Resilient Power Grid Project Report

Sahiti Bommareddy | Daniel Qian | Parv Saxena

Spring 2020

Project Goal

Extend the [Spire] intrusion-tolerant SCADA system

Three dimensions:

1. Performance optimization for single site configuration
2. Machine learning based network intrusion detection
3. Development of attack models for testing

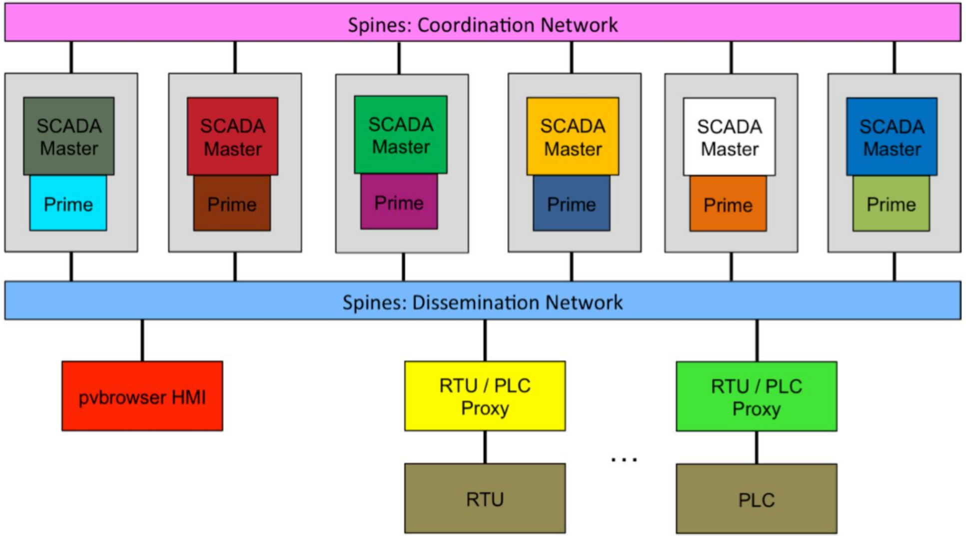
System Requirements

Critical Infrastructure Services need to address:

- System level compromises
- Network level attacks and compromises

Spire: Intrusion-Tolerant SCADA for the Power Grid

- Scada Master
- Prime
- Spines Intrusion Tolerant Network
- PLC/RTU Proxies
- HMI



Example Spire system deployment with six replicas.

Features of Spire

- BFT ($3f$+1)
- Diversity
- Proactive recovery (k)
- Proxies
- Intrusion tolerant overlay network

$$\mathbf{N = 3\boldsymbol{f} + 2\boldsymbol{k} + 1}$$

Spire's Context

- Intended for wide area
- Targeted each transaction time to be below 100ms

Part 1: Performance Optimization

New Factors

- Operate in single substation with different requirements
- Need stringent transaction times, on the order of a power cycle

Performance in One Site Setting

Benchmark of Average Transaction times in different configurations and clusters				
	Minis	Hails	DC70	
Original	**44ms**	38ms	-	Used Openssl 1.0.1, Prime Interval 20ms
Upgrade Openssl	36ms	31ms	28ms	Used Openssl 1.0.2, Prime Interval 20ms
Prime Tuning	-	20ms	**18ms**	Used Openssl 1.0.2, Prime Interval 1ms

Areas for Further Improvement

- Faster crypto using OpenSSL 1.1
  - Requires refactoring the code
- Explore real-time kernels
  - Need sub millisecond granularity
  - **However**, there is an associated overhead
- Explore alternatives to threshold crypto
  - Instead use appropriate (f + 1) number of identical messages
  - **However,** lose advantages of threshold crypto
- BFT Protocols other than Prime
  - Protocols that emphasize timeliness
  - **However,** tradeoff throughput because of no aggregation

Part 2: ML-based Network Intrusion Detection

Background

Previous work in this area for SCADA exists: MANA

Machine Learning vs. Signature Based

- Signature based can only detect **known** attacks
- MANA experiments showed superiority of ML methods

Many different methods have shown success in research

- Deep learning, decision trees, clustering
- Expert vote to reduce false positive
- **Generally done on well-known, prelabeled, datasets**

