Detecting Malicious Hosts Using Traffic Flows

Miguel Pupo Correia
joint work with Luís Sacramento
NavTalks, Lisboa, June 2017

Outline

• Motivation
• Approach
• Evaluation
• Conclusion
Outline

- Motivation
- Approach
- Evaluation
- Conclusion

Motivation

- Scenario:
  - Large national telco/ISP connected to its own provider
  - Huge amount of traffic in/out, much is encrypted
  - Possibly new attacks / new variants
Motivation

- Compromised hosts do attacks such as:
  - Distributed denial of service attacks
  - Exfiltrating confidential data
  - Sending spam
  - Mapping the network
  - Contact bot command&control centers
  - etc.

Network Intrusion Detection Systems

- Traditional NIDSs:
- **Knowledge-based**: require signatures of attacks
  - Not good for new attacks
- **Behavior-based**: require clean traffic for training
  - Where to get it with our scenario?
- Most do **deep packet inspection**, unfeasible with too much traffic
Outline

• Motivation
• Approach
• Evaluation
• Conclusion

Our approach

• Detection framework to detect malicious hosts based on real traffic
• Not knowledge-based, to avoid need for signatures
• Not behavior-based, as no training traffic exists
• No deep packet inspection, as it is slow
• Detects hots doing new attacks or new variants
Key ideas

• Collect traffic data summarized as network flows
• Extract data about hosts from flows
• Use unsupervised machine learning / clustering to
  – get information that humans can understand without
    previous knowledge about attacks
• Use supervised machine learning / classifier to
  automatically assign clusters to classes/categories
  – ex: web servers, mail servers, hosts sending spam, hosts
doing distributed denial of service,…
• Manually label new clusters

The approach

• Loop:
  – Collect flows for a period of time (e.g., 1 day)
  – Extract from the flows data about hosts with MapReduce
  – Use clustering to create groups of hosts
  – Use classifier to automatically classify hosts
  – Manually label remaining ones
  – Repeat for next period
The approach

• Loop:
  – Collect flows for a period of time (e.g., 1 day)

Flows

• Flow: sequence of related packets observed during an interval of time
  – A flow is defined in terms of a subset of src IP, dest IP, protocol, src port, dest port; ex: (*, 1.2.3.4, TCP, *, 80)
• Netflow: monitoring approach created by Cisco
  – Idea is to capture data about network flows
  – Data: begin/end of flow timestamps, n. packets, n. bytes
  – Variants: IPFIX (standard based on Netflow 9), sFlow,...
Flow collection

- Flows collected on NetFlow-enabled border routers

The approach

- Loop:
  - Collect flows for a period of time (e.g., 1 day)
  - Extract from the flows data about hosts with MapReduce
Host data extraction

- Flow format:
  `<Source IP, Destination IP, Source Port, Destination Port, Protocol, TCP Flags, #Bytes, #Packets, Duration>`
- Use MapReduce for extracting data per host (IP)
  – aggregated by source or destination IP address
The approach

- Loop:
  - Collect flows for a period of time (e.g., 1 day)
  - Extract from the flows data about hosts with MapReduce
  - Use clustering to create groups of hosts

Unsupervised ML / clustering

- Idea: group similar hosts in clusters (sets)
- Why? Humans can understand and classify a few clusters, not zillions of hosts
- How?
  -Normalize every feature into range [0,1]
  -Run clustering algorithm, e.g., K-Means, to get k clusters
  -k can be defined, e.g., with the elbow method
    (finds the “elbow”, i.e., when adding more clusters does not improve the modelling of the data)
The approach

• Loop:
  – Collect flows for a period of time (e.g., 1 day)
  – Extract from the flows data about hosts with MapReduce
  – Use clustering to create groups of hosts
  – Use classifier to automatically classify hosts
  – Manually label remaining ones

Intrusion detection with flows

• Each cluster contains hosts with similar behavior
  – ex: web servers, mail servers, hosts sending spam, hosts doing denial of service,…
• What to do with them? (at cruise speed)
• Already seen? Use classifier to classify automatically
• Never seen?
  – Label manually, with help of the features’ values
  – Focus attention on smaller clusters with odd feature distribution; often malicious
  – Retrain classifier
Supervised ML / classification

• Naïve solution: use labelled hosts to train a Binary Support Vector Machine (SVM) classifier
  – Samples/hosts classified as benign or malicious
  – Finds an hyperplane that separates samples
  – Classifies new samples (hosts)

The approach

• Loop:
  – Collect flows for a period of time (e.g., 1 day)
  – Extract from the flows data about hosts with MapReduce
  – Use clustering to create groups of hosts
  – Use classifier to automatically classify hosts
  – Manually label remaining ones
  – Repeat for next period
Outline

