National College of Ireland

Configuration Manual

MSc Research Project
Master's in cyber security

Jordan Cogan
Student ID: 15392331

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Jordan Cogan
Student ID:	15392331
Module:	Research Project/Internship
Programme:	Master's in cyber security
Year:	1
Supervisor:	Vikas Sahni
Submission Due Date:	6th September 2021
Project Title:	Configuration Manual
Word Count:	3179
Page Count:	16

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: 

Date: 02/09/2021

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Jordan Cogan
15392331

1 Introduction

The purpose of this document is to explain the implementation process undertaken during this research project, this document also includes the hardware and software requirements used during implementation. This configuration manual also contains snippets of the Python code which was used during the development of the project "**Using feature selection to improve intrusion detection?**"

1.1 System Hardware

The system hardware which was used during this research project development was as follows:

- **Processor:** Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz, 1992 Mhz, 4 Core(s), 8 Logical Processor(s)
- **Installed Physical Memory (RAM):** 16.0 GB
- **Operating System:** Microsoft Windows 10 Home
- **GPU:** NVIDIA GeForce GTX 1050
- **Storage:** 1TB HDD and 256 SSD

1.2 System Software

Microsoft Excel and Google Collaboratory was the software which was used during the implementation of the research project:

- **Microsoft Excel:** Excel was used during the entirety of the research project. Excel was used to read the datasets and visualise results. Excel was also used to convert the worksheets into CSV files to be used by the machine learning algorithms.
- **Google Collaboratory:** Google Collaboratory or Google Collab for short, is a free cloud-based service by Google which allows users to write and execute Python code via browser. Google collab also makes use Googles dedicated Graphics processing unit (GPU) and Tensor Processing Unit (TPU) which allows programmes to run faster and not rely on local machine resources. Google collab was useful for being able to write code and work on the research project from any device without the limitations of local resources.

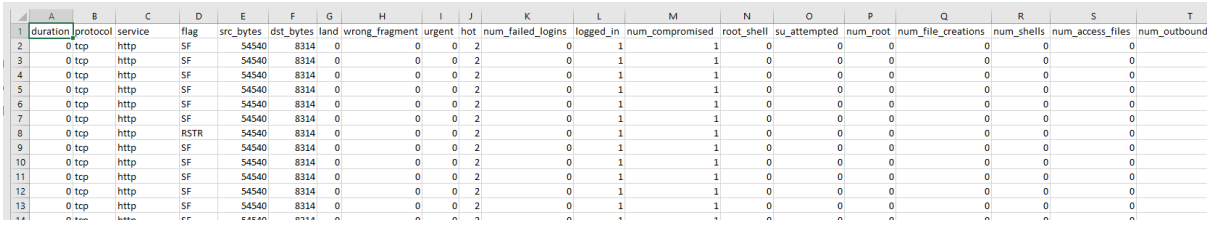
2 Project Development

The entire project was coded in the Python language with the use of Google Collab. Some dataset manipulation was performed using Microsoft Excel. The implementation is separated into Dataset preparation, Feature Selection, Classification models and Evaluation.

2.1 Dataset preparation

Initially the dataset was one large excel file containing all of the records. The dataset contained 494,022 rows and 41 columns of records.

