**Introduction**

This project presents an active learning model which attempts to block early-stage cyber attacks.
- Cyber-attacks are a set of discrete, observable steps called a ‘kill chain’.
- Data produced by the security stack from early kill chain steps can be used to automate defensive decisions.
- A machine makes low-level decisions and human analysts only see signals elevated with sufficient importance.
- A successful early-step decision avoids more severe downstream consequences by disrupting attacks at the beginning of the kill chain.
- Goal: Use deep learning to develop a classification system to differentiate legitimate network traffic from bad behavior.

**Background**

The agent must decide which gate to keep open and which to keep closed. With every kill chain stage, there is a set of observable features (activity behavior) that can be used as gates. In practice, the set of blocks are referred to as the ‘Access Control List’ and deployed through firewalls. Deep Learning is an excellent candidate to explore what is the best gate policy.

Possible features (gates) include:
- IP/MAC Address
- Port
- Geo-Location
- Packet Payload
- Temporal / Diurnal
- Protocol
- Process Application

Example: Block activity (close gate) on inbound port 80 and 443 from Nigeria

**Network Architecture Alternatives**

There is no standard for creating cyber threat detection networks, so this project explores from first principles Early models are fully-connected feed-forward network with 3 hidden layers. Later iterations expanded the network to 11 hidden layers in order to reduce bias. Similarly, we used dropout as a regularization method to reduce variance. We tested these strategies independently and as a combination to determine the overall effect.

**Data:** Self Generated
- Honeywell’s deployed on 4 cloud providers on every continent totaling 600,000 network events.
- Threats versus non-threat labels created using open-source blacklists of IPs.

**Features:**
- Time Encoding
- Port Encoding
- IP Encoding
- Average Hour
- Standard Dev Hour
- Frequency
- Latitude
- Longitude
- Regular / Irregular
- Source
- Destination
- Number unique
- Database
- Standard Network
- Other

**Methods**

**Features: Managing the Scale**
- One-hot vectors transformed into encodings

**Results**

Costs improve over epochs but level out quickly with marginal improvement.
- **Learning Rate**: 0.001
- **Epochs**: 1500
- **Minibatch Size**: 32

**Network Architecture Alternatives**

<table>
<thead>
<tr>
<th>Training</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>540,000</td>
<td>30,000</td>
<td>30,000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic Regression</td>
<td>0.59</td>
<td>0.40</td>
<td>0.59</td>
</tr>
<tr>
<td>3 Layers</td>
<td>0.64</td>
<td>0.44</td>
<td>0.64</td>
</tr>
<tr>
<td>Dropout</td>
<td>0.63</td>
<td>0.42</td>
<td>0.63</td>
</tr>
<tr>
<td>11 layers</td>
<td>0.64</td>
<td>0.47</td>
<td>0.62</td>
</tr>
<tr>
<td>11 Layer w/ Dropout</td>
<td>0.64</td>
<td>0.45</td>
<td>0.64</td>
</tr>
</tbody>
</table>

*Best Performance of Tested Values

**Conclusions/Future Work**

- The initial performance of these models against the collected data is not strong. However, these appear to be some promising early signs.
- Individual events (log data from sensors) is very noisy and may not contain enough information without aggregating.
- Expansion of labeling to other known threatening behaviors outside blacklists.
- The most important improvement this project and use is more and better quality data. Collecting raw internet data introduces some bias in the data that might cause the performance degradation.

Stanford University