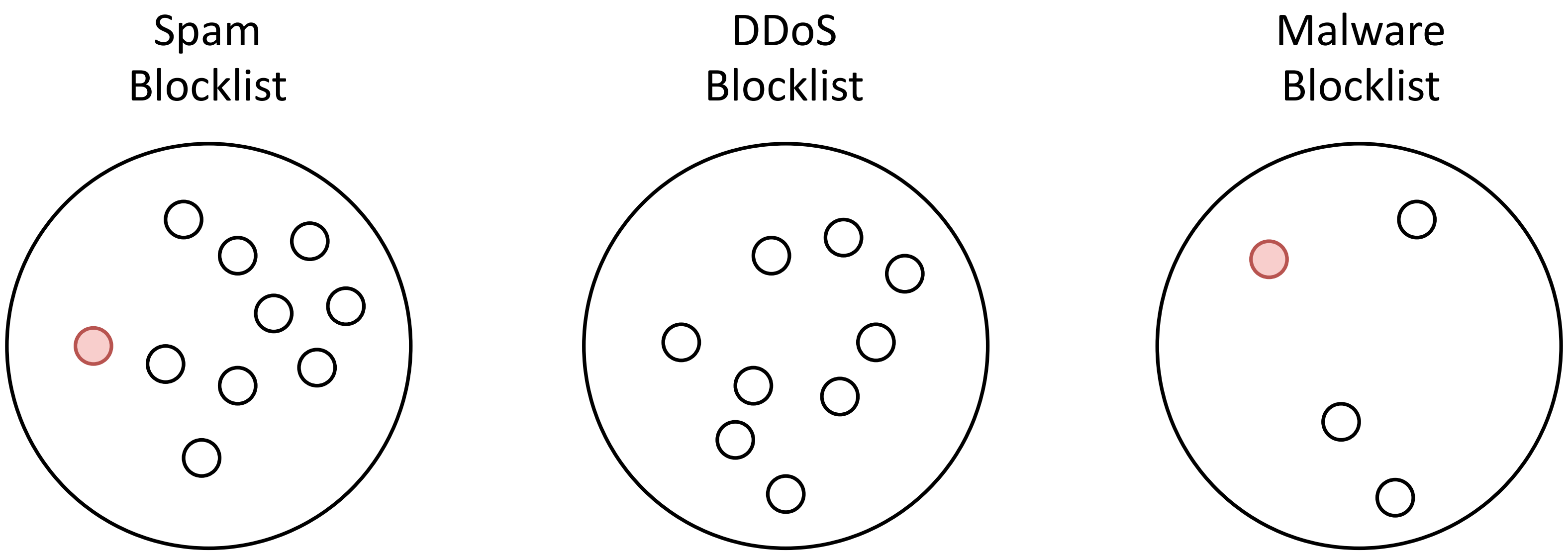


Blocklists miss many attacks[1,2] and may monitor only specific a type of attack.

[1] Kührer, Marc, Christian Rossow, and Thorsten Holz. "Paint it black: Evaluating the effectiveness of malware blocklists." International Workshop on Recent Advances in Intrusion Detection. Springer, Cham, 2014.
[2] Pitsillidis, Andreas, et al. "Taster's choice: a comparative analysis of spam feeds." *Proceedings of the 2012 Internet Measurement Conference*. ACM, 2012.

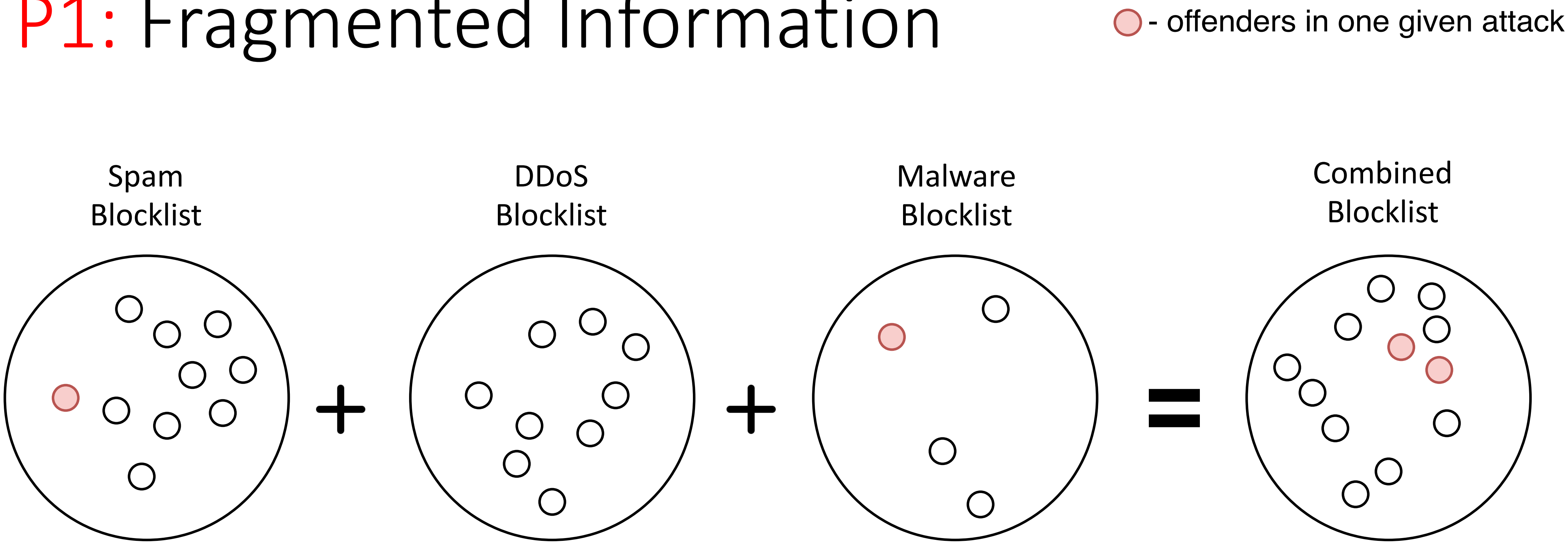
P1: Fragmented Information

- offenders in one given attack



Compromised machines are constantly re-used for initiating different types of attacks over time.

P1: Fragmented Information

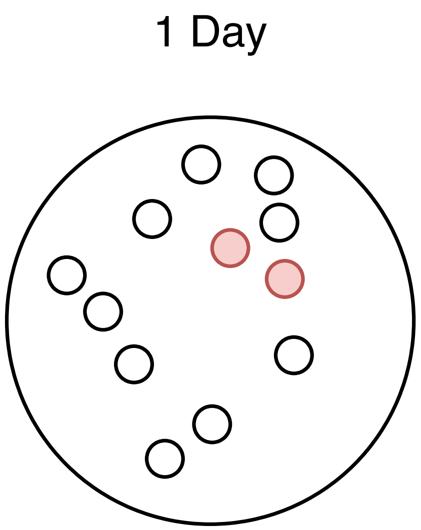


Compromised machines are constantly re-used for initiating different types of attacks over time.

A Possible solution: Combining different types of blocklists can improve attack coverage.

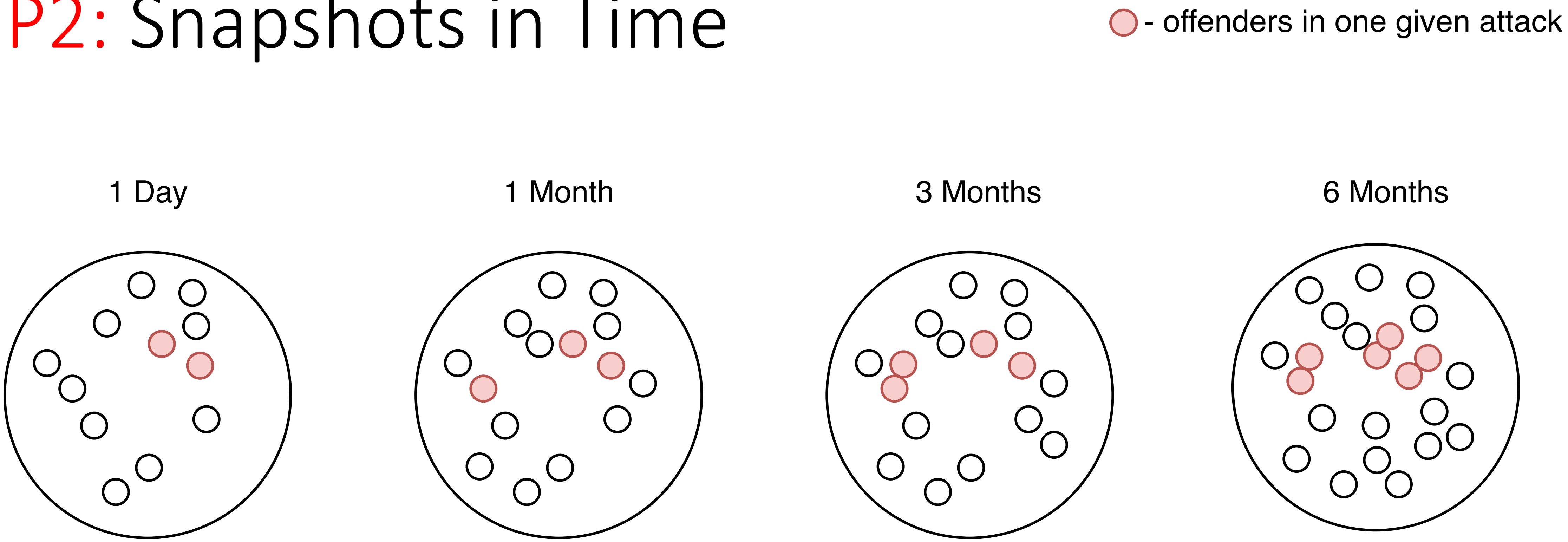
P2: Snapshots in Time

- offenders in one given attack



Historical blocklist data (union of all offenders over time) can further be useful to improve offender detection.

P2: Snapshots in Time

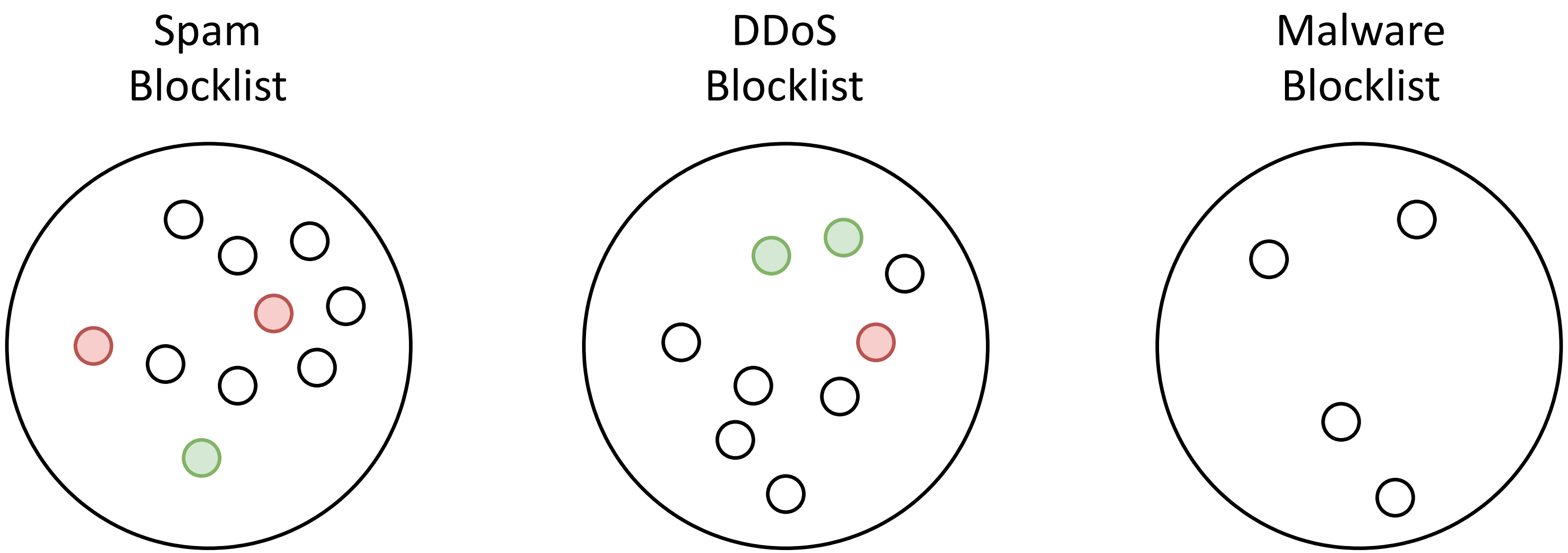


Historical blocklist data (union of all offenders over time) can further be useful to improve offender detection.

P2: Careful Aggregation

- offenders in one given attack

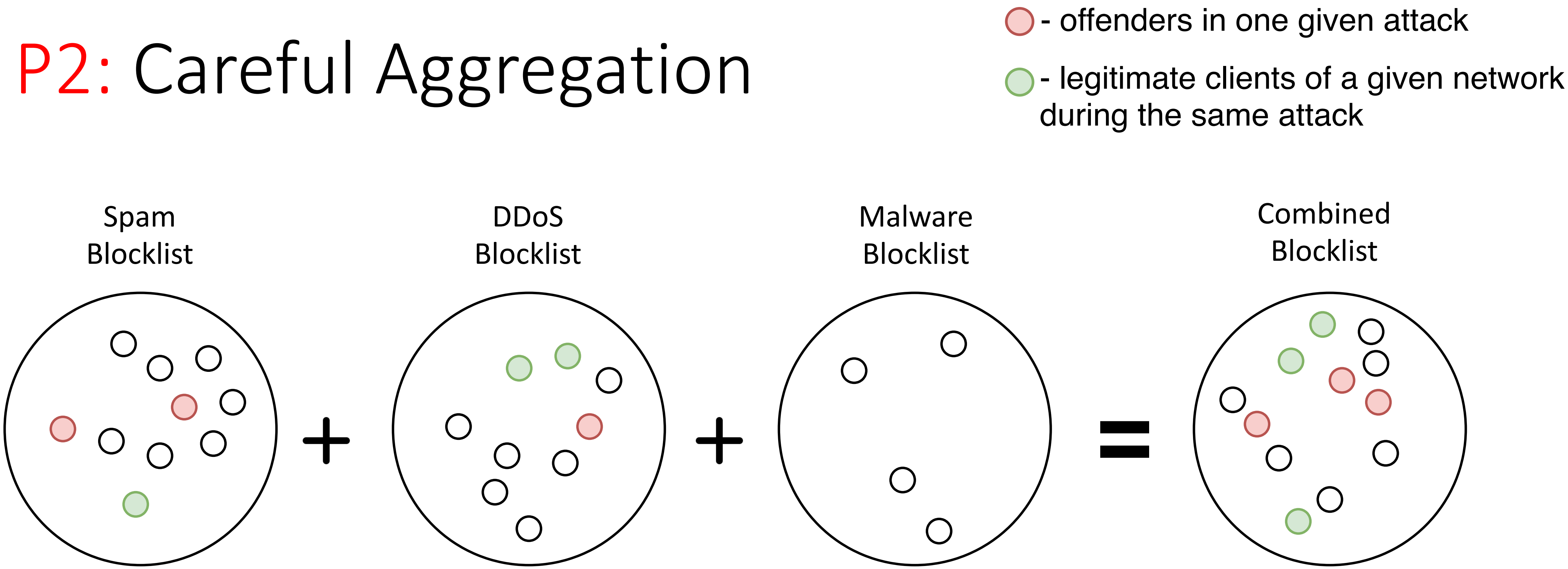
- legitimate clients of a given network during the same attack



Blocklists accuracy varies spatially

- Blocklists are maintained by individuals or organizations that use proprietary algorithms to include or exclude an address.
- Blocklists could list some legitimate addresses.

P2: Careful Aggregation

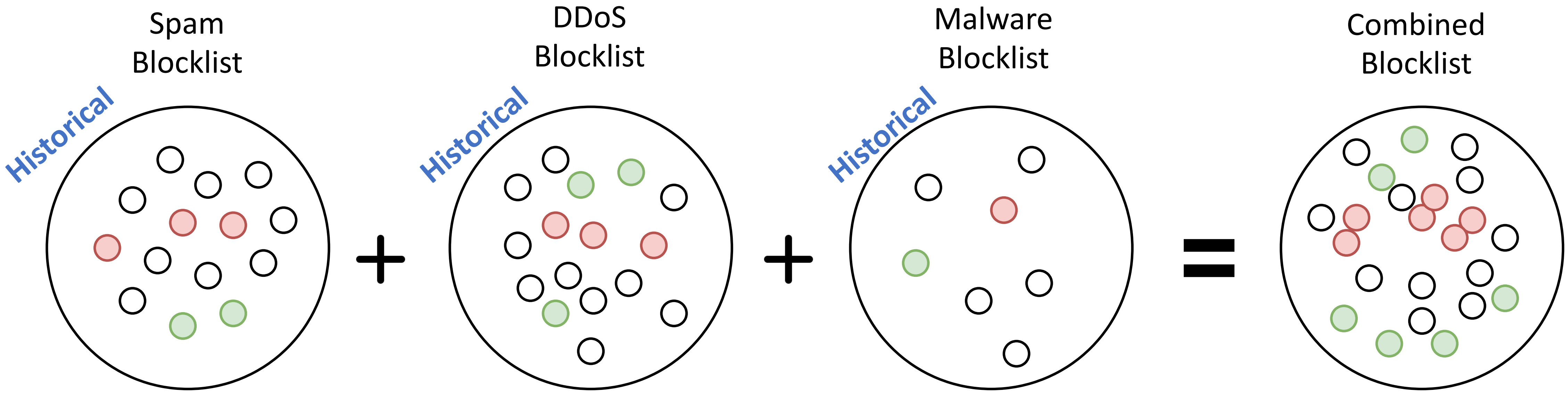


Combining blocklists can potentially amplify the number of misclassifications.

P2: Careful Aggregation

- offenders in one given attack

- legitimate clients of a given network during the same attack



Combining blocklists can further potentially amplify the number of misclassifications.

