Cyber-RF Anomaly Detector Challenge

Wireless networks provide mission-critical infrastructure for public safety, national security, and military communications. The value of wireless networks in these applications is derived from their accessibility and availability. However, this accessibility is increasingly threatened by cyber-attacks such as jamming and spoofing.

The challenge we identified for this research problem is with respect to the classification and detection of anomalous co-channel signals at the physical layer using both the IQ and network traffic data. We called this challenge the *Cyber-RF Anomaly Detector*. As a starting point, for this challenge our initial focus is Zigbee. Zigbee was chosen for its low data rate, low power consumption, and low-cost wireless network protocol that is commonly used for industrial control systems and IoT devices (e.g., Amazon Echo Plus and Samsung SmartThings).

For the Cyber-RF Anomaly Detector challenge, the participants will be tasked with developing machine learning algorithms for the detection of anomalous Zigbee transmissions. The anomalous Zigbee transmissions were represented by simulating a replay attack and a rogue transmitter use case. The replay attack is when a malicious actor listens to the transmitter and duplicates the transmissions that are eventually sent to the coordinator (i.e., the malicious actor acts as a repeater). While for the rogue transmitter use case, a malicious actor is creating and transmitting new packets that are sent to the coordinator (i.e., the malicious actor acts as an independent transmitter). To evaluate the performance of the machine learning algorithms, several metrics can be computed from the following confusion matrix:

	Actual: Legitimate	**Actual:** Anomalous
Predicted: Legitimate	TN	FN
Predicted: Anomalous	FP	TP

where *TN*, *FN*, *FP*, and *TP* refer to the numbers of true negatives, false negatives, false positives, and true positives, respectively. At the moment for this challenge, we are interested in computing the following performance metrics that assess different aspects of the classification:

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

$$Recall = TP/(TP + FN)$$

$$Precision = TP/(TP + FP)$$

$$FPR = FP/(FP + TN)$$

The provided dataset for the Cyber-RF Anomaly Detector challenge consists of features extracted from both legitimate Zigbee and anomalous Zigbee transmissions. The Zigbee data was collected from our Software-Defined Radio (SDR)-based signals testbed whose goal is to create a complex RF environment for signal classification and anomaly detection. The SDR-based testbed consists of multiple SDRs (i.e., two Ettus X310s and one B210) as shown in Figure 1. For the legitimate Zigbee transmissions, we have one B210 serving as a coordinator and one X310 serving as the transmitter. In the case of the anomalous Zigbee transmissions, it is the same as the legitimate setup, but with an additional transmitter (an X310) who serves as the malicious actor.

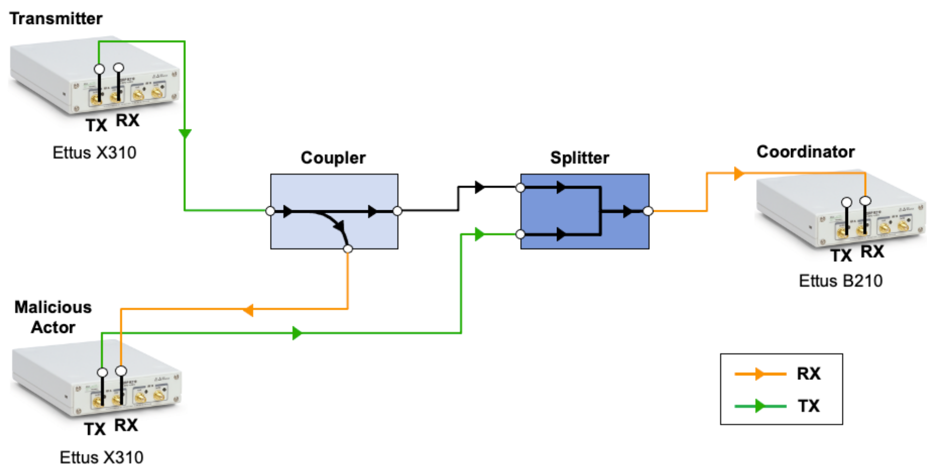

Figure 1: Wiring setup for the SDR-based signal testbed

Each SDR is transmitting or receiving a signal using the Zigbee protocol. This Zigbee signal (i.e., either legitimate or anomalous) is sent to the receiver. The receiver is running a GNURadio implementation that records both the IQ data and network traffic data from the Zigbee signal. The IQ data was saved as a .dat file while the network traffic data was stored as a .pcap file. The anomalous Zigbee transmissions were represented by simulating a replay attack and a rogue transmitter use case. As seen in Figure 1, the coupler samples the transmitter's transmission to pass to the malicious actor. In the case of the replay attack, the malicious actor will retransmit the signal passed to it by this coupler connection. In our setup, the splitter acts as a combiner (i.e., it adds the two signals from the transmitter and malicious actor together).

