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.
Problems with IP Blocklists

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

Fragmented information
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

Blocklists miss many attacks\textsuperscript{1,2} and may monitor only specific a type of attack.


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

1 Day

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

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.

- offenders in one given attack
- legitimate clients of a given network during the same attack
P2: Careful Aggregation

Combining blocklists can further potentially amplify the number of misclassifications.
P2: Careful Aggregation

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\(^1\).

P3: Careful Expansion

Expansion can further amplify misclassifications!
P3: Careful Expansion

Goal: Expand some addresses into prefixes that do not cause more misclassifications.
P4: Blocklisting Reused Addresses: NAT

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

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

https://community.cloudflare.com/t/cloudflare-blocking-my-ip/65453/57
P4: Blocklisting Reused Addresses: NAT

**Goal:** Accurately identify NATed reused address to prevent unjust blocking.
P4: Blocklisting Reused Addresses: Dynamic Addressing
P4: Blocklisting Reused Addresses: Dynamic Addressing
P4: Blocklisting Reused Addresses: Dynamic Addressing
P4: Blocklisting Reused Addresses: Dynamic Addressing

![Diagram showing public internet, device A with IP A, device B with IP B, and a blocklisted message.]

Blocklisted!
P4: Blocklisting Reused Addresses: Dynamic Addressing
Goal: Accurately identify dynamic reused address to prevent unjust blocking.
Problems with IP Blocklists

Problems

- Fragmented information
- Snapshots in time
- Reactive
- Address reuse

BLAG: Aggregation + Estimate Misclassification + Selective Expansion

Problems with IP Blocklists

- Fragmented information
- Snapshots in time
- Reactive

Address reuse

Quantifying the impact of Blocklisting

Outline

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

Aggregation

157 Blocklists
How BLAG Works

157 Blocklists

Aggregation

....

Estimate misclassification

Sample inbound traffic for a network

+   

Recommendation System
How BLAG Works

Aggregation

Estimate misclassification

Selective Expansion

Sample inbound traffic for a network

Recommendation System

157 Blocklists
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.

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
For address $a$ listed in blocklist $b$, the relevance score $r$, is given by:

$$r = 2^{t - t_{\text{out}}} \cdot l$$

Where,

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

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

- 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

<table>
<thead>
<tr>
<th></th>
<th>0.8</th>
<th>0.6</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.8</td>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>1</td>
<td>0.6</td>
<td>1</td>
</tr>
<tr>
<td>0.6</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td></td>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>0.8</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
### Estimate Misclassifications—Recommendation System

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>0.6</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.8</td>
<td>0.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>1</td>
<td>0.6</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td></td>
<td>0.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>0.8</td>
<td></td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Likes green books.
Estimate Misclassifications—Recommendation System

Likes green books.
Dislikes yellow books.
## Estimate Misclassifications—Recommendation System

<table>
<thead>
<tr>
<th></th>
<th>0.8</th>
<th>0.6</th>
<th>?</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.8</td>
<td>0.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>1</td>
<td>0.6</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td></td>
<td>0.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>0.8</td>
<td></td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Estimate Misclassifications—Recommendation System

<table>
<thead>
<tr>
<th>0.8</th>
<th>0.6</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>0.8</td>
<td>1</td>
<td>0.6</td>
</tr>
<tr>
<td>0.6</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>0.8</td>
<td>0.4</td>
<td>0.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0.8</th>
<th>0.59</th>
<th>0.6</th>
<th>0.7</th>
<th>0.99</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.99</td>
<td>0.97</td>
<td>0.8</td>
<td>0.92</td>
<td>0.8</td>
<td>1</td>
</tr>
<tr>
<td>0.8</td>
<td>0.85</td>
<td>0.99</td>
<td>0.59</td>
<td>0.99</td>
<td>1</td>
</tr>
<tr>
<td>0.7</td>
<td>0.6</td>
<td>0.6</td>
<td>0.66</td>
<td>0.72</td>
<td>0.99</td>
</tr>
<tr>
<td>0.66</td>
<td>0.79</td>
<td>0.5</td>
<td>0.6</td>
<td>0.29</td>
<td>0.8</td>
</tr>
<tr>
<td>0.77</td>
<td>0.85</td>
<td>0.4</td>
<td>0.79</td>
<td>0.55</td>
<td>0.99</td>
</tr>
</tbody>
</table>
Estimate Misclassifications—Recommendation System