P2: Careful Aggregation

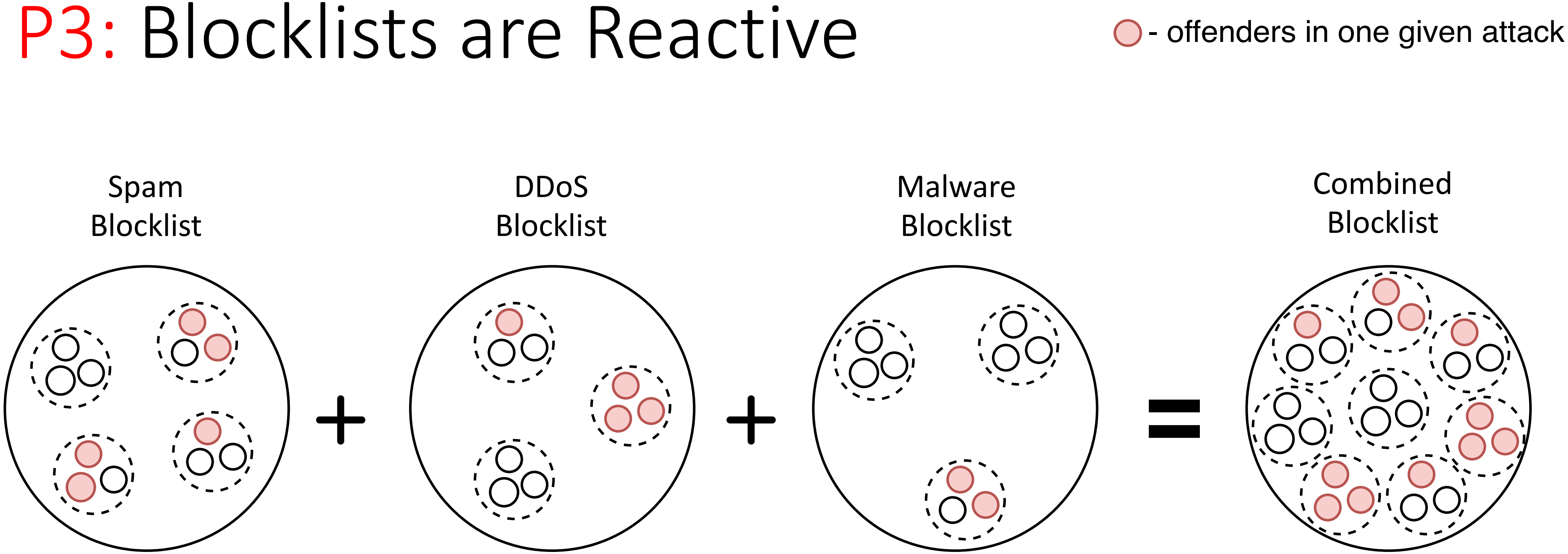
- offenders in one given attack

- legitimate clients of a given network during the same attack

Goal: Aggregate historical blocklists and reduce misclassifications.

Combining blocklists can further potentially amplify the number of misclassifications.

P3: Blocklists are Reactive



Addresses are usually listed after an attack takes place, cannot be used for prevention.

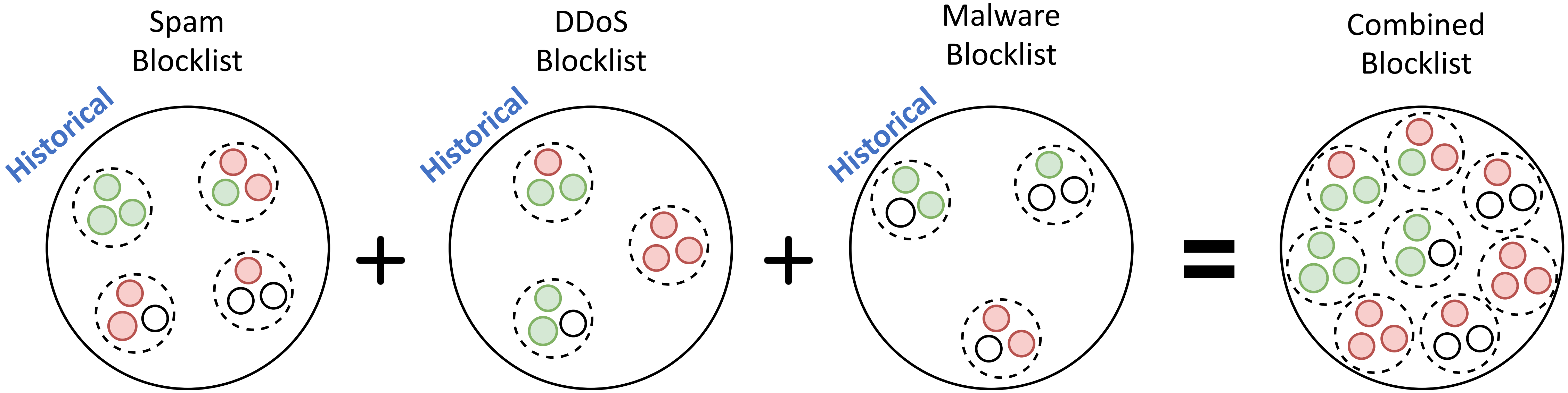
Possible solution: we could list groups of addresses in the same subnet (IP prefixes), hoping to capture future attackers - expansion[1].

[1] Zhang, Jing, et al. "On the Mismanagement and Maliciousness of Networks." NDSS. 2014.

P3: Careful Expansion

- offenders in one given attack

- legitimate clients of a given network during the same attack



Expansion can further amplify misclassifications!

P3: Careful Expansion

- offenders in one given attack

- legitimate clients of a given network during the same attack

Spam Blocklist

DDoS Blocklist

Malware Blocklist

Combined Blocklist

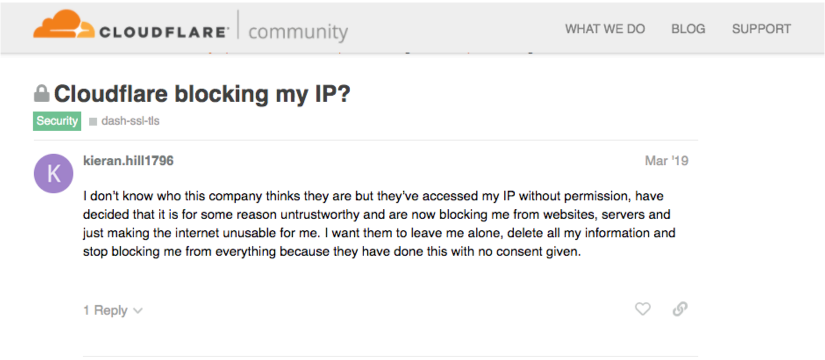
Goal: Expand some addresses into prefixes that do not cause more misclassifications.

Expansion can further amplify misclassifications!

P4: Blocklisting Reused Addresses: NAT

https://community.cloudflare.com/t/cloudflare-blocking-my-ip/65453/57

P4: Blocklisting Reused Addresses: NAT


CLOUDFLARE | community WHAT WE DO BLOG SUPPORT

Cloudflare blocking my IP?

Security dash-ssl-tls

kieran.hill1796 Mar '19

I don't know who this company thinks they are but they've accessed my IP without permission, have decided that it is for some reason untrustworthy and are now blocking me from websites, servers and just making the internet unusable for me. I want them to leave me alone, delete all my information and stop blocking me from everything because they have done this with no consent given.

1 Reply

https://community.cloudflare.com/t/cloudflare-blocking-my-ip/65453/57

P4: Blocklisting Reused Addresses: NAT

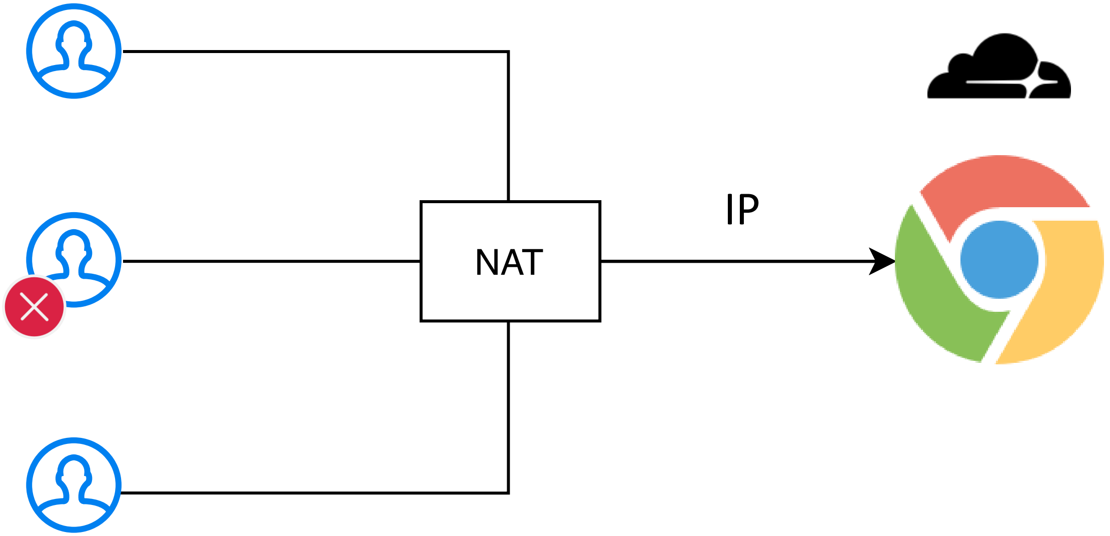
https://community.cloudflare.com/t/cloudflare-blocking-my-ip/65453/57

P4: Blocklisting Reused Addresses: NAT

Cloudflare uses
Dshield blocklist.

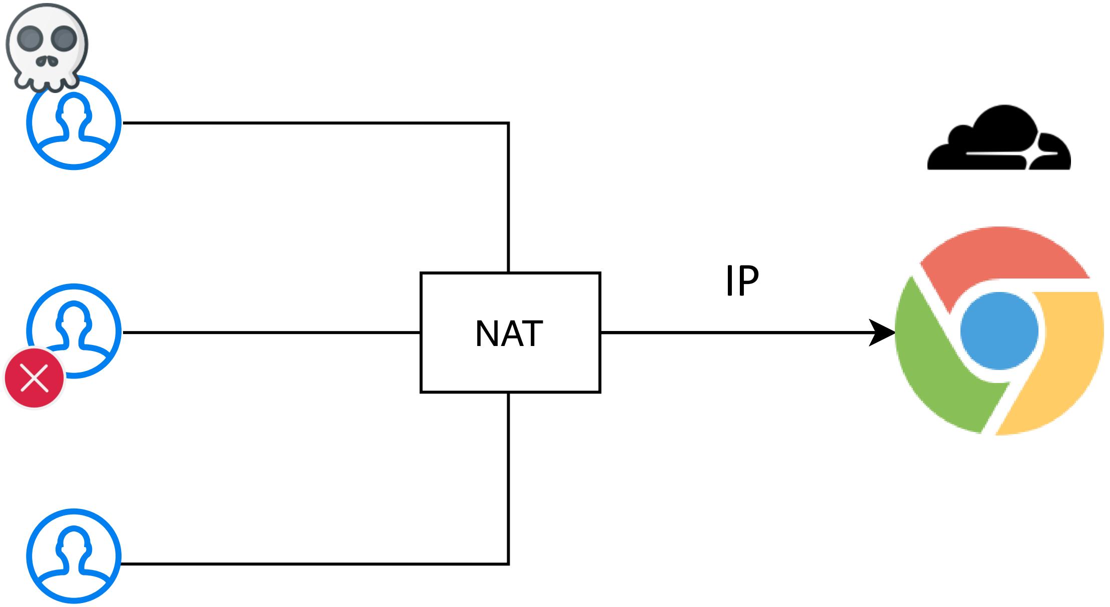
https://community.cloudflare.com/t/cloudflare-blocking-my-ip/65453/57

P4: Blocklisting Reused Addresses: NAT



https://community.cloudflare.com/t/cloudflare-blocking-my-ip/65453/57

P4: Blocklisting Reused Addresses: NAT

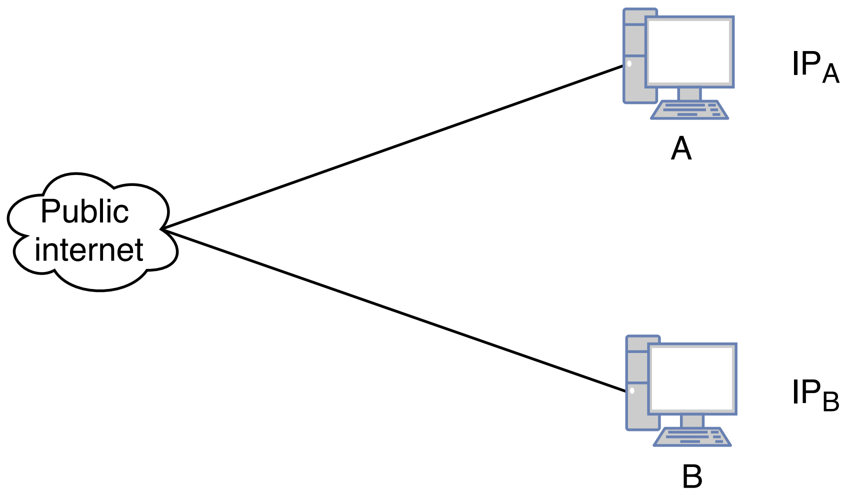


P4: Blocklisting Reused Addresses: NAT

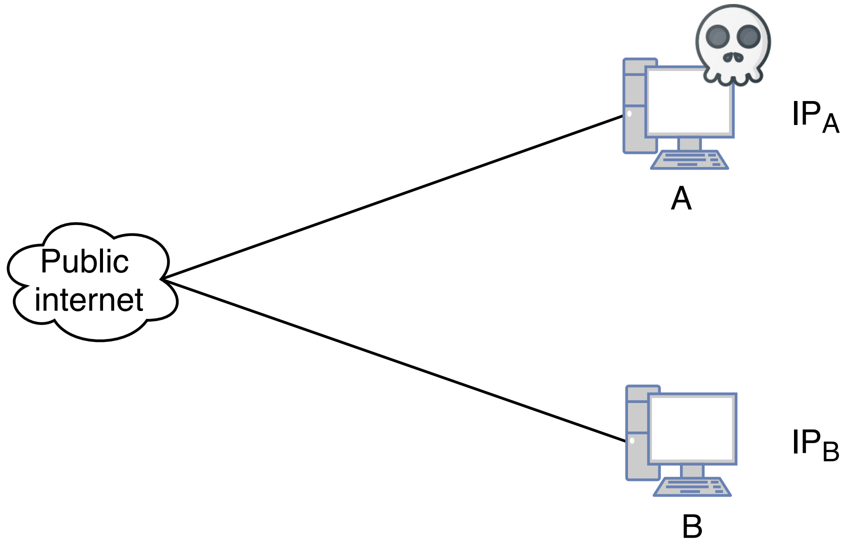
Goal: Accurately identify NATed reused address to prevent unjust blocking.

https://community.cloudflare.com/t/cloudflare-blocking-my-ip/65453/57

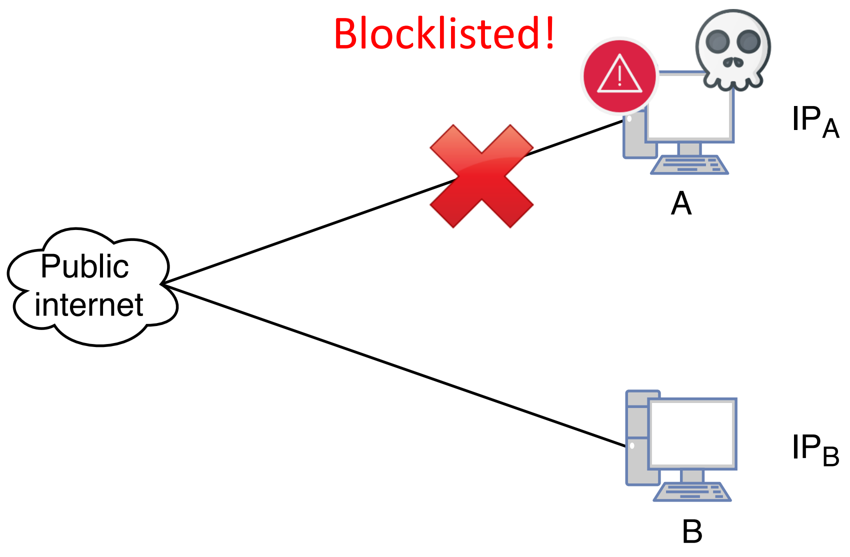
P4: Blocklisting Reused Addresses: Dynamic Addressing



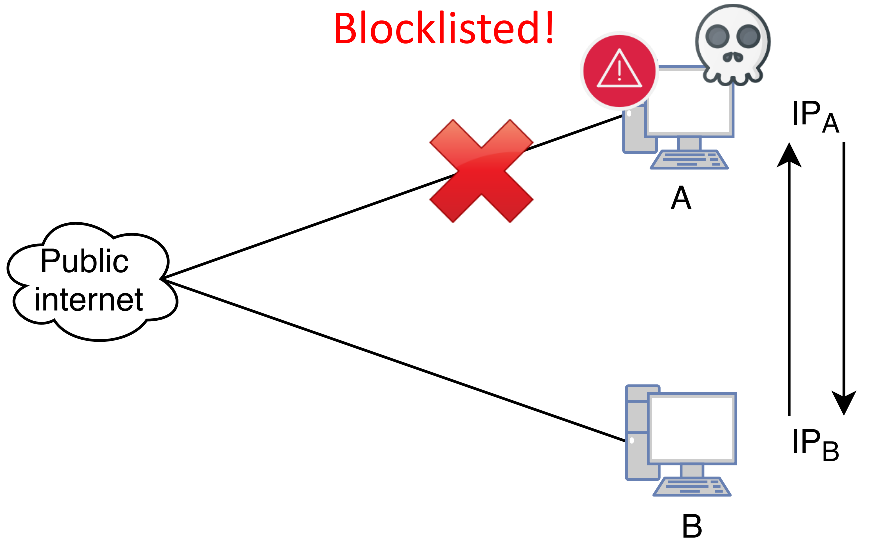
P4: Blocklisting Reused Addresses: Dynamic Addressing



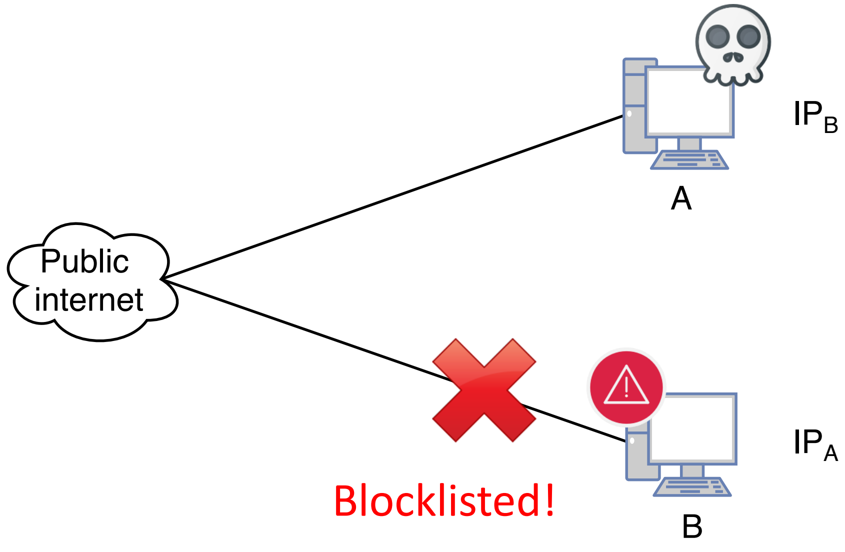
P4: Blocklisting Reused Addresses: Dynamic Addressing



