# **Cyber Security Intrusion Analysis**

## **Coding Dojo – Capstone Project**

Team Members:

* Francisco Piedra S.
* Sharon Alvarado B.

**Business Problem**

Everyday increases the amount of traffic over the network, there are a lot of devices with internet connection around the world, and the quantity of this devices will continue increasing day by day. But also, since every day the amount of traffic over the network is a lot bigger, for sure not all the packages that are sent through the global network are created with good intentions. That’s why we have identified a business problem that every company around the world could be facing and it’s to identify and classify the type (class) of package that is sent through the network in order to detect if the package is normal one or a suspicious one.

**Data**

In order to solve the business problem, we have acquired a couple of datasets that will allow us to train and test the model. This data set was used for intrusion detector learning task in the Third International Knowledge Discovery and Data Mining Tools Competition, which was held in conjunction with KDD-99, The Fifth International Conference on Knowledge Discovery and Data Mining. The intrusion detector learning task is to build a predictive model (i.e. a classifier) capable of distinguishing between “**bad**” connections, called intrusions or attacks, and “**good**” normal connections.  
  
The 1998 DARPA Intrusion Detection Evaluation Program was prepared and managed by MIT Lincoln Labs. The objective was to survey and evaluate research in intrusion detection. A standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment, was provided. The 1999 KDD intrusion detection contest uses a version of this dataset.  
  
Lincoln Labs set up an environment to acquire nine weeks of raw TCP dump data for a local-area network (LAN) simulating a typical U.S. Air Force LAN. They operated the LAN as if it were a true Air Force environment but peppered it with multiple attacks.  
  
The raw training data was about four gigabytes of compressed binary TCP dump data from seven weeks of network traffic.    This was processed into about five million connection records.

**Potential Features**

The following listed features are the ones that we selected as the most representatives in order to perform the analysis of the data:

* Duration
* Protocol\_type
* Service
* Flag
* Src\_bytes
* Dst\_bytes
* Land
* Wrong\_fragment
* Urgent
* Hot
* Num\_failed\_logins
* Logged\_in
* Num\_compromised
* Root\_shell
* Su\_attempted
* Num\_root
* Num\_file\_creations
* Num\_shells
* Num\_access\_files
* Num\_outbound\_cmds
* Is\_host\_login
* Is\_guest\_login
* Count
* Srv\_count
* Serror\_rate
* Srv\_serror\_rate
* Rerror\_rate
* Srv\_rerror\_rate
* Same\_srv\_rate
* Diff\_srv\_rate
* Srv\_diff\_host\_rate
* Dst\_host\_count
* Dst\_host\_srv\_count
* Dst\_host\_same\_srv\_rate
* Dst\_host\_diff\_srv\_rate
* Dst\_host\_same\_src\_port\_rate
* Dst\_host\_srv\_diff\_host\_rate
* Dst\_host\_serror\_rate
* Dst\_host\_srv\_serror\_rate
* Dst\_host\_rerror\_rate
* Dst\_host\_srv\_rerror\_rate

**Visualization (Plots)**

*############################*

*# Source Bytes vs Class*

*# Detection of Anomalies.*

*# We can compare the difference in the Source Bytes Size, between the Normal and Suspicious packages.*

*############################*

sns.set(style = 'whitegrid')

sns.stripplot(x="class", y="src\_bytes",hue="protocol\_type", data=df, jitter=0.2, marker="D", palette="Set2",edgecolor="gray", alpha=.75)

plt.savefig('./plots/01. SRC\_BYTES-CLASS.pdf')

plt.savefig('./plots/01. SRC\_BYTES-CLASS.jpg')

plt.show()

*############################*

*# Protocol Type*

*# Count= number of connections to the same destination host as the current connection in the past 2 seconds (quant.).*

*# Protocol Type. Protocol used in connection.*

*############################*

ax = sns.boxplot(data=df, x="count", y="protocol\_type", whis=np.inf)

ax = sns.stripplot(data=df, x="count", y="protocol\_type", color=".3")

plt.savefig('./plots/02. COUNT-PROTOCOL\_TYPE.pdf')

plt.savefig('./plots/02. COUNT-PROTOCOL\_TYPE.jpg')

plt.show()

*############################*

*# Protocol Type, Class, Duration*

*# Count= number of connections to the same destination host as the current connection in the past 2 seconds (quant.).*

*# Protocol Type. Protocol used in connection.*

*############################*

g = sns.catplot(x="class", y="duration",

hue="class", col="protocol\_type",

data=df, kind="strip",

height=4, aspect=.7);

plt.savefig('./plots/03. DURATION-CLASS-PROTOCOL\_TYPE.pdf')

plt.savefig('./plots/03. DURATION-CLASS-PROTOCOL\_TYPE.jpg')

plt.show()

*############################*

*# Service, Class, Duration*

*# Count= number of connections to the same destination host as the current connection in the past 2 seconds (quant.).*

*# Protocol Type. Protocol used in connection.*

*############################*

g = sns.catplot(x="class", y="duration",

hue="flag", col="protocol\_type",

data=df, kind="strip",

height=4, aspect=.7);

plt.savefig('./plots/04. DURATION-CLASS-PRT\_TYPE\_Flag.pdf')

plt.savefig('./plots/04. DURATION-CLASS-PRT\_TYPE\_Flag.jpg')

plt.show()