A 42nd column contains a label for the type of traffic that record is related to, i.e., Normal, Apache, Nmap, Back etc. An accompanying document provided was used to identify which traffic belonged to each category i.e., 'DOS', 'Probe', 'R2L', 'U2R', 'Other' and 'Normal'.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	duration	protocol	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
2	0	tcp	http	SF	54540	8314	0	0	0	2	0	1	1	0	0	0	0	0	0	
3	0	tcp	http	SF	54540	8314	0	0	0	2	0	1	1	0	0	0	0	0	0	
4	0	tcp	http	SF	54540	8314	0	0	0	2	0	1	1	0	0	0	0	0	0	
5	0	tcp	http	SF	54540	8314	0	0	0	2	0	1	1	0	0	0	0	0	0	
6	0	tcp	http	SF	54540	8314	0	0	0	2	0	1	1	0	0	0	0	0	0	
7	0	tcp	http	SF	54540	8314	0	0	0	2	0	1	1	0	0	0	0	0	0	
8	0	tcp	http	RSTR	54540	8314	0	0	0	2	0	1	1	0	0	0	0	0	0	
9	0	tcp	http	SF	54540	8314	0	0	0	2	0	1	1	0	0	0	0	0	0	
10	0	tcp	http	SF	54540	8314	0	0	0	2	0	1	1	0	0	0	0	0	0	
11	0	tcp	http	SF	54540	8314	0	0	0	2	0	1	1	0	0	0	0	0	0	
12	0	tcp	http	SF	54540	8314	0	0	0	2	0	1	1	0	0	0	0	0	0	
13	0	tcp	http	SF	54540	8314	0	0	0	2	0	1	1	0	0	0	0	0	0	

Figure 1. KDD99 dataset

f_host_rate	dst_host_serror_rate	dst_host_srv_serror_rate	dst_host_rerror_rate	dst_host_srv_rerror_rate	
0	0	0	0	0	back.
0	0	0	0	0	back.
0	0	0	0	0	back.
0	0	0	0	0	back.
0	0	0	0	0	back.
0	0	0	0	0	back.
0	0	0	0.14	0.14	back.
0	0	0	0.12	0.12	back.
0	0	0	0.11	0.11	back.
0	0	0	0.1	0.1	back.
0	0	0	0.09	0.09	back.
0	0	0	0.08	0.08	back.
0	0	0	0.08	0.08	back.
0	0	0	0.07	0.07	back.
0	0	0	0.07	0.07	back.
0	0	0	0.06	0.06	back.
0	0	0	0.06	0.06	back.
0	0	0	0.06	0.06	back.
0	0	0	0.11	0.11	back.
0	0	0	0.1	0.1	back.
0	0	0	0.1	0.1	back.
0	0	0	0.09	0.09	back.

Figure 2. KDD99 dataset traffic type

Figure 2 shows the 42nd column which originally contained the traffic type for each record i.e., 'back'. Figure 3 below shows the accompanying document which was used to identify which traffic belonged to which category.

```
back dos
buffer_overflow u2r
ftp_write r2l
guess_passwd r2l
imap r2l
ipsweep probe
land dos
loadmodule u2r
multihop r2l
neptune dos
nmap probe
perl u2r
phf r2l
pod dos
portsweep probe
rootkit u2r
satan probe
smurf dos
spy r2l
teardrop dos
warezclient r2l
warezmaster r2l
```

Figure 3. KDD99 dataset categories

The next step was to split the dataset into subsets based on the categories determined in figure 3. Excel was used to sort by traffic type and then a subset was created for each category as follows:

- DOS
- Probe
- U2R
- R2L
- Other
- Normal

Using the traffic type column provided in the original dataset in conjunction with the provided categories, the dataset was separated.

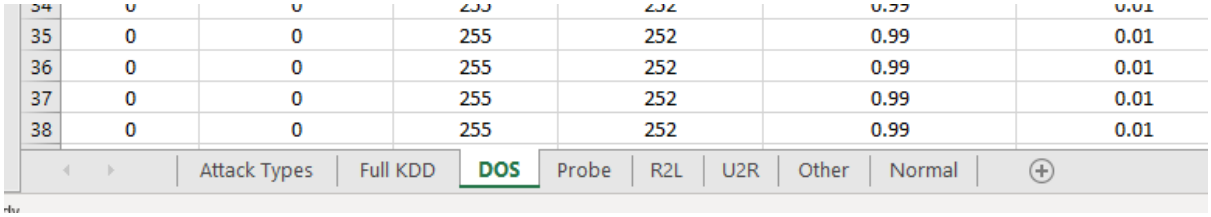


Figure 4. KDD99 dataset split into subsets

In order for machine learning algorithms to interpret the dataset, all values must be numerical, however the current datasets include string values for the variables, Protocol, Service and Flag. These needed to be converted to numerical and as such a Python script was used.

