BACKORDERS: Using Random Forests to Detect DDoS Attacks in Programmable Data Planes

Bruno Coelho, Alberto Schaeffer-Filho

Federal University of Rio Grande do Sul (UFRGS), Brazil
Context

- Distributed Denial of Service (DDoS) attacks remain an issue
- Even short downtime can result in losses
  - Amazon's 1 hour of downtime cost over $72 million on Prime Day 2018
- Detection is difficult
  - IP and Port Spoofing
  - Application-layer exploits
  - Accuracy vs Scalability
Motivation

- Programmable Data Planes (PDP)
  - Custom logic defined by software artifacts
  - Designed to process packets at line-rate

- Random Forests (RF)
  - Able to identify patterns to classify network traffic
  - Requires simple logic and arithmetic operations
  - Processing classification trees can be parallelized
  - Relatively compact data structures
Classification Tree Nodes

- **Internal Nodes**
  - Feature
  - Threshold value
  - Children
- **Node structures are naturally recursive**
  - A node contains another node (children)
- **P4 does not support recursion**
  - Cannot predict number of calls
- **Leaf Nodes**
  - Classification

Root Node R
- Feature: Packet Count
- Value: 7
- Children: A, B

Internal Node A
- Feature: Total Length
- Value: 114
- Children: C, D

Internal Node A
- Feature: Total Length
- Value: 114
- Children: C, D

Leaf Node J
- Classification: Malicious

Leaf Node O
- Classification: Legitimate
Mapping nodes to the Data Plane

**Root Node R**
- Feature: Packet Count
- Value: 7
- Children: A B

**Internal Node A**
- Feature: Total Length
- Value: 114
- Children: C D

**Internal Node B**
- Feature: Packet Count
- Value: > 7

**Leaf Node J**
- Classification: Malicious

**Leaf Node O**
- Classification: Legitimate

**Match Value**
- **Node ID**
- **Action**
- **Parameters**
  - **Threshold**
  - **Child 1**
  - **Child 2**

<table>
<thead>
<tr>
<th>Node ID</th>
<th>Action</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>compare_pkt_count</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>compare_total_length</td>
<td>114</td>
</tr>
<tr>
<td>2</td>
<td>compare_feature_B</td>
<td>y</td>
</tr>
<tr>
<td>8</td>
<td>compare_feature_H</td>
<td>z</td>
</tr>
</tbody>
</table>

**Match Value**
- **Node Identifier**
- **Action**
- **Parameters Classification**

<table>
<thead>
<tr>
<th>Node Identifier</th>
<th>Action</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>classify_flow</td>
<td>LEGITIMATE</td>
</tr>
<tr>
<td>9</td>
<td>classify_flow</td>
<td>LEGITIMATE</td>
</tr>
<tr>
<td>10</td>
<td>classify_flow</td>
<td>MALICIOUS</td>
</tr>
<tr>
<td>11</td>
<td>classify_flow</td>
<td>LEGITIMATE</td>
</tr>
<tr>
<td>12</td>
<td>classify_flow</td>
<td>MALICIOUS</td>
</tr>
<tr>
<td>13</td>
<td>classify_flow</td>
<td>MALICIOUS</td>
</tr>
</tbody>
</table>
BACKORDERS Architecture
Feature extraction in the Data Plane

- RFs require flow features as input
- Most statistical features are simple
  - Sum, max, min, duration
- Some statistical features require complex operations
  - Quantiles, means, variance
- We focused on approximating moving means (averages)
  - P4 does not support division
## Approximating Means

<table>
<thead>
<tr>
<th>$i$</th>
<th>$V_i$</th>
<th>$S_e(i)$</th>
<th>$S_a(i)$</th>
<th>$M_a(i)$</th>
<th>Mean</th>
<th>Formula: $S_a(i)$</th>
<th>Formula: $M_a(i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>15</td>
<td>160</td>
<td>160</td>
<td>20</td>
<td>20</td>
<td>$S_e(8)$</td>
<td>$S_e(8)/8$</td>
</tr>
</tbody>
</table>

\[
S_e(7) = 145 \quad V_8 = 15 \quad M_a(8) = \frac{160}{8} = 20
\]

\[
S_e(8) = S_a(8) = 145 + 15
\]
# Approximating Means

<table>
<thead>
<tr>
<th>$i$</th>
<th>$V_i$</th>
<th>$S_e(i)$</th>
<th>$S_a(i)$</th>
<th>$M_a(i)$</th>
<th>Mean</th>
<th>Formula: $S_a(i)$</th>
<th>Formula: $M_a(i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>15</td>
<td>160</td>
<td>160</td>
<td>20</td>
<td>20</td>
<td>$S_e(8)$</td>
<td>$S_e(8)/8$</td>
</tr>
<tr>
<td>9</td>
<td>25</td>
<td>185</td>
<td>165</td>
<td>20.625</td>
<td>20.5</td>
<td>$S_a(8) - M_a(8) + V_9$</td>
<td>$S_a(9)/\text{prev_pow2}(9)$</td>
</tr>
</tbody>
</table>

