National College of Ireland

Quantitative security assessment of power-grid using Common Vulnerability Scoring System (CVSS) with attack traffic analysis

Configuration Manual

MSc Research Project
Cyber Security

Vinayak Mate
Student ID: x20214791

School of Computing
National College of Ireland

Supervisor: Mr. Michael Prior

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Vinayak Mate
Student ID:	x20214791
Programme:	MSc in Cybersecurity **Year:** 2021-22
Module:	MSc Research Project
Supervisor:	Michael Prior
Submission Due Date:	15 August 2022
Project Title:	Quantitative security assessment of power-grid using Common Vulnerability Scoring System (CVSS) and attack traffic analysis
Word Count:	705
Page Count:	9

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: Vinayak Mate

Date: 15 August 2022

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

Vinayak Mate
x20214791

1 Introduction

This document is the configuration manual submitted as the supporting document for the project Quantitative security assessment of power-grid cyber-physical system (CPS) using Common Vulnerability Scoring System (CVSS) method. Documentation of all the steps taken to setup the environment, the analysis and the hardware-software configuration is detailed in this document.

2 System Configuration

2.1 Hardware Configuration

The environment was setup on a Windows laptop, with the details of the hardware configuration listed in Table 1.

Feature	Description
Operating System	Windows 11 version 21H1, Patched up to August 2022
Processor	AMD Ryzen 7 5700U @ 1.80 GHz with Radeon Graphics
System Information	64-bit OS
Memory	16 GB
Storage	150 GB OS drive; 300 GB Data drive

Table 1: Hardware Configuration

2.2 Software Configuration

The coding platform was setup on Jupyter Notebook at version 3.4.3 (*Project Jupyter*, no date). The coding was performed in Python version 3.10.5 (*Welcome to Python.org*, no date). Python libraries used are Pandas, Numpy, Matplotlib, and CVSS (Security, no date).

3 Data Selection

Network based attacks were chosen for the analysis. The attack methods selected were a DoS attack attempted through SYN packet flooding and PortMap attack for port scanning. The datasets for attack traffic were obtained from the DDoS Evaluation Dataset (CIC-DDoS2019)[1] research conducted by the Canadian Institute of Cybersecurity based in University of New Brunswick, Canada. This data was statistically analysed with respect to ideal traffic expected during a day-to-day Smart Meter operation. The ideal traffic dataset

[1] https://www.unb.ca/cic/about/index.html

was obtained from Firewall Dataset[2] collated by University of California Irvine (UCI) Machine Learning Repository.

4 Implementation

4.1 Baseline for vulnerabilities selected

a. CVSS3 module is imported in Jupyter Notebook.

```
from cvss import CVSS3
```

Figure 1: Import CVSS3 module

b. The CVSS score and severity of selected CVEs is baselined using the CVSS module and the CVSS vector published by the vendor.

```
CVE_22713 = 'CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H'
c = CVSS3(CVE_22713)

print('CVE-2021-22713 vector = ', c.clean_vector())
print('CVSS score = ', c.scores())
print('Severity is', c.severities())
```

```
CVE-2021-22713 vector =  CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H
CVSS score =  (7.5, 7.5, 7.5)
Severity is ('High', 'High', 'High')
```

Figure 2: CVSS baseline for CVE-2021-27713

```
CVE_6048 = 'CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H'
c = CVSS3(CVE_6048)

print('CVE-2017-6048 vector = ',c.clean_vector())
print('CVSS score = ', c.scores())
print('Severity is', c.severities())
```

```
CVE-2017-6048 vector =  CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H
CVSS score =  (8.8, 8.8, 8.8)
Severity is ('High', 'High', 'High')
```

Figure 3: CVSS baseline for CVE-2017-6048

4.2 Analysis of attack datasets

a. Import the Pandas module for statistical analysis and read the data from CSV files.

[2] https://archive.ics.uci.edu/ml/datasets/Internet+Firewall+Data

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

ideal=pd.read_csv('D:/NCI/5_Research/Datasets/Ideal.csv')
syn=pd.read_csv('D:/NCI/5_Research/Datasets/Syn.csv')
pm=pd.read_csv('D:/NCI/5_Research/Datasets/Portmap.csv')
```

Figure 4: Import modules and read data

b. Select attributes of interest which are the Sent and Received packets (Total Fwd Packets and Total Backward Packets) columns.

