Configuration Manual

MSc Research Project
Programme Name

Uche Lawrence Irobunda
X21116113

School of Computing
National College of Ireland

Supervisor: Rohit Verma

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Uche Irobunda Lawrence

Student ID: X21116113

Programme: M.Sc. Cyber Security **Year:** 2022-2023

Module: M.Sc. Research Project

Lecturer: Rohit Verma

Submission Due Date: 25/04/2023

Project Title: Securing and Detecting Attacks in Industrial IoT: An Efficient Intrusion Detection System (IDS) to detect the DOS Attack in IIoT

Word Count: 540 **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: ………

Date: ………

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

Uche Lawrence Irobunda
Student ID: X2111l113

1 Introduction

To conduct this study, I used the NSL-KDD dataset to carry out this study, which was amended to include cases of SQL injection, MITM, false data injection, and DOS attacks.
This manual will walk you through setting up your system to support the project, preventing problems from arising during implementation, and producing a better result. The manual will cover the necessary tools for the project, including the necessary hardware and software, as well as the necessary settings to be adjusted. It will also show you how to install and set up any extra dependencies that the project requires. It is designed to provide instructions and information required to build the ns3 3.26.

1.1 System Requirements:

OS: ubuntu-14.04 LTS(32 bit)
RAM: minimum 2GB
Processor: 2.5 GHz and above

2 Installation

2.1 Download the Ubuntu image for your virtual machine.

https://releases.ubuntu.com/14.04/
Follow the steps for installation.
After installation, launch the Ubuntu software on your PC/Virtual machine and run update.

2.2 Updating the Ubuntu repo and existing application.

```
//Launch your terminal and run
Sudo apt update
Sudo apy -y upgrade
```

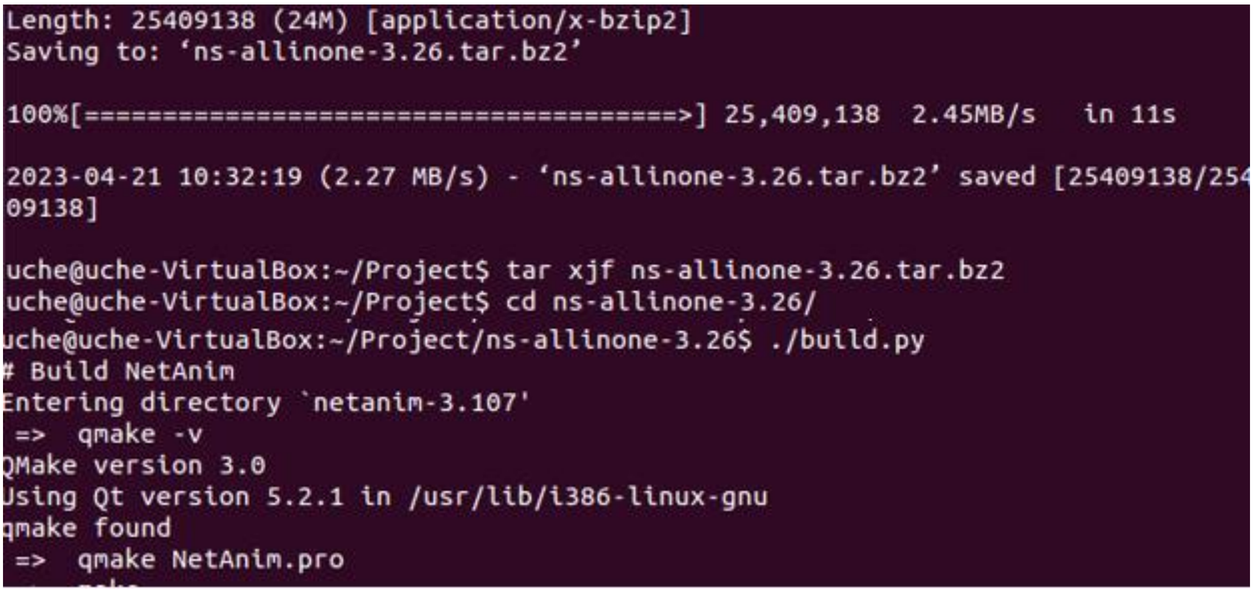
2.3 Installing core dependencies for Ns 3 build:

```
sudo apt-get install gcc g++ python python-dev mercurial bzr gdb valgrind gsl-bin libgsl0-dev libgsl0ldbl flex bison tcpdump sqlite sqlite3 libsqlite3-dev libxml2 libxml2-dev libgtk2.0-0 libgtk2.0-dev uncrustify doxygen graphviz imagemagick texlive texlive-latex-extra texlive-generic-extra texlive-generic-recommended texinfo dia texlive texlive-latex-
```

