Leveraging Prefix Structure to Detect Volumetric DDoS Attack Signatures with Programmable Switches

Chris Misa
Ramakrishnan Durairajan
Reza Rejaie
Arpit Gupta — UCSB
Walter Willinger — NIKSUN, Inc.
Background: IP Addresses

IPv4 “Address space”

\[\{0.0.0.0, \ldots, 128.0.0.0, \ldots, 255.255.255.255\}\]
Background: IP Address Prefixes

IPv4 “Address space”

{0.0.0.0,... ...,128.0.0.0,... ...,255.255.255.255}

Internet

Sources in Common “Child” Prefix

Prefix in Common “Parent” Prefix

Benign source IP address

0.0.0.0/2

64.0.0.0/2

0.0.0.0/1

128.0.0.0/1

0.0.0.0/0
Setting: Edge Networks

External source IP addresses

Internet

Benign source IP address

Protected Server

Edge Network

Border Switch

cmisa@cs.uoregon.edu
Setting: Edge Networks

External source IP addresses

Internet

Border Switch

Protected Server

○ Benign source IP address / traffic
Setting: Volumetric DDoS Against Edge Networks

Benign traffic is affected!

- Benign source IP address / traffic
- Attack source IP address / traffic
Setting: Dynamic Volumetric DDoS Against ...

Attack sources can change!

- Benign source IP address / traffic
- Attack source IP address / traffic
Need: Volumetric DDoS Signature Detection

- Benign source IP address / traffic
- Attack source IP address / traffic
(Prior) Sketch-Based Detection has Random FPs

For example, Euclid and Jaqen get 4% - 50% false-positive rate in a realistic attack scenario with 50k attack source addresses.
(Prior) ML-Based Detection has High Overheads

For example, LUCID would require ~80MB switch hardware memory.
(Prior) Prefix-Level Detection has High FNs

For example, DREAM only allows counter-based detection.
## Prior Efforts Fail to Meet All Requirements

<table>
<thead>
<tr>
<th></th>
<th>Accurate</th>
<th>Scalable</th>
<th>Robust</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sketch</td>
<td>Jaqen, Euclid</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Flow-level ML</td>
<td>LUCID</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Prefix refinement</td>
<td>RADAR</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
ZAPDOS: Prefix-Level ML + Iterative Refinement

Accurate →
Scalable →
Robust →
ZAPDOS: Prefix-Level ML + Iterative Refinement

Control plane

*Trained Model*

*Update Algo.*

Decision-making

Traffic Monitor

Mitigation

Protected Server

Attacker

External source

IP addresses
ZAPDOS: Prefix-Level ML

Prefix-level model can better “see through” ambiguous prefixes.
ZAPDOS: Iterative Refinement

Accurate → prefix-level ML

Scalable →

Robust →

Improved refinement
• Look-Ahead
• Look-Back
ZAPDOS: Refine with Look-Ahead

Real addresses are clustered and sparse at long prefix lengths.

With look-ahead only non-empty children are monitored.

Source IP prefix

Structure of 50k mirai sources (attack) and ~5M MAWILab sources.
ZAPDOS: Refine with Look-Ahead

ZAPDOS achieves fast, accurate refinement using limited hardware resources.

Accurate → prefix-level ML
Scalable → look-ahead
Robust →
ZAPDOS: Look-Back to Catch Changes

Accurate → prefix-level ML
Scalable → look-ahead
Robust → look-back

Un-monitored Prefix

Bloom filter

X1, X2, X3

?
ZAPDOS: Look-Back to Catch Changes

Look-back quickly re-focuses on the changed attack source address.

Accurate → prefix-level ML
Scalable → look-ahead
Robust → look-back
ZAPDOS: Look-Back to Catch Changes

Look-back enables quickly re-focusing refinement when the attack changes.

Accurate → prefix-level ML
Scalable → look-ahead
Robust → look-back
ZAPDOS: See More in Our Paper

Data-fusion for training on >100 attack scenarios with realistic prefix-level distributions.

Implemented a Tofino prototype.

- Batch-round-robin updates.
- Packet ferries for feature collection.

- Accurate → prefix-level ML
- Scalable → look-ahead
- Robust → look-back

Extended evaluation.

- <1% FPR and FNR across attack vectors.
- Works for spoofed attack sources.
- Works when the upstream border link is flooded.

cmisa@cs.uoregon.edu
Thanks!

Chris Misa
cmisa@cs.uoregon.edu

See our project page:
onrg.gitlab.io/projects/zapdos/