On Optimizing Load Balancing of Intrusion Detection and Prevention Systems

Anh Le, Ehab Al-Shaer, and Raouf Boutaba
Outline

1. Motivation
2. Approach Overview
3. Problem Formalization
4. Online Clustering Technique
5. Flow Correlation
6. Implementation
7. Evaluation
8. Conclusion
Motivation

- Gbps traffic requires the use of multiple NIDSs and NIPSs
- Static traffic distribution causes uneven load of the systems
- Distribution of traffic causes loss of correlation information
- Some detections fail (port scan, DDoS, etc.)

How to maintain load-balancing of the systems while minimizing the loss of correlation information?
Approach Overview

- **Clusters** capture correlations of flows and to provide structures to flows
  - i.e. Flows within a cluster have some correlation

- **Benefits** measure how much correlation information gained by assigning new flows to existing groups of flows
  - “I gain this much correlation if I assign this flow to this system”
Approach Overview, con’t

- Flows in systems are organized as clusters
  - A system has many clusters

- Desired load balancing level is specified as a variance constraint
  - i.e. load of the systems must be close

- When a flow comes:
  - Find candidate systems based on the variance constraint
  - Assign the flow to systems which give the best benefits
Problem Formalization

Maximize:
\[ \sum_{i=1}^{n} (L_i + L_f (\bar{X} \cdot \bar{G_i}) - (\mu + L_f \bar{X} \cdot \bar{I}_n)) \leq V \]

1) Maximize the total benefit

2) The new flow could be sent to at most F systems

3) For each system, the new flow could be sent to at most 1 cluster

4) Variance after the assignment must be less than the predefined variance V

Where:
\( \bar{X} \): Solution vector of size \( m \)
\( \bar{B} \): Benefit vector of size \( m \)
\( \bar{G_i} \): Cluster-ownership vector of size \( m \) of NIDPS \( i \)
\( \bar{I} \): Vector of 1’s of size \( m \)
\( F \): Maximum number of NIDPSs to assign \( f \)
\( L_i \): Load of NIDPS \( i \)
\( \mu \): Average load of all NIDPSs
\( L_f \): Predicted load of \( f \)
\( V \): Upper bound for the new variance
Problem Formalization, con’t

- Could favor security if needed:
  1. Relax variance constraint: increase \( V \)
     - i.e. “I sacrifice some load balancing for better benefit”
  2. Duplicate flows: increase \( F \)
     - i.e. “I have many resources, copy flows to send if needed for better benefit”
  3. Use threshold-based constraint: replace variance constraint
     - i.e. “Assign flows as long as load values of all systems are below a threshold”
Online Clustering Technique

- Real-time requirement

- Cluster has a weight between 0 and 1

- Decay of weight:
  - Weight of cluster decays overtime

- Adding a new flow:
  - Weight of a cluster changes based on how much the new added flow correlates with the centroid of the cluster
Listing 3 Benefit-based Load Balancing Algorithm

1. use k-Means to create n clusters
2. while there is a new flow $f$
3. \[ C = \text{solveP}(f) \]
4. \[ \text{if } C = \emptyset \]
5. \[ \text{if number of clusters} > m_{\text{max}} \]
6. \[ \text{delete clusters whose weights} < th_w \]
7. \[ \text{end if} \]
8. \[ \text{create a cluster (centroid } f, \text{ weight } 1) \]
9. \[ \text{assign it to lowest load NIDPS} \]
10. \[ \text{else} \]
11. \[ \text{assign } f \text{ to clusters in } C \]
12. \[ \text{update those clusters} \]
13. \[ \text{end if} \]
14. end while
Flow Correlation

- Basis to determine the benefit

- Distance between two flows:
  - The closer the two flows are, the more correlated they are
  - Weighted sum of logical distances between addresses and port numbers
    \[ D(f_1, f_2) = \sum_{i \in F} \alpha_i d_i(f_1, f_2) \]
  - Logical distances between addresses and port numbers are determined by their correlations
Flow Correlation, con’t

- By IP addresses:
  - **Identical correlation:**
    - Source IP addresses or destination IP addresses of two flows are the same
    - E.g. DDoS
  - **Subnet correlation:**
    - Destination IP addresses of two flows belong to the same subnets/vlan
    - E.g. Attack to a subnet
Flow Correlation, con’t

- **By port number**
  - **Identical correlation:**
    - Two flows have the same destination port number
      - E.g. DoS a webserver
  - **Functional correlation:**
    - Two destination port numbers are functionally related
      - E.g. Port 20 and 21
  - **Configuration correlation:**
    - A set of port numbers provided by administrators
      - E.g. Custom interest to group FTP and HTTP traffic together
Implementation

- **Load-balancer** with Benefit-based Load Balancing and Round Robin algorithm
  - Round Robin assigns flows to systems in a round robin manner
  - Libpcap to capture/send packet from/to NICs

- **DDoS detector** using CUSUM algorithm
  - CUSUM detects the change of the mean value of the percentage of the number of new source IP addresses overtime
  - If the accumulated change is bigger than a predefined threshold, an alert is raised
Evaluation

- To evaluate how BLB supports DDoS detection comparing to Round Robin
- Large scale UDP flood attack, single victim
- 3 settings:
  - Single CUSUM detector
  - 10 CUSUM detectors with BLB
  - 10 CUSUM detectors with Round Robin
Yn: the accumulated change overtime

- **Single detector:**
  - All packets go to 1 detector
  - Yn increases with the fastest rate

- **Round Robin:**
  - Packets are scattered, small change
  - Yn increases with a slow rate

- **BLB:**
  - Most of the packets go to the same detector, large change
  - Yn increases with a faster rate
Conclusion

- A novel Benefit-based Load Balancing algorithm, which thoroughly considers:
  - The load variation of NIDPSs
  - The loss of information due to flow distribution

- BLB distributes flows in real-time such that:
  - Correlated flows are grouped together
  - Load of the systems are maintained close within a desired bound

- BLB increases the detection accuracy of DDoS