extra texlive-extra-utils texlive-generic-recommended texi2html python-pygraphviz python-kiwi python-pygoocanvas libgoocanvas-dev python-pygccxml

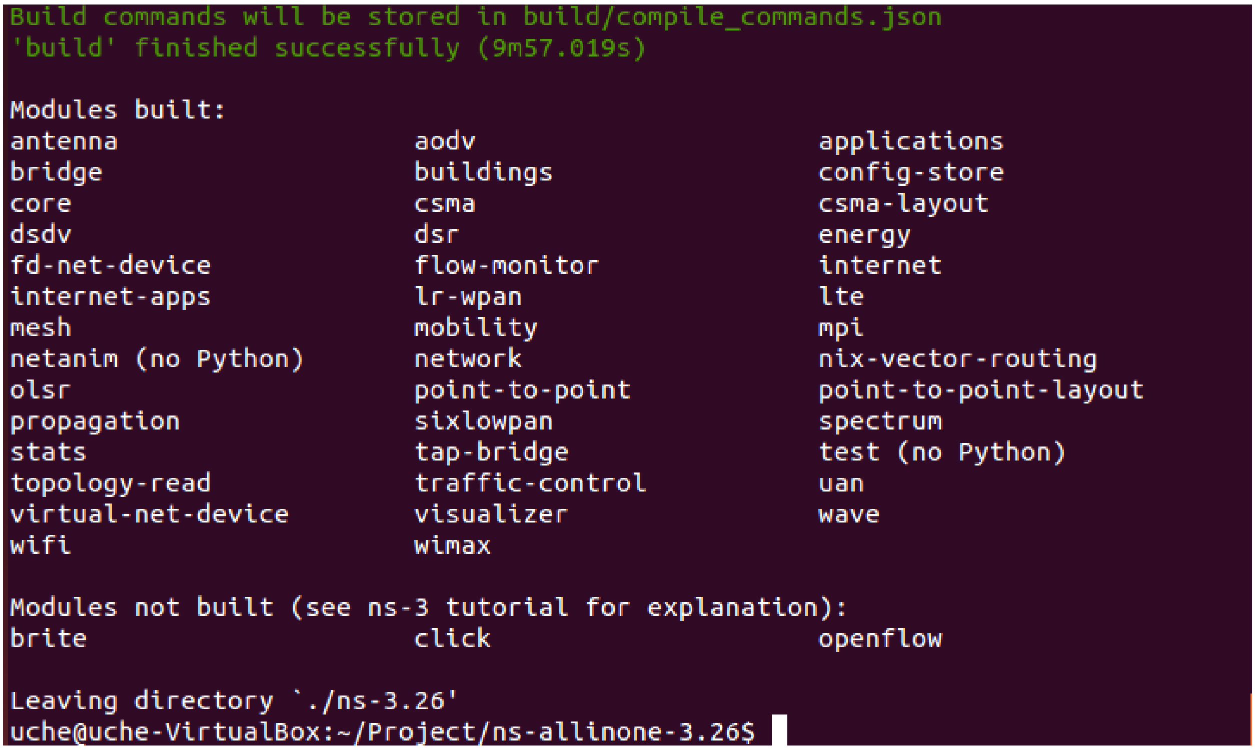
2.4 Download NS-3.26

mkdir Project // new folder (optional)
cd Project
//download the Ns3 software
wget http://www.nsnam.org/release/ns-allinone-3.26.tar.bz2
//extract the software
tar xjf ns-allinone-3.326.tar.bz2
// go to the folder directory, run
cd ns-allinone-3.26/
//build your ns3 software
./build.py



```
Length: 25409138 (24M) [application/x-bzip2]
Saving to: ‘ns-allinone-3.26.tar.bz2’

100%[======================================>] 25,409,138  2.45MB/s   in 11s

2023-04-21 10:32:19 (2.27 MB/s) - ‘ns-allinone-3.26.tar.bz2’ saved [25409138/254
09138]

uche@uche-VirtualBox:~/Project$ tar xjf ns-allinone-3.26.tar.bz2
uche@uche-VirtualBox:~/Project$ cd ns-allinone-3.26/
uche@uche-VirtualBox:~/Project/ns-allinone-3.26$ ./build.py
# Build NetAnim
Entering directory `netanim-3.107'
 =>  qmake -v
QMake version 3.0
Using Qt version 5.2.1 in /usr/lib/i386-linux-gnu
qmake found
 =>  qmake NetAnim.pro
```

Figure 1: Building Ns3.



```
Build commands will be stored in build/compile_commands.json
'build' finished successfully (9m57.019s)

Modules built:
antenna                   aodv                      applications
bridge                    buildings                 config-store
core                      csma                      csma-layout
dsdv                      dsr                       energy
fd-net-device             flow-monitor              internet