*############################*

*# 32. Dst host count: count of the connections having the same destination IP address (quant.)*

*# 33. Dst host srv count: count of connections having the same port number (quant.)*

*# 4. Flag: status of the connection (e.g. REJ = connection rejected) (cat., 11 categories)*

*# 2. Protocol type: Protocol used in connection (cat., 3 categories)*

*############################*

sns.scatterplot(data=df, x="dst\_host\_count", y="dst\_host\_srv\_count", hue="flag", style="protocol\_type", palette="deep",hue\_norm=(0, 10))

plt.savefig('./plots/05. DURATION\_SRCBYTES\_FLAG.pdf')

plt.savefig('./plots/05. DURATION\_SRCBYTES\_FLAG.jpg')

plt.show()

*############################*

*# 32. Fag*

*#*

*############################*

sns.boxplot(x='flag', y='dst\_host\_same\_srv\_rate', data=df)

plt.savefig('./plots/06. FLAG\_DST\_HOST\_SAME\_SRV.pdf')

plt.savefig('./plots/06. FLAG\_DST\_HOST\_SAME\_SRV.jpg')

plt.show()

**Model Selection**

In order to analyze the data, there were selected two main models.

* The **first** one is the Linear Regression in order to processes the previously selected features from the “Potential features” section.
* The **second** one is the K-Mean in order to perform some classification over the type of protocol, based in the information provided by the Train data.

**Additional Information**

Table

Description automatically generated

1. **Duration**: Time duration of the connection (quant.)
2. **Protocol type**: Protocol used in connection (cat., 3 categories)
3. **Service**: Destination network service used (cat., 70 categories)
4. **Flag**: status of the connection (e.g. REJ = connection rejected) (cat., 11 categories)
5. **Src bytes**: number of data bytes transferred from source to destination (quant.)
6. **Dst bytes**: number of data bytes transferred from destination to source (quant.)
7. **Land**: indicator whether port number and IP address of source and destination are equal, if yes = 1, otherwise 0 (binary)
8. **Wrong** **fragment**: number of wrong fragments in connection (quant.)
9. **Urgent**: number of urgent packets (quant.)
10. **Hot**: number of ”hot” indicators in the content such as: entering a system directory, creating programs and executing programs (quant.)
11. **Num** **failed** **logins**: number of failed login attempts (quant.)
12. **Logged** **in**: 1 if successfully logged in, 0 otherwise (binary)
13. **Num** **compromised**: number of “compromised” conditions (quant.)
14. **Root** **shell**: 1 if root shell is obtained, 0 otherwise (binary)
15. **Su** **attempted**: 1 if ”su root” command attempted or used, 0 otherwise (quant., data set contains value 2)
16. **Num** **root**: number of operations performed as a root or root accesses (quant.)
17. **Num** **file** **creations**: number of file creation operations (quant.)
18. **Num** **shells**: number of shell prompts (quant.) 51 3. Data NSL-KDD’99
19. **Num** **access** **files**: number of operations on access control files (quant.)
20. **Num** **outbound** **cmds**: number of outbound commands in an ftp session (quant.)
21. **Is** **host** **login**: 1 if the login is from root or admin, 0 otherwise (binary)
22. **Is guest** **login**: 1 if the login is from guest, 0 otherwise (binary)
23. **Count**: number of connections to the same destination host as the current connection in the past 2 seconds (quant.)
24. **Srv** **Count**: number of connections to the same service (port number) as the current connection in the past 2 seconds (quant.)
25. **Serror** **Rate**: % of connections that have activated s0, s1, s2 or s3 flag (4) among connections aggregated in count (quant.) 52 3. Data NSL-KDD’99
26. **Srv** **Serror** Rate: % of connections that have activated s0, s1, s2 or s3 flag (4) among connections aggregated in srv count (quant.)
27. **Rerror rate**: % of connections that have activated REJ flag (4) among connections aggregated in count (quant.)
28. **Srv rerror rate:** % of connections that have activated REJ flag (4) among connections aggregated in srv count (quant.)
29. **Same srv rate**: % of connections to the same service among those aggregated in count (quant.)
30. **Diff srv rate**: % of connections to the different service among those aggregated in count (quant.)
31. **Srv diff host rate**: % of connections that were to different destination machines among the connections aggregated in srv count (quant.)
32. **Dst host count**: count of the connections having the same destination IP address (quant.)
33. **Dst host srv count**: count of connections having the same port number (quant.)
34. **Dst host same srv rate**: % of connections that were to different services, among those in dst host count (quant.)
35. **Dst host diff srv rate**: % of connections that were to different services, among those in dst host count (quant.)
36. **Dst host same src port rate**: % of connections that were to the same source port, among those in dst host srv count (quant.)
37. **Dst host srv diff host rate**: % of connections that were to different destination machines, among those in dst host srv count (quant.)
38. **Dst host serror rate**: % of connections that have activated the s0, s1, s2 or s3 flag (4), among those in dst host count (quant.)
39. **Dst host srv serror rate**: % of connections that have activated the s0, s1, s2 or s3 flag (4), among those in dst host srv count (quant.)
40. **Dst host rerror rate**: % of connections that have activated the REJ flag (4), among those in dst host count (quant.)
41. **Dst host srv rerror rate**: % of connections that have activated the REJ flag (4), among those in dst host srv count (quant.)
42. **Class**: If the connection is Normal, or Anormaly.