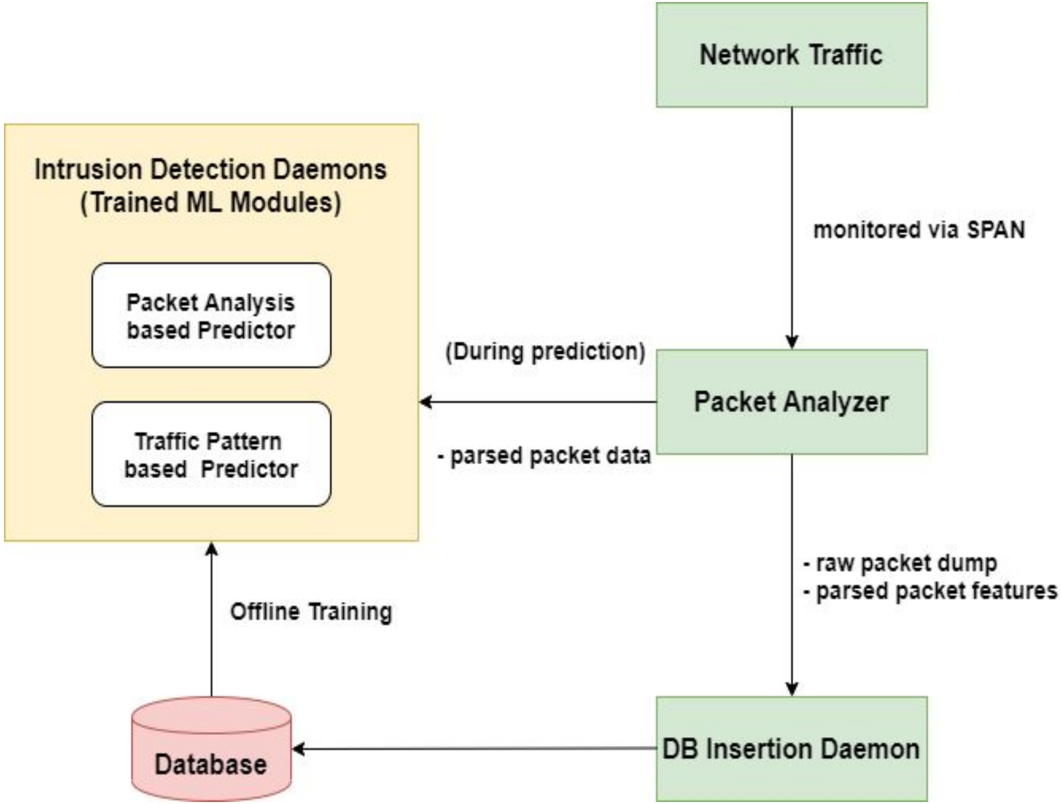
Data Pipeline

Use scripts from previous deployment (PNNL) to generate correct traffic.

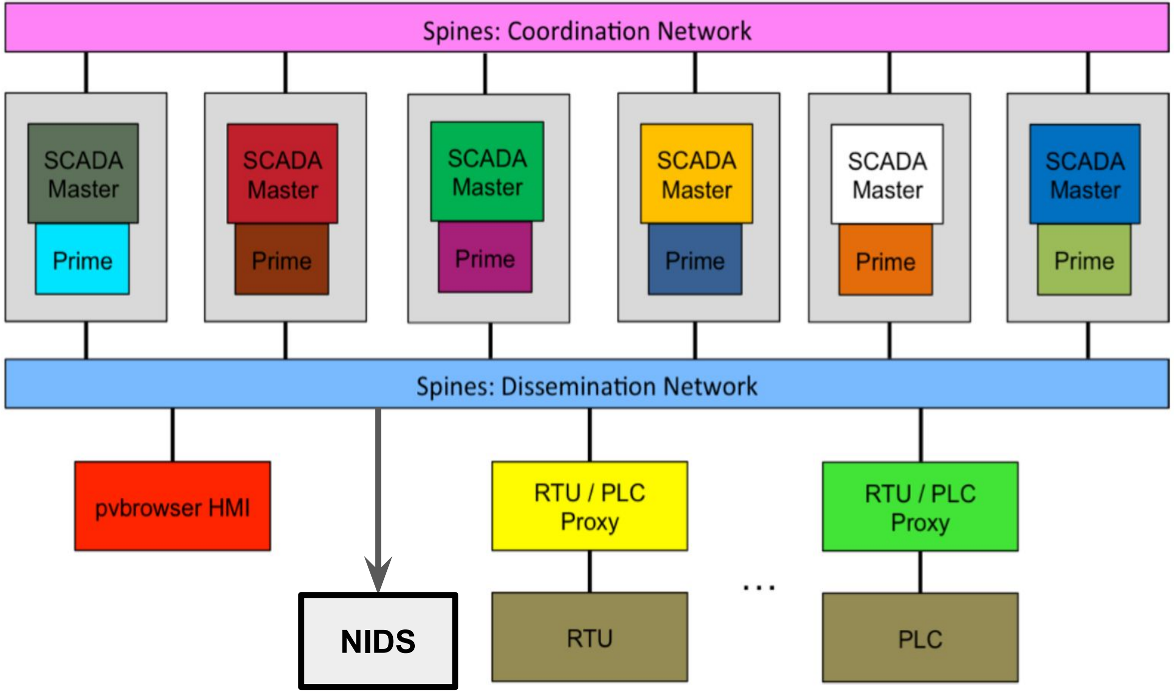
Capture network traffic on external facing switch (SPAN)

