Anomaly Detection System for Industrial Control Systems

Team: sdmay22-38 | Alex Nicolellis | Jung Ho Suh | Muhamed Stilic | Pallavi Santhosh
Client: Manimaran Govindarasu
Advisors: Manimaran Govindarasu | Moataz Abdelkhalek

Introduction

Problem: Cyber attacks on power distribution companies are debilitating and are now more common due to the increased use of IoT devices and the inadequate security on power grid systems. To feel secure in how our power is protected, our security measures must continue to advance with the constantly developing technology of cybercriminals.

Solution: ADSICS is a surveillance program that detects and prevents cyber attacks using anomaly detection.

The end goal is described as a detailed visual output of temporal anomaly detection, tracking alert information, and performing machine learning analysis quickly and efficiently using the desired platforms (Elastic Stack).

We also built our own ML model to determine whether or not Elastic, as a relatively "new"platform, could be relied upon solely as a valid solution to detect and analyze various cyberattacks.

Design Requirements

Functional Requirements

- Use machine learning to detect network anomalies
- Verify incoming alerts and detect false positives
- Display alerts for easy human understanding
- Present temporal and spatial details for each alert
- Users should be able to add or remove data visualizations on the dashboard
- Alerts should be distinguishable from each other and labled

Non-Functional Requirements

- Alerts should be presented intuitively
- Alerts should be color coded by severity
- The system should be able to handle a large volume of alerts
- The system should be reliable and consistent with analysis
- System should have an accuracy of >95% for detecting each alert

Engineering Constraints: One branch of the project must use the anomaly detection must use Elastic Stack tools.
Engineering Standards: IEEE 692-2013 (IADS Sensor & Master), ISO IEC 27039-2015 (IADS Master), ISO/IEC 27017:2015 (Cloud Server), and IEEE 802 (IADS Sensor)

Technical Details

Programming Language: Python

Libraries: pandas, numpy, matplotlib, sklearn, collections, imblearn, and joblib

Environments: Vsphere Testbed VM, Local VM, Github, and Google Colab

Development Tools: Python, Elastic Stack, Wireshark, Tshark, Editcap, Tcpreplay, and CICFlowMeter

