ECU Dataset from ahmed2021ecu Paper

The ECU-IoHT dataset was developed as part of a study to analyze the security vulnerabilities and potential cyberattacks on IoHT devices. It is the first of its kind in the healthcare domain, as most other available datasets do not cover IoHT-specific security challenges.

The ECU-IoHT dataset was created in a controlled environment following a standard white hat penetration testing methodology. The testbed used for the dataset development included components like:

- **MySignals healthcare kit:** A development platform for IoHT applications.
- **Sensors:** Temperature Sensor, Blood Pressure Sensor and Heart Rate Sensor
- **Computing setup:** Windows 10, Kali Linux (2020.2), mobile Wi-Fi hotspot, and Bluetooth adaptor.
- **Software tools:** Argus for flow monitoring, TShark for packet analysis, and Ettercap for attack simulation.

Sensor data sent wirelessly to the Libelium cloud server for transmission and then captured at the router level for attack simulation and analysis.

Dataset Fields:

NO. (Number): This is simply the index or serial number of the entry in the dataset, allowing you to reference specific records easily.

Time: This field records the time at which the packet was captured, typically in seconds since the start of the capture. It allows for the analysis of network activity over time, helping to identify patterns such as bursts of traffic or delays that could be indicative of an attack.

Source: The originating address or device of a network packet—either an IP address or a MAC address, depending on the network layer—plays an essential role in identifying the source of communication. This information is crucial for tracing back to the initiating device in case of an attack or a compromised system.

Destination: This field indicates the address of the target device the packet is trying to reach, clarifying which device is being targeted. The roles of source and destination can switch during exchanges, such as in attacks or responses.

Protocol: specifies the type of communication protocol in the packet. Each protocol serves a specific purpose. Recognizing the protocol is crucial, as specific attacks target certain protocols, aiding in the identification of potential threats.

Dataset protocols: TCP, TLS, ICMP, DNS and ARP

Length: The packet size, measured in bytes, is crucial for identifying normal or abnormal behavior. Fixed sizes are often associated with certain attacks, such as ARP Spoofing and Smurf Attacks, making it easier to flag suspicious packets.

Info: provides a detailed summary of the packet's content, which can vary significantly depending on the protocol. For example:

- Normal Traffic Example: 36954 > 110 [SYN] Seq=0 Win=1024 Len=0 MSS=1460, This represents a SYN packet, which is the first step in establishing a TCP connection. The source port 36954 is trying to connect to destination port 110 (typically used for POP3 email services). The packet initiates a connection with sequence number 0, a window size of 1024 bytes, and a maximum segment size (MSS) of 1460 bytes.
- Nmap Port Scan Example: 1720 > 36954 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0, represents a response to a port scan where the target is closing the connection with a reset (RST) flag. This typically happens when Nmap, a network scanning tool, tries to probe a closed port.

So, The Info field provides insights into the behavior of the traffic. Understanding these behaviors is crucial for distinguishing between normal and malicious activities.

Type: indicates whether the packet is part of a normal communication (Normal) or an attack (Attack).

> Dataset attacks: ARP Spoofing Instances, DoS Instances, Nmap Port Scanning Instances and Smurf Attack Instances.

Type of attack: if the packet is flagged as an attack, this field specifies the type of attack, such as ARP Spoofing, Nmap Port Scan, or Smurf Attack. Knowing the specific type of attack allows for more granular analysis and understanding of how different attacks manifest in network traffic. This can help in creating specialized detection mechanisms for each type of attack.

EHMS Dataset from hady2020intrusion Paper

Designed for intrusion detection in healthcare systems by integrating network traffic metrics with biometric data. It focuses on Man-in-the-Middle (MITM) attacks and their impact on healthcare monitoring systems. This work evaluates machine learning models for detecting intrusions.

Patient biometric data were collected using the **Six Pe Multi-Sensor Board**, which was attached to the patient. This data was then transferred to a server via Wi-Fi using the TCP/IP protocol.

For processing, the data is split with 80% used for training and 20% for testing. Class imbalance is addressed using the Synthetic Minority Oversampling Technique (SMOTE), as 88% of the data is normal.

The dataset was used to train and evaluate four different machine learning models. Each model has unique strengths for detecting patterns in the data and identifying attacks:

1. Random Forest (RF)

Random Forest (RF) is an ensemble learning method that builds multiple decision trees during the training process. It combines the predictions (or votes) from these trees to classify samples as either "normal" or "attack." This technique effectively detects anomalies in both biometric readings and network traffic patterns by establishing decision boundaries across multiple features.

2. K-Nearest Neighbor (KNN)

A non-parametric algorithm classifies a data point based on the majority vote of its k nearest neighbors. Proximity is measured using distance metrics, such as Euclidean distance. The algorithm detects attacks by comparing biometric and network feature values to those of nearby "normal" instances.

3. Support Vector Machine (SVM)

A supervised learning algorithm that classifies data points into distinct classes using a hyperplane was applied. Specifically, the linear Support Vector Machine (SVM) version was used, which is effective for datasets that are linearly separable. This algorithm categorizes network flow and biometric data as either normal or suspicious based on optimized decision boundaries.