internet-apps             lr-wpan                   lte
mesh                      mobility                  mpi
netanim (no Python)       network                   nix-vector-routing
olsr                      point-to-point            point-to-point-layout
propagation               sixlowpan                 spectrum
stats                     tap-bridge                test (no Python)
topology-read             traffic-control           uan
virtual-net-device        visualizer                wave
wifi                      wimax

Modules not built (see ns-3 tutorial for explanation):
brite                     click                     openflow

Leaving directory `./ns-3.26'
uche@uche-VirtualBox:~/Project/ns-allinone-3.26$ 
```

Figure 2: Successful Build of Ns-3

2.5 Build waf

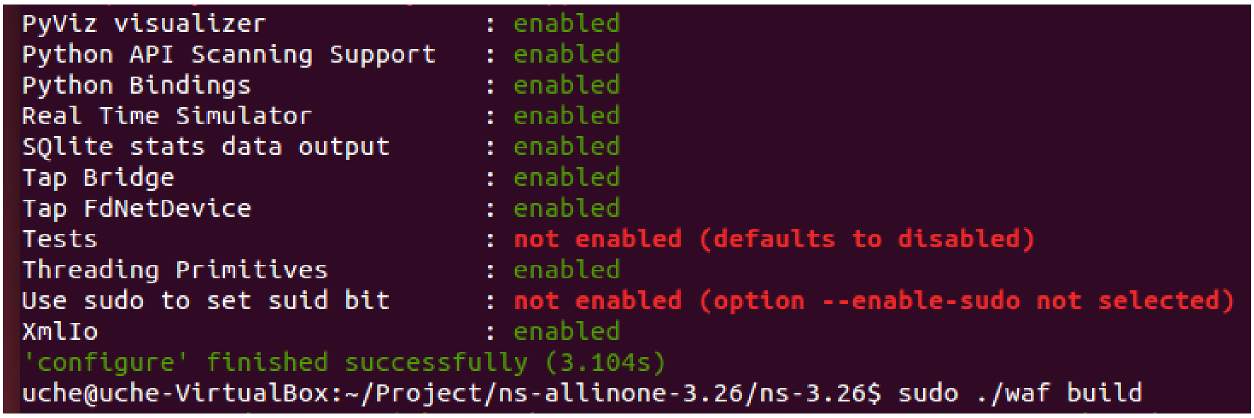
cd ns-allinone-3.26/

sudo ./waf configure

sudo ./waf build

2.6 optional

sudo./ waf check *//this will run the ns-3 tests*



```
PyViz visualizer              : enabled
Python API Scanning Support   : enabled
Python Bindings               : enabled
Real Time Simulator           : enabled
SQlite stats data output      : enabled
Tap Bridge                    : enabled
Tap FdNetDevice               : enabled
Tests                         : not enabled (defaults to disabled)
Threading Primitives          : enabled
Use sudo to set suid bit      : not enabled (option --enable-sudo not selected)
XmlIo                         : enabled
'configure' finished successfully (3.104s)
uche@uche-VirtualBox:~/Project/ns-allinone-3.26/ns-3.26$ sudo ./waf build
```

