Cyber Network Data Processing

Vijay Gadepally, Siddharth Samsi, Jeremy Kepner, Emily Do*

*Graduated, Fall 2019

Overall System Goal

- **Goal**: Detect and classify network attacks from real internet traffic

- **Dataset**:
  - Measurement and Analysis of Wide-area Internet (MAWI) working group
  - 2x48=96 hours of 1 Gigabit packet capture (PCAP) headers collected in Tokyo
  - 0.7 TB compressed; 20 TB in analysts friendly form
    - Normalized, sorted, indexed, and read optimized
  - IP addressed deterministically anonymized within each collect
    - Network analysis is still valid
Anomaly Detection

- “An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism” *
  - Outlier is sometimes referred to as anomaly, surprise, exception, ...
  - Within the context of cyber networks, these “mechanisms” can be botnets, C&C servers, insider threats or other attacks such as DDOS & Port Scan attacks, ...

![Diagram of anomalies in feature space]

Given the complexity of network traffic, we use a model based technique

---


---

Anomalies in Cyber Networks

Cyber Network Attacks

- **Probing and Scanning**
  - Port Scanning
  - SMB Scan
  - Other scanning (MS17, ...)

- **Resource Usage**
  - Network exhaustion
    - Denial of Service
  - Memory exhaustion
  - SQL Injection

- **Exploitation**
  - Trojan Horse
  - Malware

- **Command and Control**
  - Botnets
  - Ransomware

Example: Probing & Scanning

- SYN
- SYN/ACK
- RST

PORT IS OPEN

Figure Source: https://resources.infosecinstitute.com/port-scanners/

© Satyam Singh. All rights reserved. This content is excluded from our Creative Commons license. For more information, see https://ocw.mit.edu/help/faq-fair-use/
The Network Packet

**Headers converted to human readable form**

1. (1.201704132329.pcap..A.mat,,frame.time_relative|0.000000000) 1
2. (1.201704132329.pcap..A.mat,,frame.time|2017 Apr 13 10:29:59.938632000 EDT) 1
3. (1.201704132329.pcap..A.mat,,ip.dst|17.114.183.195) 1
4. (1.201704132329.pcap..A.mat,,ip.len|52) 1
5. (1.201704132329.pcap..A.mat,,ip.proto|6) 1
6. (1.201704132329.pcap..A.mat,,ip.src|163.35.157.212) 1
7. (1.201704132329.pcap..A.mat,,tcp.dstport|80) 1
8. (1.201704132329.pcap..A.mat,,tcp.flags|0x00000010) 1
9. (1.201704132329.pcap..A.mat,,tcp.srcport|47438) 1

Network packets are relatively easy to collect and often form the lowest common denominator across cyber network processing pipelines.

**Data: Public Internet Packet Capture (PCAP)**

- Measurement and Analysis of Wide-area Internet (MAWI) working group
  - 2x48=96 hours of 1 Gigabit packet capture (PCAP) headers collected in Tokyo
  - 0.7 TB compressed; 20 TB in analysts friendly form
    - Normalized, sorted, indexed, and read optimized
  - IP addressed deterministically anonymized within each collect
    - Network analysis is still valid
Labelled Data – Using Synthetic Attacks

Synthetic Attack Generation

• One of the major challenges associated with cyber network analysis is the lack of data with known attacks
• ID2T is a toolkit developed at TU Darmstadt that allows users to inject synthetic attacks directly into PCAP data
  – Still difficult to use but it is one of the best (open source) tools we have come across
  – Supports a number of attacks
• For our pipeline:
  – We use ID2T to generate a series of synthetic attacks that are embedded into PCAP data
  – We focus on DDOS and Port Scan attacks

ID2T: a DIY Intrusion Detection Dataset Creation Toolkit

Source: Copyright © 2017: Emmanouil Vasilomanolakis, Carlos Garcia, Emmanouil MIT License

Pipeline for Cyber Network Anomaly Detection

Data Conditioning

Machine Learning

Classify using ML
Data Conditioning (1)

Raw Data

- Packet Capture data is typically generated from a network traffic analyzer or a utility such as tcpdump

- For our pipeline:
  - Data is downloaded from the MAWI Lab website in .tar.gz
  - Data is uncompressed into the .pcap files
  - Typical size of a single .tar.gz: 2.5GB
  - Typical size of uncompressed .pcap: 10 GB
  - A single .pcap file corresponds to 900 seconds (15 minutes)
  - Corresponds ~150,000 packets/second

Data Conditioning (2)

