

Detection of Brute-Force Attacks in End-to-End Encrypted Network Traffic

Team Members:

Manisha Mahapatra
[CS22MTECH14009]

Julakuntla Madhuri
[CS20BTECH11023]

Guidance:

Prof. Bheemarjuna Reddy
Tamma

Prof, CSE-IITH

(Course Instructor)

T.A. Mentor:

Harinder Kaur

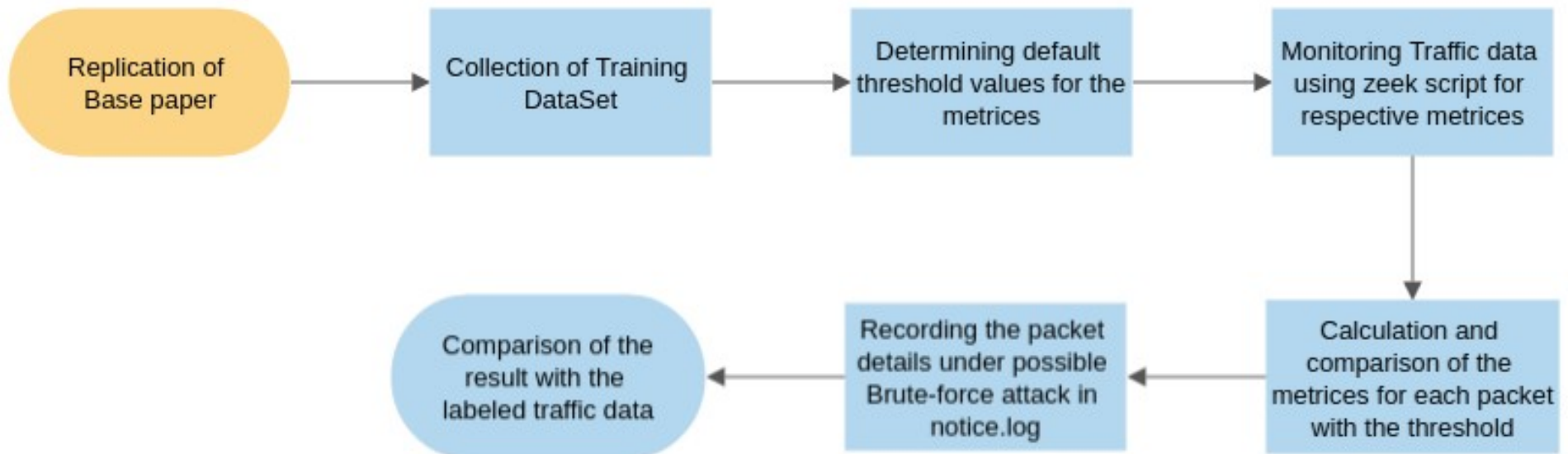
Problem Statement



- Intrusion detection systems (IDSs) are extensively used by organisations to safeguard their IT systems against assaults, but they are constrained by the rising usage of encrypted communication on the internet, which makes network IDSs (NIDSs) less effective.
- To address this constraint, organisations can utilise man-in-the-middle proxies in conjunction with NIDSs to preserve some network traffic analysis capabilities for encrypted data, but this involves issuing TLS certificates that clients can trust. However, network infrastructure providers have limited influence over destination systems, and VPNs and Tor exit nodes may be incentivized to avoid assaults in encrypted data.
- This paper provides a novel method for detecting brute-force assaults in encrypted network traffic. The method employs five novel indications to quantify traffic patterns vital for detecting brute-force attacks without the need for decryption of encrypted payload.

