SELF-LEARNING CLASSIFIER FOR INTERNET TRAFFIC

Luigi Grimaudo*, Marco Mellia*, Elena Baralis* and Ram KeralaPura^
INTERNET TRAFFIC CLASSIFICATION

• Provide network visibility for operators:
  • Deep Packet Inspection
    Look for specific tokens inside the packet payload
  • Does not work if encryption is in place

• Behavioral classifier

  Exploit some description of application behaviors by means of statistical characteristics (e.g., length of the first packets of a flow)
DISAVANTAGES

• Both require **training**
  • Define patterns to be matched
  • Define a training set
• Identify only application they have been trained for
• No new application or change in the application protocol or behavior
• Regular update or retraining
**EXPERIMENTAL ANALYSIS (1)**

- **Datasets**

<table>
<thead>
<tr>
<th>Name</th>
<th>Date Time</th>
<th>Place</th>
<th>Type</th>
<th>IP</th>
<th>Flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS-1</td>
<td>Aug05 1pm</td>
<td>South America</td>
<td>backbone</td>
<td>108k</td>
<td>527k</td>
</tr>
<tr>
<td>DS-2</td>
<td>Sep10 10am</td>
<td>Asia</td>
<td>backbone</td>
<td>111k</td>
<td>1.8M</td>
</tr>
<tr>
<td>DS-3</td>
<td>Aug11 2am</td>
<td>Europe</td>
<td>access</td>
<td>111k</td>
<td>885k</td>
</tr>
<tr>
<td>DS-4</td>
<td>Aug11 5pm</td>
<td>Europe</td>
<td>access</td>
<td>190k</td>
<td>2.3M</td>
</tr>
</tbody>
</table>

- Each dataset 1h long
- Split into client to server and server to client dataset (indicated as C and S in the following)
EXPERIMENTAL ANALYSIS (2)

• DPI as oracle:
  • Tstat [1]
  • NarusInsight

• 23 different protocols, like:

<table>
<thead>
<tr>
<th>HTTP/S</th>
<th>POP3/S</th>
<th>YAHOOIM</th>
<th>FASTTRACK</th>
<th>TELNET</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTSP</td>
<td>IMAP/S</td>
<td>BITTORRENT</td>
<td>ARES</td>
<td>IRC</td>
</tr>
<tr>
<td>TLS</td>
<td>XMPP</td>
<td>EMULE</td>
<td>SMB</td>
<td></td>
</tr>
<tr>
<td>SMTP/S</td>
<td>MSN</td>
<td>GNUTELLA</td>
<td>FTP</td>
<td></td>
</tr>
</tbody>
</table>

• Performance metrics:
  • Overall accuracy
  • Recall
  • Precision

SeLeCT
Self-Learning Classifier for Internet Traffic

• **Behavioral** classifier based on:
  • Simple *layer-4 metrics* (segments size, inter-arrival time)
  • Iterative clustering
  • Filtering phase
  • Adaptive/progressive learning

• Advantages:
  • **Few** and **very pure** clusters
    • Quick inspection
    • Easy manual labeling
THE SeLeCT APPROACH

INTERNET

FLOWS ➔ SeLeCT ➔ GOOD ➔ DEEP ANALYSIS

UNKNOWN
FEATURES

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAT_1</td>
<td>pay1_size</td>
</tr>
<tr>
<td>IAT_2</td>
<td>pay2_size</td>
</tr>
<tr>
<td>IAT_3</td>
<td>pay3_size</td>
</tr>
<tr>
<td>IAT_4</td>
<td>pay4_size</td>
</tr>
<tr>
<td>IAT_5</td>
<td>pay5_size</td>
</tr>
<tr>
<td>IAT_6</td>
<td>pay6_size</td>
</tr>
</tbody>
</table>