	A	B	C	D	E	
1	duration	protocol	service	flag	src_bytes	d
2	0	tcp	http	SF	54540	
3	0	tcp	http	RSTR	39420	
4	0	tcp	http	SF	54540	
5	0	tcp	http	SF	54540	
6	0	tcp	http	SF	54540	
7	0	tcp	http	SF	54540	
8	0	tcp	http	SF	54540	
9	0	tcp	http	SF	54540	

Figure 4. KDD99 dataset string values

Figure 5 below shows the Python code written to convert the string values to numerical values.

```
[ ] #Count Values in protocol
    data["protocol"].value_counts()

[ ] #Mapping String Values to Numerical
    mapping_dict = {"protocol": {"icmp":1, "tcp":2, "udp":3}}

    #Replace the values
    data.replace(mapping_dict, inplace=True)

[ ] #Count Values in service
    data["service"].value_counts()

[ ] #Mapping String Values to Numerical
    mapping_dict = {"service": {"http":1, "finger":2, "telnet":3, "private":4, "echo":5, "discard":6, "systat":7, "daytime":8, "netstat":9, "ftp":10, "ftp_data":11,
                    "ssh":12, "smtp":13, "time":14, "whois":15, "name":16, "domain":17, "mtp":18, "gopher":19, "remote_job":20, "rje":21, "ctf":22, "link":23, "supdup":24, "hostnames":25,
                    "iso_tsap":26, "csnet_ns":27, "pop_2":28, "sunrpc":29, "pop_3":30, "auth":31, "uucp_path":32, "nntp":33, "netbios_ns":34, "netbios_dgm":35, "netbios_ssn":36, "imap4":37,
                    "sql_net":38, "vmnet":39, "bgp":40, "Z39_50":41, "ldap":42, "nnsp":43, "http_443":44, "exec":45, "shell":46, "login":47, "printer":48, "efs":49, "courier":50, "uucp":51,
                    "klogin":52, "kshell":53, "other":54, "ecr_i":55, "tim_i":56 }}

    #Replace the values
    data.replace(mapping_dict, inplace=True)

[ ] #Count Values in flag
    data["flag"].value_counts()

[ ] #Mapping String Values to Numerical
    mapping_dict = {"flag": {"SF":1, "RSTR":2, "S2":3, "S1":4, "S0":5, "REJ":6, "RSTO":7 }}

    #Replace the values
    data.replace(mapping_dict, inplace=True)
```

Figure 5. Python code snippet converting strings to numerical values

This process was applied to each of the subsets created until all String values were converted. Once each file was converted the code which can be seen below in figure 6, was used to export these converted files to CSV files for usage.

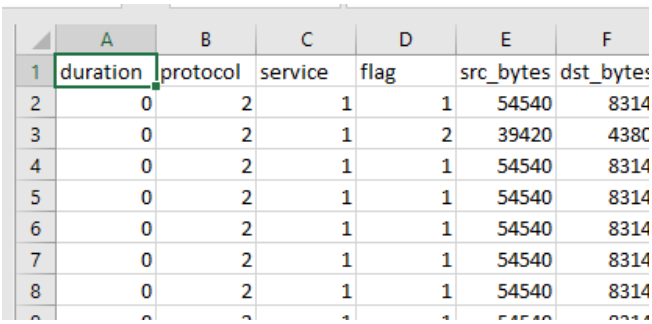
```
#Mapping String Values to Numerical
mapping_dict = {"flag": {"SF":1, "RSTR":2, "S2":3, "S1":4, "S0":5, "REJ":6, "RSTO":7 }}

#Replace the values
data.replace(mapping_dict, inplace=True)

data.to_csv('DosDataConverted.csv')
```

Figure 6. Python code snippet exporting converted csv file

Figure 7 below shows and example of the output from the CSV file once the strings have been converted to numerical.



