

Siddhartha Chowdhuri (15CO246)

Aswin Manoj (15CO209)

Ved Choupane (15CO212)

DoS Attack Detection Using Machine Learning

INTRODUCTION :

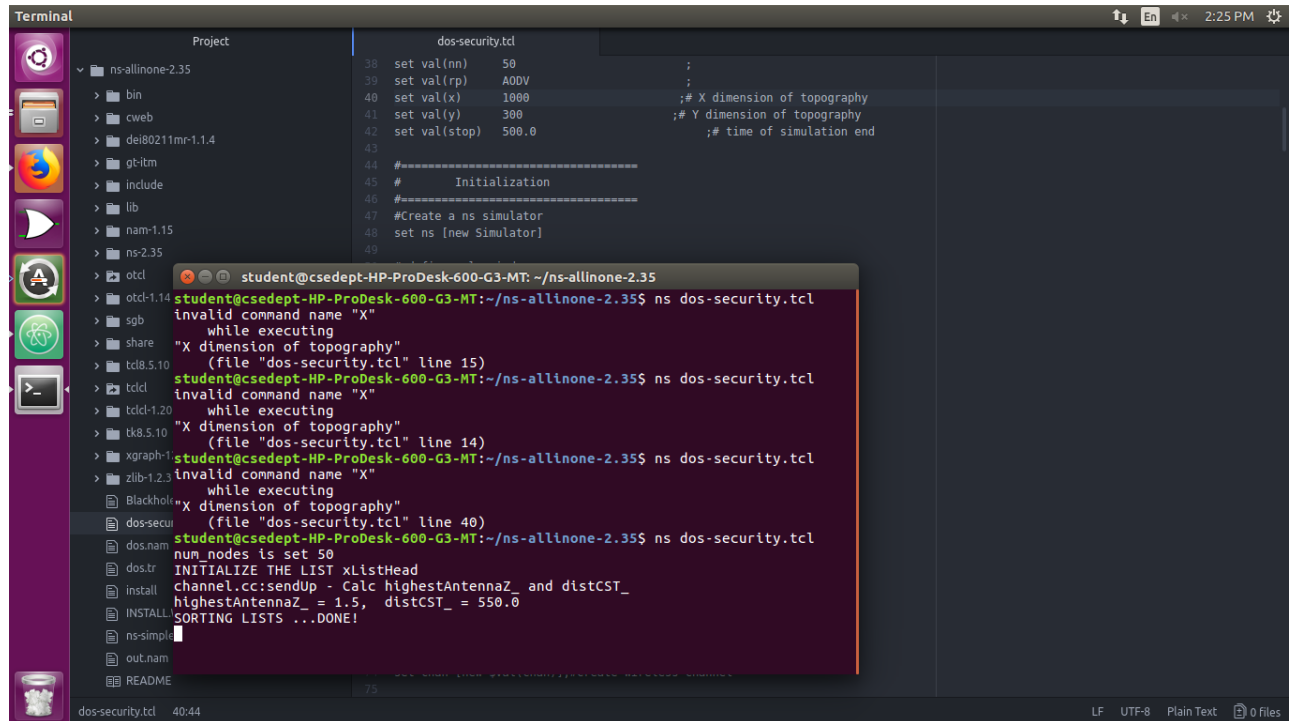
One of the major security flaws experienced by WSNs is denial of service (DoS) which can even lead to the breakdown of the complete system or to wrong decisions being made by the system that can cause adverse results. Our project aims to detect the probability of a DoS attack by using Machine Learning techniques. A sensor is an object used to gather information about a physical object or the occurrence of events. Together, many sensors can be used to collect data and communicate wirelessly to a processing station. A Wireless Sensor Network (WSN) is formed when these sensors are deployed cooperatively to monitor large physical environments. Major constraints for WSN include: security, energy (where sensor nodes are powered through either batteries or solar power), memory, computational capability and communication bandwidth.

IMPLEMENTATION :

We generated the dataset by using ns-2 which simulates a DoS attack. In the simulator we take the normalized key parameters as features for our model. We will take 100 training examples and 30 testing examples as our train-test split (70-30) and build a model with it. Primarily, we focus on doing a comparative study of models such as SVM, Neural Networks and XGBoost and build a model with the highest accuracy.

