

INTRODUCTION

- What is an IDS?
- *Intrusion Detection System* monitors network traffic for signs of malicious activity.
- Two types:
 - *Rule-based IDS* (e.g., Snort): Uses predefined attack signatures.
 - *AI-based IDS*: Uses machine learning to detect unknown patterns.



INTRODUCTION

- **Project Purpose**
 - Compare Snort (traditional) vs. an AI-based IDS (custom-trained).
 - Evaluate which system is better in detecting modern cyber threats.
 - Test using *CIC-IDS2017* dataset.
 - Assess performance, adaptability, false positives, and ease of use.





PROJECT OBJECTIVES

- Detect different cyberattack types using both systems.
- Test performance in terms of:
- Detection accuracy
- Setup complexity
- Ability to adapt to new threats
- Understand each system's limitations and practical use.

Dataset and Environment

- **CIC-IDS2017 Dataset**
- Real-world traffic + labeled attacks
- Types of attacks: DDoS, brute-force, infiltration, botnet
- **Testing Setup**
- Ubuntu VM with:
 - Snort (rule-based IDS)
 - Python & Scapy (attack simulation)
 - tcpreplay (replay PCAP)
 - Wireshark (packet capture)



SNORT IMPLEMENTATION

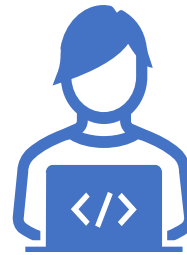


Custom Snort Rules Created

SSH Brute Force Detection

MySQL Root Login Attempt

Unencrypted MySQL Password



Example Rule:

Detection of SSH brute force attacks:

```
alert tcp any any -> any 22 (msg:"SSH Brute-force detected"; flags:S; threshold:type threshold, track by_src, count 60, seconds 60; sid:1000004; rev:1;)
```

[illegible]

```

● snort.service - LSI: Lightweight network intrusion detection system
Loaded: loaded (/etc/init.d; snort; generated)
Active: active (running) since Mon 2023-03-03 07:37:39 EST; 15s ago
Docs: man:snortd.srv.generator(3)
Process: 14546 /usr/sbin/etc/init.d/snort start (code=exited, status=0/SUCCESS)
Tasks: 4 (limit: 65536)
Memory: 159.6M (peak: 176.0M)
CPU: 1.36s
CGroup: /system.slice/snort.service
└─45462 /usr/sbin/snort -n 827 -d -d -l /var/log/snort -v snort -s snort -pid-path /run/snort/ < /etc/snort/snort.conf -i "HOME_NET=192.168.0.0/16" -l /tmp/3
└─45462 /usr/sbin/snort -n 827 -d -d -l /var/log/snort -v snort -s snort -pid-path /run/snort/ < /etc/snort/snort.conf -i "HOME_NET=192.168.0.0/16" -l /tmp/3

Preprocessor Object: SF_DBDP2C Version 1.1 -build 3e
Preprocessor Object: SF_DMP3 Version 1.1 -build 1e
Preprocessor Object: SF_SOF Version 1.1 -build 1e
Preprocessor Object: SF_DNS Version 1.1 -build 4e
Preprocessor Object: SF_HEURATDION Version 1.1 -build 1e
Preprocessor Object: SF_SMP Version 1.1 -build 1e
Preprocessor Object: SF_PDP Version 1.0 -build 1e
Compiling packet processing (pid=45462)
Mar 03 07:37:39 ubuntu-virtualbox systemd[1]: Started snort.service - LSI: Lightweight network intrusion detection system.

```

SNORT IMPLEMENTATION



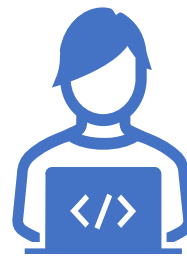
Testing with Scapy

Sent traffic matching attack profiles to trigger alerts.

Verified results using Wireshark and log inspection.