Project Description

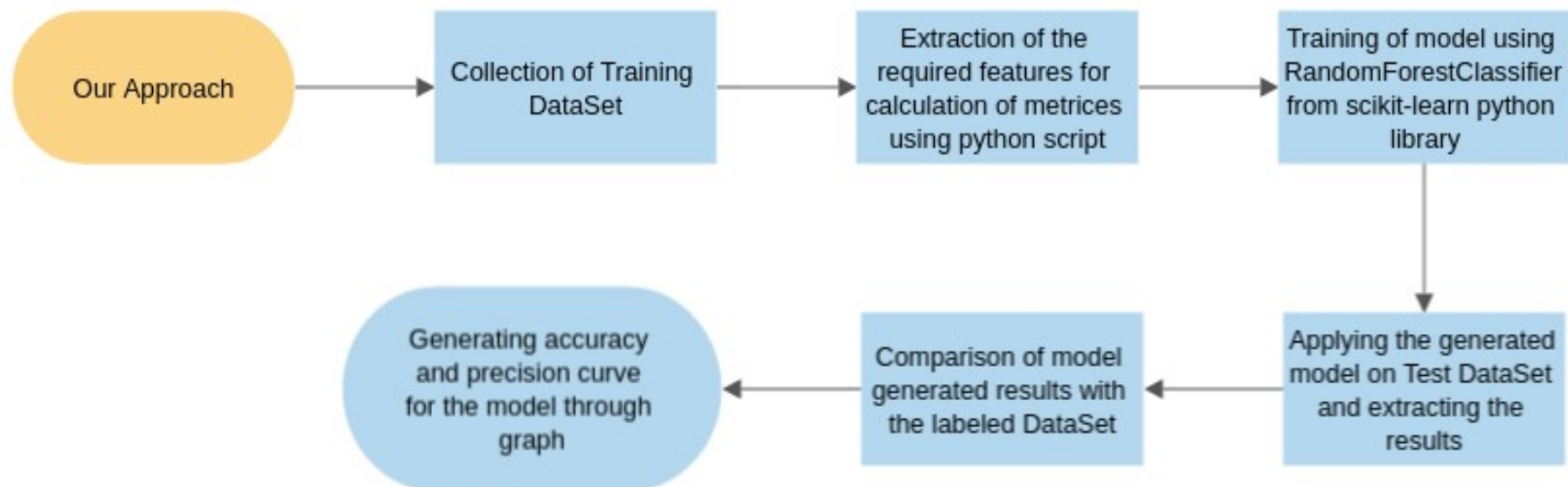
- 1) The goal was to implement mechanism for Brute-force attack detection on the basis of specified metrics and their threshold obtained through analysis of known traffic dataset.
- 2) We have implemented the setup through two scenarios:
 - Implementing the setup using zeek scripting language which directly monitors the traffic from the pcap file on the basis of predefined threshold values. This provides a complete replication of the method proposed by our base paper.