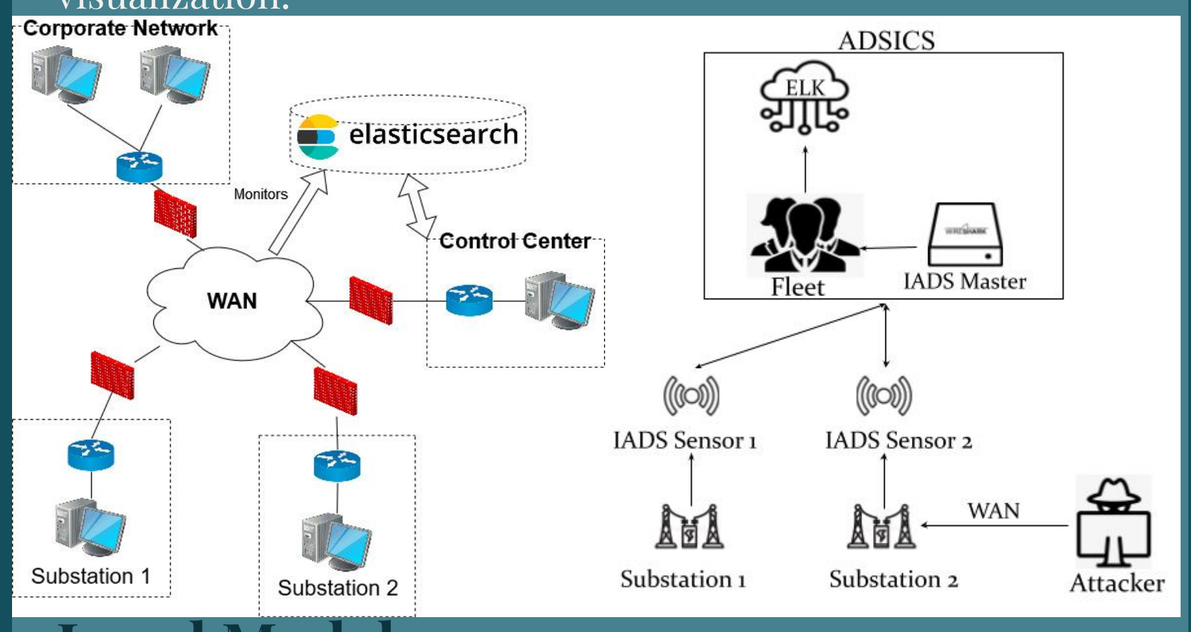
Intended Users/Uses

Users: Power distribution/utility companies

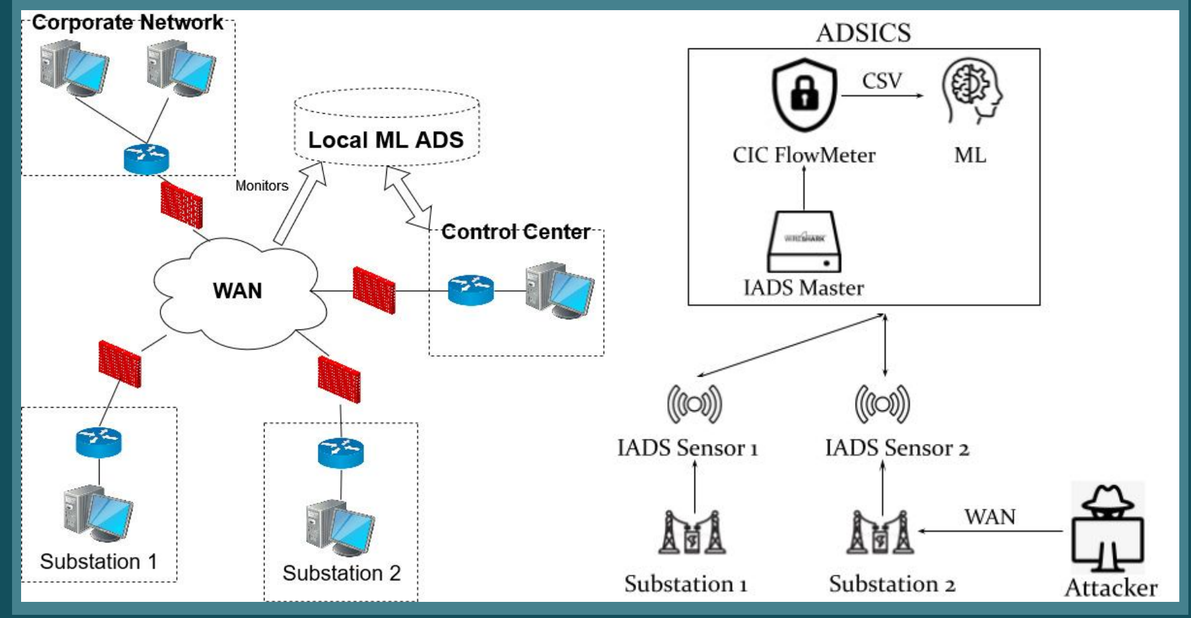
Uses: Power distribution security

Design Approach

Elastic Stack: Fleet agents send attack information from the IADS Master to the ELK Cloud for analysis and visualization.



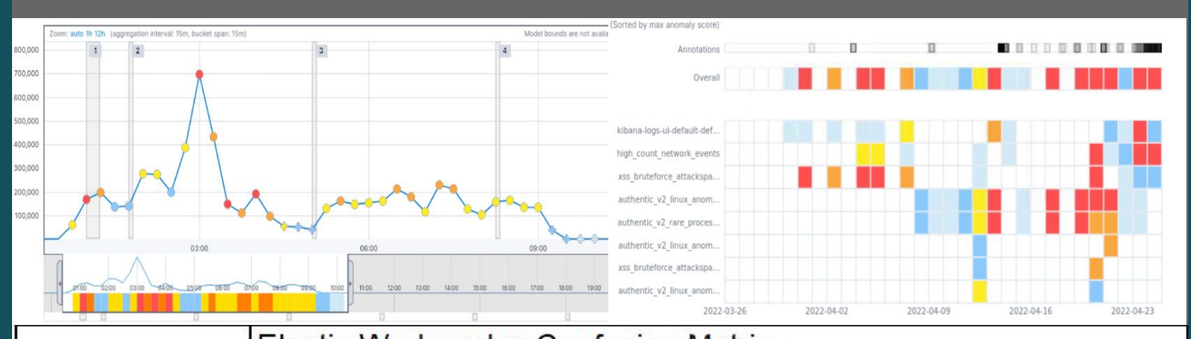
Local Model: Using CICFlowMeter we converted the network traffic into CSV format.



Testing Results

Environment: Elastic Testbed VM

Strategy: PCAP file is emulated by TCPReplay.



	Elastic Wednesday Confusion Matrix			
	Benign	DoS Hulk	DoS SlowHTTP	DoS Slowloris
Benign	99%	1%	0%	0%
DoS Hulk	0%	100%	0%	0%
DoS SlowHTTP	1%	0%	99%	1%
DoS Slowloris	0%	0%	0%	100%
	Elastic Thursday Confusion Matrix			
	Benign	Brute Force	SQL Injection	XSS
Benign	100%	0%	0%	0%
Brute Force	1%	19%	0%	80%
SQL Injection	0%	43%	0%	57%
XSS	5%	0%	0%	95%

Environment: Local ML Testbed VM

Wednesday - DDoS					
	Benign	Hulk	Slowhttp	Slowloris	Accuracy
Benign	17380	428	26	8	97%
Hulk	10	17760	14	1	100%
Slowhttp	3	2120	6787	0	76%
Slowloris	10	0	16	8902	100%
Thursday - Webattack				Overall	95%
	Benign	Brute Force	XSS	SQL Injection	Accuracy
Benign	167627	6	496	57	100%
Brute Force	3	219	1213	72	15%
XSS	21	0	630	1	97%
SQL Injection	1	0	0	20	95%
				Overall	99%

	Elasticsearch		Local ML			
	Wednesday	Thursday	Wednesday	Thursday	Modbus	
					Combined	Separate
Max Precision	100.0%	100.0%	100.0%	100.0%	99.0%	95.0%
Min Precision	99.0%	0.0%	87.0%	23.0%	92.0%	24.0%
Accuracy	99.8%	99.0%	95.0%	99.0%	96.0%	68.0%
Macro Average	99.4%	53.4%	95.0%	48.0%	96.0%	67.0%

Modbus - Combined: Normal vs DoS Anomaly
Modbus - Separate: Normal vs MBquery/ping/tcpSYN flood