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>0.6</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.8</td>
<td>0.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>1</td>
<td>0.6</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td></td>
<td>0.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>0.8</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>0.59</td>
<td>0.6</td>
<td><strong>0.7</strong></td>
<td>0.99</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0.99</td>
<td>0.97</td>
<td>0.8</td>
<td>0.92</td>
<td>0.8</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>0.85</td>
<td>0.99</td>
<td>0.59</td>
<td>0.99</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0.7</td>
<td>0.6</td>
<td>0.6</td>
<td>0.66</td>
<td>0.72</td>
<td>0.99</td>
<td></td>
</tr>
<tr>
<td>0.66</td>
<td>0.79</td>
<td>0.5</td>
<td>0.6</td>
<td>0.29</td>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td>0.77</td>
<td>0.85</td>
<td>0.4</td>
<td>0.79</td>
<td>0.55</td>
<td>0.99</td>
<td></td>
</tr>
</tbody>
</table>
Estimate Misclassifications–Recommendation System

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>0.6</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>0.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>1</td>
<td>0.6</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td></td>
<td>0.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>0.8</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>0.59</td>
<td>0.6</td>
<td>0.7</td>
<td>0.99</td>
</tr>
<tr>
<td>0.99</td>
<td>0.97</td>
<td>0.8</td>
<td>0.92</td>
<td>0.8</td>
</tr>
<tr>
<td>0.8</td>
<td>0.85</td>
<td>0.99</td>
<td>0.59</td>
<td>0.99</td>
</tr>
<tr>
<td>0.7</td>
<td>0.6</td>
<td>0.6</td>
<td>0.66</td>
<td>0.72</td>
</tr>
<tr>
<td>0.66</td>
<td>0.79</td>
<td>0.5</td>
<td>0.6</td>
<td>0.29</td>
</tr>
<tr>
<td>0.77</td>
<td>0.85</td>
<td>0.4</td>
<td>0.79</td>
<td>0.55</td>
</tr>
</tbody>
</table>
Estimate Misclassifications—Recommendation System
Estimate Misclassifications

- 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.
BLAG uses legitimate traffic traces of a network to introduce a new blocklist called the **Misclassification Blocklist (MB)**, which consists only of misclassifications.
For every known misclassification from the training data, BLAG allocates a score of 1.
## Estimate Misclassifications

<table>
<thead>
<tr>
<th>Blocklist 1</th>
<th>Blocklist 2</th>
<th>Blocklist 3</th>
<th>Blocklist m</th>
<th>MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>169.231.140.10</td>
<td>..</td>
<td>..</td>
<td>0.8</td>
<td>..</td>
</tr>
<tr>
<td>169.231.140.68</td>
<td>0.3</td>
<td>0.1</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>193.1.64.5</td>
<td>..</td>
<td>0.5</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>193.1.64.8</td>
<td>0.7</td>
<td>0.5</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>216.59.0.8</td>
<td>..</td>
<td>..</td>
<td>0.04</td>
<td>..</td>
</tr>
<tr>
<td>216.59.16.171</td>
<td>..</td>
<td>0.7</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>243.13.0.23</td>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>243.13.222.203</td>
<td>..</td>
<td>0.7</td>
<td>1</td>
<td>..</td>
</tr>
</tbody>
</table>

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


## Estimate Misclassifications

<table>
<thead>
<tr>
<th>IP</th>
<th>Blocklist 1</th>
<th>Blocklist 2</th>
<th>Blocklist 3</th>
<th>Blocklist m</th>
<th>MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>169.231.140.10</td>
<td>0.3</td>
<td>0.1</td>
<td>..</td>
<td>..</td>
<td>0.8</td>
</tr>
<tr>
<td>169.231.140.68</td>
<td>..</td>
<td>0.5</td>
<td>..</td>
<td>..</td>
<td>0.1</td>
</tr>
<tr>
<td>193.1.64.5</td>
<td>..</td>
<td>0.5</td>
<td>..</td>
<td>..</td>
<td>?</td>
</tr>
<tr>
<td>193.1.64.8</td>
<td>0.7</td>
<td>0.5</td>
<td>..</td>
<td>0.9</td>
<td>?</td>
</tr>
<tr>
<td>216.59.0.8</td>
<td>..</td>
<td>..</td>
<td>0.04</td>
<td>..</td>
<td>0.1</td>
</tr>
<tr>
<td>216.59.16.171</td>
<td>..</td>
<td>0.7</td>
<td>..</td>
<td>0.9</td>
<td>1</td>
</tr>
<tr>
<td>243.13.0.23</td>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
<td>1</td>
</tr>
<tr>
<td>243.13.222.203</td>
<td>..</td>
<td>0.7</td>
<td>1</td>
<td>..</td>
<td>0.9</td>
</tr>
</tbody>
</table>

