

# A Machine Learning Approach to Forensic Investigations (KNN)

Jashi Jeyaantony IR Engineer



# **Problem/Model?**

- Large datasets are time consuming to manually comb through, ML models can be applied for classification (i.e. Malicious, Not malicious)
- We will be looking at Windows event logs (4624 Successful log on) as a PoC

- Model selection K- Nearest neighbours
  - Supervised (labelled) ML model, Uses feature similarity for classification, Ignores data distribution patterns



#### Data

- Models are only as good as the data they are trained on.
- Collect and label data
  - Honeypot; naturally generates varied malicious activity
  - Pre-existing data sets; Labelled/Not Labelled
  - Simulation of attacks in a controlled Lab; highly specific
- My data: Simulated Active Directory lab, where lateral movement (pass-the-hash) and User and Device enumeration (Bloodhound)
- Raw data was extracted using EvtxECmd, Labelled with the help of APT



## **Feature Generation**

- Time, Account name, Workstation name, Source IP
- Raw data -> Meaningful data
- Features of this data:
  - Islan: Is it a Lan IP?
  - Isnewip: Are there reoccurring logins in past 7 days from this IP?
  - Isvpn: Is it a VPN?
  - Precent: % of logins in the past day?
  - Src\_ip\_c: Successful login in 15min window?
  - Tag: Malicious or not?



### Code

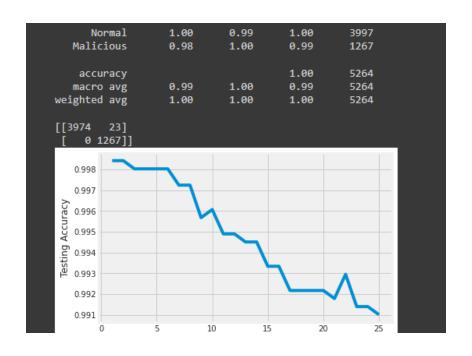
```
feature cols = ['isnewuser','isnewip','isvpn','islan','percent','src ip c']
X = df[feature cols]
y = df['tag']
# train and split data for testing
X train, X test, y train, y test = train test split(X,y, test size=0.4)
k range= range(1,26)
scores = []
# Finding the best K
for k in k range:
    knn = KNeighborsClassifier(n neighbors=k)
    knn.fit(X train, y train)
    y pred = knn.predict(X test)
    scores.append(metrics.f1 score(y test, y pred))
    targetname = ['Normal', 'Malicious']
    result = metrics.classification report(y test,y pred,target names=targetname)
    matrix = metrics.confusion matrix(y test, y pred, labels = [0,1])
    print ('Currently running K:' + str(k))
    print (result)
    print (matrix)
# Plotting as graph
plt.plot(k range, scores)
plt.xlabel('Value of K for KNN')
plt.ylabel('Testing Accuracy')
plt.show()
```

- Models tested:
  Xgboost, KNN, ANN
- Snippet of the KNN model
- 60/40, train test split
- 25 epochs
- Code/data available on github



# **Training and Results**

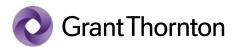
islan	1	-0.07	-0.088	-0.012	-0.39	-0.34	-0.22	-0.32
isnewip	-0.07	1	0.98	0.036	-0.37	-0.62	0.85	-0.63
isnewuser	-0.088	0.98	1	-0.02	-0.37	-0.62	0.87	-0.63
isvpn	-0.012	0.036	-0.02	1	-0.076	-0.029	-0.017	-0.024
percent	-0.39	-0.37	-0.37	-0.076	1	0.5	-0.38	0.48
src_ip_c	-0.34	-0.62	-0.62	-0.029	0.5	1	-0.54	1
tag	-0.22	0.85	0.87	-0.017	-0.38	-0.54	1	-0.55
total_c	-0.32	-0.63	-0.63	-0.024	0.48	1	-0.55	1
	islan	isnewip	isnewuser	isvpn	percent	src_ip_c	tag	total_c





### **Future works**

- Create more complex model (deep learning) with larger databases and more inputs.
- -> Leverage currently archived investigation data using investigation notes, kape, apt...etc to create the *holy grail* in Al models for IR
- Application of ML in SOC? Open call for ideas
- Ongoing ML projects Malware classification using tagging
- Thank you for your attention!



© 2023 Grant Thornton UK LLP.

'Grant Thornton' refers to the brand under which the Grant Thornton member firms provide assurance, tax and advisory services to their clients and/or refers to one or more member firms, as the context requires. Grant Thornton UK LLP is a member firm of Grant Thornton International Ltd (GTIL). GTIL and the member firms are not a worldwide partnership. GTIL and each member firm is a separate legal entity. Services are delivered by the member firms. GTIL does not provide services to clients. GTIL and its member firms are not agents of, and do not obligate, one another and are not liable for one another's acts or omissions.