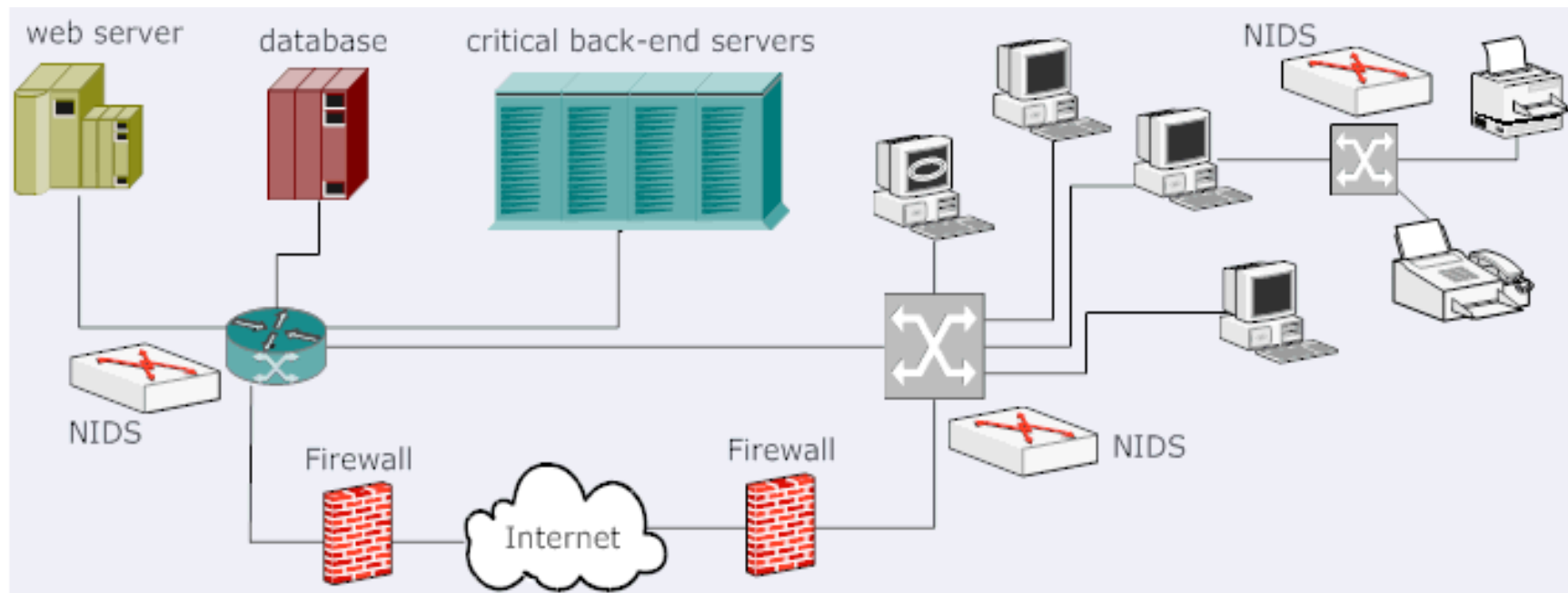
Project Description

- Implementation of mechanism for Brute-force attack detection through Random-Forest Model with the help of python scripting. This approach is aimed towards automating the process for detection of Brute-force attack packets with the help of a trained classifier model and thus avoiding any hard-coded values.

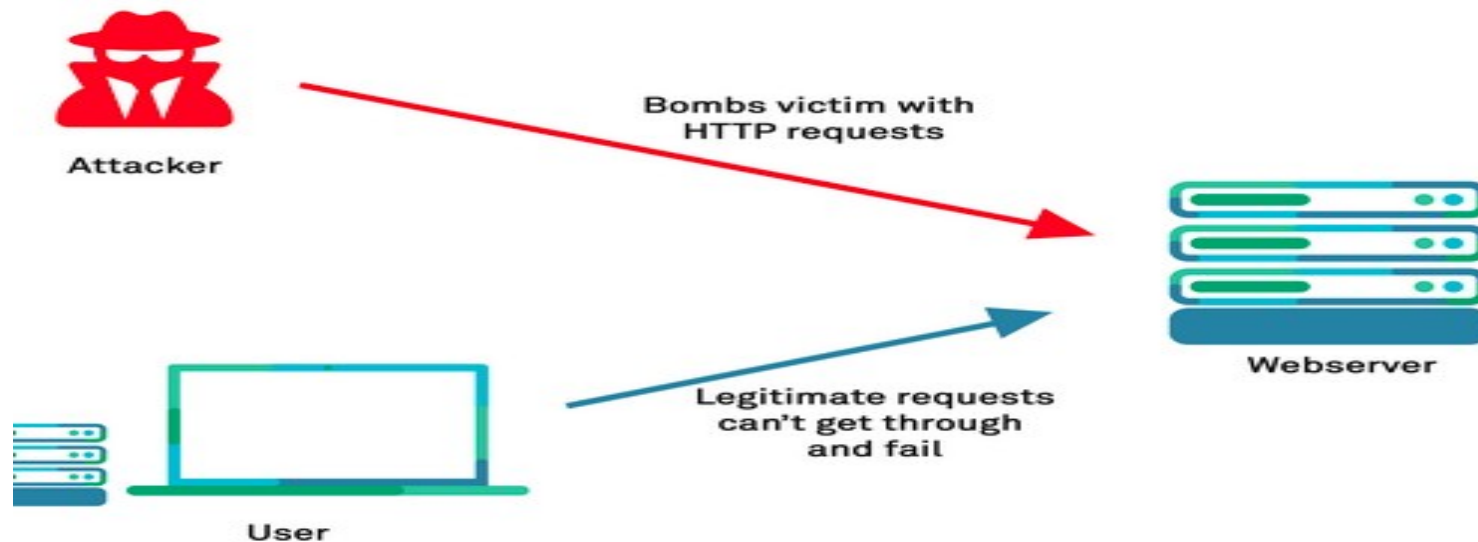
Logs are generated to obtain relevant information as per script execution. Result was compared with the labelled dataset to check for accuracy and precision with the help of relevant graphs.