4. Artificial Neural Network (ANN)

This biologically inspired model features interconnected layers to learn complex data patterns, including an input layer with 40 neurons, multiple hidden layers (40, 40, 20, and 10 neurons), and an output layer with 1 neuron for binary classification (normal or attack). It processes both biometric and network data to identify subtle patterns that may indicate spoofing or data tampering.

The dataset combines two categories of features:

1. **Biometric Features (Healthcare Data):**

Patient vital signs collected from sensors, including:

- **Temp**: **Body temperature** of the patient (°C).
- **SpO2**: **Blood oxygen saturation level** (percentage). Normal range: 95-100%.
- **Pulse_Rate**: Patient's **pulse rate** (beats per minute).
- **SYS**: **Systolic blood pressure** (higher value). Normal: ~120 mmHg.
- **DIA**: **Diastolic blood pressure** (lower value). Normal: ~80 mmHg.
- **Heart_rate**: Patient's **heart rate** (beats per minute). Measures cardiovascular activity.
- **Resp_Rate**: **Respiration rate**: Breaths per minute. Normal: 12-20 breaths/min.
- **ST**: **Stress index**: Indicates patient's stress level, calculated from biometric features.

These features provide real-time patient health data, helping identify abnormal behavior caused by potential attacks.

2. **Network Traffic Features:**

Specific metrics extracted using the Argus software for real-time network flow monitoring.

- **Dir**: Describes the **direction** of the traffic flow.

  Example: -> for outgoing traffic, <- for incoming traffic.

- **Flgs**: Represents the **TCP flags** used in the network connection.

  Example: SYN, ACK, FIN, RST, etc., which indicate the state of the connection.

- **SrcAddr**: Source IP address of the packet's origin.
- **DstAddr**: Destination IP address of the packet.
- **Sport**: Source port number. Indicates the application or service on the sender's side. crucial for tracing attack origins.
- **Dport**: Destination port number. Represents the application/service receiving the packet. helping identify targeted systems or ports.
- **SrcBytes**: Number of bytes sent by the source device. useful for detecting abnormal traffic patterns.
- **DstBytes**: Number of bytes received by the destination device. highlighting potential data exfiltration.
- **SrcLoad**: Traffic load (bytes/second) sent from the source. It is important to identify when resources are overused, such as during Denial of Service (DoS) attacks.
- **DstLoad**: Traffic load received at the destination. useful for detecting traffic floods or overloads.

- **SrcGap**: Time gap between two consecutive packets sent by the source. helping detect irregular communication patterns.
- **DstGap**: Time gap between two consecutive packets received by the destination. identifying anomalies like delays or packet bursts.
- **SIntPkt**: **Source inter-packet time**: Time interval between two packets sent by the source. detecting malicious activity like packet injection.
- **DIntPkt**: **Destination inter-packet time**: Time interval between two packets received by the destination. useful for recognizing abnormal communication timing.
- **SIntPktAct**: Active source inter-packet time, measuring activity bursts. detecting irregular traffic spikes.
- **DIntPktAct**: Active destination inter-packet time, indicating active network behavior. helping identify sudden data surges.
- **SrcJitter**: Jitter for packets sent by the source. Measures variations in packet delivery time. useful for detecting spoofing or timing attacks.
- **DstJitter**: Jitter for packets received at the destination. identifying communication instability or replay attacks.
- **sMaxPktSz**: Maximum packet size sent by the source. useful for identifying unusual data transmissions.
- **dMaxPktSz**: Maximum packet size received at the destination. highlighting potential malicious payloads.
- **sMinPktSz**: Minimum packet size sent by the source. identifying fragmented or abnormal packets.
- **dMinPktSz**: Minimum packet size received at the destination. useful for identifying probing or scanning attempts.
- **Dur**: Duration of the connection or session. critical for analyzing prolonged or suspicious connections.
- **Trans**: Number of transactions (data exchanges) in the network session. helping identify abnormal activity levels during attacks.
- **TotPkts**: Total number of packets exchanged during the session. useful for detecting traffic anomalies or floods.
- **TotBytes**: Total number of bytes exchanged between source and destination. indicating possible data theft or suspicious activity.
- **Load**: Combined network load for the connection. useful for detecting system or network stress.
- **Loss**: Number of lost packets in the session. helping detect disruptions caused by attacks.
- **pLoss**: Packet loss percentage during the session. identifying instability caused by spoofing or DoS.
- **pSrcLoss**: Percentage of packets lost from the source side. highlighting sender-side issues

- **pDstLoss**: Percentage of packets lost at the destination side. identifying network or target-side issues.
- **Rate**: Data **rate** (bytes per second) of the connection.
- **SrcMac**: MAC address of the source device.
- **DstMac**: MAC address of the destination device.
- **Packet_num**: Sequential packet number in the dataset. essential for analyzing packet order and patterns.

3. Attack Features

- **Attack Category:** Label indicating the type of **attack**:

  Normal: No attack.

  Spoofing: ARP/IP spoofing attack.

  DoS: Denial of Service attack.

  Replay: Data replay attack.

- **Label:** Binary **attack label**: 0 → Normal, 1 → Attack