	A	B	C	D	E	F
1	duration	protocol	service	flag	src_bytes	dst_bytes
2	0	2	1	1	54540	8314
3	0	2	1	2	39420	4380
4	0	2	1	1	54540	8314
5	0	2	1	1	54540	8314
6	0	2	1	1	54540	8314
7	0	2	1	1	54540	8314
8	0	2	1	1	54540	8314

Figure 7. Converted CSV file example

Provided below it the table of all conversions for each variable, Protocol, Service and Flag.

Flag' Conversions:

Flag	**Conversion**
SF	1
RSTR	2
S2	3
S1	4
S0	5
REJ	6
RSTO	7
S3	8
SH	9
OTH	10
RSTOS0	11

'Service' Conversions:

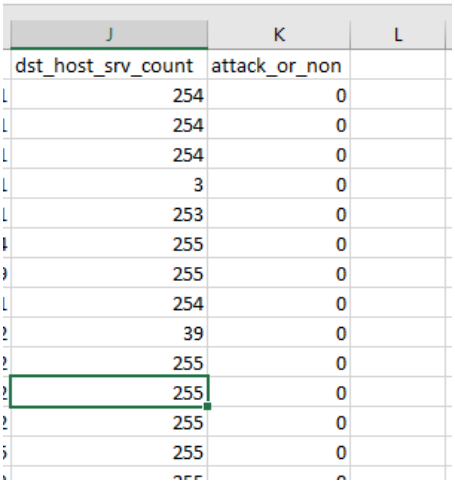
Service	**Conversion**
http	1
finger	2
telnet	3
private	4
echo	5
discard	6
systat	7
daytime	8
netstat	9
ftp	10
ftp_data	11
ssh	12
smtp	13
time	14
whois	15
name	16
domain	17
mtp	18
gopher	19

remote_job	20
rje	21
ctf	22
link	23
supdup	24
hostnames	25
iso_tsap	26
csnet_ns	27
pop_2	28
sunrpc	29
pop_3	30
auth	31
uucp_path	32
nntp	33
netbios_ns	34
netbios_dgm	35
netbios_ssn	36
imap4	37
sql_net	38
vmnet	39
bgp	40
Z39_50	41
ldap	42
nnsp	43
http_443	44
exec	45
shell	46
login	47
printer	48
efs	49
courier	50
uucp	51
klogin	52
kshell	53
other	54
ecr_i	55
tim_i	56
urp_i	57
eco_i	58
domain_u	59
IRC	60
pm_dump	61
X11	62
ntp_u	63
icmp	64
tftp_u	65

'Protocol' Conversions:

Protocol	Conversion
icmp	1
tcp	2
udp	3

Finally, a new column was added to each of the datasets called 'Attacking Or Non' with the Value 1 being assigned to all the attacking traffic and the value 0 being assigned to all Normal traffic as can be seen in figure 8 below.



J	K	L
dst_host_srv_count	attack_or_non	
254	0	
254	0	
254	0	
3	0	
253	0	
255	0	
255	0	
254	0	
39	0	
255	0	
255	0	
255	0	
255	0	

Figure 8. Attacking Or Non column added

2.2 Feature Selection

Once the data was prepared the next stage of the implementation was the feature selection. Feature selection was implemented in two layers.

Figure 9. Multi-layer feature selection model

The first layer was to apply the Chi2 feature selection to each of the subsets individually. This was done using a Python code which can be seen in figure 10 below.

```
#Import Dependencies
import pandas as pd
import numpy as np

#Import Feature libraries
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2

#Read CSV File
data = pd.read_csv("/content/drive/MyDrive/Masters/Thesis/Dataset/DosDataConverted.csv")

#Create Dataframe
X = data.iloc[:,0:40]
y = data.iloc[:,0]

#Select 10 features with highest Chi2 Value
bestfeatures = SelectKBest(score_func=chi2, k=10)
fit = bestfeatures.fit(X,y)

dfscores = pd.DataFrame(fit.scores_)
dfcolumns = pd.DataFrame(X.columns)

#Add Features to table
featureScores = pd.concat([dfcolumns,dfscores],axis=1)
featureScores.columns = ['Heading','Chi2 Score']

#Print All Feature Scores
featureScores

#Print Top 10 Feature Scores
print(featureScores.nlargest(10,'Chi2 Score'))
```