Network based Intrusion Detection System (NIDS)



Brute-force attack



Project Implementation details

A yellow pencil and a pink eraser are positioned in the top right corner of the slide, appearing to be part of the presentation's design.

- Metrics for Brute-force attack detection
- Modes of applying metrics
- Collection of Traffic Data
- Implementation of base paper method
- Implementation of modification based on our approach

Metrics for Brute-force attack detection



- $m_{\text{esr}}(t)$ = no. of equally sized response packets within a specific time interval t
- $m_{\text{ssr}}(t, d)$ = no. of similarly sized responses within a specific time interval t , assuming that the size of responses differs at most by d bytes
- $m_{\text{cc}}(t)$ = no. of TCP connections within a specific time interval t (connection count)
- $m_{\text{pc}}(t)$ = no. of TCP packets within a specific time interval t (packet count)
- $m_{\text{sc}}(t, s)$ = no. of short-living TCP connections within a specific time interval t (short connection)

Modes of applying metrics



- The attack detection method based on these metrics can be applied in three different modes:
 - Source-oriented (applied per source) : classifies traffic sources as attackers if the metric's value for that source exceeds a threshold.
 - Destination-oriented (applied per destination) : classifies destination as being attacked, especially useful when information on traffic sources is not available
 - Both source- and destination-oriented : allows to detect distributed brute-force attacks as well as to track the origins of attacks performed from individual systems.

Collection of Traffic Data



- Requirement of the traffic dataset pcap and csv was to include Brute-force attacks on the packets sent over different protocols including HTTPS (considered in this project).
- For this purpose, we have considered CICIDS2017 dataset which contains benign and the most up-to-date common attacks, which resembles the true real-world data (PCAPs). It also includes CSV files for machine and deep learning purpose.
- Thursday, July 6, 2017 (Morning) -> Web Attack – Brute Force (9:20 – 10 a.m.)

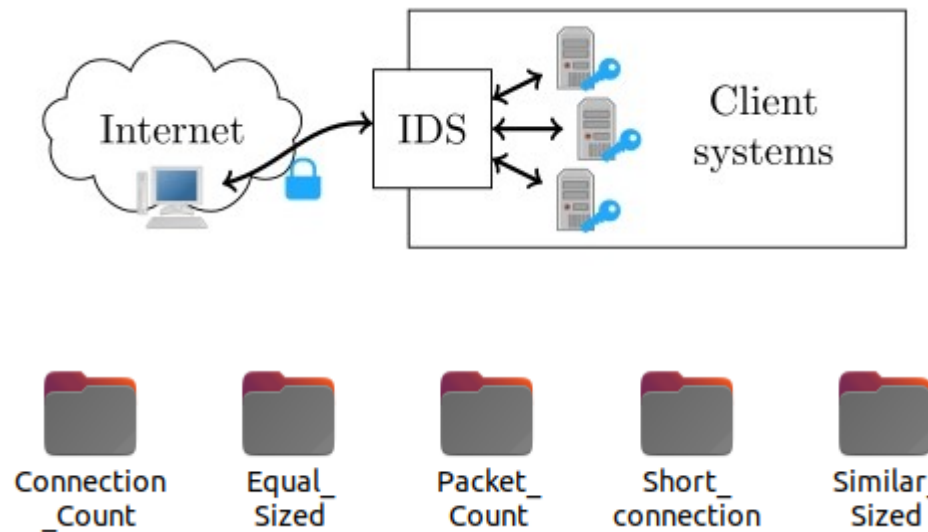
Attacker: Kali, 205.174.165.73

Victim: WebServer Ubuntu, 205.174.165.68 (Local IP 192.168.10.50)

- Initial evaluation of metrics on synthetic traffic with existing dataset.