Figure 3: Running Test to Confirm All Dependencies Were Installed

2.7 Additional software required:

sudo apt-get install gnuplot

2.8 Installing Net Animator

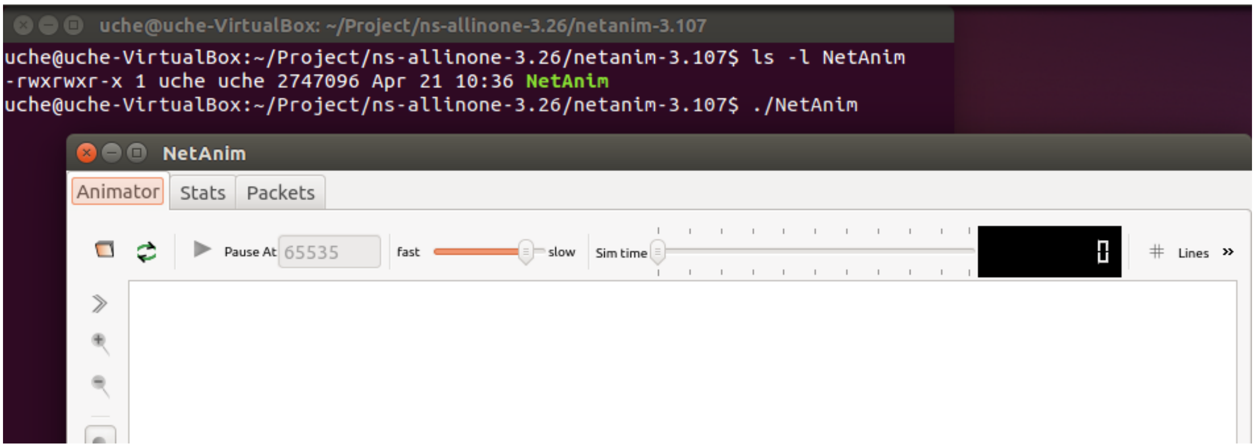
```
sudo apt-get install qt5-default
sudo cd netanim
sudo  qmake NetAnim.pro
sudo  make
```

This should create an executable named "NetAnim" in the same directory: us the ls -l to confirm the permission and owner are the same as the below.

```
$ ls -l NetAnim
-rwxr-xr-x 1 uche r john 390395 2012-05-22 08:32 NetAnim
```

2.9 launch the animator to test.

```
//launch your animator to test the installation
./NetAnim
```