- Motivation
- Approach
- Evaluation
- Conclusion

Tool – interface
Tool – interactive visualiz. of cluster

Evaluation

- Two parts:
  - Synthetic dataset (ISCX)
    - Designed for IDSs
    - Flows are labelled
    - Allows validating the approach
  - Real dataset collected at the telco
    - No ground truth
ISCX dataset evaluation

- Brute-Force SSH attack found during this day (cluster 3)
  - Maximum for SSH connections (and high, not seen in table)

Telco dataset evaluation

Cluster data with source aggregation key (i.e., aggregated by IP inside the ISP) cluster data – 1st part
Telco dataset evaluation

<table>
<thead>
<tr>
<th>Cluster #</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
<th>25</th>
<th>26</th>
<th>27</th>
<th>28</th>
<th>29</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td># Hosts</td>
<td>714</td>
<td>3</td>
<td>4406</td>
<td>1864</td>
<td>1826</td>
<td>12</td>
<td>557</td>
<td>13</td>
<td>2293</td>
<td>2253</td>
<td>8091</td>
<td>10</td>
<td>13897</td>
<td>14943</td>
<td>30</td>
<td></td>
</tr>
</tbody>
</table>

Features

1. 0.367
2. 0.367
3. 0.367
4. 0.367
5. 0.346
6. 0.237
7. 0.158
8. 0.346
9. 0.346
10. 0.346
11. 0.346
12. 0.346
13. 0.346
14. 0.346
15. 0.346
16. 0.346
17. 0.346

Cluster data with source aggregation key – 2nd part

Source aggregation key – cluster 15

i.e., aggregated by IP inside the ISP

<table>
<thead>
<tr>
<th>Cluster #</th>
<th>15</th>
<th>Cluster #</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td># Hosts</td>
<td>1</td>
<td># Hosts</td>
<td>1</td>
</tr>
<tr>
<td>Features</td>
<td></td>
<td>Features</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1.0</td>
<td>9</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>10</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>1.0</td>
<td>11</td>
<td>1.0</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>12</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>0.667</td>
<td>13</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>14</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>15</td>
<td>0.843</td>
</tr>
<tr>
<td>8</td>
<td>0.843</td>
<td>16</td>
<td>-</td>
</tr>
</tbody>
</table>

- Spammer or denial of service (?)
  - High connectivity to various users, many ports, receiving communication on IRC port, communication through HTTP, high number of packets sent, high number of bytes
Source aggregation - Cluster 21

<table>
<thead>
<tr>
<th>Cluster #</th>
<th>Hosts</th>
<th>Features</th>
<th>Cluster #</th>
<th>Hosts</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9</td>
<td>0.541, 0.181</td>
<td>2</td>
<td>10</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>-</td>
<td>3</td>
<td>11</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>-</td>
<td>5</td>
<td>13</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>14</td>
<td>-</td>
<td>7</td>
<td>15</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>0.261, 0.026</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Bot communicating with C&C server
  - High IRC communication + high average packet size
  - Confirmed by accessing the IP of the C&C server

Telco dataset evaluation summary

<table>
<thead>
<tr>
<th>Cluster #</th>
<th>Aggregation Key</th>
<th>Highlighted Features</th>
<th>Type of Attack</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>Source</td>
<td>1, 3, 5, 8, 11, 15</td>
<td>Spam / DoS</td>
</tr>
<tr>
<td>16</td>
<td>Destination</td>
<td>1, 3, 6</td>
<td>DoS</td>
</tr>
<tr>
<td>17</td>
<td>Source</td>
<td>10</td>
<td>Brute-Force SSH</td>
</tr>
<tr>
<td>20</td>
<td>Destination</td>
<td>1, 2, 15</td>
<td>Network Scan</td>
</tr>
<tr>
<td>21</td>
<td>Source</td>
<td>9, 16</td>
<td>Botnet Communication</td>
</tr>
<tr>
<td>22</td>
<td>Destination</td>
<td>1, 3, 8, 15</td>
<td>Web Application Probing</td>
</tr>
<tr>
<td>27</td>
<td>Source</td>
<td>1, 2, 5, 8, 11, 15</td>
<td>DDoS IRC Botnet</td>
</tr>
<tr>
<td>29</td>
<td>Destination</td>
<td>1, 2, 4, 11, 15</td>
<td>DDoS Botnet</td>
</tr>
</tbody>
</table>
Outline

• Motivation
• Approach
• Evaluation
• Conclusion

Conclusion

• Network Intrusion Detection for identifying malicious hosts using flows
• ...without having to say how entities misbehave
• Use clustering (unsupervised ML) to reduce the size of the problem and
• a classifier (supervised ML) to automatize classification
• Keep humans in the loop; mandatory w/evolving threats
• Detects attacks involving many packets, not low traffic attacks like buffer overflows or SQL injection
Thanks! Questions?

This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013 (INESC-ID)