**Goal:** Find the relevance scores for remaining addresses in MB.
# Estimate Misclassifications

<table>
<thead>
<tr>
<th>IP</th>
<th>Blocklist 1</th>
<th>Blocklist 2</th>
<th>Blocklist 3</th>
<th>Blocklist m</th>
<th>MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP1</td>
<td>169.231.140.10</td>
<td>..</td>
<td>..</td>
<td>0.8</td>
<td>..</td>
</tr>
<tr>
<td></td>
<td>169.231.140.68</td>
<td>0.3</td>
<td>0.1</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td></td>
<td>193.1.64.5</td>
<td>..</td>
<td>0.5</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td></td>
<td>193.1.64.8</td>
<td>0.7</td>
<td>0.5</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td></td>
<td>216.59.0.8</td>
<td>..</td>
<td>..</td>
<td>0.04</td>
<td>..</td>
</tr>
<tr>
<td></td>
<td>216.59.16.171</td>
<td>..</td>
<td>0.7</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td></td>
<td>243.13.0.23</td>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IP2</th>
<th>Blocklist 1</th>
<th>Blocklist 2</th>
<th>Blocklist 3</th>
<th>Blocklist m</th>
<th>MB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>243.13.222.203</td>
<td>..</td>
<td>0.7</td>
<td>1</td>
<td>..</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IP</th>
<th>Blocklist 1</th>
<th>Blocklist 2</th>
<th>Blocklist 3</th>
<th>Blocklist m-1</th>
<th>MB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>169.231.140.10</td>
<td>..</td>
<td>..</td>
<td>0.78</td>
<td>..</td>
</tr>
<tr>
<td></td>
<td>169.231.140.68</td>
<td>0.28</td>
<td>0.11</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td></td>
<td>193.1.64.5</td>
<td>..</td>
<td>0.46</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td></td>
<td>193.1.64.8</td>
<td>0.72</td>
<td>0.23</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td></td>
<td>216.59.0.8</td>
<td>..</td>
<td>..</td>
<td>0.32</td>
<td>..</td>
</tr>
<tr>
<td></td>
<td>216.59.16.171</td>
<td>..</td>
<td>0.58</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td></td>
<td>243.13.0.23</td>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td></td>
<td>243.13.222.203</td>
<td>..</td>
<td>0.79</td>
<td>0.87</td>
<td>..</td>
</tr>
</tbody>
</table>