Figure 10. Feature selection code snippet

The feature selection code was applied to each of the attacking categories, DOS, Probe, U2R, R2L and Other. The top ten features were then extracted.

```
[11] #Print Top 10 Feature Scores
     print(featureScores.nlargest(10,'Chi2 Score'))

                  Heading    Chi2 Score
4               src_bytes  2.585959e+07
5               dst_bytes  1.260810e+07
0                duration  1.280612e+06
22                 count  5.088164e+03
23           srv_count    4.746115e+03
11           logged_in    2.819402e+03
9                  hot    2.261863e+03
2                service  5.609249e+02
12     num_compromised    4.537381e+02
32  dst_host_srv_count    4.510814e+02
```

Figure 11. Top 10 Features for DOS traffic

Once the top ten features from each attacking traffic was discovered, they were combined to create the Initial Features list, which is the top features of each attack type not including duplicate features, the initial features were as follows:

Feature Number	Feature Name
1	count
2	dst_bytes
3	dst_host_count
4	dst_host_srv_count
5	duration

6	flag
7	hot
8	logged_in
9	num_access_files
10	num_compromised
11	num_failed_logins
12	num_file_creations
13	num_root
14	service
15	src_bytes
16	srv_count
17	srv_rerror_rate
18	urgent

The Initial features dataset was then created, this dataset comprised of all the attacking traffic put together but removing any features which were not selected in the key features list above.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	duration	service	flag	src_bytes	dst_bytes	urgent	hot	num_failed_logins	logged_in	num_compromised	num_root	num_file_creations	num_access_files	count	srv_count	srv_rerror_rate	dst_host_count	dst_host_srv_count
2	0	1	1	54540	8314	0	2	0	1	1	0	0	0	1	1	0	255	250
3	0	1	2	39420	4380	0	1	0	1	0	0	0	0	2	2	0.5	255	250
4	0	1	1	54540	8314	0	2	0	1	1	0	0	0	3	3	0.33	255	250
5	0	1	1	54540	8314	0	2	0	1	1	0	0	0	2	2	0	255	250

Figure 12. Initial features dataset

The previously used feature selection code was again used, and this time applied to this Initial Features dataset in order to find the final top 10 features of attacking traffic. The results of the final top 10 features of attacking traffic can be seen in figure 13.

```
[10] #Print Top 10 Feature Scores for Final Feature Selection
     print(featureScores.nlargest(10,'Chi2 Score'))

                  Heading    Chi2 Score
3             src_bytes  2.533436e+12
4             dst_bytes  7.841052e+09
0              duration  1.737232e+09
10             num_root  1.671366e+07
9       num_compromised  5.649862e+06
13                count  3.772507e+06
14            srv_count  3.286356e+06
11  num_file_creations  7.417003e+05
12    num_access_files  3.217854e+05
17  dst_host_srv_count  2.789314e+05
```

Figure 13. Final top features of attacking data

The Final features dataset was then created, this dataset comprised of all the attacking traffic put together but removing any features which were not selected in the final key features list above. This dataset was then combined with the Normal traffic while continuing to only select the top features to create the final dataset.

	A	B	C	D	E	F	G	H	I	J	K
1	duration	src_bytes	dst_bytes	num_compromised	num_root	num_file_creations	num_access_files	count	srv_count	dst_host_srv_count	attack_or_non
2	0	105	146	0	0	0	0	1	1	254	0
3	0	105	146	0	0	0	0	1	1	254	0
4	0	105	146	0	0	0	0	1	1	254	0

Figure 14. Final features dataset

2.3 Classification models

Now that the final dataset was created using the top key features of attacking traffic, the next step was to apply classification models to the dataset. The two models used during this research project were Random Forest and K-nearest neighbours (KNN).

