Security Threat Intelligence Analytics

Sibora Seranaj, Rijish Ganguly
Department of Statistical Science, Department of ECE, Duke University

Abstract

There are around 1.5 billion malicious connection attempts per day on the Duke network. One untapped area is research into these types of attacks and learning how academic institutions are targeted.

Duke OIT has deployed honeypots with sensors to collect data which can be analyzed to predict behavior of attackers. A honeypot is a network-attached system set up as a decoy to lure cyber attackers and to detect, deflect or study patterns of hacking attempts.

Through the STINGAR[1] project, Duke has also partnered with several other universities who have agreed to deploy this system of honeypots and share the collected data. This enables low friction generation of threat intelligence.

Objectives

1. Find the differences between attacks on cloud and local honeypots
2. Classify honeypots based on generated features and collected data
3. Build statistical models to cluster attacking IPs based on similar features
4. Make inferences from geographical data

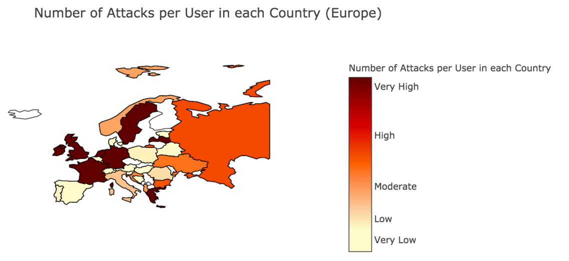
Feature engineering

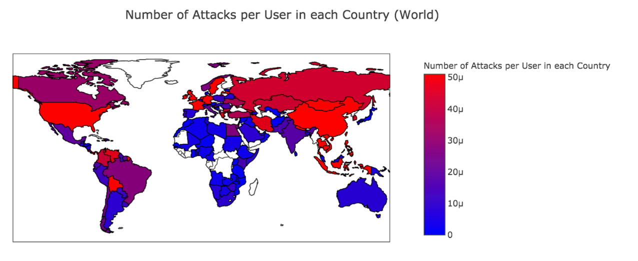
Feature engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work. According to Marcin Nawrocki et al (2016)[2], IP, port, attack-frequency, time until first attack, number of sources per attack, session duration and time between sessions are a few features that can be useful for honeypot data analysis.

Therefore, we generated several features from the data that was provided to us by Duke OIT. The features used throughout our research are**:** *mean time difference between attacks, standard deviation of time difference between attacks, sensor numbers, daily frequency, length of username, length of password, length of command, honeypot type and signature of attack.*

Inference from Country Data

We used the attacker's IP to locate the country of attack. Afterwards, we generated features with respect to a particular country such as frequency of attack, population, GDP, percentage of population with internet access and number of attacks per user. Then we built a linear regression model to see the relationship between number of attacker per user and other features. One limitation of our method was that the attacker might alter or hide their actual IP address. The linear regression model was insignificant with a mean absolute error of 2073.20 and mean squared error of 3613.84.

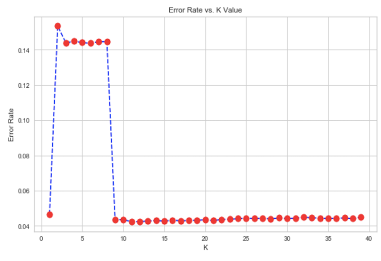

Number of Attacks per User in each Country (Europe)
Number of Attacks per User in each Country
Very High
High
Moderate
Low
Very Low

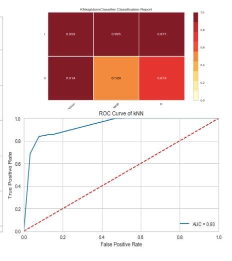

Number of Attacks per User in each Country (World)
Number of Attacks per User in each Country
50μ
40μ
30μ
20μ
10μ
0

Classification

K - Nearest Neighbors

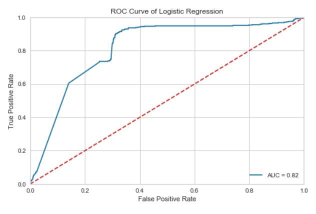
Our objective was to classify the type of honeypot – Cowrie or Dionaea based on the features we generated. Cowrie was assigned 0 and Dionaea was assigned 1. The ROC curve had an AUC of 0.92.

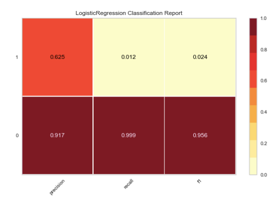




Logistic Regression

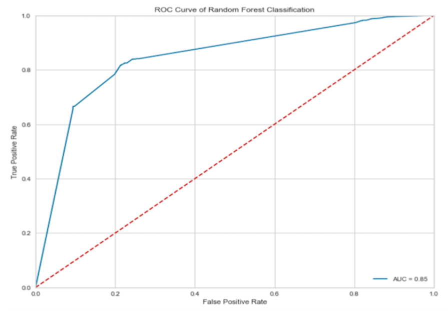
Our objective was to classify the honeypots based on whether they were cloud-based or local. Cloud was assigned 0 and local was assigned 1. The ROC curve had an AUC of 0.82. The precision for local is low because of a smaller training dataset.

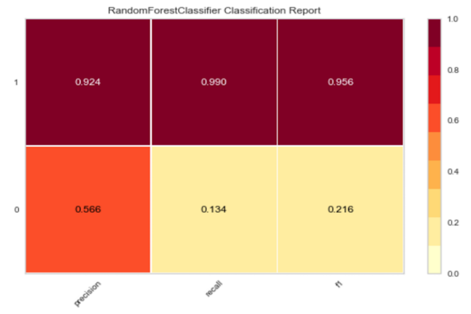




Random Forest

Our objective was to classify the type of honeypot – attacks happening on cloud or local honeypot based on the features we generated. Cloud was assigned 1 and local was assigned 0. The variables of highest importance were: mean time difference between attacks, standard deviation of time difference between attacks, and daily frequency. There was a 92.4% precision for predicting attacks on cloud and a 56.6% precision for predicting attacks on local honeypots. The precision for local is low because of a smaller training dataset. The ROC curve had an AUC of 0.85.