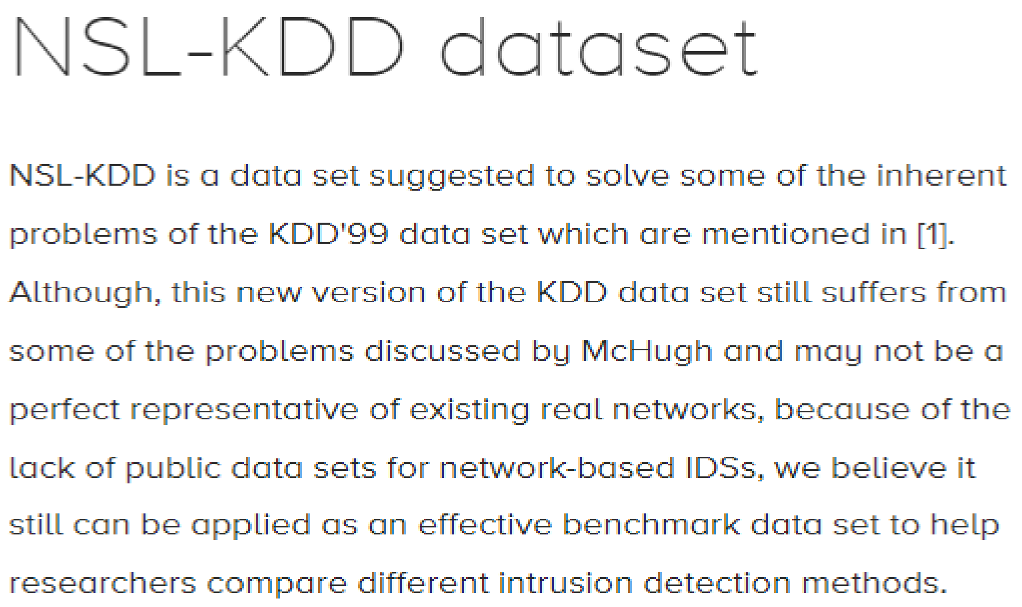
uche@uche-VirtualBox: ~/Project/ns-allinone-3.26/netanim-3.107

```
uche@uche-VirtualBox:~/Project/ns-allinone-3.26/netanim-3.107$ ls -l NetAnim
-rwxrwxr-x 1 uche uche 2747096 Apr 21 10:36 NetAnim
uche@uche-VirtualBox:~/Project/ns-allinone-3.26/netanim-3.107$ ./NetAnim
```

Figure 4: Net Amin Animator Interface

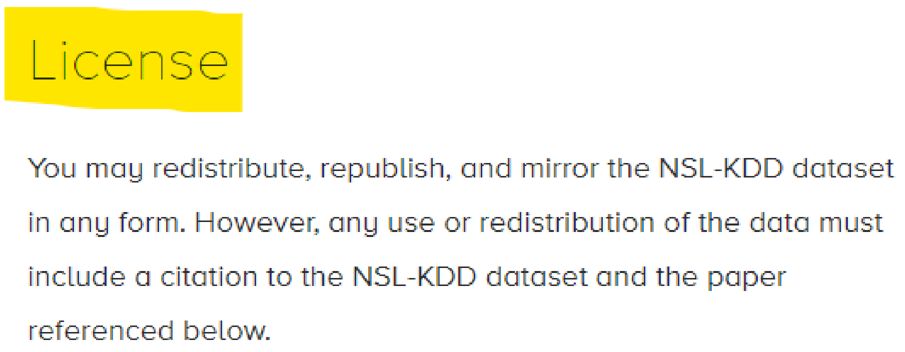
3 Dataset

This study made use of the NSL-KDD dataset. The dataset is a CSV file with regular traffic that has been modified to detect SQL injection, MITM, false data injection, and DOS attacks. It is open source and can be downloaded from the CIC website ("cic/datasets/nsl.html").



NSL-KDD dataset

NSL-KDD is a data set suggested to solve some of the inherent problems of the KDD'99 data set which are mentioned in [1]. Although, this new version of the KDD data set still suffers from some of the problems discussed by McHugh and may not be a perfect representative of existing real networks, because of the lack of public data sets for network-based IDSs, we believe it still can be applied as an effective benchmark data set to help researchers compare different intrusion detection methods.



License

You may redistribute, republish, and mirror the NSL-KDD dataset in any form. However, any use or redistribution of the data must include a citation to the NSL-KDD dataset and the paper referenced below.