Alerts logged correctly for:

- Repeated login attempts
- Root login attempts
- Plaintext password strings

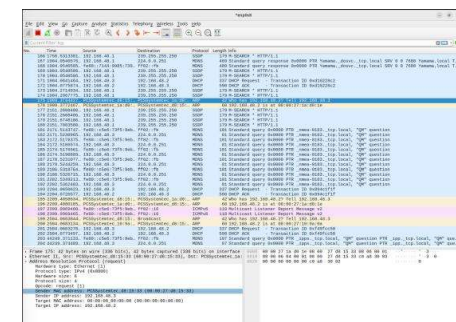


Example Rule:

For SSH brute force:

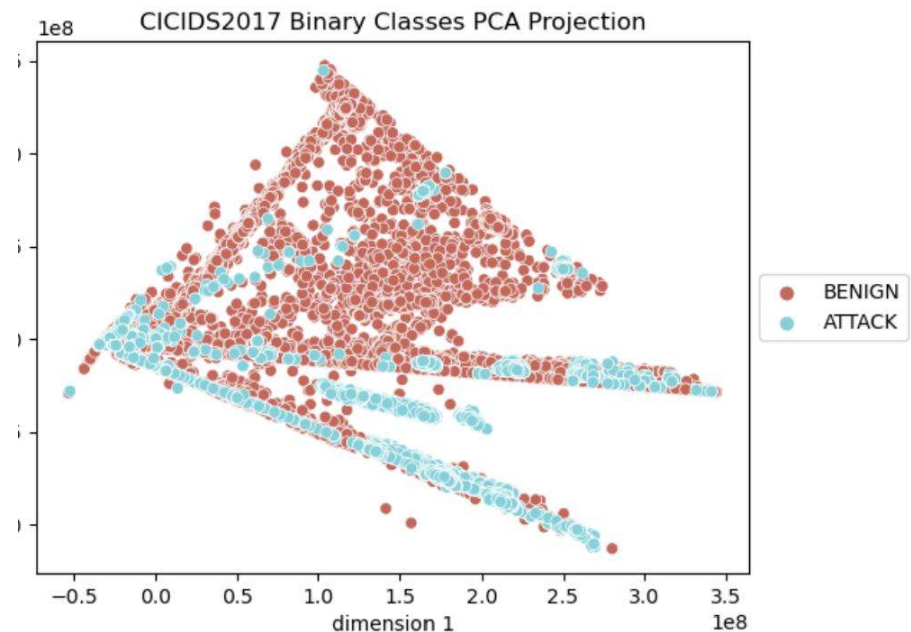
```
from scapy.all import *  
for i in range(65):  
    send(IP(dst="192.168.0.102")/TCP(dport=22,  
        flags="S"))
```

Test Snort Rules: Trigger SSH brute force detection



AI-IDS SETUP AND TESTING

- **AI-IDS Approach**
 - Trained on labeled CSV flow data (from CIC-IDS2017).
 - No manual rules – learns patterns in traffic.
 - Binary classification: *benign* vs. *malicious*
- **Why we switched from Stratosphere IPS:**
 - Compatibility issues and limited control
 - Our custom AI model gave us more flexibility



TRAINING THE AI-IDS

Flow Data Extraction

- Used **CICFlowMeter** to convert PCAP files into labeled CSV flow data.
- Extracted features like duration, packet size, and protocol info.

Model Selection

- Started with **Isolation Forest** (unsupervised)
- Switched to **XGBoost** for better accuracy and control.

Data Cleaning

- Trimmed output to show only key details:
 - *Attack type, source IP, timestamp*

[INFO] Evaluation Report:

	precision	recall	f1-score	support
BENIGN	1.00	1.00	1.00	454620
MALICIOUS	0.99	1.00	1.00	111529
accuracy			1.00	566149
macro avg	1.00	1.00	1.00	566149
weighted avg	1.00	1.00	1.00	566149

[INFO] Model saved to models/xgboost_model.pkl

TRAINING THE AI-IDS

Coding & Debugging

- Frequent errors during model building
- Required extensive debugging and validation

Training Time

- XGBoost training was slow and unpredictable
- Had to monitor progress manually over hours

Feature Extraction Issues

- CICFlowMeter worked for training
- Manual extraction needed during testing due to tool limits

[INFO] Evaluation Report:

	precision	recall	f1-score	support
BENIGN	1.00	1.00	1.00	454620
MALICIOUS	0.99	1.00	1.00	111529
accuracy			1.00	566149
macro avg	1.00	1.00	1.00	566149
weighted avg	1.00	1.00	1.00	566149