```
ideal=ideal[['pkts_sent','pkts_received']]
```

```
syn=syn[[' Total Fwd Packets', ' Total Backward Packets']]
```

```
pm=pm[[' Total Fwd Packets', ' Total Backward Packets']]
```

Figure 5: Selection of attributes of interest

c. Data cleaning performed by identifying any missing values. Missing values identified only in expected places.

```
missing_zero_values_table(ideal)
```

```
Your selected dataframe has 2 columns and 65532 Rows.
There are 1 columns that have zero and missing values. Rows.
There are 1 columns that have zero values. Rows.
There are 0 columns that have missing values.
```

	Zero_Values	Missing_Values	%_of_Total_Values	Total_Zero_Missing_Values	%_Total_Zero_Missing Values	Data_Type
pkts_sent	0	0	0.0	0	0.0	int64
pkts_received	31574	0	0.0	31574	48.2	int64

Figure 6: Missing zero values in Ideal traffic dataset

```
missing_zero_values_table(syn)
```

```
Your selected dataframe has 2 columns and 4320541 Rows.
There are 1 columns that have zero and missing values. Rows.
There are 1 columns that have zero values. Rows.
There are 0 columns that have missing values.
```

	Zero_Values	Missing_Values	%_of_Total_Values	Total_Zero_Missing_Values	%_Total_Zero_Missing Values	Data_Type
Total Fwd Packets	0	0	0.0	0	0.0	int64
Total Backward Packets	2523495	0	0.0	2523495	58.4	int64

Figure 7: Missing zero values in SYN attack dataset

```
missing_zero_values_table(pm)
```

```
Your selected dataframe has 2 columns and 191694 Rows.
There are 1 columns that have zero and missing values. Rows.
There are 1 columns that have zero values. Rows.
There are 0 columns that have missing values.
```

	Zero_Values	Missing_Values	%_of_Total_Values	Total_Zero_Missing_Values	%_Total_Zero_Missing Values	Data_Type
Total Fwd Packets	0	0	0.0	0	0.0	int64
Total Backward Packets	187468	0	0.0	187468	97.8	int64

Figure 8: Missing zero values in PortMap attack dataset

d. Descriptive statistics are generated.

```
ideal.describe(include='all')
```

	pkts_sent	pkts_received
count	65532.000000	65532.000000
mean	41.399530	61.466505
std	3218.871288	2223.332271
min	1.000000	0.000000
25%	1.000000	0.000000
50%	1.000000	1.000000
75%	3.000000	2.000000
max	747520.000000	327208.000000

Figure 9: Descriptive statistics of Ideal traffic dataset

```
syn.describe(include='all')
```

	Total Fwd Packets	Total Backward Packets
count	4.320541e+06	4.320541e+06
mean	2.903221e+00	1.150095e+00
std	1.079579e+01	7.724044e+00
min	1.000000e+00	0.000000e+00
25%	2.000000e+00	0.000000e+00
50%	2.000000e+00	0.000000e+00
75%	2.000000e+00	2.000000e+00
max	1.561400e+04	8.029000e+03

Figure 10: Descriptive statistics of SYN attack dataset

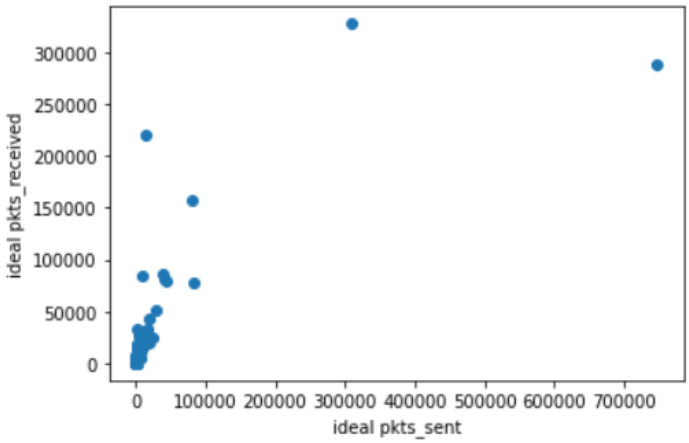
```
pm.describe(include='all')
```

	Total Fwd Packets	Total Backward Packets
count	191694.000000	191694.000000
mean	2.269868	0.381493
std	46.812214	72.545552
min	1.000000	0.000000
25%	2.000000	0.000000
50%	2.000000	0.000000
75%	2.000000	0.000000
max	20444.000000	31700.000000

Figure 11: Descriptive statistics of PortMap attack dataset

e. Scatter plot is plotted, and Pearson's correlation is calculated for Ideal traffic dataset.