\[ V_9 = 25 \]

\[ S_a(9) = S_a(8) - M_a(8) + V_9 \]

\[ S_a(9) = 160 - 20 + 25 = 165 \]

\[ M_a(9) = \frac{S_a(9)}{\text{prev\_pow2}(9)} \]

\[ M_a(9) = \frac{165}{8} = 20.625 \]
Approximating Means

![Graph showing approximating means](image)
Evaluation - Dataset

- **CICIDS 2017 Dataset**
  - 692,703 flows
    - 440,031 legitimate (63.52%)
    - 5,796 DoS Slowloris
    - 5,499 DoS SlowHTTPTest
    - 231,073 DoS Hulk
    - 10,293 DoS GoldenEye
    - 11 Heartbleed
  - Binary division of classes
    - Legitimate
    - DoS (including all classes)
F1-Score for RF configurations
Conclusion

- **BACKORDERS**
- Classification of network flow in programmable data planes
  - Assisted by Machine Learning technique
- Maps nodes into match+action table entries
  - Sequential evaluation as opposed to recursive
- Extraction of features in the data plane
  - Approximation of means
- Proof-of-concept for utilizing ML in the data plane
  - Small forests with over 90% accuracy
Thank you for your time!

Bruno Coelho, Alberto Schaeffer-Filho
blcoelho@inf.ufrgs.br, alberto@inf.ufrgs.br

Federal University of Rio Grande do Sul (UFRGS), Brazil
## Approximating Means

<table>
<thead>
<tr>
<th>$i$</th>
<th>$V_i$</th>
<th>$S_e(i)$</th>
<th>$S_a(i)$</th>
<th>$M_a(i)$</th>
<th>Mean</th>
<th>Formula: $S_a(i)$</th>
<th>Formula: $M_a(i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>15</td>
<td>160</td>
<td>160</td>
<td>20</td>
<td>20</td>
<td>$S_e(8)$</td>
<td>$S_e(8)/8$</td>
</tr>
<tr>
<td>9</td>
<td>25</td>
<td>185</td>
<td>165</td>
<td>20.625</td>
<td>20.5</td>
<td>$S_a(8) - M_a(8) + V_9$</td>
<td>$S_a(9)/prev_pow2(9)$</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>195</td>
<td>154.375</td>
<td>19.29875</td>
<td>19.5</td>
<td>$S_a(9) - M_a(9) + V_{10}$</td>
<td>$S_a(10)/prev_pow2(10)$</td>
</tr>
</tbody>
</table>

\[
V_{10} = 10
\]

\[
S_a(10) = 165 - 20.625 + 10 = 154.375
\]

\[
M_a(10) = \frac{154.375}{8} = 19.296875
\]
Scalability Analysis

- Processing time is limited by maximum depth
  - $O(M)$ per tree
  - $O(NM)$ per forest
- Memory
  - Each node is mapped into a single match+action entry
  - Table entry number is limited by maximum depth
    - 1 layer = 1 node
    - 2 (full) layers = 3 nodes
    - 3 (full) layers = 7 nodes
  - $O(2^M)$ per tree
  - $O(N(2^M))$ per forest
Scalability Analysis

- Processing time is limited by maximum depth
  - $O(M)$ per tree
  - $O(NM)$ per forest

- Memory
  - Each node is mapped into a single match+action entry
  - Table entry number is limited by maximum depth
    - 1 layer = 1 node
    - 2 (full) layers = 3 nodes
    - 3 (full) layers = 7 nodes
  - $O(2^M)$ per tree
  - $O(N(2^M))$ per forest
## Scalability Analysis

<table>
<thead>
<tr>
<th># Trees</th>
<th>Max. Depth</th>
<th>Comparisons/tree</th>
<th>Total comparisons</th>
<th>Memory/tree</th>
<th>Total memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>63</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>127</td>
<td>127</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>5</td>
<td>15</td>
<td>31</td>
<td>93</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>6</td>
<td>18</td>
<td>63</td>
<td>189</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>7</td>
<td>21</td>
<td>127</td>
<td>381</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
<td>25</td>
<td>31</td>
<td>155</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>6</td>
<td>30</td>
<td>63</td>
<td>315</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>7</td>
<td>35</td>
<td>127</td>
<td>635</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>6</td>
<td>54</td>
<td>63</td>
<td>567</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>7</td>
<td>63</td>
<td>127</td>
<td>1143</td>
</tr>
</tbody>
</table>
Future Work

- Optimize memory utilized per feature
  - Current implementation may not scale for a high number of flows
- Include only the features that were selected by trees
  - Less memory utilization per flow
- Feature selection
  - Less registers
  - Lower depth