[INFO] Model saved to models/xgboost_model.pkl

AI-IDS DEMO

Execution & Results

- Ran model on new PCAPs
- Classified traffic as *benign* or *malicious*

```
sudo pip install .
```

```
(env) salman@Salman-MacBook-Pro-2 network_anomaly_detector_updated % sudo scan-live-traffic
[INFO] Sniffing for 10 seconds on interface: en0
[INFO] Done sniffing.
[INFO] Captured packets saved to captured_traffic.pcap
[RESULT] Live Traffic Prediction Results:


|    | Source IP      | Destination IP | Source Port | Destination Port | Protocol | Label  |
|----|----------------|----------------|-------------|------------------|----------|--------|
| 0  | 149.154.175.57 | 10.72.101.247  | 443         | 58056            | TCP      | BENIGN |
| 1  | 10.72.101.247  | 149.154.175.57 | 58058       | 443              | TCP      | BENIGN |
| 2  | 10.72.101.247  | 34.120.195.249 | 58105       | 443              | TCP      | BENIGN |
| 3  | 34.120.195.249 | 10.72.101.247  | 443         | 58105            | TCP      | BENIGN |
| 4  | 10.72.101.247  | 35.186.224.9   | 58106       | 443              | TCP      | BENIGN |
| 5  | 35.186.224.9   | 10.72.101.247  | 443         | 58106            | TCP      | BENIGN |
| 6  | 10.72.101.247  | 140.82.113.22  | 58882       | 443              | TCP      | BENIGN |
| 7  | 140.82.113.22  | 10.72.101.247  | 443         | 58882            | TCP      | BENIGN |
| 8  | 10.72.101.247  | 35.186.224.30  | 58053       | 443              | TCP      | BENIGN |
| 9  | 35.186.224.30  | 10.72.101.247  | 443         | 58053            | TCP      | BENIGN |
| 10 | 10.72.101.247  | 35.186.224.24  | 58086       | 443              | TCP      | BENIGN |
| 11 | 35.186.224.24  | 10.72.101.247  | 443         | 58086            | TCP      | BENIGN |
| 12 | 10.72.101.247  | 4.249.131.160  | 58887       | 443              | TCP      | BENIGN |
| 13 | 4.249.131.160  | 10.72.101.247  | 443         | 58887            | TCP      | BENIGN |
| 14 | 140.82.112.25  | 10.72.101.247  | 443         | 58047            | TCP      | BENIGN |
| 15 | 10.72.101.247  | 140.82.112.25  | 58047       | 443              | TCP      | BENIGN |
| 16 | 10.72.101.247  | 31.13.80.53    | 58048       | 443              | TCP      | BENIGN |
| 17 | 31.13.80.53    | 10.72.101.247  | 443         | 58048            | TCP      | BENIGN |
| 18 | 10.72.101.247  | 142.250.31.108 | 58897       | 505              | TCP      | BENIGN |
| 19 | 10.72.101.247  | 35.186.224.30  | 56168       | 443              | TCP      | BENIGN |
| 20 | 35.186.224.30  | 10.72.101.247  | 443         | 56168            | TCP      | BENIGN |
| 21 | 10.72.101.247  | 142.250.31.109 | 58898       | 505              | TCP      | BENIGN |


(env) salman@Salman-MacBook-Pro-2 network_anomaly_detector_updated %
```

SNORT PROS AND CONS

- **Pros:**

- Transparent rule logic
- Lightweight and efficient
- Real-time detection
- Widely used and documented

- **Cons:**

- Only detects known threats
- Needs constant rule updates
- Difficult to scale in large networks
- High manual effort for tuning



AI-IDS PROS AND CONS

- **Pros:**

- Learns from data, detects novel threats
- Low manual rule management
- Can reduce false positives
- Scales well with large traffic

- **Cons:**

- Needs large, clean training data
- High resource consumption
- Can act like a “black box”
- Risk of false negatives if not tuned



Feature	Snort (Rule-Based)	AI-IDS
Detection Method	Signature-based	ML/Anomaly-based
Adaptability	Low	High
New Threat Detection	Weak	Strong
Transparency	High	Low
Maintenance	High (Manual)	Medium (Model Tuning)
Scalability	Medium	High
Resource Needs	Low	High

Comparative Table

Challenges Faced



Stratosphere IPS was difficult to deploy → switched to custom model.