L4 DATA → CLUSTERING → FEATURES → FILTERING

GOOD CLUSTERS → READY FOR INSPECT

DISCARDED

UNCLUSTERED FLOWS
ITERATIVE CLUSTERING

- Work on **batches**: every 10000 flows or similar...
- Based on **k-means** clustering algorithm
  - Simple, well understood, and it works!
- Group flows into clusters possibly originated by the same application
- How to deal with Server Port?
  - It is know to carry valuable information [2]
- **KEY IDEA**: Use port number for **filtering**:
  - Dominated port protocol
  - Random port protocol
- Clustering and filtering phase alternate for a fixed number of iterations

---

FILTERING PROCEDURE

Clusters after k-means execution

For each cluster finds DOMINANT DESTINATION PORT

Create a new cluster for the dominant port of each cluster

Flows from original cluster going to dominant port

UNCLUSTERED FLOWS

ADD TO NEW CLUSTER

NO

YES

NO

YES
FINAL ITERATIVE STEP

N-1 STEPS

- Filtering procedure
- Collect all dominant destination ports
- Port Filtering
- UNCLUSTERED FLOWS
- Remove flows going to the collected ports
- CLUSTERING
- DISCARD SMALL CLUSTERS
- OUTLIERS

FINAL CLUSTERS

Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5
Cluster 6
Cluster 7
Cluster 8
Cluster 9
Cluster n

FINAL STEP
ITERATIVE CLUSTERING PERFORMANCE
SeLeCT vs k-means [3]

Overall accuracy on average ~ 98%

Interestingly – Select does **BETTER** than the ground truth
- Fooled by non-English welcome messages of SMTP server
• Clusters from iterative clustering:
  • Ready to be inspected by operators
  • Clusters are labeled by means of seeding flows and voting scheme
  • Otherwise clusters are labeled as “unknown”
• **Bootstrapping:**
  - Labels can be assigned to each “unknown” cluster by operator using domain knowledge
  - Turns out to be **trivial** for most of port-dominated clusters
    - Cluster of flows all going to port 80 -> HTTP 😊
    - Cluster of flows all going to login.skype.com -> skype login protocol 😊
### INTERESTING FINDINGS (2)

<table>
<thead>
<tr>
<th></th>
<th>BT</th>
<th>HTTP</th>
<th>HTTPS</th>
<th>MSN</th>
<th>POP3</th>
<th>POP3S</th>
<th>SMB</th>
<th>SMTP</th>
<th>SSH</th>
<th>Telnet</th>
<th>UNK</th>
<th>XMPP</th>
</tr>
</thead>
<tbody>
<tr>
<td>BT</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>HTTP</td>
<td>0</td>
<td>5769</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>HTTPS</td>
<td>0</td>
<td>0</td>
<td>530</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MSN</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>POP3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>42</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>POP3S</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>46</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMB</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMTP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2217</td>
<td>0</td>
<td>0</td>
<td>102</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SSH</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Telnet</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>43</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>UNK</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>83</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>XMPP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
</tbody>
</table>

Pure Clusters (not identified by free and commercial DPIs) for:
- Apple push notification over TLS
- Microsoft Media Server (MMS)
- ...
SELF-SEEDING

• SeLeCT is able to automatically **reuse the knowledge** from previous batches
• Seeding flows from labeled clusters are extracted
  • Stratified sampling technique
• Seeding flows are used to label flows of the next batch

• Self training process:
  • System grows the set of labeled data
  • Augment the coverage of classification process
ITERATIVE CLUSTERING PERFORMANCE
Accuracy over different batches

- For server – 99% accuracy
- Server features are stronger than client features
- For client – 90% accuracy
- Only P2P traffic is “confused”
SeMule clusters are used at bootstrap.
HOW FAST IS SeLeCT TO DETECT NEW CLASS?

HTTPS traffic added at batch 3 and POP3 traffic added at batch 6
CONCLUSION

• SeLeCT:
  • Semi-automated Internet flow traffic classifier
  • Based on clustering and filtering
  • Adapts the model to traffic changes
  • Able to increase its knowledge
  • Extremely good performance

• Based on simple behavioral features
  • Can be extended to include other features
Thanks for attention!

Questions?

Luigi Grimaudo – luigi.grimaudo@polito.it