Flow Extraction

- A network flow is defined as a sequence of packets from a source to a destination.
- RFC 2722 describes flows as “an artificial logical equivalent to a call or connection”

- For our pipeline:
  - We convert the 15 minute .pcap files into network flow representation using YAF (yet another flowmeter)
  - .yaf output is a binary format
  - Size of 15 minutes worth of flows: 2GB
  - Defined between a source ip, destination ip with the same port. Flow timeout is set to a default 5 min

Image Source MAWILAB: http://www.fukuda-lab.org/mawilab/data.html

Parse Flows

- Machine learning models require conversion of binary flow format into some tabular form.
- YAF comes with a tool: yafscii to convert binary flows into a human readable form.

For our pipeline:
- We convert each of the .yaf files into a .txt file using yafscii.
- Typical size of this ascii table is: 8GB.
- Each line of the output text file corresponds to a single flow.
- Following fields are recorded for each flow:

  - start-time|end-time|duration|rtt|proto|sip|sp|dip|dp|iflags|uflags|riflags|ruflags|isn|risn|tag|rtag|pkt|oct|rpkt|roct|end-reason

Yaf: [https://tools.netsa.cert.org/yaf/yaf.html](https://tools.netsa.cert.org/yaf/yaf.html)

Tabular Flow Fields

<table>
<thead>
<tr>
<th>Features of Interest</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source IP</td>
<td>Source IP address</td>
</tr>
<tr>
<td>Source Port</td>
<td>Source port</td>
</tr>
<tr>
<td>Destination IP</td>
<td>Destination IP address</td>
</tr>
<tr>
<td>Destination Port</td>
<td>Destination port</td>
</tr>
<tr>
<td>Protocol</td>
<td>IP protocol</td>
</tr>
<tr>
<td>Initial Flags</td>
<td>Forward first-packet TCP flags</td>
</tr>
<tr>
<td>Union Flags</td>
<td>Forward n-th-packet TCP flags union</td>
</tr>
<tr>
<td>Reverse Initial Flags</td>
<td>Reverse first-packet TCP flags</td>
</tr>
<tr>
<td>Reverse Union Flags</td>
<td>Reverse n-th-packet TCP flags union</td>
</tr>
<tr>
<td>End reason</td>
<td>Indicate whether the flow was ended normally (i.e., by TCP RST or FIN), expired by idle timeout, or expired by active timeout.</td>
</tr>
<tr>
<td>Destination IP</td>
<td>Combination of Destination IP and Destination Port</td>
</tr>
<tr>
<td>Destination Port</td>
<td>Combination of Destination IP and Initial Flags</td>
</tr>
<tr>
<td>Source IP</td>
<td>Combination of Source IP and Destination IP</td>
</tr>
<tr>
<td>Source IP Initial Flags</td>
<td>Combination of Source IP and Initial Flags</td>
</tr>
</tbody>
</table>
Data Conditioning
Feature Engineering (1)

- Each flow contains 21 features such as IP addresses, ports, ...
- Many of these are either unchanging or unlikely to help us look for anomalous behavior
- We used domain knowledge, trial-and-error and luck to pick features

Data Conditioning
Feature Engineering (2)

Using Entropy

- Entropy is a measure of uncertainty associated with a random variable
- Used extensively in information theory
- Typically measured using Shannon Entropy:
  \[ H = - \sum p_i \log p_i \]

- For our pipeline:
  - We compute the Shannon entropy associated with each of our features: \( H_{\text{src.ip}}, H_{\text{dst.ip}}, \ldots \)

Intuition is that the overall entropy of the system should stay reasonably constant without external "mechanisms"
### Machine Learning - Deep Neural Networks -

\[ y_{i+1} = f(W_i y_i + b_i) \]

Equation relating inputs and outputs

---

### Machine Learning Model - Training -

- Our model is a fully connected feed-forward network that takes in the 14 entropies as input features and attempts to classify them as one of 5 different classes
  - Input layer, 3 hidden layers (100, 30, and 100 nodes), and an output layer
  - ReLU activation
- Five output classes: No-Attack, DDoS-Attack, Port-Scan-Attack, Point2Point-DoS-Attack, and Network-Scan-Attack
Evaluation

On going research!

Summary

• Good results on detecting and classifying network attacks from internet backbone traffic

• Key notes:
  – Data conditioning consisted of a number of steps:
    • Cleaning up collected data
    • Generating "labeled" data using a synthetic attack generator
    • Feature engineering to determine which features and form of the features were likely to get the best results