Implementation of base paper method

- We have implemented mechanism for detection of Brute-force attack as per the approach described in the base paper using Zeek scripting language.
- This involved hardcoding of threshold values for the metrics on the basis of which the traffic packets were compared to detect Brute-force attack.
- As per the setup, traffic between clients is encrypted. The Intrusion Detection System analyzes it without decrypting the traffic.
- Helps to detect attacks and avoid higher computational cost & overhead from decrypting the Brute-force traffic.





Zeek code implementation

Challenges Faced



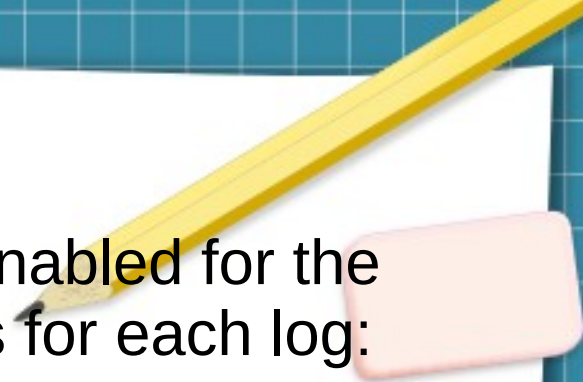
- Implemented zeek setup and related dependencies with python. To use Zeek with Python, installed Zeek Python API i.e. pyzeek. Setup not successful due to pip issue. The same was not available in conda.

```
File "/usr/local/lib/python3.8/dist-packages/pyOpenSSL-23.1.1-py3.8.egg/OpenSSL/
L/__init__.py", line 8, in <module>
    from OpenSSL import SSL, crypto
File "/usr/local/lib/python3.8/dist-packages/pyOpenSSL-23.1.1-py3.8.egg/OpenSSL
L/SSL.py", line 19, in <module>
    from OpenSSL.crypto import (
File "/usr/local/lib/python3.8/dist-packages/pyOpenSSL-23.1.1-py3.8.egg/OpenSSL
L/crypto.py", line 3258, in <module>
    utils.deprecated(
TypeError: deprecated() got an unexpected keyword argument 'name'
```

- Used VM for setting up the environment (with Zeek and Python integration), but the configuration still faced issues due to no **prof.log** , **packet_filter.log**, **loaded_scripts.log**

Updated **'/opt/zeek/etc/zeekctl.cfg'** to enable Logging and set below parameters to **'1'**:

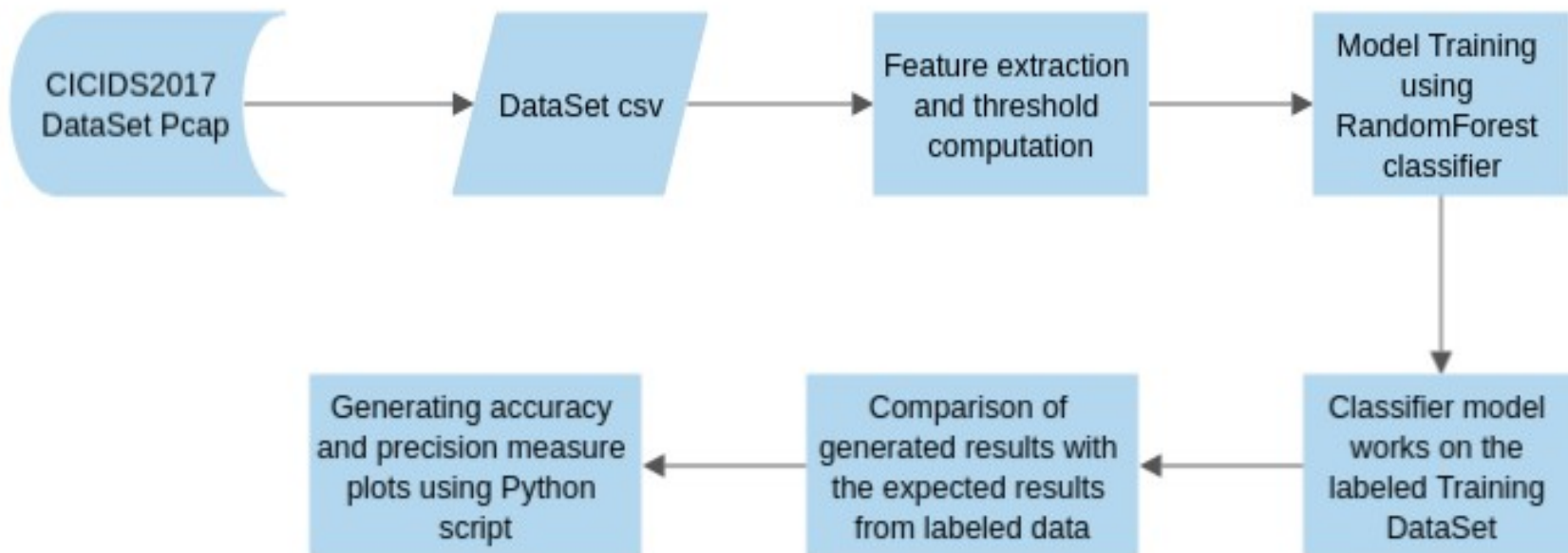
CompressLogs, LogAscii, LogBinary

- 
- Checking Zeek scripts to ensure that logging is enabled for the specific logs. The paths to the relevant script files for each log:
 - **prof.log:** /opt/zeek/share/zeek/site/local.zeek
 - **packet_filter.log:** /opt/zeek/share/zeek/site/local.zeek
 - **loaded_scripts.log:** /opt/zeek/share/zeek/site/local.zeek
 - Setting up PATH variable at **~/.bashrc** using super-user, still the integration issue could not be resolved
 - Similar issue observed for virtual environment in local system using conda:

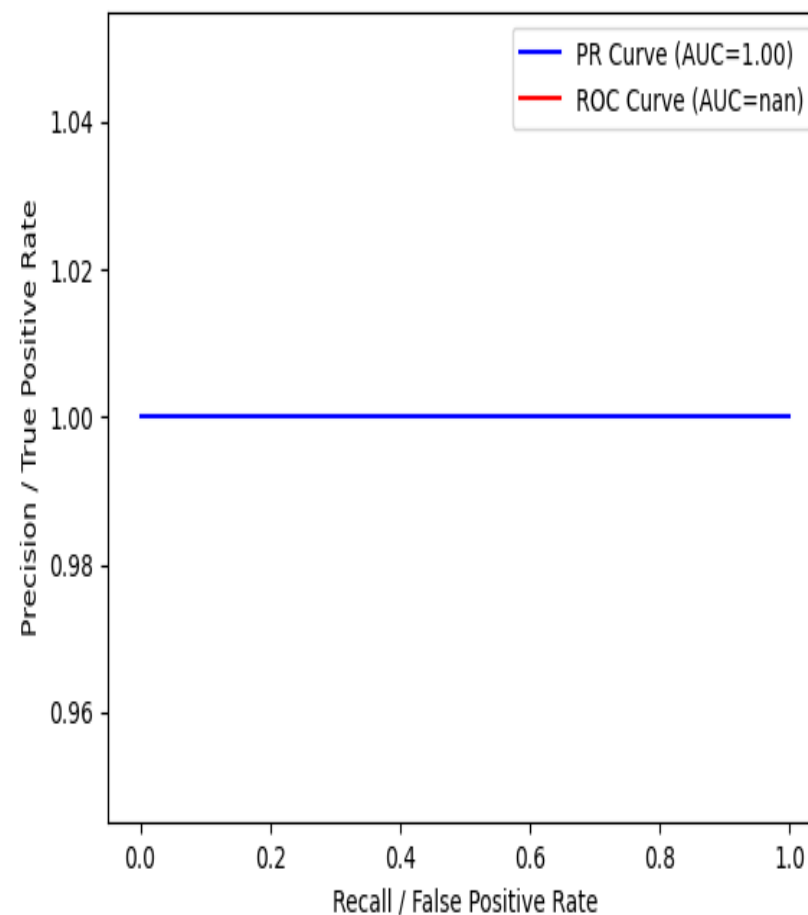
(myenv) manisha@manisha-HP-Laptop-15-bs1xx:~/Documents/Sem2/NS/TermProject\$ zeek zk1.zeek
fatal error in ./zk1.zeek, line 16: can't find python3