Snort's network interface needed tuning to capture replayed traffic.



Writing and validating custom rules took time.



Data preprocessing for AI training was complex.

Challenges Faced

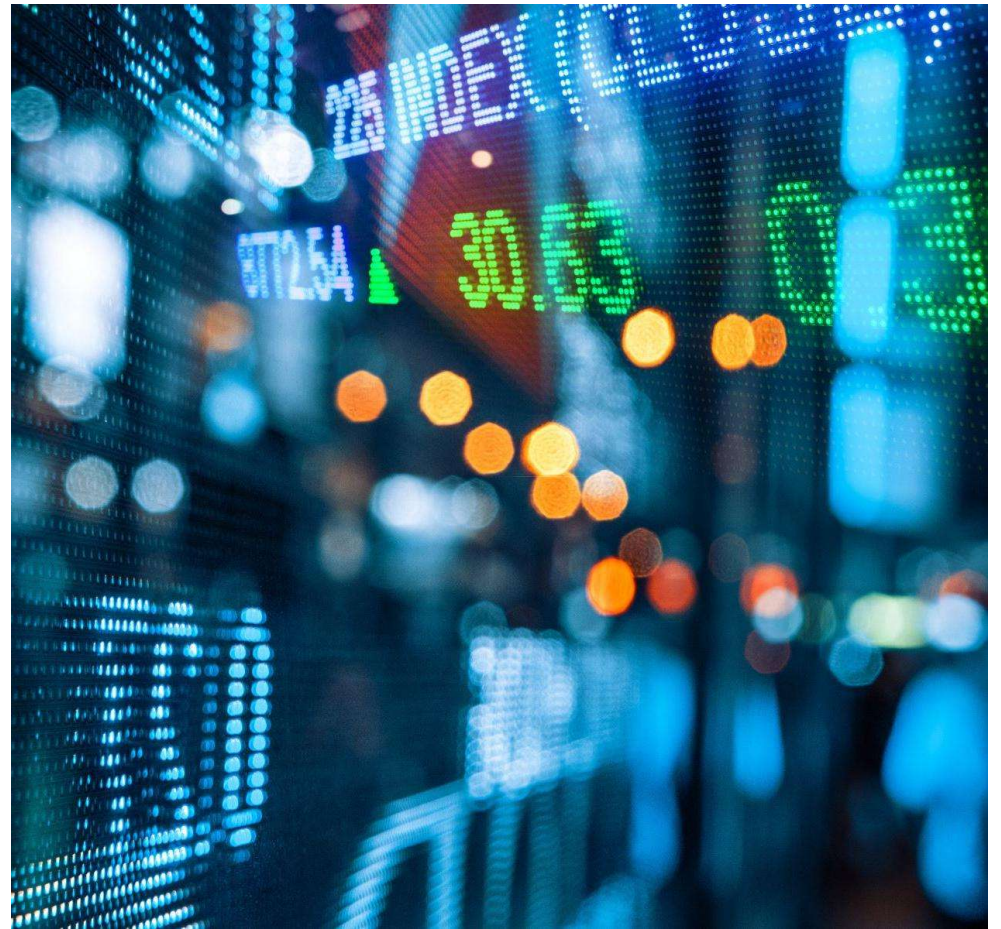
```
--
19 def extract_features_with_cicflowmeter(input_pcap, output_csv):
20     """
21     Use CICFlowMeter to extract features from a PCAP file.
22     """
23     print(
24         f"[INFO] Extracting features from {input_pcap} using CICFlowMeter..."
25     )
26     command = f"cicflowmeter -f {input_pcap} -c {output_csv}"
27     result = subprocess.run(command, shell=True,
28                             capture_output=True, text=True)
29     if result.returncode == 0:
30         print(f"[INFO] Features saved to {output_csv}")
31     else:
32         print(f"[ERROR] Failed to extract features: {result.stderr}")
33
34 def predict_traffic(output_csv, model_path):
35     """
36     Predict traffic labels using the trained model.
37     """
38
39 PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS zsh + v [ ] |
(3.12.0) salman@Salmans-MacBook-Pro-2 network_anomaly_detector_updated % cicflowmeter -f live_traffic.pcap -c extracted_f
es.csv
PermissionError: [Errno 13] Permission denied: 'extracted_features.csv'
Exception ignored in: <function CSVWriter.__del__ at 0x1084bd4eb>
Traceback (most recent call last):
  File "/Users/salman/.pyenv/versions/3.12.0/lib/python3.12/site-packages/cicflowmeter/writer.py", line 27, in __del__
AttributeError: 'CSVWriter' object has no attribute 'file'
(3.12.0) salman@Salmans-MacBook-Pro-2 network_anomaly_detector_updated % sudo cicflowmeter -f live_traffic.pcap -c extrac
tures.csv
reading from file live_traffic.pcap, Link-type EN10MB (Ethernet)
(3.12.0) salman@Salmans-MacBook-Pro-2 network_anomaly_detector_updated %
naster 0 0 0 1 file and 0 cells to analyze Ln 76, Col 11 Spaces: 4
```