4 Execution

4.1 Execution Steps

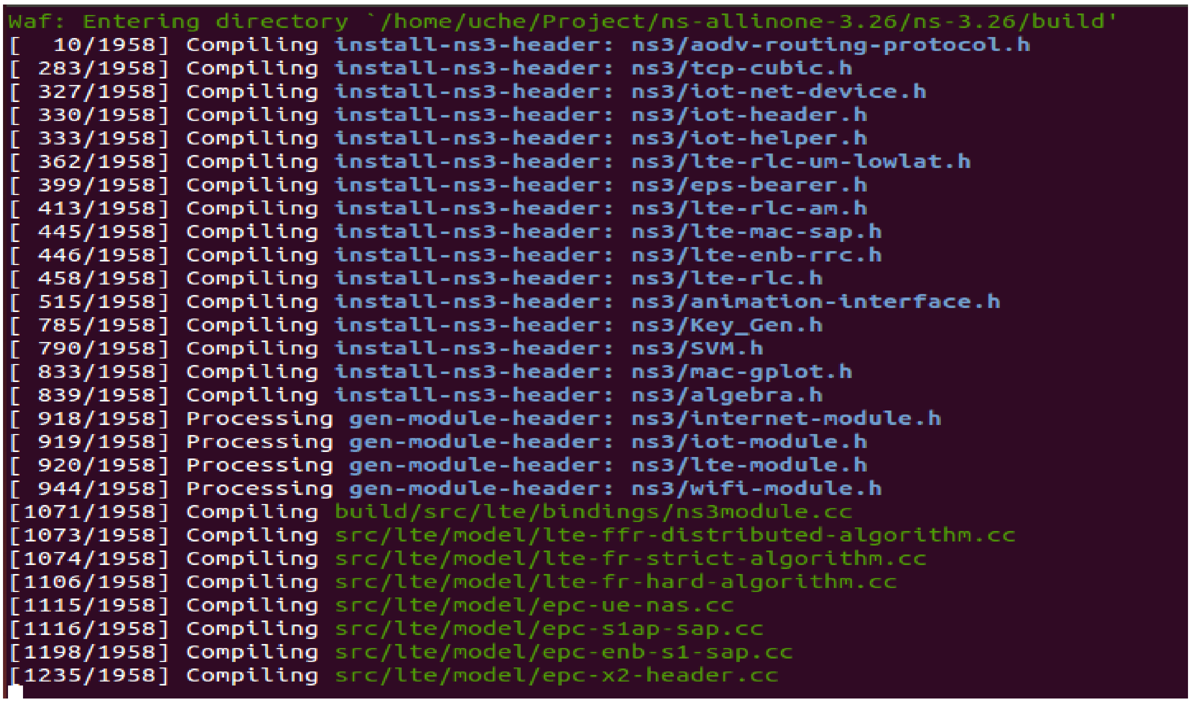
cd Project/ns-allinone-3.26/ns-3.26/

4.2 Run

//filename is the name of your .cc program you intent to build e.g. sudo ./waf --run IDS_for_DOS_Attack –vis

./waf --run “filename” –vis

Ns3 compiles the code and launches the visualizer.