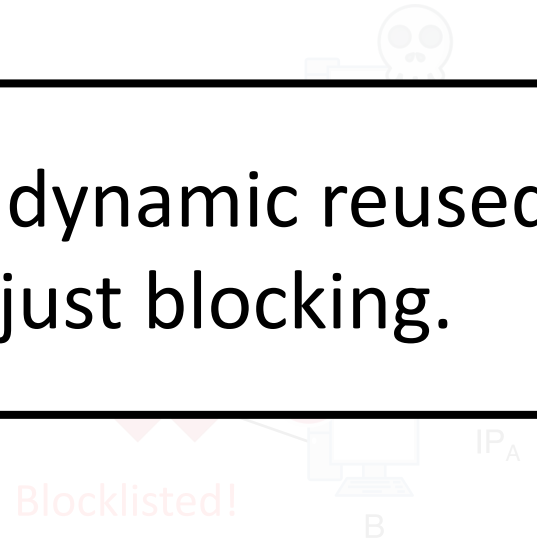
P4: Blocklisting Reused Addresses: Dynamic Addressing



P4: Blocklisting Reused Addresses: Dynamic Addressing

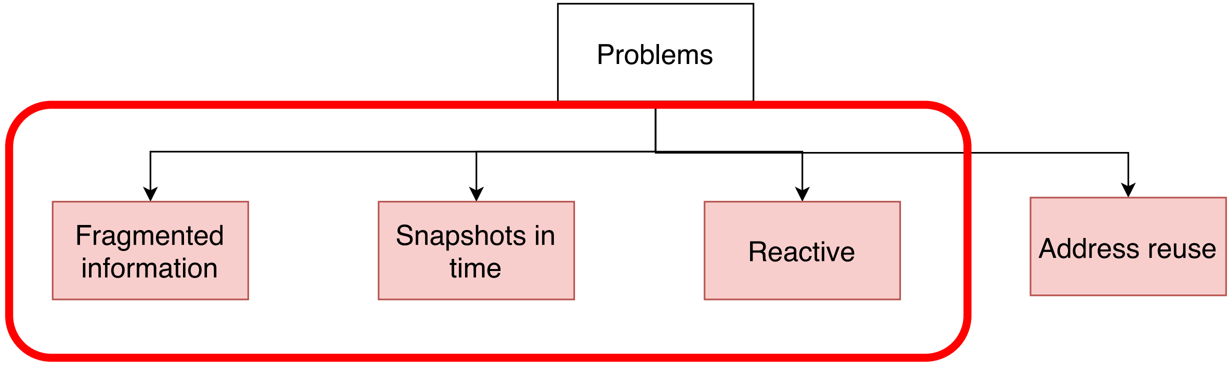


P4: Blocklisting Reused Addresses: Dynamic Addressing



Goal: Accurately identify dynamic reused address to prevent unjust blocking.

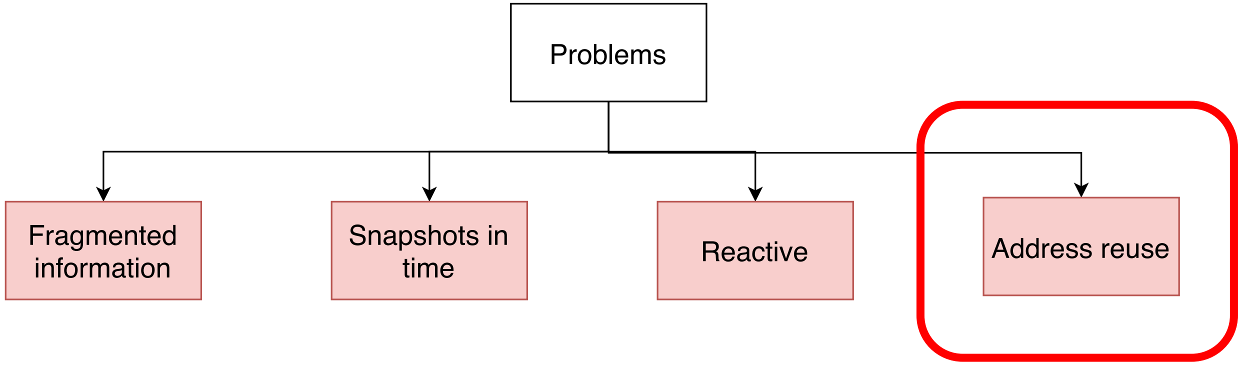
Problems with IP Blocklists



BLAG: Aggregation + Estimate Misclassification + Selective Expansion[1]

[1] BLAG: Improving the Accuracy of Blocklists; Sivaram Ramanathan, Jelena Mikovic and Minlan Yu; NDSS 2020.

Problems with IP Blocklists



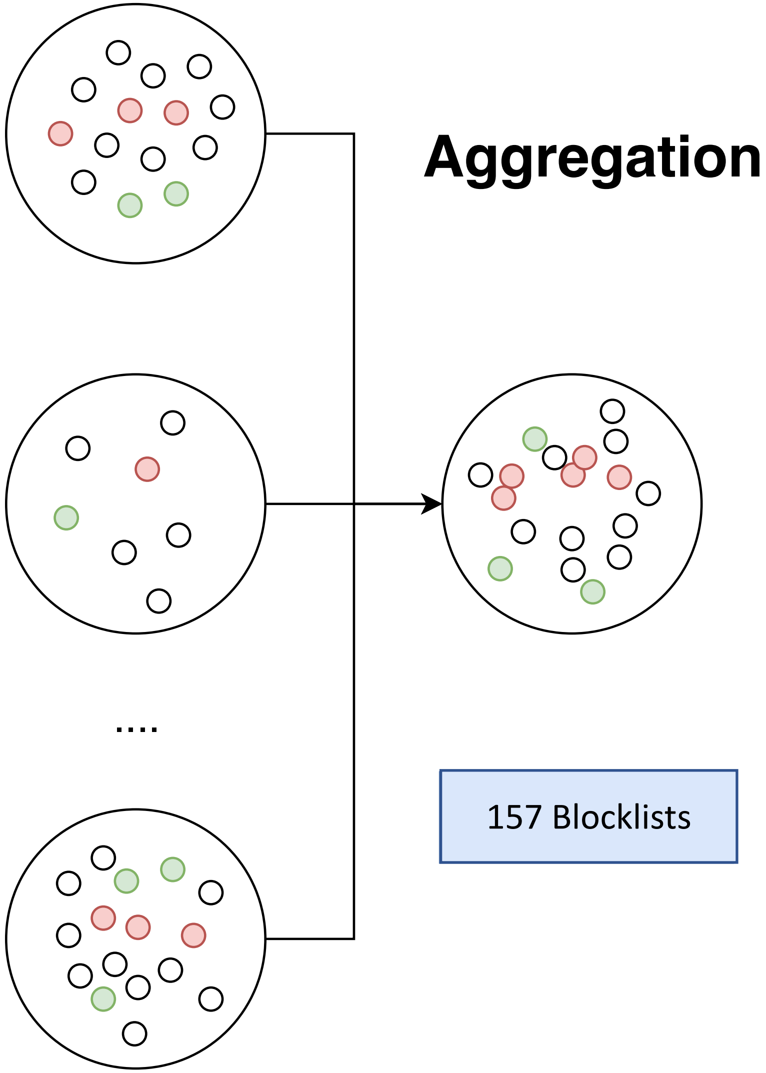
Quantifying the impact of Blocklisting[2]

[1] BLAG: Improving the Accuracy of Blocklists; Sivaram Ramanathan, Jelena Mikovic and Minlan Yu; NDSS 2020.
[2] Quantifying the Impact of Blocklisting in the Age of Address Reuse: Sivaram Ramanathan, Anushah Hossain, Jelena Mirkovic, Minlan Yu and Sadia Afroz; IMC 2020

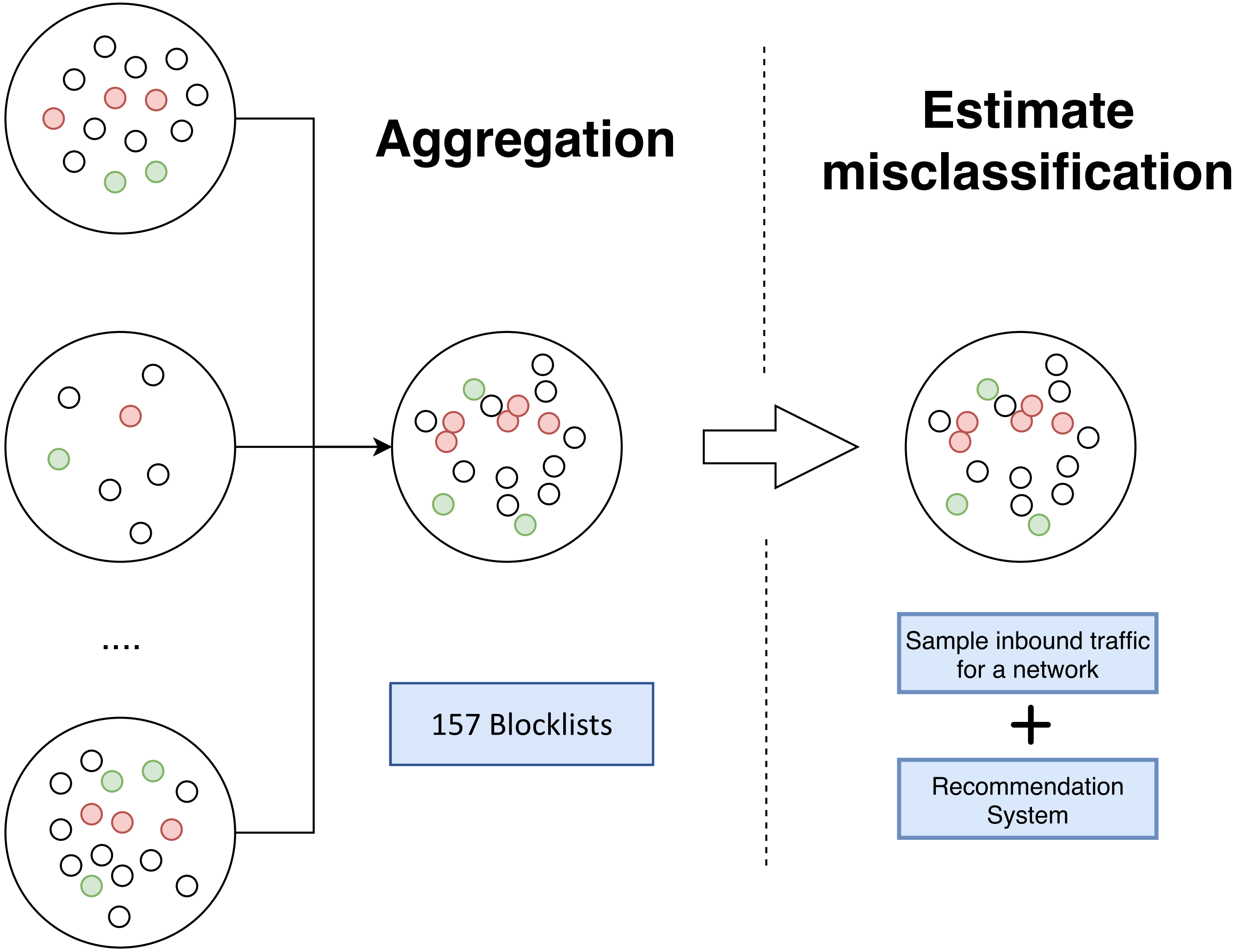
Outline

- Introduction
- Quantifying problems faced by blocklists
- BLAG
  - Datasets
  - Evaluation
- Identifying reused addresses
  - Detecting NATed addresses
  - Detecting dynamic addresses
  - Evaluation
- Summary

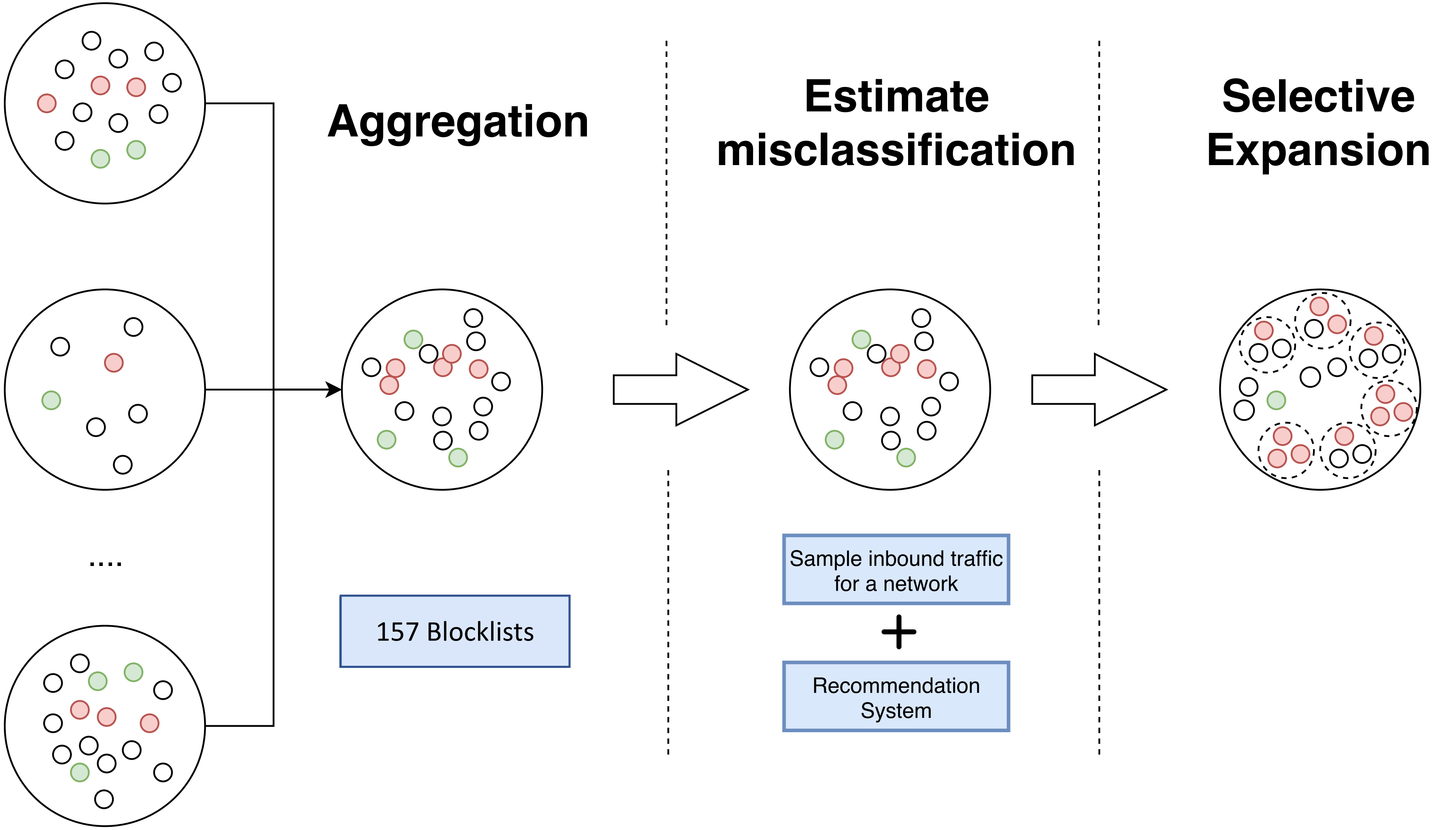
How BLAG Works



How BLAG Works



How BLAG Works



Aggregation of Blocklists

- Historical blocklist data can be useful.
- However, including addresses reported way back in the past can increase the misclassifications.
- PRESTA[1] showed that recently listed addresses have a higher tendency to be malicious than older ones.
- BLAG uses the same metric as that of PRESTA to assign a relevance score, based on when the address was listed in a blocklist
  - Recently listed addresses have a higher score.

[1] West, Andrew G., et al. "Spam mitigation using spatio-temporal reputations from blocklist history." Proceedings of the 26th Annual Computer Security Applications Conference. ACM, 2010.

Aggregation of Blocklists: Relevance Scores

- For address *a* listed in blocklist *b*,

$$r_{a,b} = 2^{\frac{t_{out} - t}{l}}$$

Aggregation of Blocklists: Relevance Scores

- For address *a* listed in blocklist *b*,

$$r_{a,b} = 2^{\frac{t_{out} - t}{l}}$$

Where,

- t is the current time

Aggregation of Blocklists: Relevance Scores

- For address *a* listed in blocklist *b*,

$$r_{a,b} = 2^{\frac{t_{out} - t}{l}}$$

Where,

- t is the current time
- t_{out} is the last time when an address *a* was listed in blocklist *b*

Aggregation of Blocklists: Relevance Scores

- For address *a* listed in blocklist *b*,

$$r_{a,b} = 2^{\frac{t_{out}-t}{l}}$$

Where,

- t is the current time
- t_{out} is the last time when an address *a* was listed in blocklist *b*
- l is constant, which ensures that the score decays over time

Aggregation of Blocklists: Relevance Scores

- For address *a* listed in blocklist *b*,

A high relevance score means that an IP has been recently listed and has a higher tendency of being malicious.

- t is the current time
- t_{out} is the last time when address *a* was listed in blocklist *b*
- l is constant, which ensures that the score decays exponentially over time

Estimate Misclassifications–Recommendation System

0.8	0.6			1	
1		0.8		0.8	
0.8		1	0.6	1	
	0.6				1
	0.8				0.8
		0.4	0.8		1

- Commonly found in popular services like Netflix, Amazon, and YouTube to improve user retention and increase revenue.
- Recommend new items to users based on their or similar users' previous ratings of similar items.

Estimate Misclassifications–Recommendation System

0.8	0.6			1	
1		0.8		0.8	
0.8		1	0.6	1	
	0.6				1
	0.8				0.8
		0.4	0.8		1

Estimate Misclassifications–Recommendation System

0.8	0.6			1	
1		0.8		0.8	
0.8		1	0.6	1	
	0.6				1
	0.8				0.8
		0.4	0.8		1

Likes green books.

Estimate Misclassifications–Recommendation System

	0.8	0.6			1	
	1		0.8		0.8	
	0.8		1	0.6	1	
		0.6				1
		0.8				0.8
			0.4	0.8		1

Likes green books.

Dislikes yellow books.