```
#Relationship between ideal pkts_sent and ideal pkts_received
x = ideal['pkts_sent'].to_list()
y = ideal['pkts_received'].to_list()
plt.scatter(x, y)
plt.xlabel('ideal pkts_sent')
plt.ylabel("ideal pkts_received")
plt.show()
```



There seems to be a significant relation between two variables with a correlation of 0.77154954

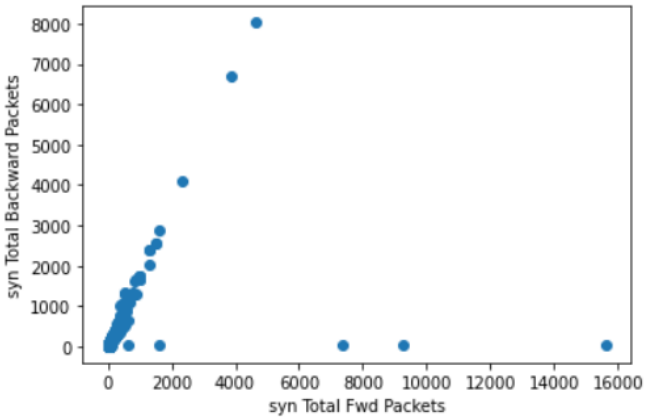
```
#Correlation
pearsons_coefficient = np.corrcoef(x, y)
print("The pearson's coeffient of the x and y inputs are: \n" ,pearsons_coefficient)
```

```
The pearson's coeffient of the x and y inputs are:
 [[1.         0.77154954]
 [0.77154954 1.        ]]
```

Figure 12: Scatter plot and Pearson's coefficient of Ideal traffic dataset

f. Scatter plot is plotted, and Pearson's correlation is calculated for SYN flood attack traffic dataset.

```
#Relationship between syn Total Fwd Packets and syn Total Backward Packets
x = syn[' Total Fwd Packets'].to_list()
y = syn[' Total Backward Packets'].to_list()
plt.scatter(x, y)
plt.xlabel("syn Total Fwd Packets")
plt.ylabel("syn Total Backward Packets")
plt.show()
```



Except few outliers, there is a linear relationship between two variables. The correlation is low because of few outliers.

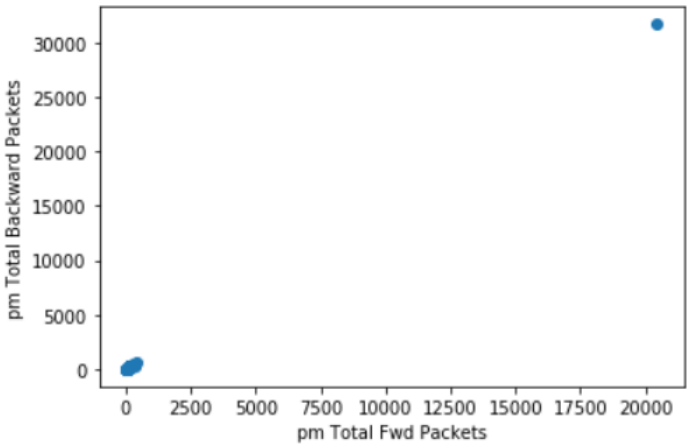
```
#Correlation
pearsons_coefficient = np.corrcoef(x, y)
print("The pearson's coeffient of the x and y inputs are: \n" ,pearsons_coefficient)
```

```
The pearson's coeffient of the x and y inputs are:
 [[1.         0.44239957]
 [0.44239957 1.        ]]
```

Figure 13: Scatter plot and Pearson's coefficient of SYN attack dataset

g. Scatter plot is plotted, and Pearson's correlation is calculated for PortMap traffic dataset.

