DeepSecurity

Cybersecurity Threat Behavior Classification

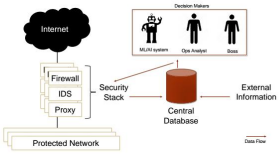
Isaac Faber, Giovanni Malloy

Introduction

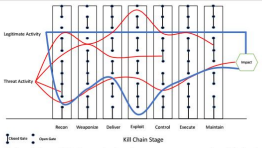
This project presents an active learning model which attempts to block early-stage cyber attacks.

- 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

Typical Cyber Posture



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 is deployed through the firewall. 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

Methods

Data: Self Generated

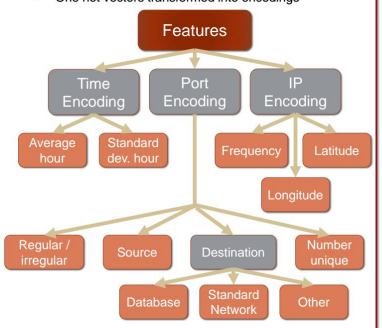
- Honeypot's deployed on 4 cloud providers on every continent totaling 600,000 network events:
- Threats versus non-threat labels created using open-source blacklist of IPs: criticalstack

Observed Threats by Country

Train	Dev	Test
540,000	30,000	30,000

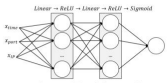
Features: Managing the Scale

- One hot vectors transformed into encodings

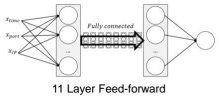


Network Architecture Alternatives

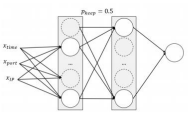
There is no standard for creating cyber threat detection networks so this project explores from first principals Early models are fully-connected feed-forward network with 3 hidden layers. Later iterations expanded the network to 11 hidden layers in order to reduce bias. Similarly, we used dropout as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



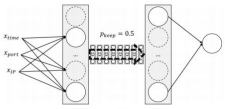
3 Layer Feed-forward



11 Layer Feed-forward



3 Layer Dropout



11 Layer Dropout

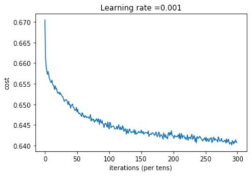
Loss function: $\mathcal{L}\{y, \hat{y}\} = -[y \log \hat{y} + (1-y) \log(1-\hat{y})]$

Optimization Method: AdamOptimizer

Results

Costs improve over epochs but level out quickly with marginal improvement.

- **Learning Rate*:** 0.001
- **Epochs*:** 1500
- **Minibatch Size*:** 32



11 Layer Feed-forward (best performance)

	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.59	0.40	0.59	0.46
3 Layers	0.64	0.44	0.64	0.51
Dropout	0.63	0.42	0.63	0.49
11 layers	0.64	0.47	0.62	0.53
11 Layer w/ Dropout	0.64	0.45	0.64	0.53

*Best Performance of Tested Values

Conclusions/Future Work

- The initial performance of these models against the collected data is not strong. However, there does 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 blacklist
- 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 be causing some of the performance degradation.