Estimate Misclassifications– Recommendation System

0.8	0.6		?	1	
1		0.8		0.8	
0.8		1	0.6	1	
	0.6				1
	0.8				0.8
		0.4	0.8		1

Estimate Misclassifications–Recommendation System

0.8	0.6			1	
1		0.8		0.8	
0.8		1	0.6	1	
	0.6				1
	0.8				0.8
		0.4	0.8		1

0.8	0.59	0.6	0.7	0.99	1
0.99	0.97	0.8	0.92	0.8	1
0.8	0.85	0.99	0.59	0.99	1
0.7	0.6	0.6	0.66	0.72	0.99
0.66	0.79	0.5	0.6	0.29	0.8
0.77	0.85	0.4	0.79	0.55	0.99

Estimate Misclassifications– Recommendation System

0.8	0.6			1	
1		0.8		0.8	
0.8		1	0.6	1	
	0.6				1
	0.8				0.8
		0.4	0.8		1

0.8	0.59	0.6	0.7	0.99	1
0.99	0.97	0.8	0.92	0.8	1
0.8	0.85	0.99	0.59	0.99	1
0.7	0.6	0.6	0.66	0.72	0.99
0.66	0.79	0.5	0.6	0.29	0.8
0.77	0.85	0.4	0.79	0.55	0.99

Estimate Misclassifications–Recommendation System

0.8	0.6			1	
		0.8			
0.8		1	0.6	1	
	0.6				1
	0.8				0.8
		0.4	0.8		1

0.8	0.59	0.6	**0.7**	0.99	1
0.99	0.97	0.8	0.92	0.8	1
0.8	0.85	0.99	0.59	0.99	1
0.7	0.6	0.6	0.66	0.72	0.99
0.66	0.79	0.5	0.6	0.29	0.8
0.77	0.85	0.4	0.79	0.55	0.99

Estimate Misclassifications– Recommendation System

0.8	0.6			1	
		0.8			
0.8		1	0.6	1	
	0.6				1
	0.8				0.8
		0.4	0.8		1

0.8	0.59	0.6	**0.7**	0.99	1
0.99	0.97	0.8		0.8	1
0.8	0.85	0.99	0.59	0.99	1
0.7	0.6	0.6	0.66	0.72	0.99
0.66	0.79	0.5	0.6	0.29	0.8
0.77	0.85	0.4	0.79	0.55	0.99

Estimate Misclassifications

	Blocklist 1	Blocklist 2	Blocklist 3	..	Blocklist m
169.231.140.10	..	..	0.8	..	..
169.231.140.68	0.3	0.1	..	..	0.1
193.1.64.5	..	0.5	..	..	..
193.1.64.8	0.7	0.5	..	..	0.9
216.59.0.8	..	..	0.04	..	0.1
216.59.16.171	..	0.7	..	..	0.9
243.13.0.23	..	..	..	..	..
243.13.222.203	..	0.7	1	..	0.9

- BLAG arranges IP addresses and blocklists in a matrix, where rows are addresses and columns are blocklists.
- If an address *a* is listed in blocklist *b*, BLAG assigns the relevance score $r_{a,b}$ to the cell.

Estimate Misclassifications

	Blocklist 1	Blocklist 2	Blocklist 3	..	Blocklist m	MB
169.231.140.10	..	..	0.8	..	..	
169.231.140.68	0.3	0.1	..	..	0.1	
193.1.64.5	..	0.5	..	..	..	
193.1.64.8	0.7	0.5	..	..	0.9	
216.59.0.8	..	..	0.04	..	0.1	
216.59.16.171	..	0.7	..	..	0.9	
243.13.0.23	..	..	..	..	..	
243.13.222.203	..	0.7	1	..	0.9	

BLAG uses legitimate traffic traces of a network to introduce a new blocklist called the Misclassification Blocklist (MB), which consists only of misclassifications.

Estimate Misclassifications

	Blocklist 1	Blocklist 2	Blocklist 3	..	Blocklist m	MB
169.231.140.10	..	..	0.8	..	..	
169.231.140.68	0.3	0.1	..	..	0.1	
193.1.64.5	..	0.5	..	..	..	
193.1.64.8	0.7	0.5	..	..	0.9	
216.59.0.8	..	..	0.04	..	0.1	
216.59.16.171	..	0.7	..	..	0.9	1
243.13.0.23	..	..	..	..	..	1
243.13.222.203	..	0.7	1	..	0.9	1

For every known misclassification from the training data, BLAG allocates a score of 1.

Estimate Misclassifications

	Blocklist 1	Blocklist 2	Blocklist 3	..	Blocklist m	MB
169.231.140.10	..	..	0.8	..	..	?
169.231.140.68	0.3	0.1	..	..	0.1	?
193.1.64.5	..	0.5	..	..	..	?
193.1.64.8	0.7	0.5	..	..	0.9	?
216.59.0.8	..	..	0.04	..	0.1	?
216.59.16.171	..	0.7	..	..	0.9	1
243.13.0.23	..	..	..	..	..	1
243.13.222.203	..	0.7	1	..	0.9	1

Goal: Find the relevance scores for remaining addresses in MB.

Estimate Misclassifications

		Blocklist 1	Blocklist 2	Blocklist 3	..	Blocklist m	MB
IP_1	169.231.140.10	..	..	0.8	..	..	?
	169.231.140.68	0.3	0.1	..	..	0.1	?
	193.1.64.5	..	0.5	..	..	..	?
	193.1.64.8	0.7	0.5	..	..	0.9	?
	216.59.0.8	..	..	0.04	..	0.1	?
	216.59.16.171	..	0.7	..	..	0.9	1
	243.13.0.23	..	..	..	..	..	1
IP_2	**243.13.222.203**	..	0.7	1	..	0.9	1

Recommendation system

	Blocklist 1	Blocklist 2	Blocklist 3	..	Blocklist m-1	MB
169.231.140.10	..	..	0.78	..	..	0.75
169.231.140.68	0.28	0.11	..	..	0.15	0.22
193.1.64.5	..	0.46	..	..	..	0.4
193.1.64.8	0.72	0.23	..	..	0.87	0.6
216.59.0.8	..	..	0.32	..	0.25	0.12
216.59.16.171	..	0.58	..	..	0.95	0.91
243.13.0.23	..	..	..	..	..	0.92
243.13.222.203	..	0.79	0.87	..	0.81	0.99

Goal: Find the relevance scores for remaining addresses in MB.

Estimate Misclassifications

		Blocklist 1	Blocklist 2	Blocklist 3	..	Blocklist m	MB
IP_1	*169.231.140.10*	..	..	0.8	..	..	?
	169.231.140.68	0.3	0.1	..	..	0.1	?
	193.1.64.5	..	0.5	..	..	..	?
	193.1.64.8	0.7	0.5	..	..	0.9	?
	216.59.0.8	..	..	0.04	..	0.1	?
	216.59.16.171	..	0.7	..	..	0.9	1
	243.13.0.23	..	..	..	..	..	1
IP_2	***243.13.222.203***	..	0.7	1	..	0.9	1

Recommendation system

	Blocklist 1	Blocklist 2	Blocklist 3	..	Blocklist m-1	MB
169.231.140.10	..	..	0.78	..	..	0.75
169.231.140.68	0.28	0.11	..	..	0.15	0.22
193.1.64.5	..	0.46	..	..	..	0.4
193.1.64.8	0.72	0.23	..	..	0.87	0.6
216.59.0.8	..	..	0.32	..	0.25	0.12
216.59.16.171	..	0.58	..	..	0.95	0.91
243.13.0.23	..	..	..	..	..	0.92
243.13.222.203	..	0.79	0.87	..	0.81	0.99

Goal: Find the relevance scores for remaining addresses in MB.

Estimate Misclassifications

		Blocklist 1	Blocklist 2	Blocklist 3	..	Blocklist m	MB
IP_1	169.231.140.10	..	..	0.8	..	..	?
	169.231.140.68	0.3	0.1	..	..	0.1	?
	193.1.64.5	..	0.5	..	..	..	?
	193.1.64.8	0.7	0.5	..	..	0.9	?
	216.59.0.8	..	..	0.04	..	0.1	?
	216.59.16.171	..	0.7	..	..	0.9	1
	243.13.0.23	..	..	..	..	..	1
IP_2	243.13.222.203	..	0.7	1	..	0.9	1

Recommendation system

		Blocklist 1	Blocklist 2	Blocklist 3	..	Blocklist m-1	MB
IP_1	169.231.140.10	..	..	0.78	..	..	0.75
	169.231.140.68	0.28	0.11	..	..	0.15	0.22
	193.1.64.5	..	0.46	..	..	..	0.4
	193.1.64.8	0.72	0.23	..	..	0.87	0.6
	216.59.0.8	..	..	0.32	..	0.25	0.12
	216.59.16.171	..	0.58	..	..	0.95	0.91
	243.13.0.23	..	..	..	..	..	0.92
IP_2	243.13.222.203	..	0.79	0.87	..	0.81	0.99

Likely to be a misclassification!

Goal: Find the relevance scores for remaining addresses in MB.

Estimate Misclassifications

	Blocklist 1	Blocklist 2	Blocklist 3	..	Blocklist m	MB
169.231.140.10	..	..	0.8	..	..	?
169.231.140.68	0.3	0.1	..	..	0.1	?
193.1.64.5	..	0.5	..	..	..	?
193.1.64.8	0.7	0.5	..	..	0.9	?
216.59.0.8	..	..	0.04	..	0.1	?
216.59.16.171	..	0.7	..	..	0.9	1
243.13.0.23	..	..	..	..	..	1
243.13.222.203	..	0.7	1	..	0.9	1

Recommendation system

	Blocklist 1	Blocklist 2	Blocklist 3	..	Blocklist m-1	MB
169.231.140.10	..	..	0.78	..	..	0.75
169.231.140.68	0.28	0.11	..	..	0.15	0.22
193.1.64.5	..	0.46	..	..	..	0.4
193.1.64.8	0.72	0.23	..	..	0.87	0.6
216.59.0.8	..	..	0.32	..	0.25	0.12
216.59.16.171	..	0.58	..	..	0.95	0.91
243.13.0.23	..	..	..	..	..	0.92
243.13.222.203	..	0.79	0.87	..	0.81	0.99

Prune

Master blocklist candidates
169.231.140.68
193.1.64.5
193.1.64.8
216.59.0.8

Using a defined threshold customized for every network (0.7 in this case), BLAG prune out addresses that are potentially misclassified.

Why Recommendation System?

- Given the incomplete view of the address space, there are many addresses that cannot be determined to be a misclassification (or not).
- Several latent factors influence an address to be a misclassification.
  - Proprietary algorithms historical data or overall reputation of the blocklist
- The recommendation system helps us identify other addresses:
  - Which “behave” similar to our known misclassifications.
  - They are listed on same or similar blocklists as our known misclassifications, with similar scores.

Selective Expansion

	Blocklist 1	Blocklist 2	Blocklist 3	..	Blocklist m	MB
169.231.140.10	..	..	0.8	..	..	?
169.231.140.68	0.3	0.1	..	..	0.1	?
193.1.64.5	..	0.5	..	..	..	?
193.1.64.8	0.7	0.5	..	..	0.9	?
216.59.0.8	..	..	0.04	..	0.1	?
216.59.16.171	..	0.7	..	..	0.9	1
243.13.0.23	..	..	..	..	..	1
243.13.222.203	..	0.7	1	..	0.9	1

Recommendation system

	Blocklist 1	Blocklist 2	Blocklist 3	..	Blocklist m-1	MB
169.231.140.10	..	..	0.78	..	..	0.75
169.231.140.68	0.28	0.11	..	..	0.15	0.22
193.1.64.5	..	0.46	..	..	..	0.4
193.1.64.8	0.72	0.23	..	..	0.87	0.6
216.59.0.8	..	..	0.32	..	0.25	0.12
216.59.16.171	..	0.58	..	..	0.95	0.91
243.13.0.23	..	..	..	..	..	0.92
243.13.222.203	..	0.79	0.87	..	0.81	0.99

Prune

Master blocklist candidates	Check 1
169.231.140.68	OK
193.1.64.5	OK
193.1.64.8	OK
216.59.0.8	OK

Check 1: If a prefix has any known misclassification, it is excluded from expansion.

Selective Expansion

	Blocklist 1	Blocklist 2	Blocklist 3	..	Blocklist m	MB
169.231.140.10	..	..	0.8	..	..	?
169.231.140.68	0.3	0.1	..	..	0.1	?
193.1.64.5	..	0.5	..	..	..	?
193.1.64.8	0.7	0.5	..	..	0.9	?
216.59.0.8	..	..	0.04	..	0.1	?
216.59.16.171	..	0.7	..	..	0.9	1
243.13.0.23	..	..	..	..	..	1
243.13.222.203	..	0.7	1	..	0.9	1

Recommendation system

	Blocklist 1	Blocklist 2	Blocklist 3	..	Blocklist m-1	MB
169.231.140.10	..	..	0.78	..	..	0.75
169.231.140.68	0.28	0.11	..	..	0.15	0.22
193.1.64.5	..	0.46	..	..	..	0.4
193.1.64.8	0.72	0.23	..	..	0.87	0.6
216.59.0.8	..	..	0.32	..	0.25	0.12
216.59.16.171	..	0.58	..	..	0.95	0.91
243.13.0.23	..	..	..	..	..	0.92
243.13.222.203	..	0.79	0.87	..	0.81	0.99

Prune

Master blocklist candidates	Check 1	Check 2
169.231.140.68	OK	!
193.1.64.5	OK	OK
193.1.64.8	OK	OK
216.59.0.8	OK	OK

Check 2: If a prefix has any likely misclassification, it is excluded from expansion.

Selective Expansion

	Blocklist 1	Blocklist 2	Blocklist 3	..	Blocklist m	MB
169.231.140.10	..	..	0.8	..	..	?
169.231.140.68	0.3	0.1	..	..	0.1	?
193.1.64.5	..	0.5	..	..	..	?
193.1.64.8	0.7	0.5	..	..	0.9	?
216.59.0.8	..	..	0.04	..	0.1	?
216.59.16.171	..	0.7	..	..	0.9	1
243.13.0.23	..	..	..	..	..	1
243.13.222.203	..	0.7	1	..	0.9	1

Recommendation system

IP_1

	Blocklist 1	Blocklist 2	Blocklist 3	..	Blocklist m-1	MB
169.231.140.10	..	..	0.78	..	..	0.75
169.231.140.68	0.28	0.11	..	..	0.15	0.22
193.1.64.5	..	0.46	..	..	..	0.4
193.1.64.8	0.72	0.23	..	..	0.87	0.6
216.59.0.8	..	..	0.32	..	0.25	0.12
216.59.16.171	..	0.58	..	..	0.95	0.91
243.13.0.23	..	..	..	..	..	0.92
243.13.222.203	..	0.79	0.87	..	0.81	0.99

Prune

Master blocklist candidates	Check 1	Check 2
169.231.140.68	OK	!
193.1.64.5	OK	OK
193.1.64.8	OK	OK
216.59.0.8	OK	OK

IP_3

Check 2: If a prefix has any likely misclassification, it is excluded from expansion.

Selective Expansion

	Blocklist 1	Blocklist 2	Blocklist 3	..	Blocklist m	MB
169.231.140.10	..	..	0.8	..	..	?
169.231.140.68	0.3	0.1	..	..	0.1	?
193.1.64.5	..	0.5	..	..	..	?
193.1.64.8	0.7	0.5	..	..	0.9	?
216.59.0.8	..	..	0.04	..	0.1	?
216.59.16.171	..	0.7	..	..	0.9	1
243.13.0.23	..	..	..	..	..	1
243.13.222.203	..	0.7	1	..	0.9	1

Recommendation system

	Blocklist 1	Blocklist 2	Blocklist 3	..	Blocklist m-1	MB
169.231.140.10	..	..	0.78	..	..	0.75
169.231.140.68	0.28	0.11	..	..	0.15	0.22
193.1.64.5	..	0.46	..	..	..	0.4
193.1.64.8	0.72	0.23	..	..	0.87	0.6
216.59.0.8	..	..	0.32	..	0.25	0.12
216.59.16.171	..	0.58	..	..	0.95	0.91
243.13.0.23	..	..	..	..	..	0.92
243.13.222.203	..	0.79	0.87	..	0.81	0.99

Prune

Master blocklist candidates	**Check 1**	**Check 2**
169.231.140.68	OK	!
193.1.64.5	OK	OK
193.1.64.8	OK	OK
216.59.0.8	OK	OK

Selective expansion

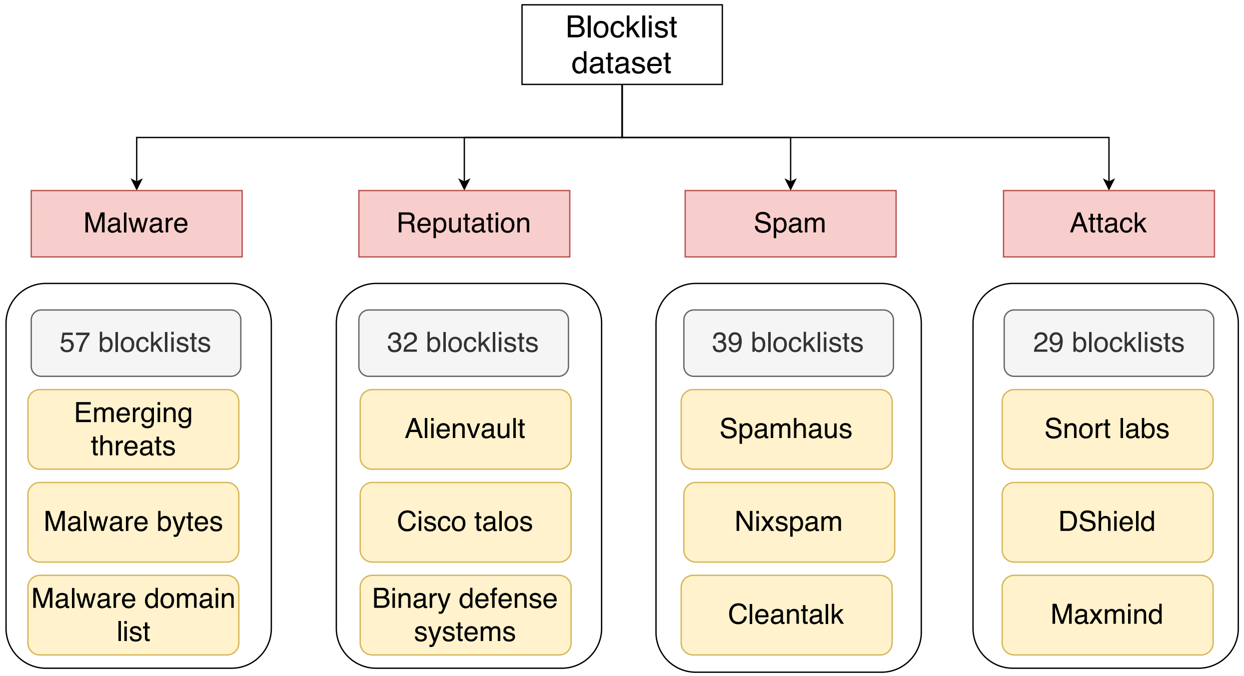
BLAG master blocklist
169.231.140.68
193.1.64.0/24
216.59.0.0/24

BLAG expands addresses to their /24 prefix only when both conditions are satisfied.