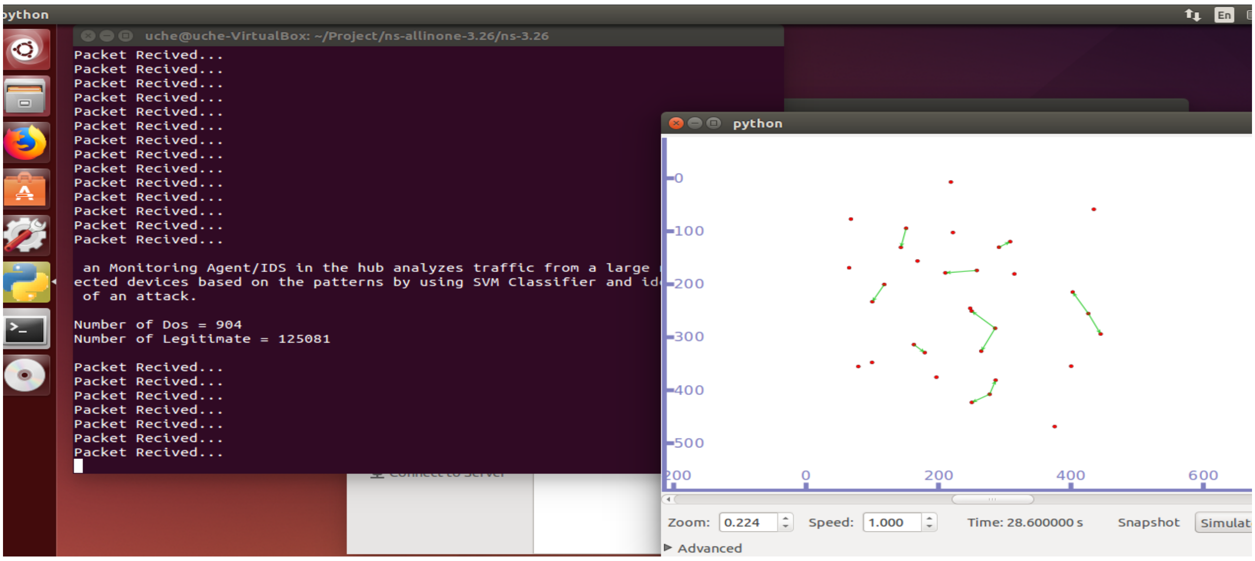


```
Waf: Entering directory `/home/uche/Project/ns-allinone-3.26/ns-3.26/build'
[  10/1958] Compiling install-ns3-header: ns3/aodv-routing-protocol.h
[ 283/1958] Compiling install-ns3-header: ns3/tcp-cubic.h
[ 327/1958] Compiling install-ns3-header: ns3/iot-net-device.h
[ 330/1958] Compiling install-ns3-header: ns3/iot-header.h
[ 333/1958] Compiling install-ns3-header: ns3/iot-helper.h
[ 362/1958] Compiling install-ns3-header: ns3/lte-rlc-um-lowlat.h
[ 399/1958] Compiling install-ns3-header: ns3/eps-bearer.h
[ 413/1958] Compiling install-ns3-header: ns3/lte-rlc-am.h
[ 445/1958] Compiling install-ns3-header: ns3/lte-mac-sap.h
[ 446/1958] Compiling install-ns3-header: ns3/lte-enb-rrc.h
[ 458/1958] Compiling install-ns3-header: ns3/lte-rlc.h
[ 515/1958] Compiling install-ns3-header: ns3/animation-interface.h
[ 785/1958] Compiling install-ns3-header: ns3/Key_Gen.h
[ 790/1958] Compiling install-ns3-header: ns3/SVM.h
[ 833/1958] Compiling install-ns3-header: ns3/mac-gplot.h
[ 839/1958] Compiling install-ns3-header: ns3/algebra.h
[ 918/1958] Processing gen-module-header: ns3/internet-module.h
[ 919/1958] Processing gen-module-header: ns3/iot-module.h
[ 920/1958] Processing gen-module-header: ns3/lte-module.h
[ 944/1958] Processing gen-module-header: ns3/wifi-module.h
[1071/1958] Compiling build/src/lte/bindings/ns3module.cc
[1073/1958] Compiling src/lte/model/lte-ffr-distributed-algorithm.cc
[1074/1958] Compiling src/lte/model/lte-fr-strict-algorithm.cc
[1106/1958] Compiling src/lte/model/lte-fr-hard-algorithm.cc
[1115/1958] Compiling src/lte/model/epc-ue-nas.cc
[1116/1958] Compiling src/lte/model/epc-s1ap-sap.cc
[1198/1958] Compiling src/lte/model/epc-enb-s1-sap.cc
[1235/1958] Compiling src/lte/model/epc-x2-header.cc
```