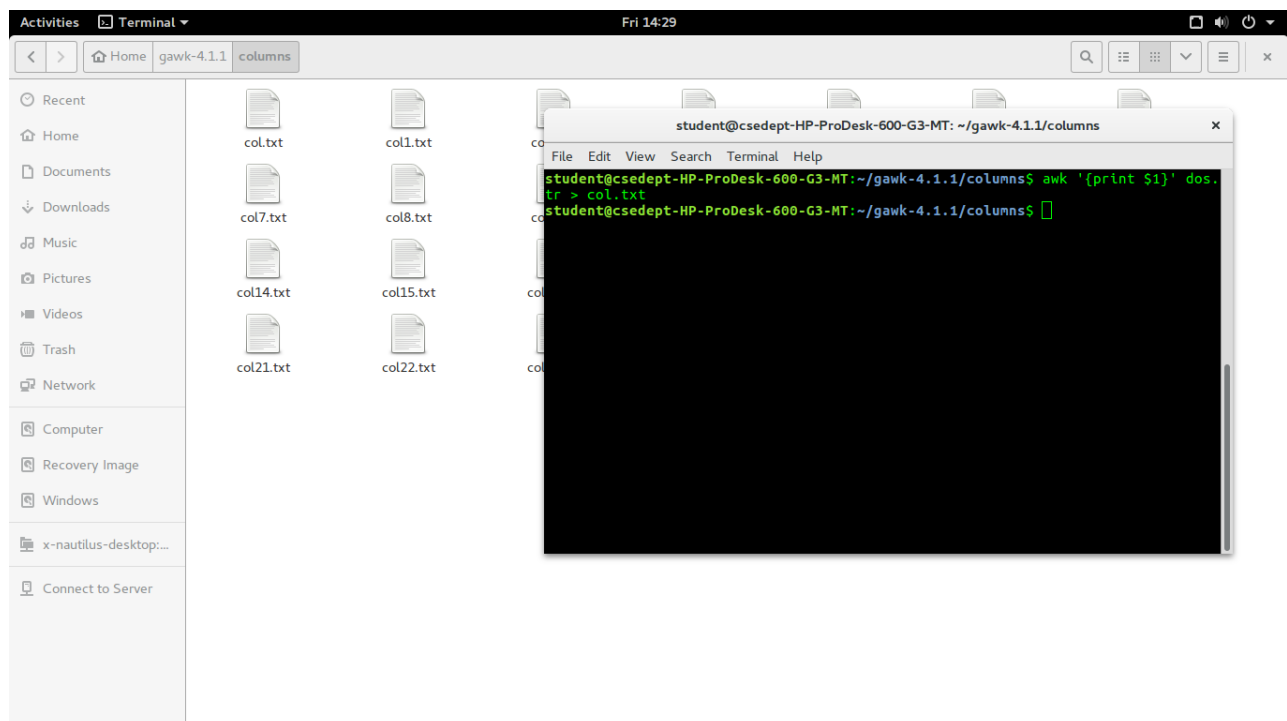
IMPLEMENTATION :

Generation of dataset is done by writing the code in ns-2 for 50 regular nodes and 1 attacker nodes. On running the code the code generates a **“.tr”** file.



```
dos-security.tcl
38 set val(nn) 50 ;
39 set val(rp) AODV ;
40 set val(x) 1000 ;# X dimension of topology
41 set val(y) 300 ;# Y dimension of topology
42 set val(stop) 500.0 ;# time of simulation end
43
44 #=====
45 # Initialization
46 #=====
47 #Create a ns simulator
48 set ns [new Simulator]
49
student@csdept-HP-ProDesk-600-G3-MT: ~/ns-allinone-2.35
student@csdept-HP-ProDesk-600-G3-MT:~/ns-allinone-2.35$ ns dos-security.tcl
invalid command name "X"
while executing
"X dimension of topology"
(file "dos-security.tcl" line 15)
student@csdept-HP-ProDesk-600-G3-MT:~/ns-allinone-2.35$ ns dos-security.tcl
invalid command name "X"
while executing
"X dimension of topology"
(file "dos-security.tcl" line 14)
student@csdept-HP-ProDesk-600-G3-MT:~/ns-allinone-2.35$ ns dos-security.tcl
invalid command name "X"
while executing
"X dimension of topology"
(file "dos-security.tcl" line 40)
student@csdept-HP-ProDesk-600-G3-MT:~/ns-allinone-2.35$ ns dos-security.tcl
num_nodes is set 50
INITIALIZE THE LIST xListHead
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 550.0
SORTING LISTS ...DONE!
```

The file is further converted into columns for every column present in the **“.tr”** file using **awk** software.



```
student@csdept-HP-ProDesk-600-G3-MT: ~/gawk-4.1.1/columns
File Edit View Search Terminal Help
student@csdept-HP-ProDesk-600-G3-MT:~/gawk-4.1.1/columns$ awk '{print $1}' dos.tr > col.txt
student@csdept-HP-ProDesk-600-G3-MT:~/gawk-4.1.1/columns$
```