~6 hours of traffic

Very regular



A note on SPAN



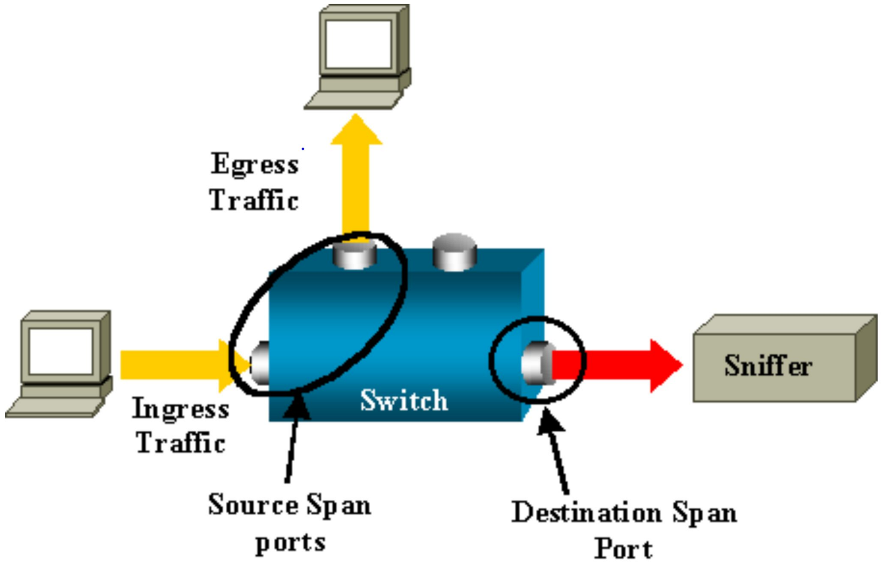
A note on SPAN

SPAN - Switched Port Analyzer

Only certain types of switches come built with this capability

The switch sends a copy of all network packets seen on one port (or an entire VLAN) to a special monitoring port

Network traffic is captured using switch to replicate the packets. So, no impact on the system.