Figure 5: Running/Compiling the Build IDS_For_DOS_Attack

5 Evaluation

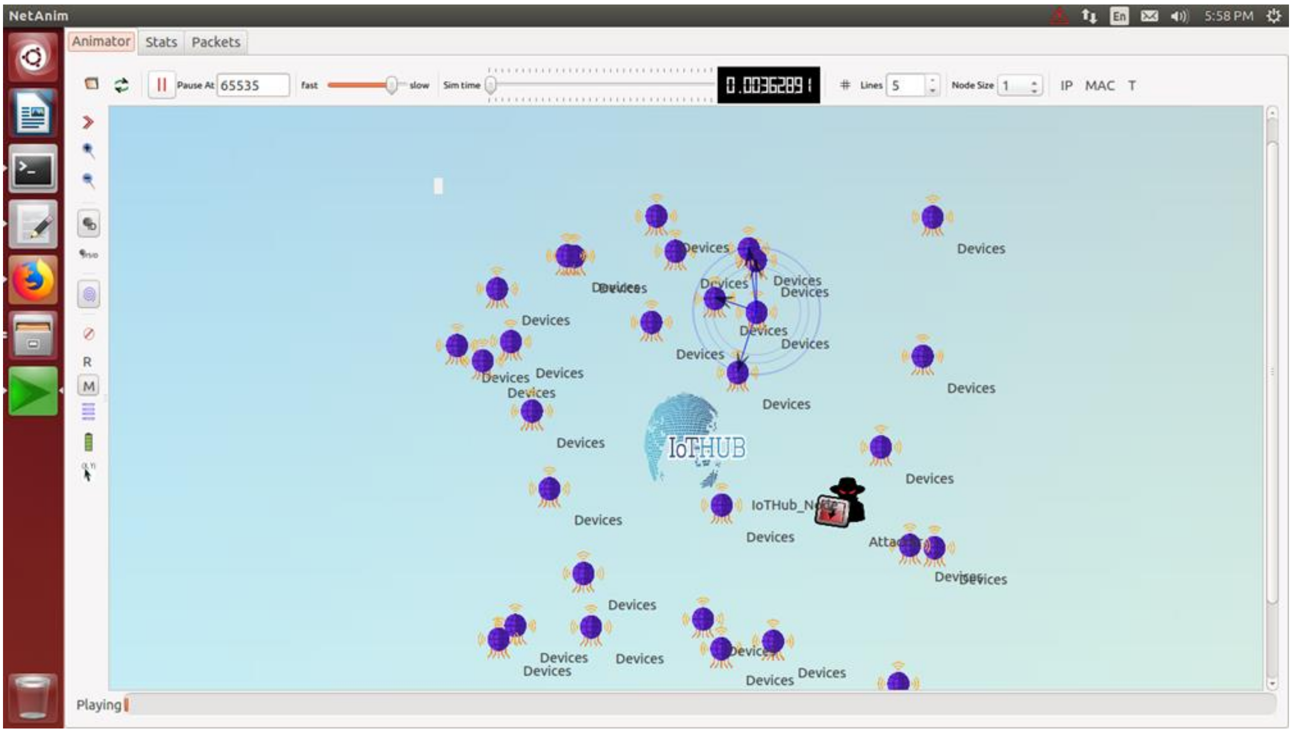
When analyzing network traffic, the code employs SVM Classifier dataset to look for indicators of an attack. The program also uses Flow Monitor to gather and examine traffic information, NetAnim to show the simulation results.



```
Packet Recived...
Packet Recived...
Packet Recived...
Packet Recived...
Packet Recived...
Packet Recived...
Packet Recived...
Packet Recived...
Packet Recived...
Packet Recived...
Packet Recived...
Packet Recived...
Packet Recived...
Packet Recived...

 an Monitoring Agent/IDS in the hub analyzes traffic from a large
ected devices based on the patterns by using SVM Classifier and id
 of an attack.

Number of Dos = 904
Number of Legitimate = 125081

Packet Recived...
Packet Recived...
Packet Recived...
Packet Recived...
Packet Recived...
Packet Recived...
Packet Recived...
```

Zoom: 0.224 Speed: 1.000 Time: 28.600000 s

Figure 6: Python Simulator Showing the Nodes.



Pause At 65535

Lines 5

Node Size 1

Figure 7: Net Animator Showing the Simulation.

6 References

NS-3 (2021) 'Installation - ns-3', [online] Available at: https://www.nsnam.org/wiki/Installation (Accessed: 21 January 2023).

NS-3. (n.d.). Animation. NS-3 Models Documentation. Available at: https://www.nsnam.org/docs/models/html/animation.html#mandatory/ (Accessed: 22 March 2023)

NS-3 Consortium. (2021). NS-3 Installation Guide (Version 3.38). Available at: https://www.nsnam.org/docs/release/3.38/installation/ns-3-installation.pdf/ (Accessed: 11 March 2023)

University of New Brunswick. Canadian Institute for Cybersecurity. (n.d.). NSL-KDD Data Set. https://www.unb.ca/cic/datasets/nsl.html/ (Accessed at: 20 March 2023)