```
#Relationship between pm Total Fwd Packets and pm Total Backward Packets
x = pm[' Total Fwd Packets'].to_list()
y = pm[' Total Backward Packets'].to_list()
plt.scatter(x, y)
plt.xlabel("pm Total Fwd Packets")
plt.ylabel("pm Total Backward Packets")
plt.show()
```



Except one outlier, all the variables are centered around zero. Correlation is also very high, i.e. 0.99973336

```
#Correlation
pearsons_coefficient = np.corrcoef(x, y)
print("The pearson's coeffient of the x and y inputs are: \n" ,pearsons_coefficient)
```

```
The pearson's coeffient of the x and y inputs are:
 [[1.         0.99973336]
 [0.99973336 1.        ]]
```

Figure 14: Scatter plot and Pearson's coefficient of PortMap attack dataset

4.3 Final CVSS score using updated Environmental Metrics

The ordinally weighted values are updated in the CVSS vector and CVSS module command is used to identify the updated CVSS severity and score. This is performed for below combinations.

a. CVE-2021-22713 with SYN attack

```
CVE_22713_syn = 'CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H/MA:H/AR:H'
c = CVSS3(CVE_22713_syn)

print('CVE-2021-22713 vector with SYN attack = ', c.clean_vector())
print('CVSS score = ', c.scores())
print('Severity is', c.severities())
```

```
CVE-2021-22713 vector with SYN attack =  CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:
U/C:N/I:N/A:H/AR:H/MA:H
CVSS score =  (7.5, 7.5, 9.3)
Severity is ('High', 'High', 'Critical')
```

Figure 15: Updated CVSS score for CVE-2021-22713 with SYN attack

b. CVE-2021-22713 with PortMap attack

```
CVE_22713_pm = 'CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H/MA:L/AR:L'
c = CVSS3(CVE_22713_pm)

print('CVE-2021-22713 vector with PortMap attack = ', c.clean_vector())
print('CVSS score = ', c.scores())
print('Severity is', c.severities())
```

```
CVE-2021-22713 vector with PortMap attack =  CVSS:3.0/AV:N/AC:L/PR:N/UI:
N/S:U/C:N/I:N/A:H/AR:L/MA:L
CVSS score =  (7.5, 7.5, 4.6)
Severity is ('High', 'High', 'Medium')
```

Figure 16: Updated CVSS score for CVE-2021-22713 with PortMap attack

c. CVE-2017-6048 with SYN attack

```
CVE_6048_syn = 'CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H/MC:H/MI:H/MA\
:H/CR:H/IR:H/AR:H'
c = CVSS3(CVE_6048_syn)

print('CVE-2017-6048 vector with SYN attack = ', c.clean_vector())
print('CVSS score = ', c.scores())
print('Severity is', c.severities())
```

```
CVE-2017-6048 vector with SYN attack =  CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/
C:H/I:H/A:H/CR:H/IR:H/AR:H/MC:H/MI:H/MA:H
CVSS score =  (8.8, 8.8, 8.8)
Severity is ('High', 'High', 'High')
```

Figure 17: Updated CVSS score for CVE-2017-6048 with SYN attack

d. CVE-2017-6048 with PortMap attack

```
CVE_6048_pm = 'CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H/MC:L/MI:L/MA\
:L/CR:L/IR:L/AR:L'
c = CVSS3(CVE_6048_pm)

print('CVE-2017-6048 vector with PortMap attack = ', c.clean_vector())
print('CVSS score = ', c.scores())
print('Severity is', c.severities())
```

```
CVE-2017-6048 vector with PortMap attack =  CVSS:3.0/AV:N/AC:L/PR:L/UI:N/
S:U/C:H/I:H/A:H/CR:L/IR:L/AR:L/MC:L/MI:L/MA:L
CVSS score =  (8.8, 8.8, 4.8)
Severity is ('High', 'High', 'Medium')
```

Figure 18: Updated CVSS score for CVE-2017-6048 with PortMap attack

5 Conclusion

The steps enumerated above were successfully implemented to perform the baseline CVSS score calculation, analyse attack traffic datasets, and calculate the updated CVSS score according to the proposed methodology in the research.

6 References

Project Jupyter (no date). Available at: https://jupyter.org (Accessed: 15 August 2022).

Security, S.K., Red Hat Product (no date) 'cvss: CVSS2/3 library with interactive calculator for Python 2 and Python 3'. Available at: https://github.com/skontar/cvss (Accessed: 15 August 2022).

Welcome to Python.org (no date) *Python.org*. Available at: https://www.python.org/ (Accessed: 15 August 2022).