Outline

- Introduction
- Quantifying problems faced by blocklists
- BLAG
  - Datasets
  - Evaluation
- Identifying reused addresses
  - Detecting NATed addresses
  - Detecting dynamic addresses
  - Evaluation
- Summary

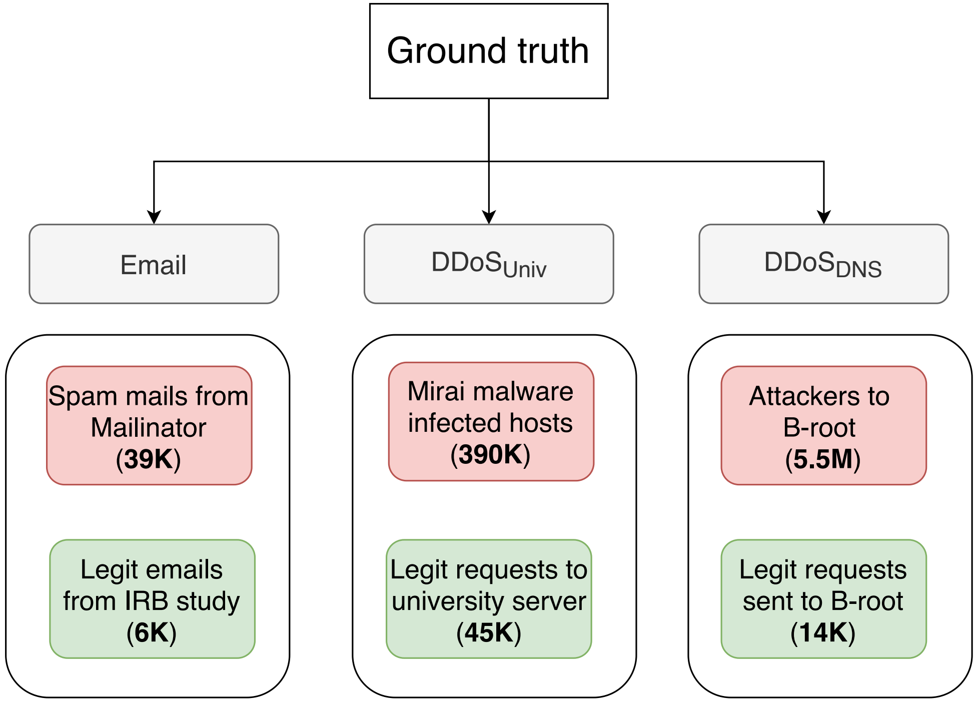
Monitored Blocklists



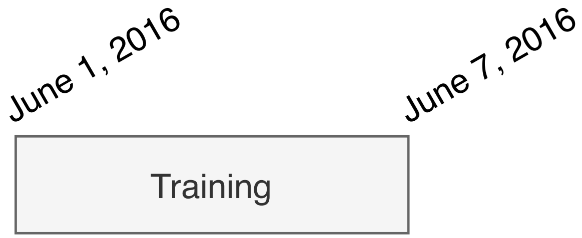
- 157 blocklists monitored from Jan 2016 to Dec 2017 roughly categorized into four attack variants.
- Collected over *176 million* IP addresses during this period.

Ground Truth for Evaluating Blocklists

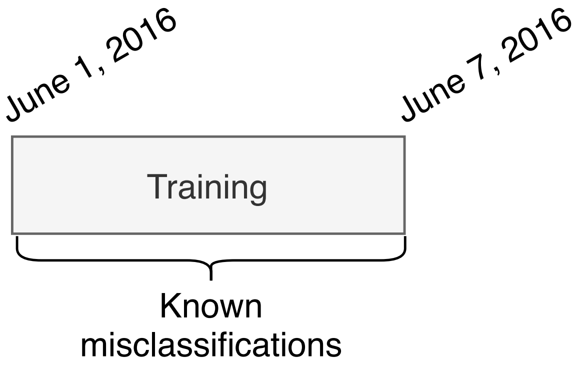
- Three types of ground truth, each with its corresponding legitimate and attack dataset.
- The legitimate portion is to validate the false detections of blocklists.
- The attack portion is to validate the accurate detections of blocklists.



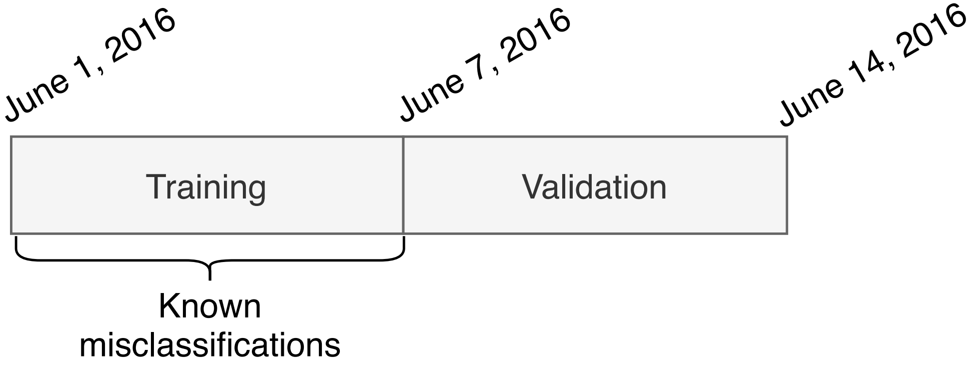
Email Dataset



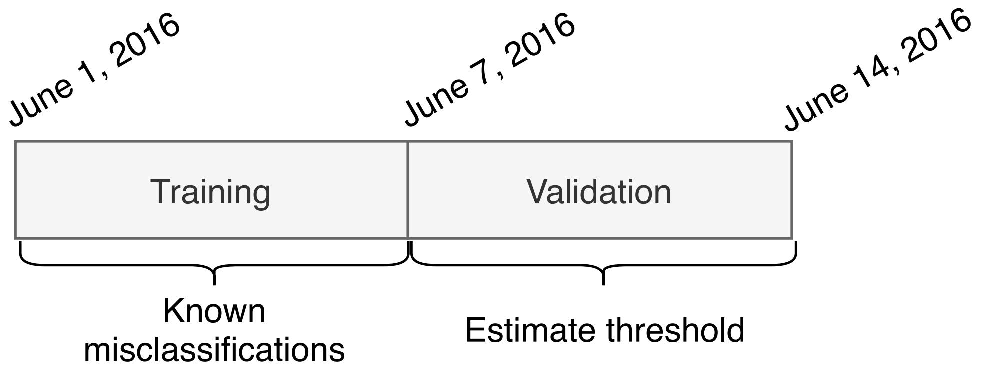
Email Dataset



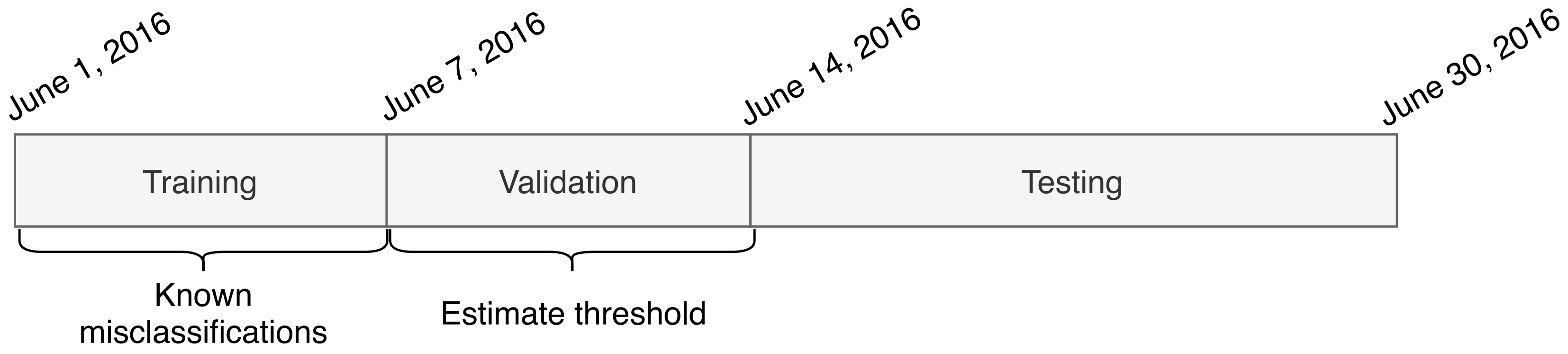
Email Dataset



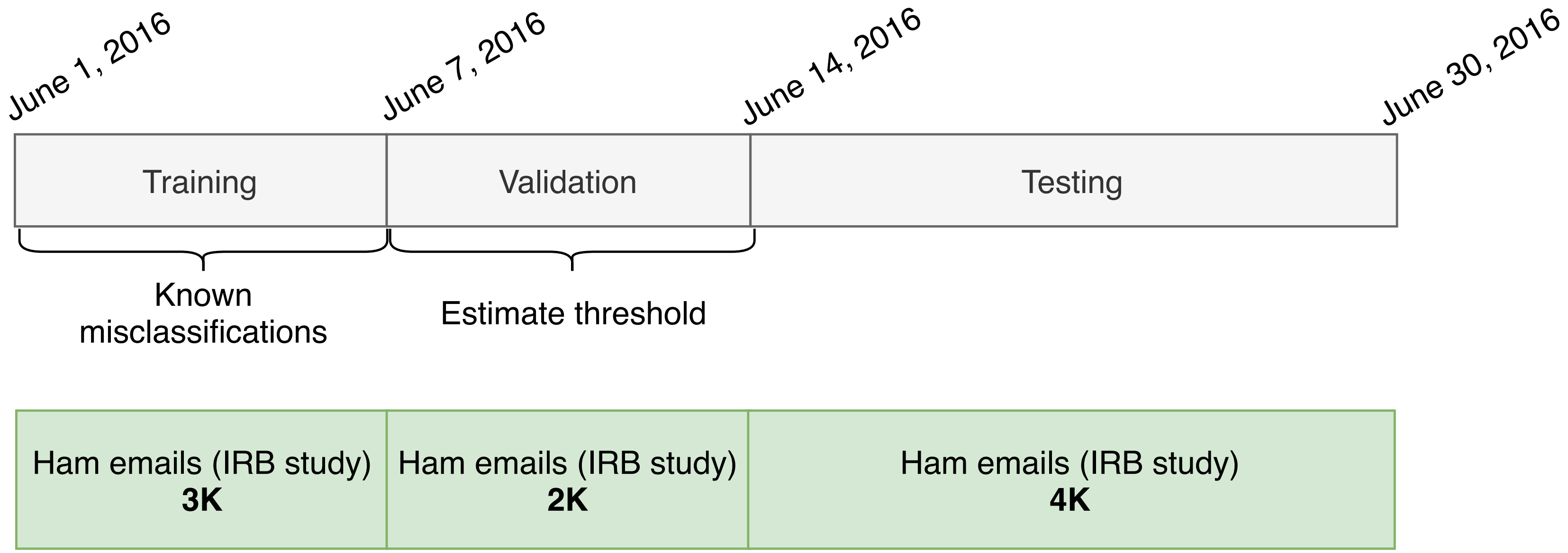
Email Dataset



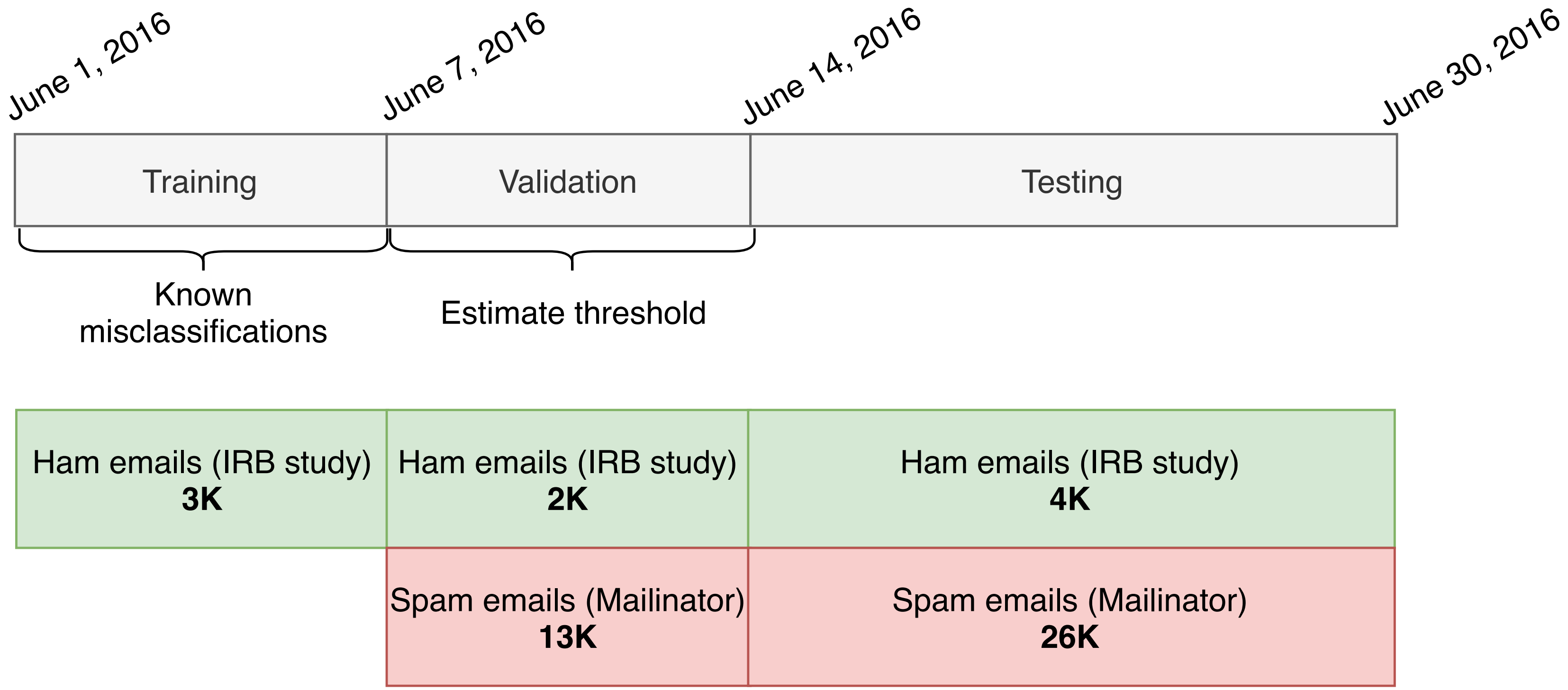
Email Dataset



Email Dataset



Email Dataset



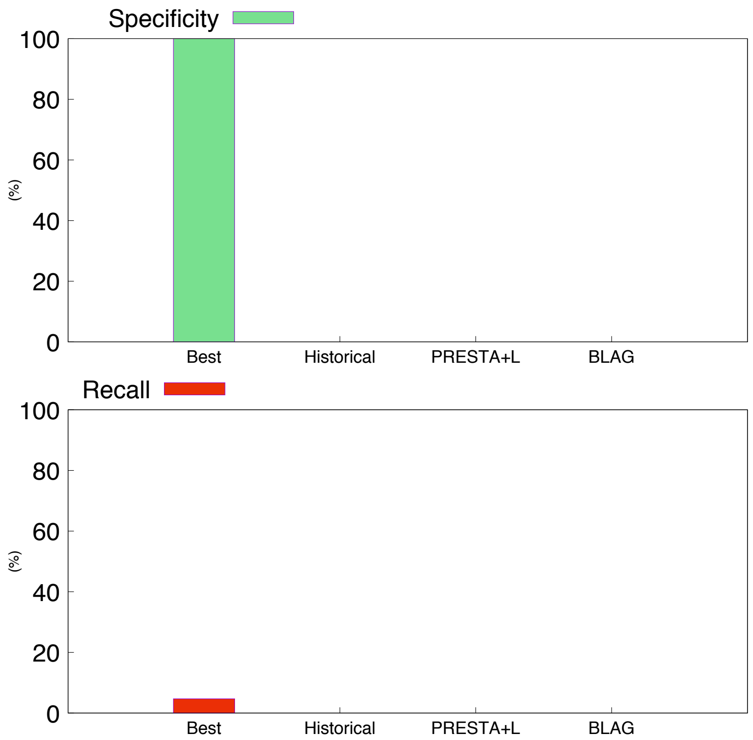
Outline

- Introduction
- Quantifying problems faced by blocklists
- BLAG
  - Datasets
  - Evaluation
- Usage and perception of blocklists
- Identifying reused addresses
  - Detecting NATed addresses
  - Detecting dynamic addresses
  - Evaluation
- Summary

Evaluation

- Accuracy of BLAG: Compare the performance of BLAG with competing approaches
  - Best: The best-performing blocklist on a given ground truth dataset (hindsight) at the given time (of the ground truth dataset).
  - Historical: All addresses listed in all blocklists up until ground truth dataset.
  - PRESTA+L: Blocklisting approach taken by PRESTA algorithm that uses spatial properties of blocklisted addresses to generate a new blocklist.
- Metrics:
  - Specificity - the percentage of legitimate addresses that were not false positives.
  - Recall - the percentage of offenders that were detected.

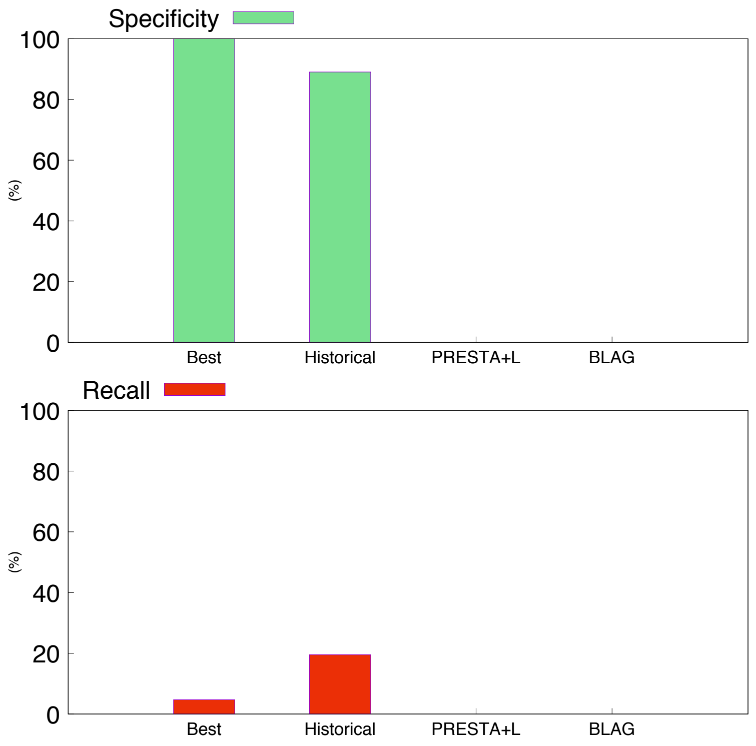
BLAG is Accurate



Email

Best blocklists have high specificity (>99%) but poor recall(< 4%) indicating that even the best blocklist is not enough to capture all attackers.