Random Forest was applied first, the code for the Random Forest can be seen below in figure 15.

```
[ ] #Read CSV File
    dataset = pd.read_csv("/content/drive/MyDrive/Masters/Thesis/Dataset/FeatureSelectionCombined.csv")

[ ] #Create dataframe
    X = dataset.iloc[:,0:11]
    y = dataset.iloc[:,0:311030]

[ ] #Split Features and Labels
    X=dataset[['duration', 'src_bytes', 'dst_bytes', 'num_compromised', 'num_root', 'num_file_creations', 'num_access_files', 'count', 'srv_count', 'dst_host_srv_count' ]] #Split Features
    y=dataset['attack_or_non'] #Split Labels

[ ] #Split dataset into training set and test set
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)

[ ] #Perfrom Random Forest
    rfc = RandomForestClassifier(n_estimators=100, random_state=0)

[ ] rfc.fit(X_train,y_train)

[ ] y_pred = rfc.predict(X_test)
```

Figure 15. Random Forest Code snippet

The features and labels were split in order to teach the algorithm which of the categories are features and which is to be considered ‘Attack or Non’. The dataset was then split into a 70/30 ratio of training data and testing data. The model was fine-tuned with n_estimator as 100 and random_state as 0.

The second model used was K-nearest neighbours (KNN), figure 16 shows the code snippet used for the KNN model.

```
[2] #Read CSV File
    dataset = pd.read_csv("/content/drive/MyDrive/Masters/Thesis/Dataset/FeatureSelectionCombined.csv")

    #Create dataframe
    X = dataset.iloc[:,0:11]
    y = dataset.iloc[:,0:311030]

[4] #Split Features and Labels
    X=dataset[['duration', 'src_bytes', 'dst_bytes', 'num_compromised', 'num_root', 'num_file_creations', 'num_access_files', 'count', 'srv_count', 'dst_host_srv_count' ]] #Split Features
    y=dataset['attack_or_non'] #Split Labels

[5] #Split dataset into training set and test set
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)

[6] #Perfrom KNN
    classifier = KNeighborsClassifier(n_neighbors = 10)

[7] classifier.fit(X_train, y_train)

[8] y_pred = classifier.predict(X_test)
```

Figure 16. K-nearest neighbours (KNN) Code snippet

Similar to the Random forest, the features and labels were split in order to teach the algorithm which of the categories are features and which is to be considered ‘Attack or Non’. The dataset was then split into a 70/30 ratio of training data and testing data. The KNN model was fine-tuned with n_neighbors as 10. The default number of n_neighbors is 5, so double was chosen for this research project.

2.4 Evaluation

In this section the classification models are evaluated. Python code was written in order to produce the accuracy of each model as well as to produce a classification report, graph a confusion matrix and graph an ROC curve.

```
[11] #Display Accuracy
     print("Accuracy:",metrics.accuracy_score(y_test, y_pred))

     #View Classifaction Report
     print(classification_report(y_test, y_pred))

                   precision    recall  f1-score   support

                0       0.95      0.94      0.95     18034
                1       0.99      0.99      0.99     75275

         accuracy                           0.98     93309
        macro avg       0.97      0.96      0.97     93309
     weighted avg       0.98      0.98      0.98     93309

     #Confusion Matrix
     confusion_matrix = pd.crosstab(y_test, y_pred, rownames=['Actual'], colnames=['Predicted'])
     ax= plt.subplot()
     sn.heatmap(confusion_matrix, annot=True, fmt='g', annot_kws={'size':15}, ax=ax);

     plt.figure(figsize=(2,2))

     #Edit Labels and Title
     ax.set_xlabel('Predicted labels');
     ax.set_ylabel('True labels');
     ax.set_title('Random Forest Confusion Matrix');

     ax.tick_params(axis='both', which='major', labelsize=10)
     ax.xaxis.set_ticklabels(['Normal Traffic', 'Attacking traffic']);
     ax.yaxis.set_ticklabels(['Normal Traffic', 'Attacking traffic']);
```

Figure 17. Evaluations Code snippet

```
#ROC Curve
fpr, tpr, thresholds = roc_curve(y_test, probas[:,0], pos_label=0)
roc_auc = auc(fpr, tpr)

plt.figure(dpi=150)
plt.plot(fpr, tpr, lw=1, color='green', label=f'AUC = {roc_auc:.3f}')
plt.title('ROC Curve for Random Forest')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate (Recall)')
plt.xlim([-0.05, 1.05])
plt.ylim([-0.05, 1.05])
plt.legend()
plt.show()
```

Figure 18. ROC Curve Code snippet

The same code was used to produce the evaluations for both the Random Forest and KNN models.