- Issue with **CICIDS2017** dataset PCAP due to large size (10Gb), could not be handled with WireShark. Using tcpdump, this was segmented into smaller PCAPs.
- Format issue with captured Traffic Data, resolved by conversion into csv format when using our modified Python script.

Implementation of modification based on our approach



```
/home/manisha/miniconda3/envs/myenv/lib/python3.8/site-packages/sklearn/metrics/_ranking.py:990: UndefinedMetricWarning: No negative samples in y_true, false positive value should be meaningless
  warnings.warn(
(myenv) manisha@manisha-HP-Laptop-15-bs1xx:~/Documents/Sem2/NS/TermProject/OurImpl/Approach_2/Scripts$ python3 tstMain.py
-----First-----
-----fE1-----
-----fE2-----
-----Second-----
-----tM1-----
Accuracy: 1.00
-----tM2-----
-----Third-----
Prediction for Brute-force attack starting
Prediction for Brute-force attack completed
Computing precision-recall curve and area under the curve
Computing ROC curve and area under the curve
/home/manisha/miniconda3/envs/myenv/lib/python3.8/site-packages/sklearn/metrics/_ranking.py:990: UndefinedMetricWarning: No negative samples in y_true, false positive value should be meaningless
  warnings.warn(
█
```



Conclusion



- Thus from the two approaches we conclude that using the proposed method can help in detection of Brute-force attacks and thus generating alerts. This would help to reduce computational costs involved in decrypting the Brute-forced packets and hence reduces overhead involved.
- Also on using our proposed ML-based classifier method can help to avoid setting any hardcoded thresholds for the detection and thus automating the process. Moreover, this also causes increase in precision and accuracy of the model.

Future Scope



- The current ML-based implementation can be extended to include other protocols like FTP, SSH, etc. as our current work is for HTTPS Traffic only.
- This implementation can also be extended to work on live Traffic as we work on stored Traffic Pcap only as a part of our implementation.

This involves better integration between Python and Zeek such that Zeek based network monitoring & analysis capabilities can be used with Python scripts.

- Any other better performing ML Algorithm can be used in place of RandomForestClassifier algorithm which has been applied as part of our implementation. This can help to achieve a better performing model.
- This mechanism can be further implemented as a setup for real-world traffic for detection of Brute-force attacks in case of encrypted traffic.



References

Base paper : Pascal Wichmann, Matthias Marx, Hannes Federrath, Mathias Fischer, “Detection of Brute-Force Attacks in End-to-End Encrypted Network Traffic”, ACM Digital Library, 2021.

Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani, “Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization”, 4th International Conference on Information Systems Security and Privacy (ICISSP), Portugal, January 2018.

www.wikipedia.org

<https://cloud.smartdraw.com/editor.aspx?templateId=490dad73-de30-42bf-9a58-1789d56c1afd&flags=128#depold=44970663&credID=-47961954>

<https://tales-from-a-security-professional.com/intrusion-detection-system-have-they-become-useless-2ecf51488fed>

Team Contribution



- **Manisha:** Finalizing the base paper for project implementation; Collection for attack DataSet for the implementation; Implementation of the base paper method using zeek; Worked on integration of zeek with python in local system, local VM, virtual environment; Worked on implementation of python based modified approach, generated related logs and graphs; Involved in writing the report and the presentation making (Mid-Term & Final).
- **Madhuri:** Went through available materials related to the project implementation; worked on Zeek implementation and integration of zeek with python through pyzeek in local system, involved in part of presentation and report writing.

Thank-you