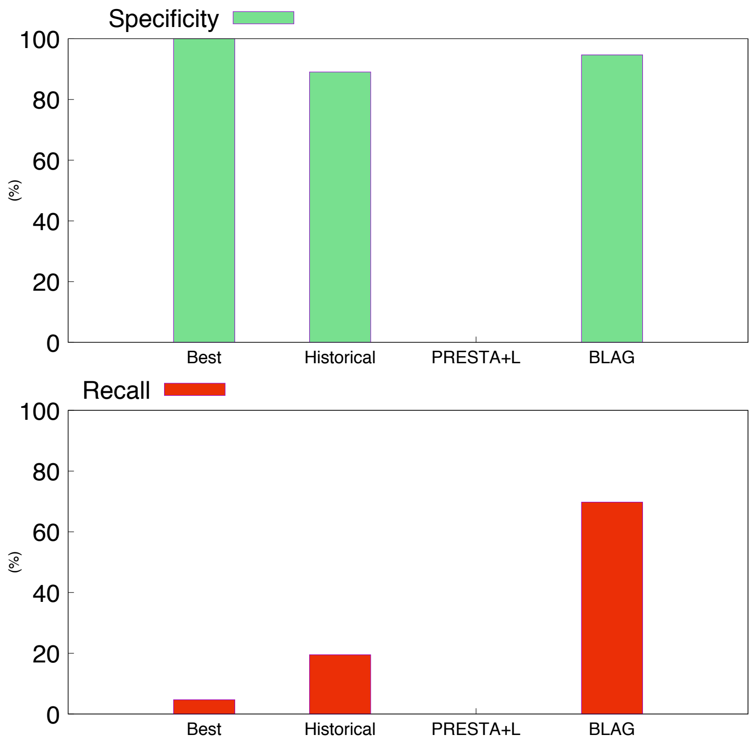
BLAG is Accurate



Email

Historical blocklists improve recall to 18% but with a drop in specificity by 12%, indicating that naïve combination of all blocklists has potential to capture attackers, but lowers specificity.

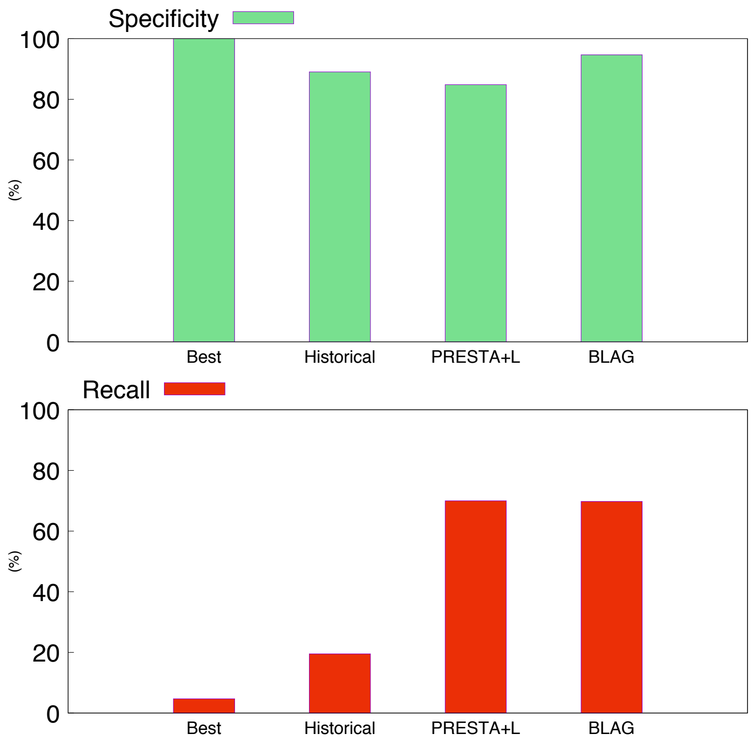
BLAG is Accurate



Email

BLAG with expansion further improves recall, with only a slight drop in specificity and has better specificity than historical blocklists.

BLAG is Accurate



Email

PRESTA+L has been tuned to have same recall as BLAG, but the specificity is lower than BLAG (82% vs 95%)

Other evaluations

- Evaluated BLAG on two other datasets: $\mathrm{DDoS}_{\mathrm{Univ}}$ and $\mathrm{DDoS}_{\mathrm{DNS}}$.
- Other expansion techniques -- expand using BGP prefixes or by autonomous systems.
- Impact of
  - Number of blocklists
  - Size of misclassification blocklists
- Contribution of recommendation system in aggregation and expansion phase.
- Parameter tuning techniques.

Outline

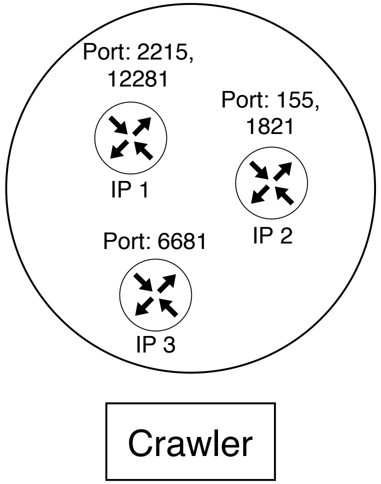
- Introduction
- Quantifying problems faced by blocklists
- BLAG
  - Datasets
  - Evaluation
- Identifying reused addresses
  - Detecting NATed addresses
  - Detecting dynamic addresses
  - Evaluation
- Summary

Detecting Reused NATed addresses

- We use the BitTorrent Network to identify users that are allocated the same IP address.
- The BitTorrent protocol allows two messages that helps us identify NATted users accurately.
  - *get_nodes*: Returns a list of active neighbors to a node.
  - *bt_ping*: Periodically pings active neighbors.
- The protocol mandates all BitTorrent users to reply to these messages.

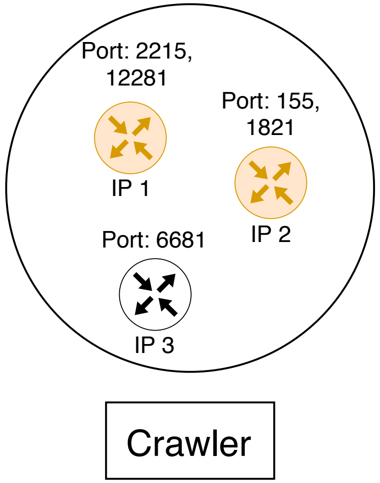
Detecting NATed addresses

Using *get_node* messages.



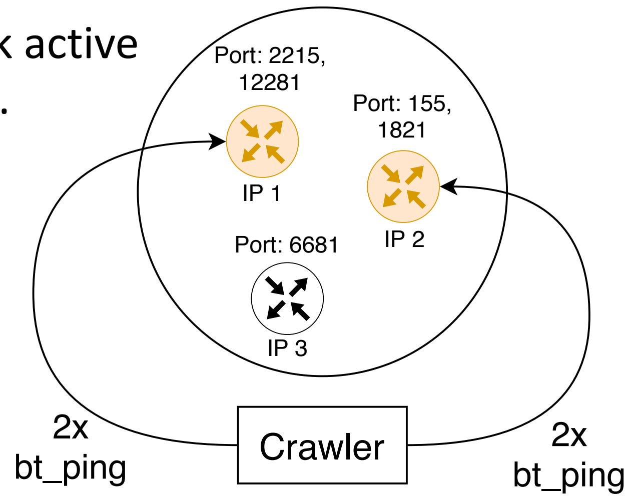
Detecting NATed addresses

Using *get_node* messages.

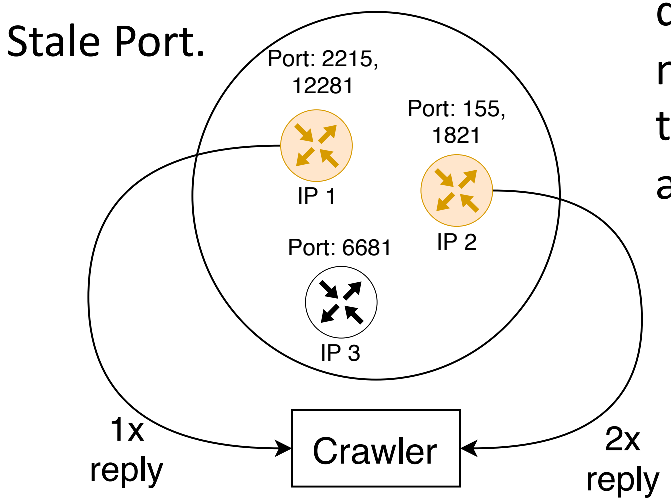


Detecting NATed addresses

Check active users.

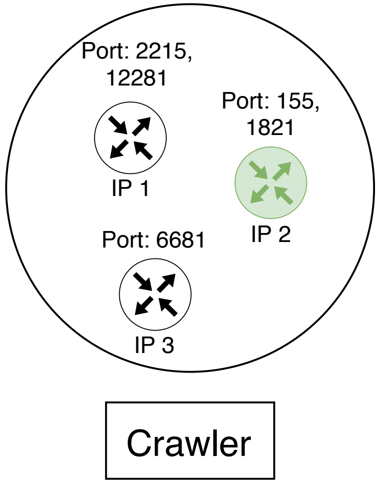


Detecting NATed addresses

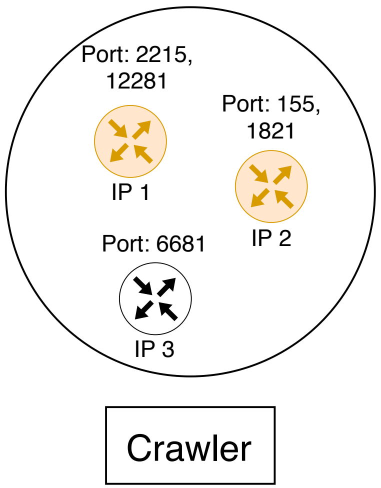


Two active users with two different port numbers using the same IP address.

Detecting NATed addresses

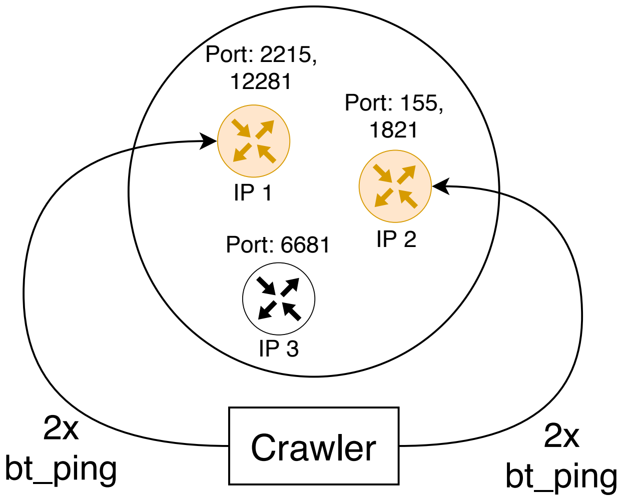


Discovered NATed addresses



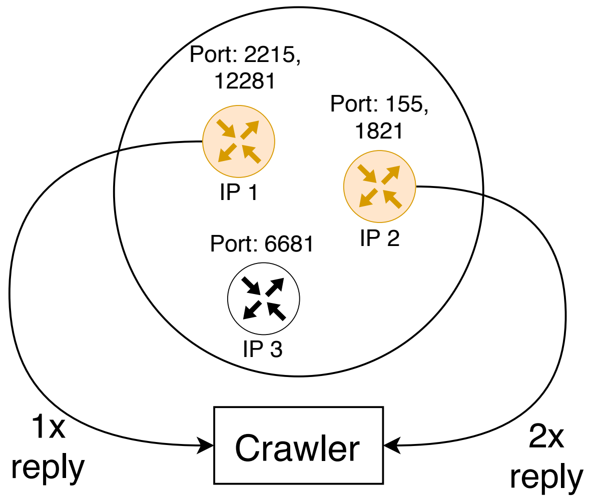
- 48.7M IP addresses that use BitTorrent.

Discovered NATed addresses



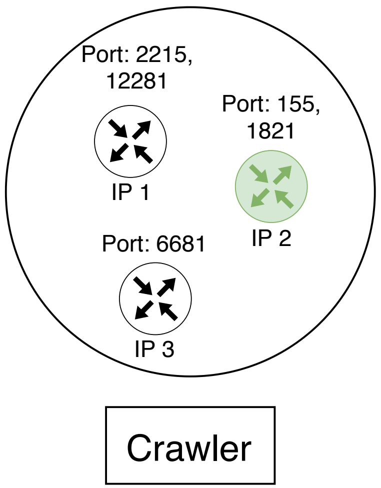
- 48.7M IP addresses that use BitTorrent.
- 1.6B bt_ping messages sent.

Discovered NATed addresses



- 48.7M IP addresses that use BitTorrent.
- 1.6B bt_ping messages sent.
- 779M responses (48.6%).

Discovered NATed addresses

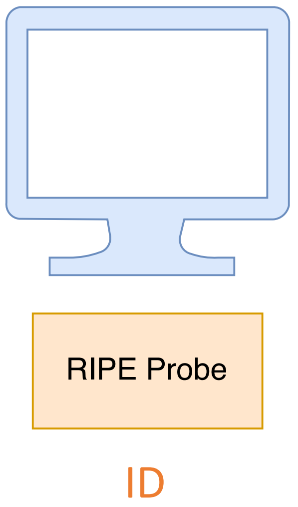


- 48.7M IP addresses that use BitTorrent.
- 1.6B bt_ping messages sent.
- 779M responses (48.6%).
- 2M IP addresses that are NATed.

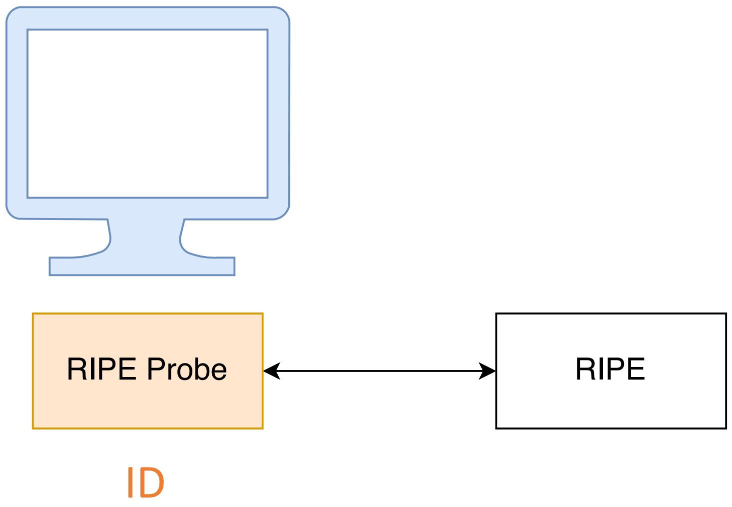
Outline

- Introduction
- Quantifying problems faced by blocklists
- BLAG
  - Datasets
  - Evaluation
- Usage and perception of blocklists
- Identifying reused addresses
  - Detecting NATed addresses
  - Detecting dynamic addresses
  - Evaluation
- Summary

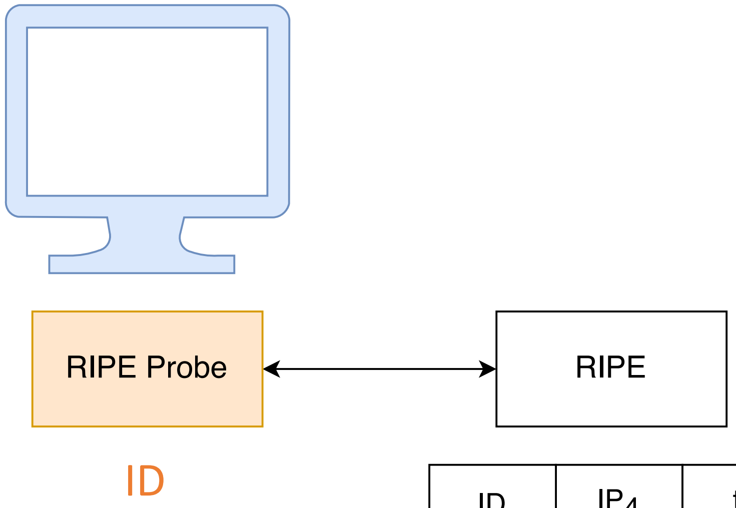
Detecting Dynamic Addresses



Detecting Dynamic Addresses

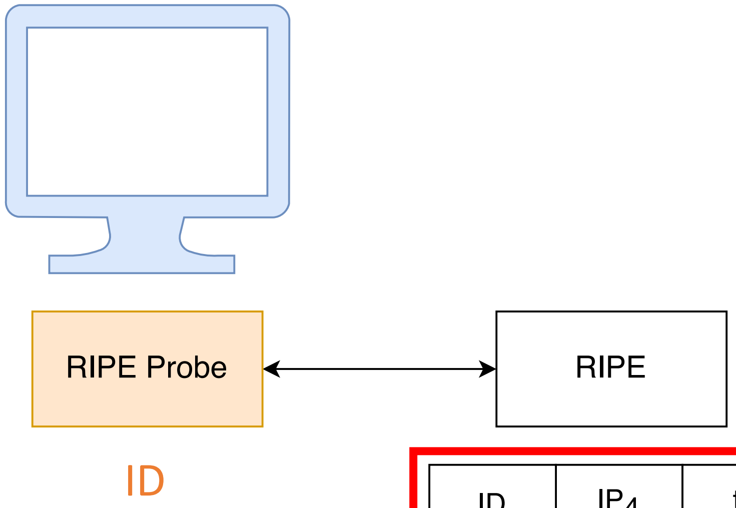


Detecting Dynamic Addresses



ID	IP_4	t_1
ID	IP_4	t_2
ID	IP_4	t_3

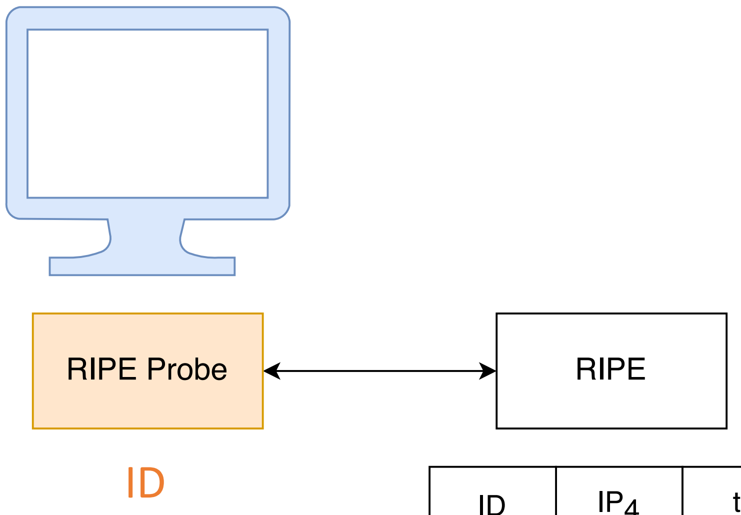
Detecting Dynamic Addresses



ID	IP_4	t_1
ID	IP_4	t_2
ID	IP_4	t_3

Measurement logs to determine dynamically allocated addresses.