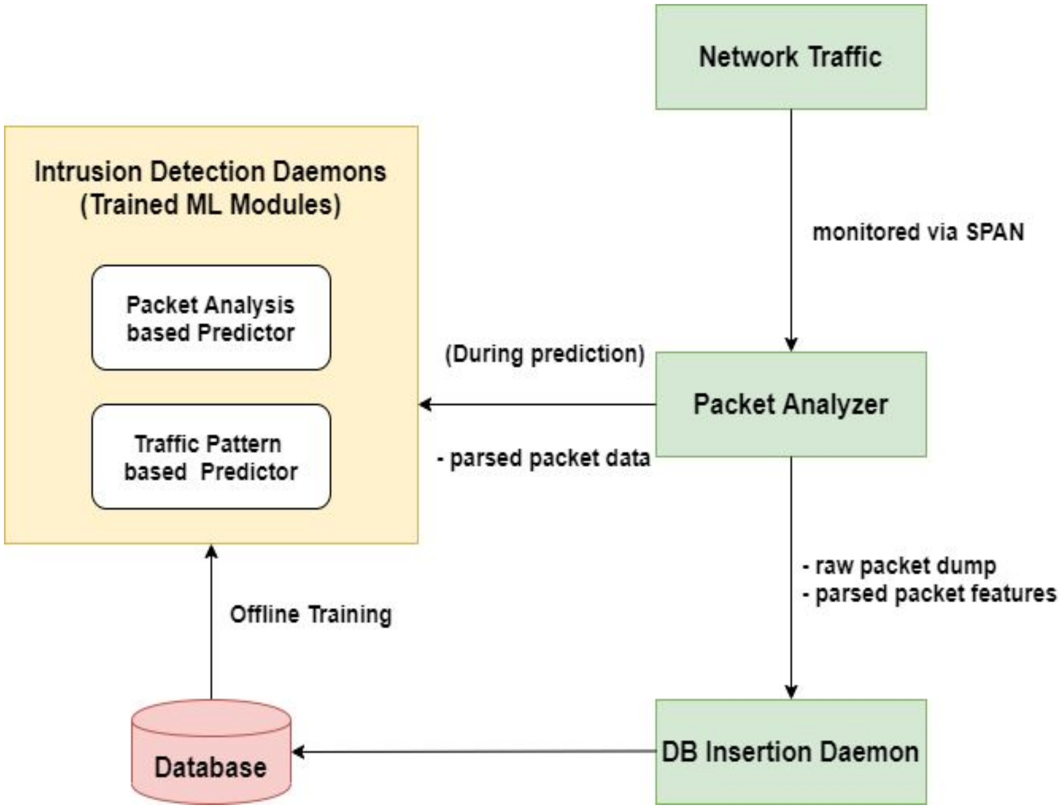
Data Pipeline

Use scripts from previous deployment (PNNL) to generate correct traffic.

Capture network traffic on external facing switch (SPAN)

~6 hours of traffic

Very regular



Feature Engineering

Two approaches that complement each other:

Packet Analysis Based

- Directly use information from packet headers.
- Features are ips, macs, port, protocol type, length etc.

Traffic/Flow Pattern Based

- Use counts of different packet types over minute-long intervals
- Feature are counts of ip/mac destination/sources, packet sizes, protocol etc.

(One Class) Classification

Situation: We only have "good" data in both approaches

Idea 1: Create "bad" data

- Not useful for new types of attacks (zero day)
- Would have to label data ourselves
- Still imbalanced

(One Class) Classification

Situation: We only have “good” data in both approaches

Idea 1: Create “bad” data

- Not useful for new types of attacks (zero day)
- Would have to label data ourselves
- Still imbalanced

Idea 2: Unsupervised Learning

- One Class Classification
- Algorithms that are suited for novelty/outlier detection

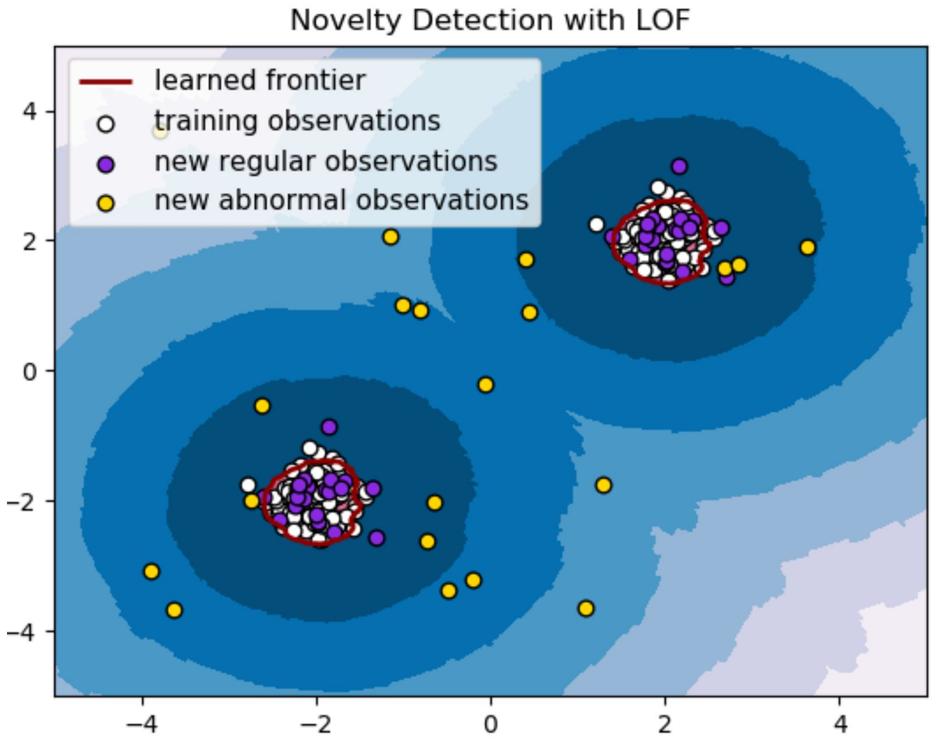
Also, train a number of models and take **majority vote** for final decision