Several columns are taken inside the dataset and data preprocessing is done in jupyter notebook.

```

df7 = pd.read_csv('col7.txt', sep='delimiter', header=None)
df8 = pd.read_csv('col8.txt', sep='delimiter', header=None)
df9 = pd.read_csv('col9.txt', sep='delimiter', header=None)
df10 = pd.read_csv('col10.txt', sep='delimiter', header=None)
df11 = pd.read_csv('col11.txt', sep='delimiter', header=None)
df12 = pd.read_csv('col12.txt', sep='delimiter', header=None)
df13 = pd.read_csv('col13.txt', sep='delimiter', header=None)
df14 = pd.read_csv('col14.txt', sep='delimiter', header=None)
df15 = pd.read_csv('col15.txt', sep='delimiter', header=None)
df16 = pd.read_csv('col16.txt', sep='delimiter', header=None)
df17 = pd.read_csv('col17.txt', sep='delimiter', header=None)
df18 = pd.read_csv('col18.txt', sep='delimiter', header=None)
df19 = pd.read_csv('col19.txt', sep='delimiter', header=None)
df20 = pd.read_csv('col20.txt', sep='delimiter', header=None)
df21 = pd.read_csv('col21.txt', sep='delimiter', header=None)
df22 = pd.read_csv('col22.txt', sep='delimiter', header=None)
df23 = pd.read_csv('col23.txt', sep='delimiter', header=None)
df24 = pd.read_csv('col24.txt', sep='delimiter', header=None)
df25 = pd.read_csv('col25.txt', sep='delimiter', header=None)

In [9]:
df1.columns = ["Column1"]
df2.columns = ["Column2"]
df3.columns = ["Column3"]
df4.columns = ["Column4"]
df5.columns = ["Column5"]
df6.columns = ["Column6"]
df7.columns = ["Column7"]
df8.columns = ["Column8"]
df9.columns = ["Column9"]
df10.columns = ["Column10"]
df11.columns = ["Column11"]
df12.columns = ["Column12"]
df13.columns = ["Column13"]
df14.columns = ["Column14"]
df15.columns = ["Column15"]
df16.columns = ["Column16"]
df17.columns = ["Column17"]
df18.columns = ["Column18"]
df19.columns = ["Column19"]
df20.columns = ["Column20"]
df21.columns = ["Column21"]
df22.columns = ["Column22"]
df23.columns = ["Column23"]
df24.columns = ["Column24"]
df25.columns = ["Column25"]

In [10]:
frames = [df1, df2, df3, df4, df5, df6, df7, df8, df9, df10, df11, df12, df13, df14, df15, df16, df17, df18, df19, df20, df21, df22, df23, df24, df25]

In [11]:
df = pd.concat(frames, axis=1)

In [12]:
df.head(5)

Out[12]:


|   | Column1 | Column2  | Column3 | Column4 | Column5 | Column6 | Column7 | Column8 | Column9 | Column10 | ... | Column16 | Column17 | Column18 | Column19 | Column20 | Column21 | Column22 | Column23 | Column24 | Column25 |
|---|---------|----------|---------|---------|---------|---------|---------|---------|---------|----------|-----|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 0 | s       | 3.000000 | 15.     | AGT     | ---     | 0       | tcp     | 40      | 0       | ...      | 32  | 0        | ...      | 32       | 0        | ...      | 32       | 0        | ...      | 32       | 0        |
| 1 | r       | 3.000000 | 15.     | RTR     | ---     | 0       | tcp     | 40      | 0       | ...      | 32  | 0        | ...      | 32       | 0        | ...      | 32       | 0        | ...      | 32       | 0        |
| 2 | s       | 3.000000 | 15.     | RTR     | ---     | 0       | AODV    | 48      | 0       | ...      | 30  | 0        | ...      | 30       | 0        | ...      | 30       | 0        | ...      | 30       | 0        |
| 3 | s       | 3.000399 | 15.     | MAC     | ---     | 0       | AODV    | 106     | 0       | ...      | 30  | 0        | ...      | 30       | 0        | ...      | 30       | 0        | ...      | 30       | 0        |
| 4 | r       | 3.000916 | 41.     | MAC     | ---     | 0       | AODV    | 48      | 0       | ...      | 30  | 0        | ...      | 30       | 0        | ...      | 30       | 0        | ...      | 30       | 0        |



5 rows x 25 columns

In [14]:
#df.to_csv('submission1.csv', index = False)

In [ ]:

```

THINGS DONE :

- Generation of Dataset using ns-2
- Used awk to extract the texts
- Data Preprocessing is still ongoing using Python frameworks.

THINGS TO DO:

- Finish Preprocessing
- Use Feature Extraction.
- Train and test models and compare their accuracy.

REFERENCES AND WORKS :

Security Enhancement in Wireless Sensor Networks using Machine Learning by Aswathy B. Raj.

The paper achieved an accuracy of 97% . We plan to improve the accuracy by our Machine Learning Models.