3 Appendix

3.1 Monthly Journal June

Monthly Internship Activity Report

Student Name: _____Jordan Cogan_____ Student number: _____15392331_____

Company: _____SecuriCentrix_____ Month Commencing: __1st June _____

Week One: Day one was mainly an introduction day, I was shown the tools and systems the company uses and shown the key attributes to look for when attempting to analyse an alarm in the SIEM. After feeling comfortable with how to navigate the SIEM, the each of the team members assigned me an alarm to analyse. The first alarm was a Potentially Unwanted Program (PUP) the alarm was found to be a true positive because a client was using a third-party website to download driver updates, so a query was raised. The second alarm to analyse was Windows Defender Update Failed, it was found that the port windows defender was attempting to update on was blocked by the client, which means windows defender was unable to complete the update. The third alarm was Multiple AWS IAM Access Denied, the alarm showed that as user was attempting to gain access to resources which they did not have access to, the client was queried on this, and it was determined a legitimate new user on their system.

Week Two: At the start of the second week, I was asked to analyse a malware which had been detected previously on a clients' network. I used a sandbox VM for my analysis and installed Flare VM for the analysis toolset. I used a number of tools such as, PPEE, PEStudio, Wireshark and CFF Explorer. Some online tools were also used once the malware Hash value was found such as Virus Total and Hybrid analysis. Using these tools, I was able to find a lot of information about the malware such as the Hostname, Hash value of the files, file size, magic value and found a .exe file hidden within a .doc file.

During this week there was also a PCI DSS assessment taking place, I was able to join the meeting and watch the assessment. The assessment lasted the full week, I attended 2 days of the assessment and was able to pick up on things such as, communication skills with clients and adapting the language used and approach for questions based on the relationship built with clients. I was also able to learn how to read and understand detailed Data flow diagrams, services diagrams, and hardware lists. I noted also for some areas which may not be in the scope of PCI are investigated for due diligence. I found that under PCI passwords need to expire every 90 days and internal logs must be able to be produced from the last 3 months instantly and if required the last 12 months logs should be obtainable.

Week Three: At the start of the third week, I was asked to analyse a Phishing mail which was sent to one of the internal employees. During my analysis I found that the mail was indeed a phishing attempt with multiple links leading to insecure websites using HTTP and claiming to be an educational company from Egypt. The website included many spelling errors also. Later I was asked to create a new asset group of the Linux assets which a client had sent to us, I created this group on the USM Anywhere and I was then shown how a scan would be performed on the group. During this week I was also continuously viewing various alarms in an attempt to further understand different types and how to interpret them.

Week Four: During the week I was again given two specific alarms to analyse, the first of these alarms was
"Desktop Software - Bitcoin", during my investigation I found the malware that was detected was
"Trojan.Monero" which is used for cryptocurrency mining purposes. The DNS RR name "pool.minexmr[.]com" was checked using online tools such as VirusTotal and IBM X Force Exchange, where it was found to be malicious. The client was emailed about the incident, and they informed the team that the DNS RR has been blocked on their Antivirus and Perimeter Firewall.
The second alarm was "Vulnerable Software Exploitation", this alarm was linked to a "HeartBleed Attack" which is due to a vulnerability in all OpenSSL versions before 1.0.1g. I was informed this had been queried to the client previously and they are running higher version of this and as such are not vulnerable to this exploit.
During this week I was also shown how to create and complete the daily security checklists which are used for internal usage to monitor all the events and sensors health for each client. I was asked to complete multiple of these checklists and send them to one of the other team members for review and then upload to the internal SharePoint. I was also asked to create a new separate checklist for all of these clients to track during the weekends, this was completed and sent to one of the team members for review and upload also.