To be certain that commonly used approaches such as an energy detector cannot be used to discern anomalous Zigbee transmissions from legitimate Zigbee transmissions, we are ensuring power balance while collecting the anomalous Zigbee data. The power balance was established by tuning several parameters from the malicious actor transmitter, such as TX, RX gain values, and the period of the message that is been sent. By first establishing the baseline maximum and mean power levels for legitimate captures done at different center frequencies, we manually tuned the gain and period values so that all anomalous captures would result in roughly the same number of packets, as well as would approximately match either the maximum or the mean of its legitimate counterpart. Later, an assessment was done to check that the attained signal power reflects the signal power from a legitimate Zigbee transmission.

Besides the power balance, for the rogue transmitter use case, we also tweak the proportion of packets that the malicious actor sends to the coordinator. Specifically, several combinations of malicious vs. legitimate packet proportions were explored (i.e., 50/50, 60/40, 70/30, 80/20). In the case of the replay attack, tweaking the packet proportion is not applicable because here the malicious actor does not act as an independent transmitter.

After the data collection and curation, the participants will be given a dataset containing features extracted from IQ and network traffic data using legitimate and anomalous Zigbee transmissions. Each of the legitimate and anomalous Zigbee transmissions was recorded for a duration of two minutes and different parameters were considered. For example, different values were used for the center frequency (i.e., 2.47, 2.48, and 2.49 GHz, respectively) and the Tx gain (i.e., 20, 25, and 30, respectively). A total of 450 captures were collected for both the legitimate Zigbee, and anomalous Zigbee transmissions.

For the IQ data, specific measurements were computed (i.e., Amplitude, Phase, RMS, Signal Power, FFT, and Periodogram). Later, features were extracted from these measurements. Some examples of the extracted IQ-based features are the skewness, Kurtosis, and entropy of these measurements. A list of the extracted IQ-based features is given in Table 1.

Table 1: IQ-based features extracted for each legitimate and anomalous Zigbee captures

Features	**Description**
Amp_min	Minimum amplitude measurement
Amp_max	Largest amplitude measurement
Amp_var	Variance of the amplitude measurement
Amp_skew	Skewness of the amplitude measurement
Amp_rango	Range of the amplitude measurement
Amp_Kurtosis	Kurtosis of the amplitude measurement
Amp_entropy	Entropy of the amplitude measurement
Phase_min	Minimum phase measurement
Phase_var	Variance of the phase measurement
Phase_skew	Skewness of the phase measurement
Phase_entropy	Entropy of the phase measurement
RMS_max	Largest RMS measurement
RMS_skew	Skewness of the RMS measurement
RMS_rango	Range of the RMS measurement
RMS_Kurtosis	Kurtosis of the RMS measurement
RMS_entropy	Entropy of the RMS measurement
SP_min	Minimum signal power measurement

SP_max	Largest signal power measurement
SP_var	Variance of the signal power measurement
SP_skew	Skewness of the signal power measurement
SP_rango	Range of the signal power measurement
SP_Kurtosis	Kurtosis of the signal power measurement
SP_entropy	Entropy of the signal power measurement
SP_stError	Standard error of the signal power measurement
FFT_min	Minimum FFT measurement
FFT_max	Largest FFT measurement
FFT_avg	Average of the FFT measurement
FFT_median	Median of the FFT measurement
FFT_var	Variance of the FFT measurement
FFT_skew	Skewness of the FFT measurement
FFT_rango	Range of the FFT measurement
FFT_Kurtosis	Kurtosis of the FFT measurement
FFT_entropy	Entropy of the FFT measurement
FFT_stError	Standard error of the FFT measurement
Pd_max	Largest periodogram measurement
Pd_avg	Average of the periodogram measurement
Pd_var	Variance of the periodogram measurement
Pd_skew	Skewness of the periodogram measurement
Pd_rango	Range of the periodogram measurement
Pd_Kurtosis	Kurtosis of the periodogram measurement
Pd_entropy	Entropy of the periodogram measurement

With respect to the network traffic data, the .pcap files were fed to a tool called Tranalyzer (a lightweight unidirectional flow exporter that collects packet information with common characteristics) to obtain network flows. Later, these network flows served as the input into a Python script to extract network traffic-based features such as the number of flows, the average of the number of bytes sent, and the average of the packet size. The extracted network traffic-based features are listed in Table 2.

Table 2: Network flows-based features extracted for each legitimate and anomalous Zigbee captures

Features	**Description**
Duration_Avg	Average time the communication lasted
SumNoPktsSent	Summation of the # of transmitted packets sent by all the network flows extracted from a pcap file
numPktSent_avg	Average of the # of transmitted packets sent by all the network flows extracted from a pcap file
NoBytesSnt_avg	Average of the # of bytes sent by all the network flows extracted from a pcap file
minPktSize_min	Minimum layer 3 packet size

maxPktSize_max	Largest layer 3 packet size
avgPktSz_avg	Average packet load ratio
pktps_avg	Average of the packets sent per second
bytps_avg	Average of bytes sent per second
maxIAT_max	Maximum of inter-arrival-time (IAT) of the flow
avgIAT_avg	Average of IAT of the flow