(One Class) Classification

Local Outlier Factor
Compares local density of point to density of near points

One Class SVM
Modified SVM: separates transformed data (kernel) from origin

Elliptic Envelope
Fits ellipse around data using assumption of Gaussian distribution



Part 3: Attack Vectors and Testing

Attack Vectors

- Replicated some well known network level attacks
- Systematically vary parameters in attack vectors to generate testbed
- Use them to measure model performance under attack

Port Scanning

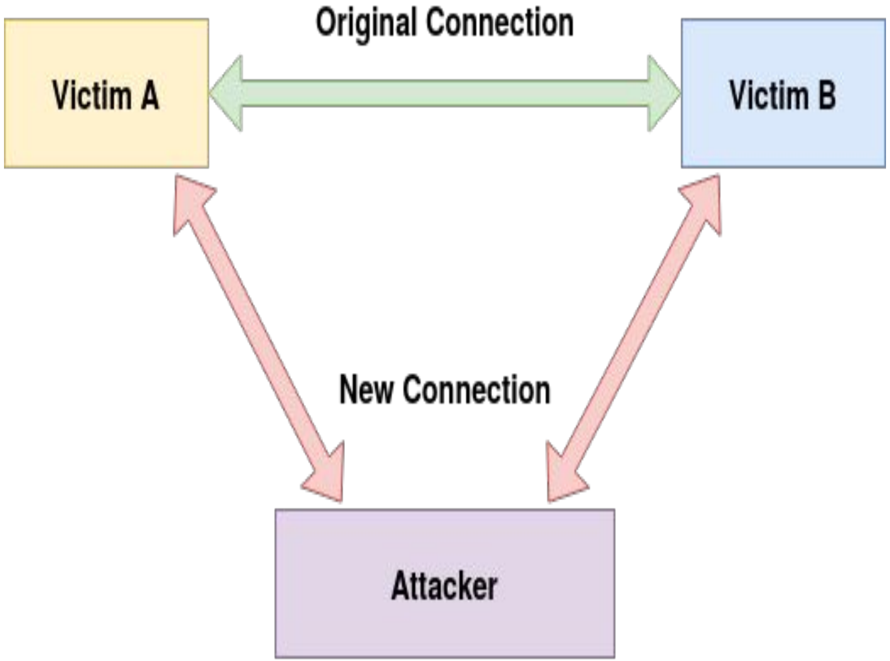
- Attacker sends requests to different ports to find active ones
- Use the active ports to launch attacks/exploit vulnerabilities

Denial Of Service (DOS)

- Attacker overwhelms the target machine with high volume of traffic
  - Deplete machine resources
  - Prevents/Delays correct transactions

Address Resolution Protocol (ARP) Poisoning

- Attacker disturbs IP to MAC mapping on LAN
- Spoofs MAC address so that traffic flows through it
- Can sniff/modify packets



Man in The Middle

Replay Attack

- Attacker eavesdrops on the network
- Fraudulently resends or delays the packets to adversely affect the target
- No need to modify the packet

Deployment, Integration and Tuning

- Train multiple ML algorithms offline
- Run Spire system with PNNL scenario
- Launch multiple attacks
- Observe and tune ML algorithms

Model Testing

- For each type of attack, randomized one or more parameters
- For Aggregate model, launched attack every alternate time bucket, randomizing both parameters and counts

	Packet Analysis Model	Traffic/Flow Pattern Model	Overall System
Accuracy	25/28(89.2%)	22/28(78.6%)	27/28(96.5%)

Demo

Attack	Characteristic	Packet analysis based ML	Traffic pattern based ML	Note when undetected
Replay	Packets mimic actual packets	Undetected	Detected	Header looks exactly same as good packets
ARP		Detected	Detected	
Probing / Scanning	Low volume; Header varies	Detected	Undetected	Certain volume would be needed for Traffic based ML to detect
DoS	High Volume; Mixed Headers	Detected	Detected	

Conclusion

1. **Optimization**
    a. Obtained significant improvements with small adjustments
    b. Identified future areas for improvement
2. **Network Intrusion Detection Component**
    a. Created monitoring system and data pipeline
    b. Demonstrated effectiveness of ML with proof of concept system
3. **Attack Vectors**
    a. Created tools for launching network - level attacks and demonstrated their detection by the IDS.

Questions

Thank You

Resources/References

1. Spire
2. Spines
3. Prime
4. Scapy
5. Sklearn
6. SPAN