Employer comments

Jordan has taken on tasks diligently and grasped the concepts quickly.

Student Signature: Jordan Cogan Date:28/06/2021

Industry David Steele Supervisor Signature: 05/07/2021

3.2 Monthly Journal July

Monthly Internship Activity Report

Student Name: _____Jordan Cogan_____ Student number: _____15392331_____

Company: _____SecuriCentrix_____ Month Commencing: __1st July _____

This month has continued as smoothly as the first. I am at a point now where I have daily tasks, one of these daily tasks is to complete a daily checklist. The daily checklist is an excel sheet which contains information about all of the clients'

environments for the last 24hrs, my task each morning is to complete this checklist by entering all of the data for the last 24hrs for each client, this includes number of alarms, alarm details, sensor status and system status. I found the checklist to be a great tool for recognising trends in each client's environment, noticing which alarms are common daily helped me to asks questions about them and then allow me to notice when a new alarm has appeared, and I then investigate it.

During this month I have also been shown how to create an incident and query report to raise with clients, these reports go hand in hand with the daily checklist as if there are any anomalies noticed during the checklist, I investigate it and then rise either a query or incident. A query report is used for lower priority events which we would just like to get more information about, incident reports are high priority events which we require more information from the client and also attach mitigations where possible.

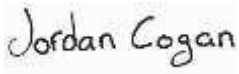
I was asked to investigate a newly released CISA Ransomware Readiness Assessment (RRA) tool, during this investigation I found:

- The RRA tools will carry out an evaluation on the strength of an organisation's cybersecurity controls and protocols, regarding ransomware attacks.
- Aims to focus on the basics then first, then progression is achieved by implementation of best practices which leads to intermediate and advanced categories.
- The analysis is presented in user friendly graphs and tables that will contain the results of the assessment in both a summary and detailed form.

As well as the daily checklists, each Monday I am tasked with creating the weekend checklists, these follow the same procedure of the daily checklist although we don't not have staff during weekends, so they are completed each Monday to cover the full weekend. Again, any anomalies are raised with the client as incidents or queries.

I was also shown how to create alarms in the SIEM this month, one of the team members demonstrated how alarms are created, after the demonstration I was tasked with creating my own alarms for a new client environment, these alarms were reviewed and then implemented in this client's environment.

Employer comments

Student Signature:  Date: 21/07/2021

 24/08/2021

Industry Supervisor Signature: _____

3.3 Monthly Journal August

Monthly Internship Activity Report

Student Name: _____Jordan Cogan_____ Student number: _____15392331_____

Company: _____SecuriCentrix_____ Month Commencing: __1st August _____

This month has again been a smooth and enjoyable learning experience. I have continued to complete the daily checklists and every Monday I complete the weekend checklists also.

I have become more comfortable with creating alarms which I learned last month and have now created multiple alarms which have all been implemented. I have also been continuing to raise incidents or queries for clients where needed.

This month I lead my first client call, this particular client has weekly calls with SecuriCentrix in order to review the tickets which are been worked on and updating with any progress since the last meeting. I lead my first call this month which was a great experience, there were two other more senior team members on the call along with me in case I needed support, the call went smoothly, and I presented the client with the weekly report and asked for updates regarding our work in progress tickets. After the call I received feedback from the rest of the team, I was informed I may have been speaking too quickly due to nerves and should try to slow down in future calls. Since this first call, I have now led multiple more calls with this same client and I believe I have been improving each call.

I have also been exposed to some penetration tests experience this month, I was tasked with performing a segmentation test for one client, the client provided me with details to access the environment and from there I ran the scans. I exported the results and then created the report using a template which I was shown. Another team member performed Quality Assurance on the report and one satisfied, the report was sent to the client.

One of the SecuriCentrix clients utilises the application version of AlienVault, due to this a backup must be completed each Monday of the environment and then stored externally, this is done to save space on the actual environment. I was shown this process by another team member and have completed this process 3 times now.

Employer comments

Student Signature: Jordan Cogan Date: 18/08/2021

Industry Supervisor Signature: David Steele 24/08/2021