Detecting Dynamic Addresses



ID	IP_4	t_1
ID	IP_4	t_2
ID	IP_4	t_3
ID	IP_6	t_4

IP4 and IP6 are potentially dynamically allocated.

Detecting Dynamic Addresses

To prevent users that have changed ISPs.

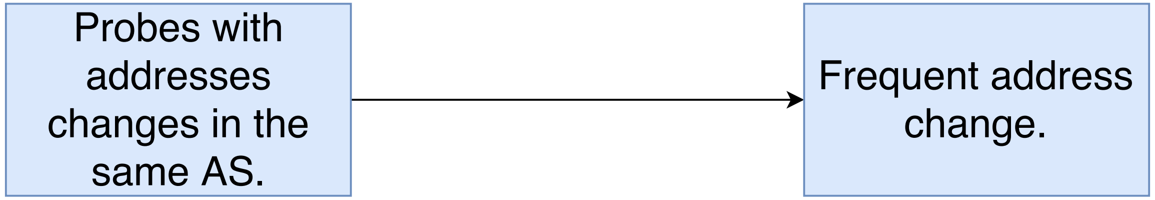
Probes with addresses changes in the same AS.

Remaining: 13.6K RIPE probes

Detecting Dynamic Addresses

To prevent users that have changed ISPs.

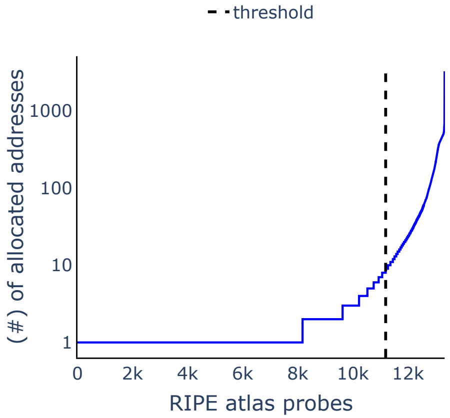
To consider probes that are potentially dynamically allocated.



Remaining: 13.6K RIPE probes

2.6K RIPE probes

Detecting Dynamic Addresses



To prevent users that have changed ISPs.

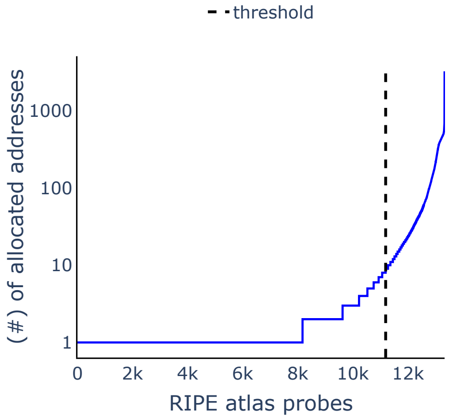
To consider probes that are potentially dynamically allocated.

Probes with addresses changes in the same AS. → Frequent address change.

Remaining: 13.6K RIPE probes

2.6K RIPE probes

Detecting Dynamic Addresses



To prevent users that have changed ISPs.

To consider probes that are potentially dynamically allocated.

Addresses that will have maximum impact on being blocklisted.

Probes with addresses changes in the same AS. → Frequent address change. → Change addresses daily.

Remaining: 13.6K RIPE probes

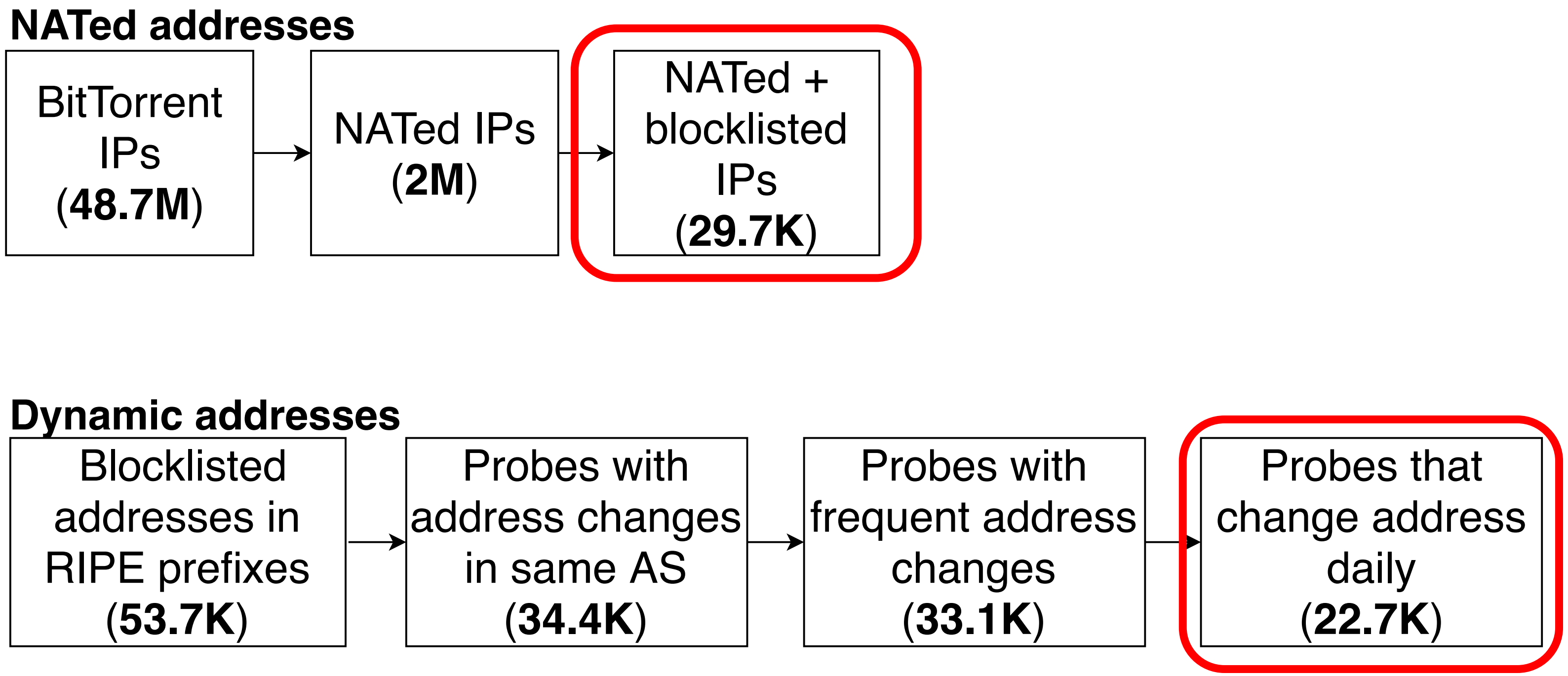
2.6K RIPE probes

629 RIPE probes

Quantifying Impact with Blocklists

- We use the BLAG dataset that actively maintains blocklisted addresses from public blocklists.
- 151 blocklists that monitor variety of attacks including Spam, DDoS, malware hosting or reputation of IP addresses.
- Monitoring period of 83 days over two measurement periods:
  - Aug 2019 – Sep 2019
  - Mar 2020 – May 2020
- Observed 2.2M blocklisted IP addresses.

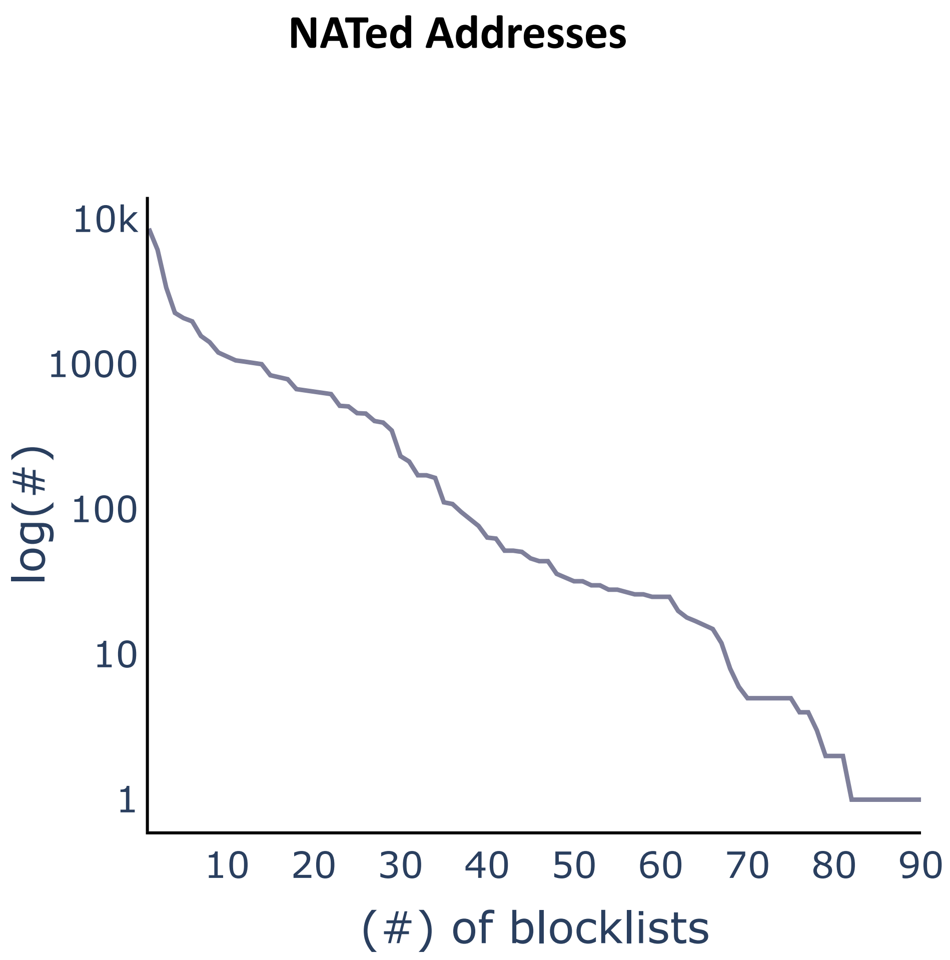
Number of Reused Addresses in Blocklists

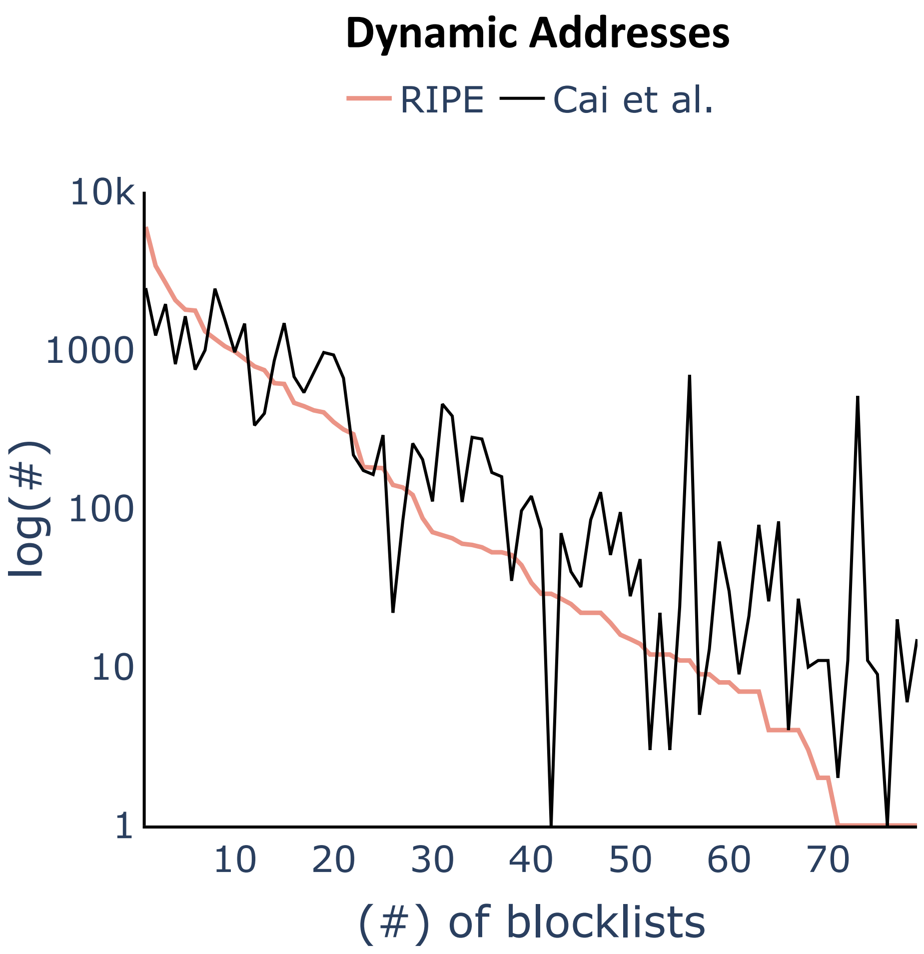


Outline

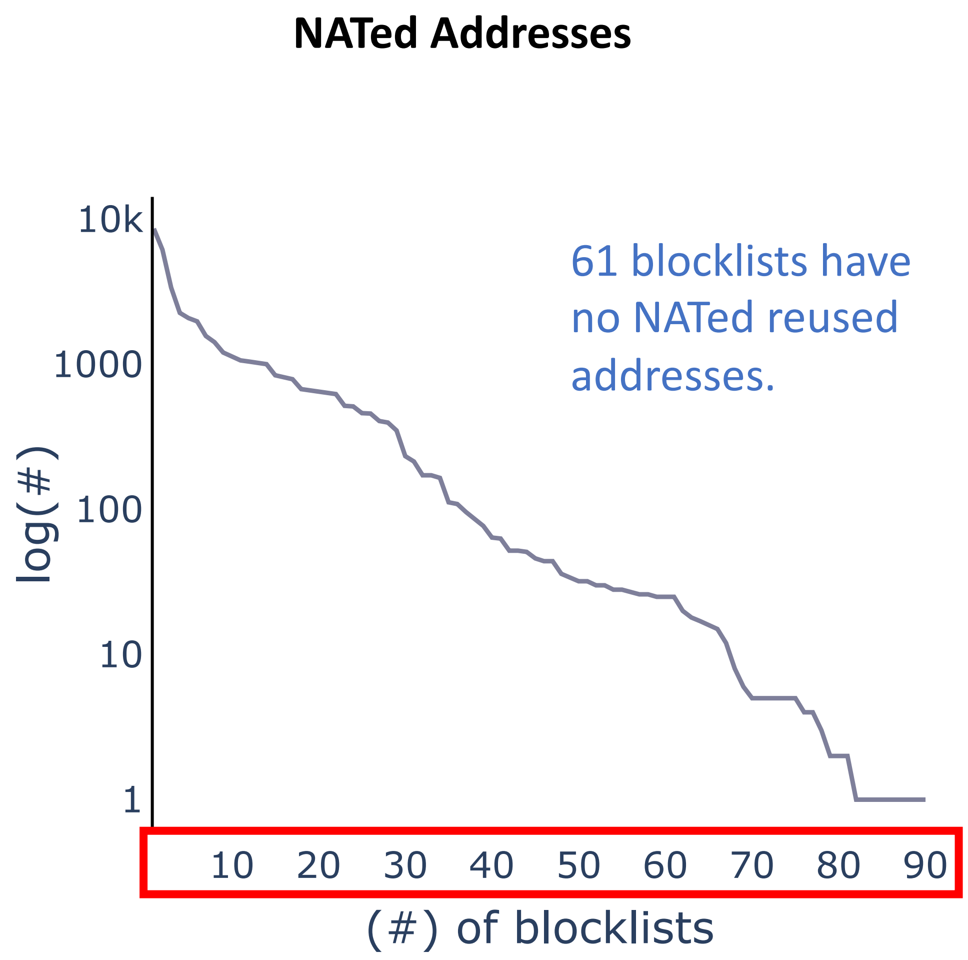
- Introduction
- Quantifying problems faced by blocklists
- BLAG
    - Datasets
    - Evaluation
- Usage and perception of blocklists
- Identifying reused addresses
    - Detecting NATed addresses
    - Detecting dynamic addresses
    - Evaluation
- Summary

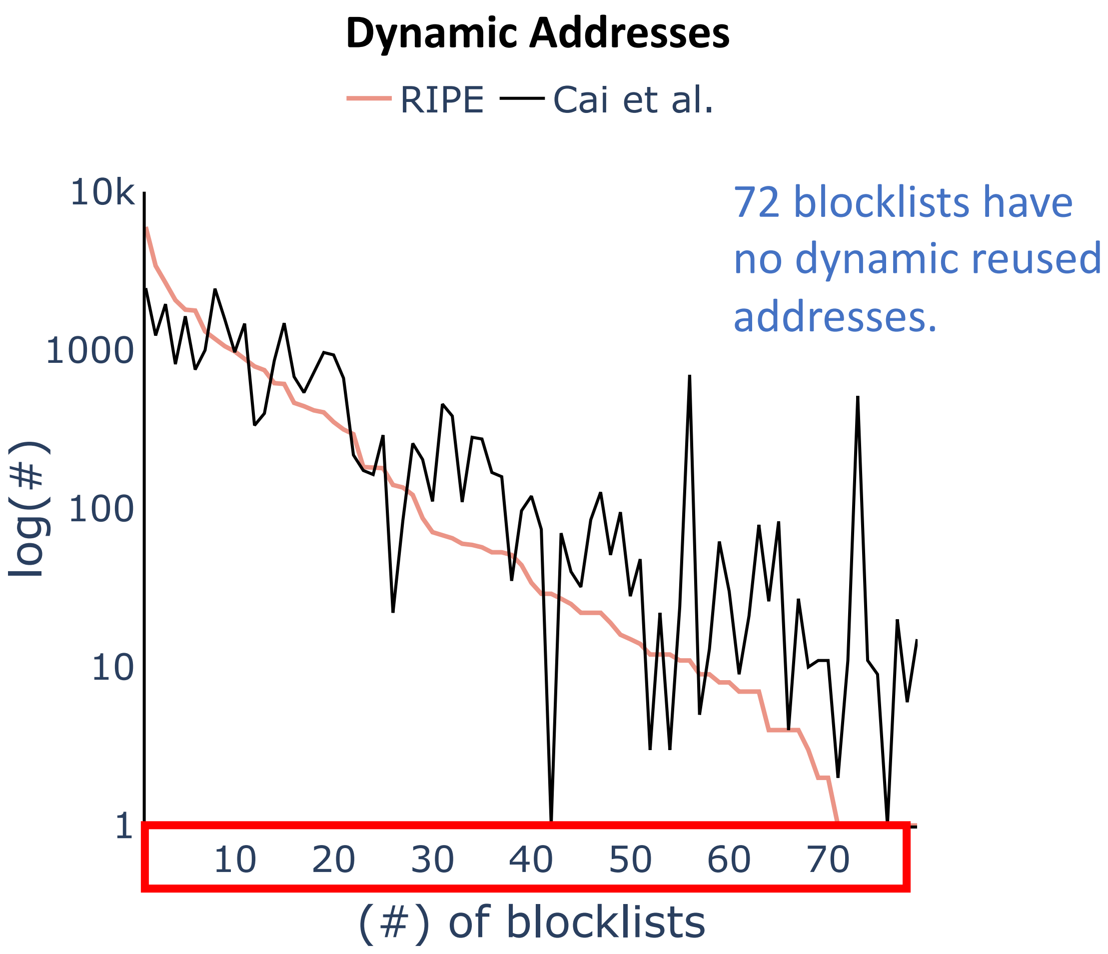
How many Blocklists list reused addresses?



