FlowHacker – An Approach for Detecting Unknown Network Attacks With Flows
Miguel Pupo Correia
joint work with Luís Sacramento
European Security Conference, Lisboa, June 2017

Outline

• Motivation
• Approach
• Evaluation
• Conclusion
Outline

- Motivation
- Approach
- Evaluation
- Conclusion

Motivation

- Scenario:
  - Large national telco: mobile commun., Internet, TV,…
  - Connected to its own provider
  - Huge amount of traffic in/out, much is encrypted
  - Possibly new attacks / new variants

![Diagram of network connections between customers, national ISPs, and international ISPs.](image)
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

FlowHacker approach

- Detection framework to detect malicious hosts based on network 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 hosts 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, hosts doing distributed denial of service,…
• Manually label new clusters

FlowHacker 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 clusters
  – Repeat for next period
FlowHacker 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

FlowHacker 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

```
Distributed File System

Input Data

Mapper

Reducer

Mapper

Reducer

Mapper

Reducer

Output
```

Host data extraction

- Host features (data) extracted by MapReduce:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregation Key</td>
<td>The IP address that will be used as an identifier, to which the below features relate to</td>
</tr>
<tr>
<td>NumSIPs / NumDIPs</td>
<td>The number of IP addresses contacted</td>
</tr>
<tr>
<td>NumSPorts</td>
<td>The number of different source ports contacted</td>
</tr>
<tr>
<td>NumDPorts</td>
<td>The number of different destination ports contacted</td>
</tr>
<tr>
<td>tx2DNumHTTP</td>
<td>The number of packets to/from port 80 (HTTP)</td>
</tr>
<tr>
<td>NumIRC</td>
<td>The number of packets to/from ports 194 or 6667 (IRC)</td>
</tr>
<tr>
<td>NumSMTP</td>
<td>The number of packets to/from port 25 (SMTP)</td>
</tr>
<tr>
<td>NumSSH</td>
<td>The number of packets to/from port 22 (SSH)</td>
</tr>
<tr>
<td>TotalNumPkts</td>
<td>The total number of packets exchanged</td>
</tr>
<tr>
<td>PktRate</td>
<td>The ratio of the number of packets sent and its duration</td>
</tr>
<tr>
<td>ICMPRate</td>
<td>The ratio of ICMP packets, and total number of packets</td>
</tr>
<tr>
<td>SyntRate</td>
<td>The ratio of packets with a SYN flag and the total number of packets</td>
</tr>
<tr>
<td>TotalNumBytes</td>
<td>The overall sum of bytes</td>
</tr>
<tr>
<td>AvgPktSize</td>
<td>The average packet size</td>
</tr>
<tr>
<td>BadSubnet</td>
<td>This field expresses whether the IP address belongs to a blacklisted subnet</td>
</tr>
<tr>
<td>MaliciousIP</td>
<td>This field expresses whether the IP address is blacklisted</td>
</tr>
<tr>
<td>OpenVaultBlacklistedIP</td>
<td>Same as the above, but checked from a trusted and well know threat database</td>
</tr>
<tr>
<td>MaliciousASN</td>
<td>This field shows if the IP address belongs to a blacklisted ASN</td>
</tr>
<tr>
<td>LocationCode</td>
<td>Code for the country associated with the address</td>
</tr>
</tbody>
</table>

Extracted from the flow directly

Based on threat intelligence
FlowHacker 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)
FlowHacker 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 clusters

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)

FlowHacker 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 clusters
  - Repeat for next period
Outline

- Motivation
- Approach
- Evaluation
- Conclusion

FlowHacker 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 telco) cluster data – 1st part
Telco dataset evaluation

Cluster data with source aggregation key – 2nd part

Source aggregation key – cluster 15
i.e., host(s) in the telco’s network

- 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>21</td>
<td>12</td>
<td></td>
<td>21</td>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>

High IRC communication

High average packet size

- Bot communicating with C&C server
  - 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, 6, 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

• FlowHacker: 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?

Daniel Gonçalves, João Bota, Miguel Correia. Big Data Analytics for Detecting Host Misbehavior in Large Logs. TrustCom 2015

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