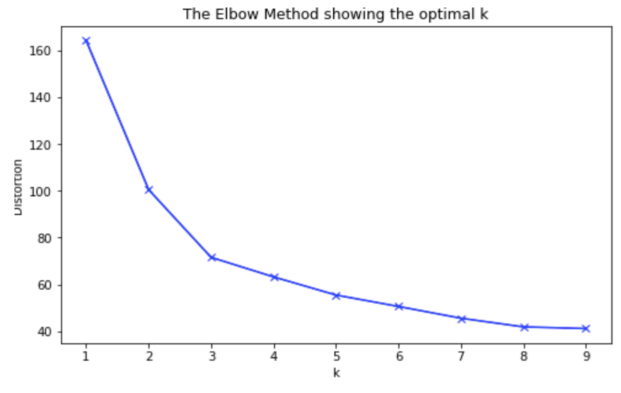


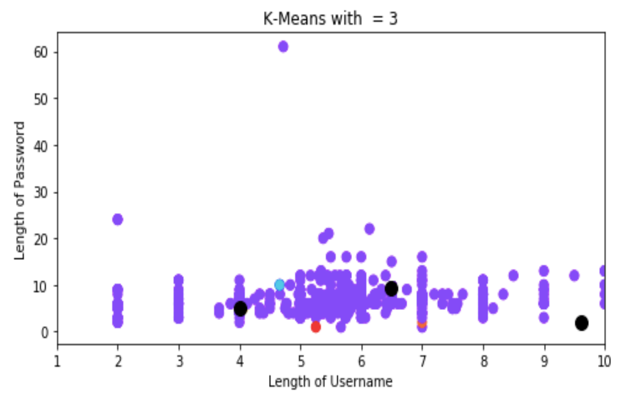


Clustering

K - Means

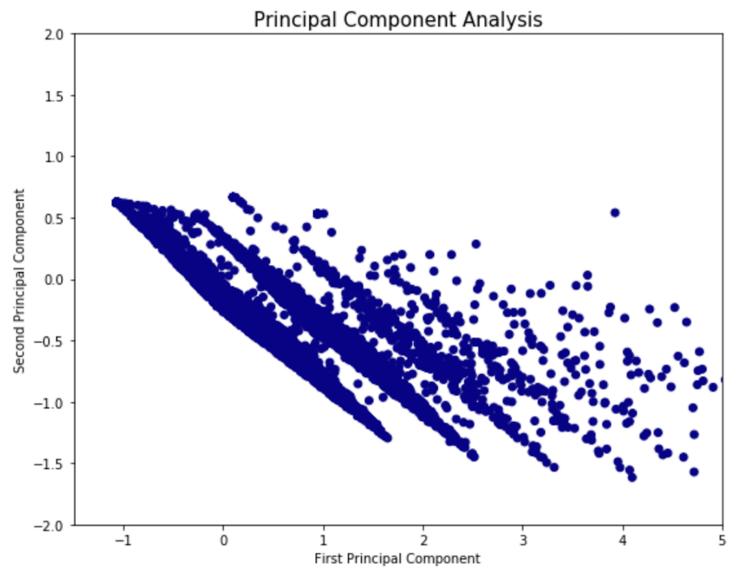
The K-Means algorithm belongs to the category of prototype-based clustering. One of the drawbacks of this clustering algorithm is that we have to specify the number of clusters, k, a priori. Our objective was to find clusters among the different attack IPs and visualize the clustering based on two features, namely – length of password and length of username. We used the elbow method to reduce the sum of squared errors and found the optimal value of k to be 3 as it was the point where the distortion reduced the most. However, we didn't observe any significant clustering.





Principal Component Analysis

The objective of PCA was to find common factors in form of linear combinations of the features under investigation, to rank them according to their importance and to observe cluster formation. Our first attempt was analysis on all honeypots based on attack signatures. We observed very distinct clustering on one signature, and after careful analysis of the data, we concluded that those attacks were happening on Dionaea honeypots. We conducted PCA only on attacks on Dionaea, and observed distinct clusters forming. However, dimensionality remained high, because two components explained around 37% of the variation in the data and seven components explained around 92% of the variation in the data, but considering that there were 8 features to start with, the reduction is not significant enough. The figure below is the plot of the first two components of our Principal Component Analysis on attacks on Dionaea.



Conclusion

1. The most important features we should consider while classifying honeypots are mean time difference between attacks, standard deviation of time difference between attacks and daily frequency of an attack on a particular honeypot.
2. It is possible to observe clusters among a particular type of honeypot based on the features we generated using PCA.
3. Logistic Regression and KNN are highly successful in classifying local and cloud-based honeypots.

Acknowledgements

We would like to acknowledge the following for their contributions to this research: Susan Jacobs – our project manager, Gagan Kaur, Eric Hope, Jesse Bowling, and John Haws from Duke OIT, Paul Bendich and Ariel Dawn from Duke Data+. Thank you for your scientific guidance, helpful advice and continuous support throughout the entire program.

References

1."Shared Threat Intelligence for Network Gatekeeping and Automated Response." Duke OIT, https://stingar.security.duke.edu/.

2. Nawrocki, Marcin, et al. "A Survey on Honeypot Software and Data Analysis." arXiv.org, vol. 10, Aug. 2016, pp. 19-29