DEEP HONEYNET ANALYSIS

Napoleon Cornel Paxton (ncpaxton@stanford.edu)
Berkay Polat (berkayp@stanford.edu)

Stanford University

Predicting

An overwhelming majority of cyberattacks follow a sequence beginning with reconnaissance activities and ending with a total infiltration of a targeted system. Each step is recorded in network traffic that returns a level of information the attacker can use in order to be successful in the next step. In this project we explore using Recurrent Neural Networks and input data from an IoT and Web Application honeypot to train a model that can learn from the observed steps attackers are using. Our results were promising, but more testing is needed to prove the validity of our approach. Successful implementation of this approach can lead to a more proactive cyber defense, which would significantly improve the security posture of a protected network.

Features

Our project has 36 features and each of them are raw. We believe these features are appropriate for this task because they represent features which can be detected on any computer network and we were able to fit our model based on them.
Using the timestamp of packets, we managed to characterize packets based on the minute and the second of their arrival. Furthermore, among these 36 features, 2 of them were classified as categorical:

Categorical:

- IP Address
- TLL

For TLL (Time-to-Live), one-hot vectorization was applied. IP addresses were presented as 32-bit numbers for model to retain each IP's meaning.

For example:

172.16.254.1 ➡
10101100.00010000.11111110.00000001

Standard feature scaling was applied before rest of the data was fed into our models.

Results

Our first approach was to feed in all the network packets after sorting them out by their timestamp in order to capture the relative time difference and the order of each packet. However, the resulting algorithms were not able to capture the distinction among 3 type of traffic. This is due to mainly the heterogeneous time difference when each packet was captured. The resulting loss values over 100 epoch can be seen in Figure 1.

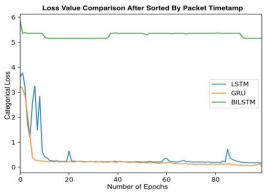


FIGURE 1

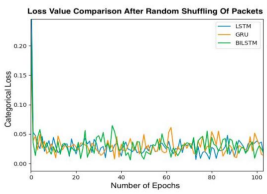


FIGURE 2

On our second approach, we experimented with randomly shuffling all the network packets in order to overcome the high bias originated from heterogeneous time captures and highly unproportional normal traffic to web/Iot traffic ratio. We obtained the resulting loss value characteristics in Figure 2 after running our models for 200 epochs. All 3 models still had a hard time trying to capture the complexity but it overcame the Bi LSTM anomaly.

MODEL TYPE	ACCURACY ON THE TEST SET
LSTM	%52
GRU	%52
BILSTM	%60

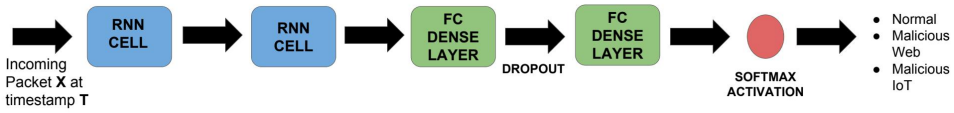
Future

Training a larger model to reduce bias is our next step. Creating a NET2VEC embedded matrix for computer networks is also an interesting idea that would benefit this project. Providing context to fields like the nocturnal patterns of detected IP addresses could help to determine user patterns and could improve the accuracy of the classifications. More sophisticated honeypots that would be able to capture more homogeneous network traffic could also help to develop the idea of using this type of method as a protection mechanism.

Data

Data for this project was generated using network traffic connecting to two honeypots and a normal host. Each column of data corresponds to a feature captured from the traffic, and each row is a separate example of a communication. All examples are labeled based on where the data was captured.

Architecture Overview:



Models

Our main goal was to classify incoming network packets based on their origins. Mainly, we wanted to classify if a packet was from normal, web (malicious), or IoT (malicious) traffic. We decided to use and compare 3 different Recurrent Network architectures for our classification. In order to compare these 3 models on the same benchmark, the architecture followed a similar pattern for each model. Replacing the "RNN CELL" with each of the 3 cells, we ran our models for 200 epochs.

1. LSTM CELL
2. GRU CELL
3. Bi-DIRECTIONAL LSTM CELL

Discussion

This project was very interesting and challenging. We generated our own data, and we had some initial difficulty operating the data generating tools and determining how much data to generate. After running the baseline model, we realized we needed to generate more data, which really improved our results. Initially we also discussed creating an embedded matrix of some of our features, but due to time decided to use one hot encoding and 32-bit encoding for our features. Overall, we believe this is a good approach to predict traffic from attackers, but more work needs to be done in terms of capturing a more meaningful relationship on packets and possibly building more complex RNN architectures to understand the complexity of multiple network features.

References

- Babu, Mohammed & R, Vinayakumar & Kp, Soman. (2018). RNNSecureNet: Recurrent neural networks for Cybersecurity use-cases. 10.13140/RG.2.2.21876.81283.
- Abien Fred M. Agarap (2019). A Neural Network Architecture Combining Gated Recurrent Unit (GRU) and Support Vector Machines (SVM) for Intrustion Detection in Network Traffic Data. arXiv:1709.03082v8
- Zhang, Hongpo & Wu, Chase & Gao, Shan & Wang, Zongmin & Xu, Yuxiao & Liu, Yongpeng. (2018). An Effective Deep Learning Based Scheme for Network Intrusion Detection. 682-687. 10.1109/ICPR.2018.8546162.
- Dhanabal, L., and S. P. Shantharajah. "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms." *International Journal of Advanced Research in Computer and Communication Engineering* 4.6 (2015): 446-452.
- Elsherif, Ahmed. "Automatic Intrusion Detection System Using Deep Recurrent Neural Network Paradigm." (2018).
- Britton, T., Liu-Johnston, I., Cugnière, I., Gupta, S., Rodriguez, D., Barbier, J., & Tricaud, S. Analysis of 24 Hours Internet Attacks.