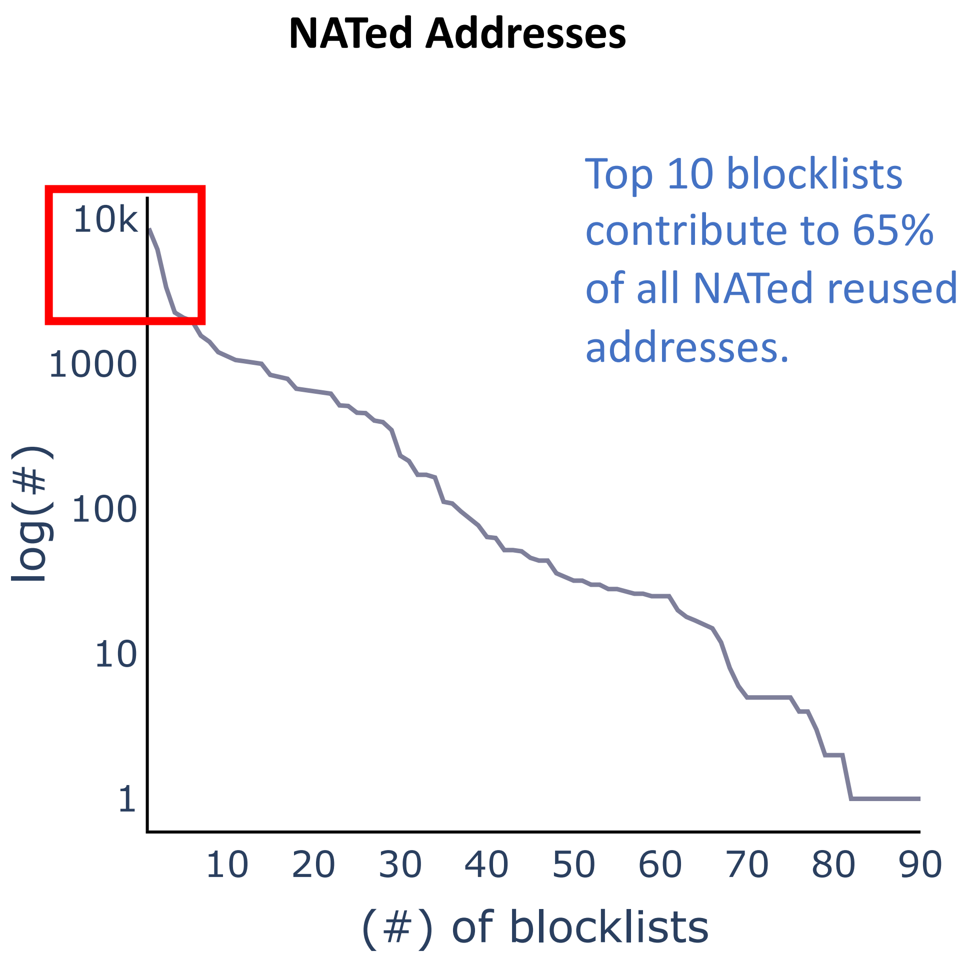


How many Blocklists list reused addresses?

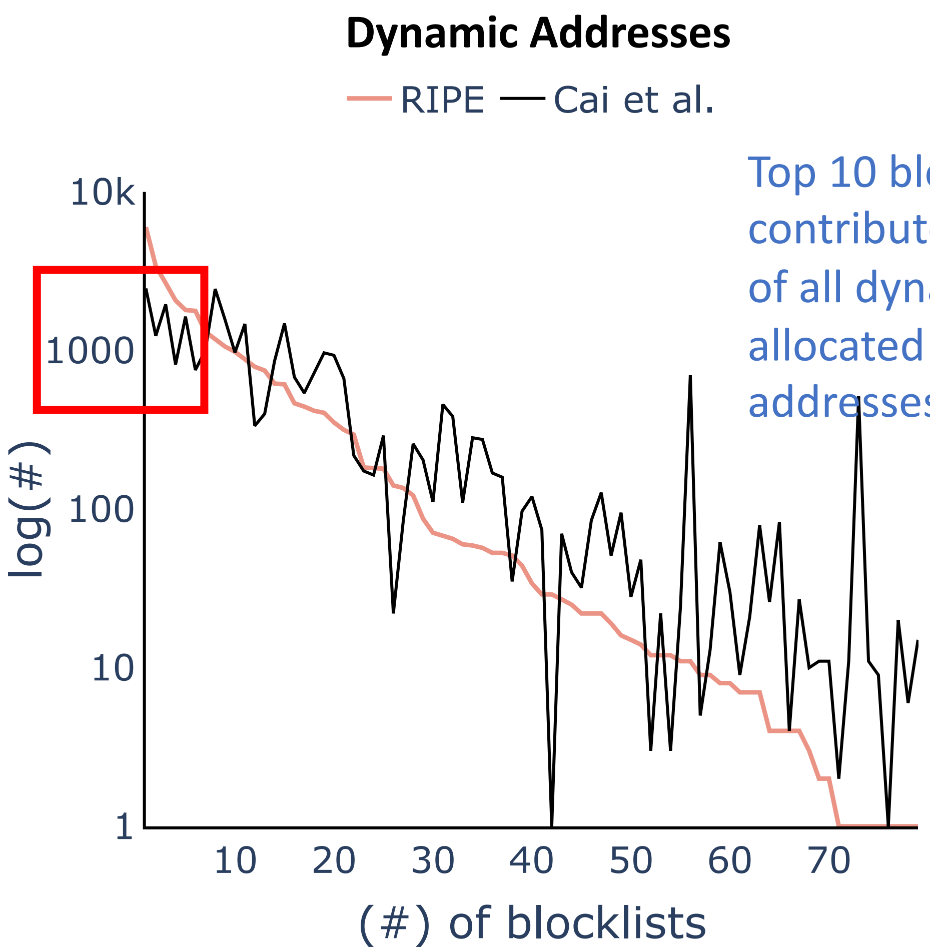




How many Blocklists list reused addresses?

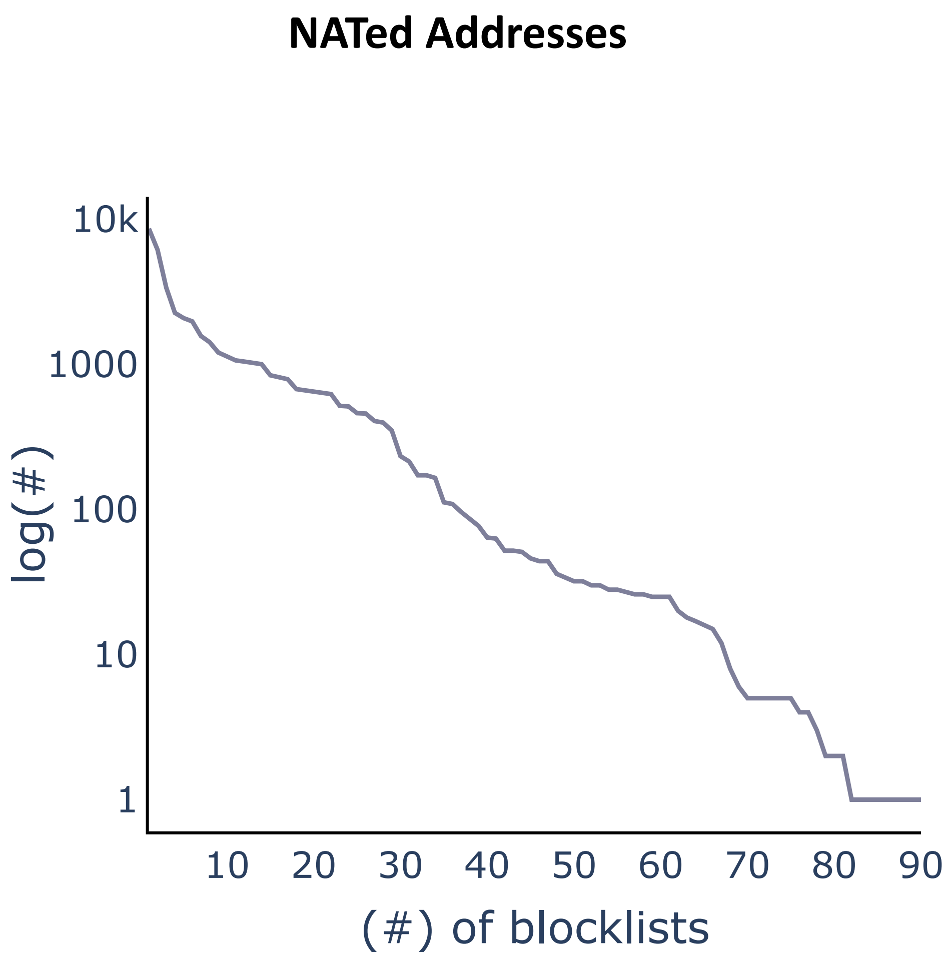


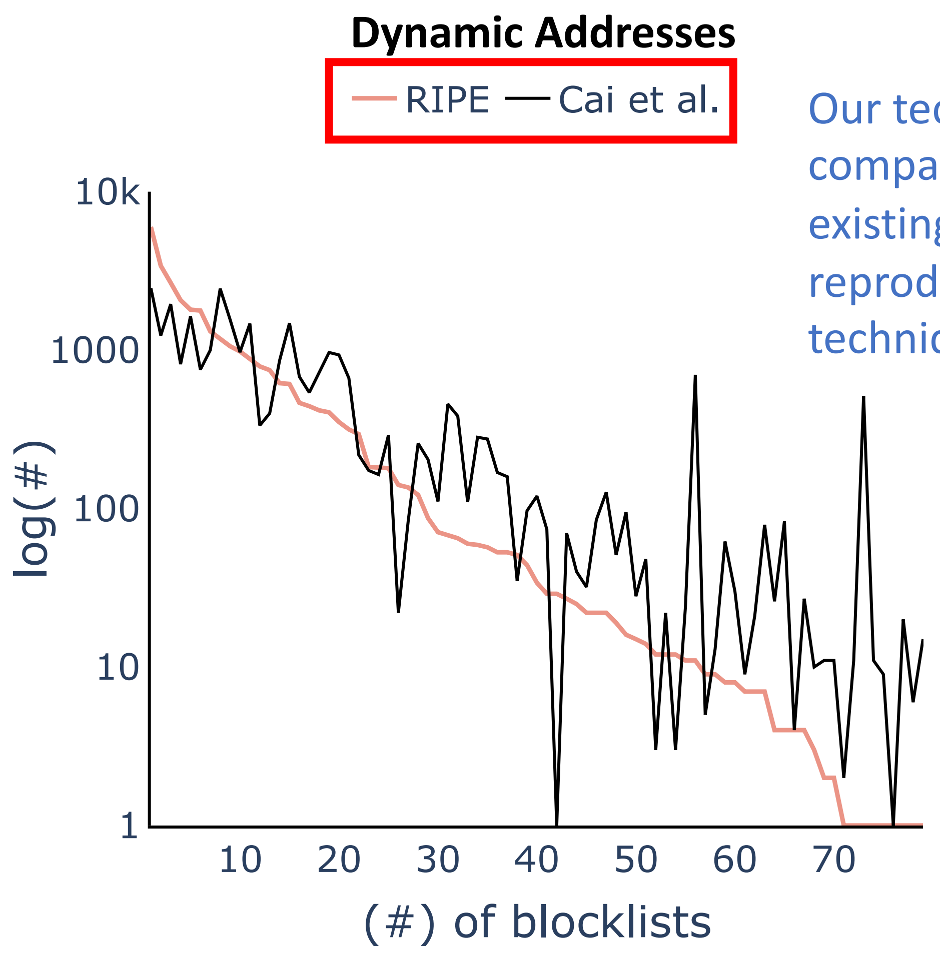
Top 10 blocklists contribute to 65% of all NATed reused addresses.



Top 10 blocklists contribute to 72% of all dynamically allocated reused addresses.

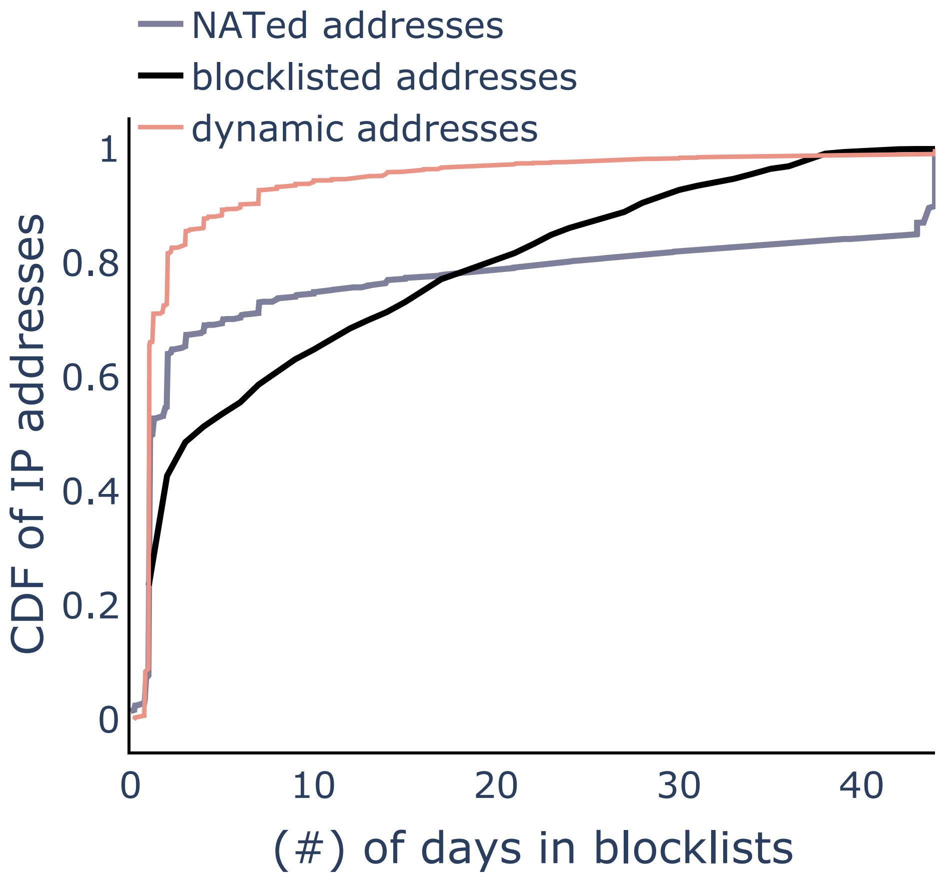
How many Blocklists list reused addresses?





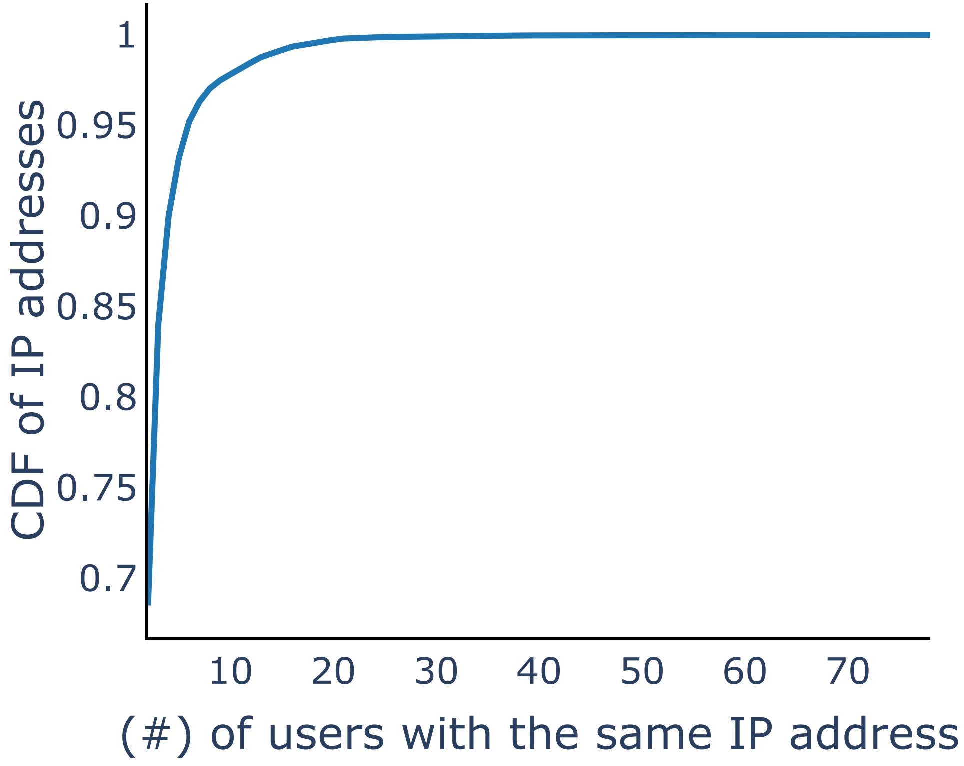
Our technique is comparable to existing reproducible technique.

How long are reused addreses in Blocklists?



- Reused addresses are removed faster than other addresses (3—9 days).
- Among reused addresses, dynamically allocated addresses are removed quicker.
- Within two days, 77% of dynamic addresses are removed compared to only 42% of all blocklisted addresses.

How many users are affected?



- Some IP addresses impact many more users, affecting as many as 78 users.
- Many IP addresses have only two active users (68.5%)
- 98% of IP addresses have less than 10 active users.

Summary

- Blocklists have poor attack detection.
- Combining blocklists from different sources improves attack detection, but also increases misclassifications.
- BLAG (blocklist aggregator)
  - Assigns relevance scores to addresses belonging to blocklists
  - Predicts addresses that are likely to be misclassifications using a recommendation system
  - Expands selective addresses into prefixes for better attack detection
- Reused addresses in blocklists can unjustly block more users.
- We propose two new techniques of identifying reused addresses in blocklists.

Thank You! Questions?

All datasets are available at:

https://steel.isi.edu/Projects/BLAG/

https://steel.isi.edu/members/sivaram/blocklisting_impact