	Destination Port	Flow	Duration	Total Fwd Packets	Total Backward Packets	...	Idle Std	Idle Max	Idle Min	label
0	443	0.026381	0	2	0	...	0	0	0	BENIGN
1	62795	0.000000	1	0	0	...	0	0	0	BENIGN
2	62821	1.129716	2	0	0	...	0	0	0	BENIGN
3	443	1.129283	2	0	0	...	0	0	0	BENIGN
4	57621	0.000000	1	0	0	...	0	0	0	BENIGN
5	0	0.000544	2	0	0	...	0	0	0	BENIGN
6	443	0.290074	4	0	0	...	0	0	0	BENIGN
7	63782	0.255747	3	0	0	...	0	0	0	BENIGN
8	443	0.490975	17	0	0	...	0	0	0	BENIGN
9	63816	0.470386	16	0	0	...	0	0	0	BENIGN
10	443	0.034640	3	0	0	...	0	0	0	BENIGN
11	63817	0.031881	2	0	0	...	0	0	0	BENIGN

[12 rows x 79 columns]

Key Observations

- Snort was effective at identifying **only known threats**.
- AI-IDS showed promise for **adaptive threat detection**, especially unknown attacks.
- A hybrid system could provide the **best of both worlds**.



FUTURE WORK

- Include anomaly detection by fine tuning the model to fit specific deployment environment.
- Use more recent pcap data or CSV.
- Utilize NTFlowLyzer(2024) for more future feature extractions as it makes the model more efficient and accurate.
- Utilize deep learning/sequential models rather than statistical models.



- Snort is simple, reliable, but outdated for modern threats.
- AI-IDS adapts to new threats but is complex and resource-heavy.
- Real-world security teams may benefit from a **combined approach**.



CONCLUSION

REFERENCES



ASIS International. (2021, April). *Breaking down the pros and cons of AI in cybersecurity*. Security Management. <https://www.asisonline.org/security-management-magazine/monthly-issues/security-technology/archive/2021/april/breaking-down-the-pros-and-cons-of-ai-in-cybersecurity/>



Bulajoul, W., James, A., & Pannu, M. (2017). Network intrusion detection systems in high-speed traffic in computer networks. *Future Generation Computer Systems*, 79, 1–9. <https://doi.org/10.1016/j.future.2017.09.058>



EC-Council. (2023). *Getting started with Snort IDS: Examining the pros and cons for network security*. EC-Council Cybersecurity Exchange. <https://www.eccouncil.org/cybersecurity-exchange/whitepaper/getting-started-snort-ids-network-security/>



Shaik, D. A. (2024, January 12). *AI-based intrusion detection systems*. Insights2Techinfo. <https://insights2techinfo.com/ai-based-intrusion-detection-systems/>



University of New Brunswick, Canadian Institute for Cybersecurity (CIC). (2017). *CICIDS 2017 dataset*. <https://www.unb.ca/cic/datasets/ids-2017.html>



Scapy. (n.d.). *Scapy – Packet crafting for Python*. <https://scapy.net/>



Snort. (n.d.). *Snort users manual and rule guides*. <https://www.snort.org/>



Wireshark Foundation. (n.d.). *Wireshark: Go deep*. <https://www.wireshark.org/>



• THANK YOU