**Goal:** Find the relevance scores for remaining addresses in MB.
### Estimate Misclassifications

<table>
<thead>
<tr>
<th>IP1</th>
<th>Blocklist 1</th>
<th>Blocklist 2</th>
<th>Blocklist 3</th>
<th>Blocklist m</th>
<th>MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>169.231.140.10</td>
<td>0.3</td>
<td>0.1</td>
<td>...</td>
<td>0.1</td>
<td>?</td>
</tr>
<tr>
<td>169.231.140.68</td>
<td>0.5</td>
<td>...</td>
<td>0.9</td>
<td>0.1</td>
<td>?</td>
</tr>
<tr>
<td>193.1.64.5</td>
<td>0.7</td>
<td>0.5</td>
<td>...</td>
<td>0.9</td>
<td>?</td>
</tr>
<tr>
<td>216.59.0.8</td>
<td>0.1</td>
<td>0.04</td>
<td>...</td>
<td>0.1</td>
<td>?</td>
</tr>
<tr>
<td>216.59.16.171</td>
<td>...</td>
<td>0.7</td>
<td>...</td>
<td>0.9</td>
<td>1</td>
</tr>
<tr>
<td>243.13.0.23</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>243.13.222.203</td>
<td>0.7</td>
<td>1</td>
<td>...</td>
<td>0.9</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IP1</th>
<th>Blocklist 1</th>
<th>Blocklist 2</th>
<th>Blocklist 3</th>
<th>Blocklist m-1</th>
<th>MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>169.231.140.10</td>
<td>0.28</td>
<td>0.11</td>
<td>...</td>
<td>0.15</td>
<td>0.22</td>
</tr>
<tr>
<td>169.231.140.68</td>
<td>0.46</td>
<td>0.23</td>
<td>...</td>
<td>0.87</td>
<td>0.4</td>
</tr>
<tr>
<td>193.1.64.5</td>
<td>0.72</td>
<td>0.32</td>
<td>...</td>
<td>0.25</td>
<td>0.12</td>
</tr>
<tr>
<td>193.1.64.8</td>
<td>0.95</td>
<td>0.58</td>
<td>...</td>
<td>0.96</td>
<td>0.91</td>
</tr>
<tr>
<td>216.59.0.8</td>
<td>0.25</td>
<td>0.58</td>
<td>...</td>
<td>0.25</td>
<td>0.12</td>
</tr>
<tr>
<td>216.59.16.171</td>
<td>0.75</td>
<td>0.99</td>
<td>...</td>
<td>0.91</td>
<td>0.92</td>
</tr>
<tr>
<td>243.13.0.23</td>
<td>0.79</td>
<td>0.87</td>
<td>...</td>
<td>0.81</td>
<td>0.99</td>
</tr>
<tr>
<td>243.13.222.203</td>
<td>0.79</td>
<td>0.87</td>
<td>...</td>
<td>0.81</td>
<td>0.99</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Blocklist 1</th>
<th>Blocklist 2</th>
<th>Blocklist 3</th>
<th>Blocklist m</th>
<th>MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>169.231.140.10</td>
<td>..</td>
<td>0.8</td>
<td>..</td>
<td>?</td>
</tr>
<tr>
<td>169.231.140.68</td>
<td>0.3</td>
<td>0.1</td>
<td>..</td>
<td>0.1</td>
</tr>
<tr>
<td>193.1.64.5</td>
<td>..</td>
<td>0.5</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>193.1.64.8</td>
<td>0.7</td>
<td>0.5</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>216.59.0.8</td>
<td>..</td>
<td>..</td>
<td>0.04</td>
<td>..</td>
</tr>
<tr>
<td>216.59.16.171</td>
<td>..</td>
<td>0.7</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>243.13.0.23</td>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>243.13.222.203</td>
<td>..</td>
<td>0.7</td>
<td>1</td>
<td>..</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Blocklist 1</th>
<th>Blocklist 2</th>
<th>Blocklist 3</th>
<th>Blocklist m-1</th>
<th>MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>169.231.140.10</td>
<td>..</td>
<td>..</td>
<td>0.78</td>
<td>..</td>
</tr>
<tr>
<td>169.231.140.68</td>
<td>0.28</td>
<td>0.11</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>193.1.64.5</td>
<td>..</td>
<td>0.46</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>193.1.64.8</td>
<td>0.72</td>
<td>0.23</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>216.59.0.8</td>
<td>..</td>
<td>..</td>
<td>0.32</td>
<td>..</td>
</tr>
<tr>
<td>216.59.16.171</td>
<td>..</td>
<td>0.58</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>243.13.0.23</td>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>243.13.222.203</td>
<td>..</td>
<td>0.79</td>
<td>0.87</td>
<td>..</td>
</tr>
</tbody>
</table>

### Recommendation System

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

### Prune

- Master blocklist candidates: 169.231.140.68, 193.1.64.5, 193.1.64.8, 216.59.0.8
- Check 1: OK
Selective Expansion

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

<table>
<thead>
<tr>
<th>Blocklist 1</th>
<th>Blocklist 2</th>
<th>Blocklist 3</th>
<th>Blocklist m</th>
<th>MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>169.231.140.10</td>
<td>..</td>
<td>..</td>
<td>0.8</td>
<td>..</td>
</tr>
<tr>
<td>169.231.140.68</td>
<td>0.3</td>
<td>0.1</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>193.1.64.5</td>
<td>..</td>
<td>0.5</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>193.1.64.8</td>
<td>0.7</td>
<td>0.5</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>216.59.0.8</td>
<td>..</td>
<td>..</td>
<td>0.04</td>
<td>..</td>
</tr>
<tr>
<td>216.59.16.171</td>
<td>..</td>
<td>0.7</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>243.13.0.23</td>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>243.13.222.203</td>
<td>..</td>
<td>0.7</td>
<td>1</td>
<td>..</td>
</tr>
</tbody>
</table>

**Recommendation system**

<table>
<thead>
<tr>
<th>Blocklist 1</th>
<th>Blocklist 2</th>
<th>Blocklist 3</th>
<th>Blocklist m-1</th>
<th>MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>169.231.140.10</td>
<td>..</td>
<td>..</td>
<td>0.78</td>
<td>..</td>
</tr>
<tr>
<td>169.231.140.68</td>
<td>0.28</td>
<td>0.11</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>193.1.64.5</td>
<td>..</td>
<td>0.46</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>193.1.64.8</td>
<td>0.72</td>
<td>0.23</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>216.59.0.8</td>
<td>..</td>
<td>..</td>
<td>0.32</td>
<td>..</td>
</tr>
<tr>
<td>216.59.16.171</td>
<td>..</td>
<td>0.58</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>243.13.0.23</td>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>243.13.222.203</td>
<td>..</td>
<td>0.79</td>
<td>0.87</td>
<td>..</td>
</tr>
</tbody>
</table>

**Master blocklist candidates**

| 169.231.140.68 | OK | ! |
| 193.1.64.5 | OK | OK |
| 193.1.64.8 | OK | OK |
| 216.59.0.8 | OK | OK |

**Prune**

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

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

June 1, 2016 to June 7, 2016 Training
Email Dataset

June 1, 2016

Training

Known misclassifications

June 7, 2016
Email Dataset

- Training: June 1, 2016
- Validation: June 7, 2016
- Known misclassifications: June 14, 2016
Email Dataset

- June 1, 2016: Training
  - Known misclassifications
- June 7, 2016: Validation
  - Estimate threshold
Email Dataset

- Training
- Validation
- Testing

- Known misclassifications
- Estimate threshold

June 1, 2016 to June 30, 2016
Email Dataset

- **Training** (June 1, 2016)
  - Known misclassifications

- **Validation** (June 7, 2016)
  - Estimate threshold

- **Testing** (June 14, 2016)

- **Ham emails (IRB study)**
  - June 1, 2016: 3K
  - June 7, 2016: 2K
  - June 14, 2016: 4K

- **June 30, 2016**
Email Dataset

- Training
  - June 1, 2016
  - Known misclassifications

- Validation
  - June 7, 2016
  - Estimate threshold

- Testing
  - June 14, 2016
  - June 30, 2016

<table>
<thead>
<tr>
<th>Category</th>
<th>June 1, 2016</th>
<th>June 7, 2016</th>
<th>June 14, 2016</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ham emails (IRB study)</td>
<td>3K</td>
<td>2K</td>
<td>4K</td>
</tr>
<tr>
<td>Spam emails (Mailinator)</td>
<td>13K</td>
<td>26K</td>
<td></td>
</tr>
</tbody>
</table>
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

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

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

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: $\text{DDoS}_{\text{Univ}}$ and $\text{DDoS}_{\text{DNS}}$.
• 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 \textit{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
Detecting Dynamic Addresses

Measurement logs to determine dynamically allocated addresses.
Detecting Dynamic Addresses

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.

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.

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

NATed addresses

<table>
<thead>
<tr>
<th>BitTorrent IPs (48.7M)</th>
<th>NATed IPs (2M)</th>
<th>NATed + blocklisted IPs (29.7K)</th>
</tr>
</thead>
</table>

Dynamic addresses

| Blocklisted addresses in RIPE prefixes (53.7K) | Probes with address changes in same AS (34.4K) | Probes with frequent address changes (33.1K) | Probes that change address daily (22.7K) |
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?

NATed Addresses

Dynamic Addresses

RIPE — Cai et al.
How many Blocklists list reused addresses?

**NATed Addresses**
- 61 blocklists have no NATed reused addresses.

**Dynamic Addresses**
- 72 blocklists have no dynamic reused addresses.

RIPE — Cai et al.
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 addresses 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