Testing an Autonomic Energy Security Management Framework for Cyber-Physical Systems

Guenevere (Qian) Chen
Savannah State University
Frederic Sheldon, Sajjan Shiva
University of Memphis

Motivation and Challenges

- Distributed systems (DSs) are a collection of independent possibly heterogeneous computers communicating and coordinating with each other to work on a single computational problem, but appearing and functioning as a centralized system.
- Advantages of DS
  - Improves system performance, reliability, availability, and scalability.
  - e.g., the failure of a single node will not disable the entire system.
  - e.g., system expansion and modification are easy to realize without changing the inherent structure of the system.
- Security Challenges of DS
  - Increased system size and complexity.
  - The widespread employment of commercial-off-the-shelf (COTS) applications, operating systems, and network protocols.
  - Security solutions are far from attack-proof.

Impact of Cyber Attacks

Web Applications

- The TJX data breach incident in 2007.
  - Man-in-the-middle attacks
    - Stole 94 million customers' personal data and credit card information.
    - More than $250 million in financial damages.
  - Vulnerabilities: Lack of security patching, anti-virus software updating, and authentication cost the company.
- Target Corporation data breach in the 2013.
  - Network credentials from a third party vendor.
    - Stole more than 40 million customers' credit and debit card information
    - Target’s profit fell 46% to $520 million (fourth-quarter 2013), and 16% to $418 million (first-quarter, 2014) from the same period the year before.

Impact of Cyber Attacks (Cont'd)

Cyber Physical Systems

- Cyber Physical attacks on the Springfield IN water system cycled power on a key water pump until it failed.
- Stuxnet Worm: subverted Iranian enrichment facilities to temporally derail Iran's nuclear program in 2010 by damaging roughly 1000 centrifuges.
  - Social Engineering
  - A root kit to compromise programmable logic controllers
  - Man-in-the-Middle Attacks

Research Contributions

- Designing a self-protecting and decision support system
  - Developed a performance model for distributed systems
  - Identified system parameters affected by cyber attacks
  - Support for fully-autonomous and decision support modes
  - Support reliable, sustainable, and resilient self-protection performance under known and unknown attacks.
- Implementing the main components of self-protecting and decision support system
  - System Model
  - Forecaster
  - Intrusion Detection
  - Network Forensics Analysis
  - Intrusion Response

Research Contributions (Cont'd)

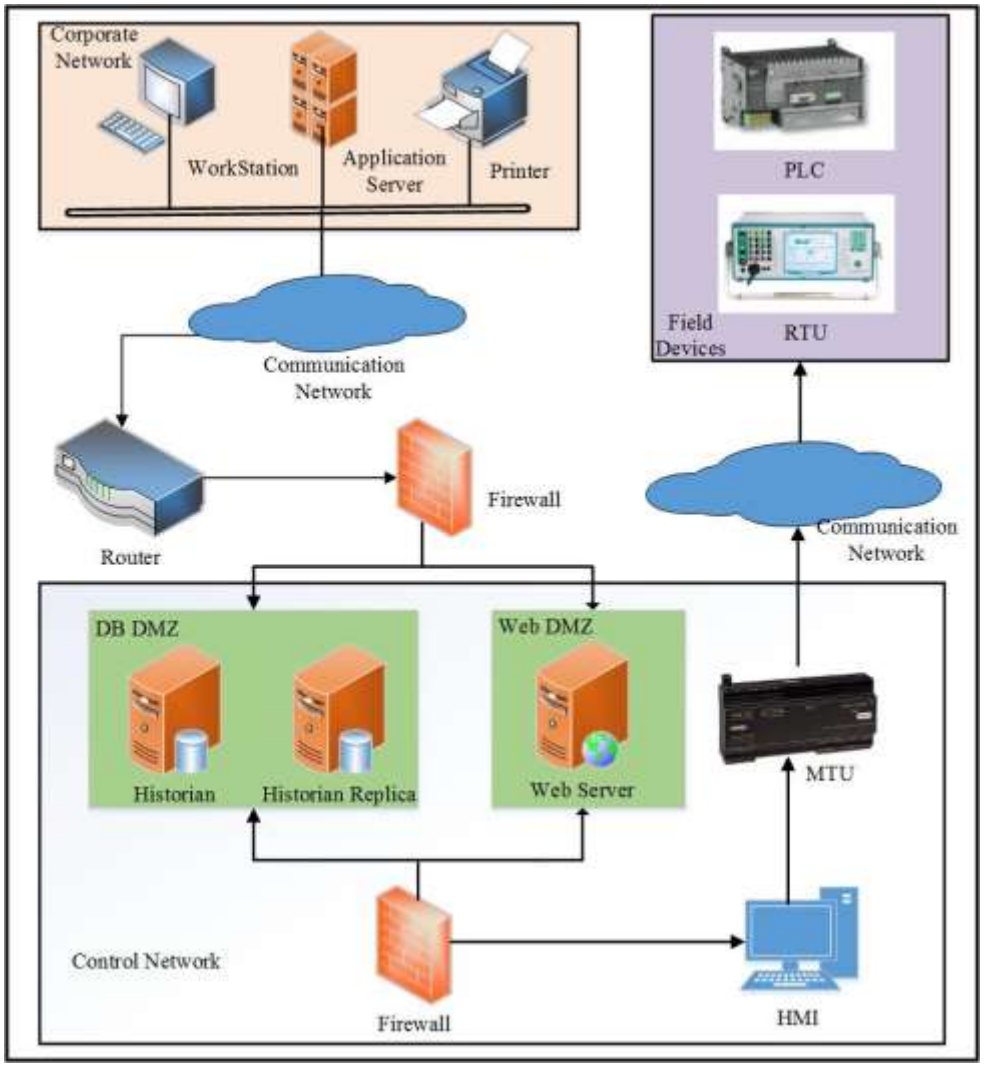
- Generic approach
  - Extendable to various application domains with few modifications
  - Simple to configure and deploy in different platforms.

- Validation of self-protecting and decision support functions
  - SCADA Testbed
  - various real-world cyber attacks are simulated
  - system security is measurably enhanced

Research Background

SCADA System Network Architecture

- SCADA systems are a type of ICSs that adopt many aspects of Information and Communications Technology (ICT/IT) to monitor and control physical processes.
  - Programmable logic controllers (PLCs) and remote terminal units (RTUs): collect and convert sensor sourced analog measurements to digital data
  - Master terminal units (MTUs): issues commands to RTUs, gathers and stores required data, process information and display the information in HMI
  - Human machine interface (HMI): operators control the process and presents data to operators.

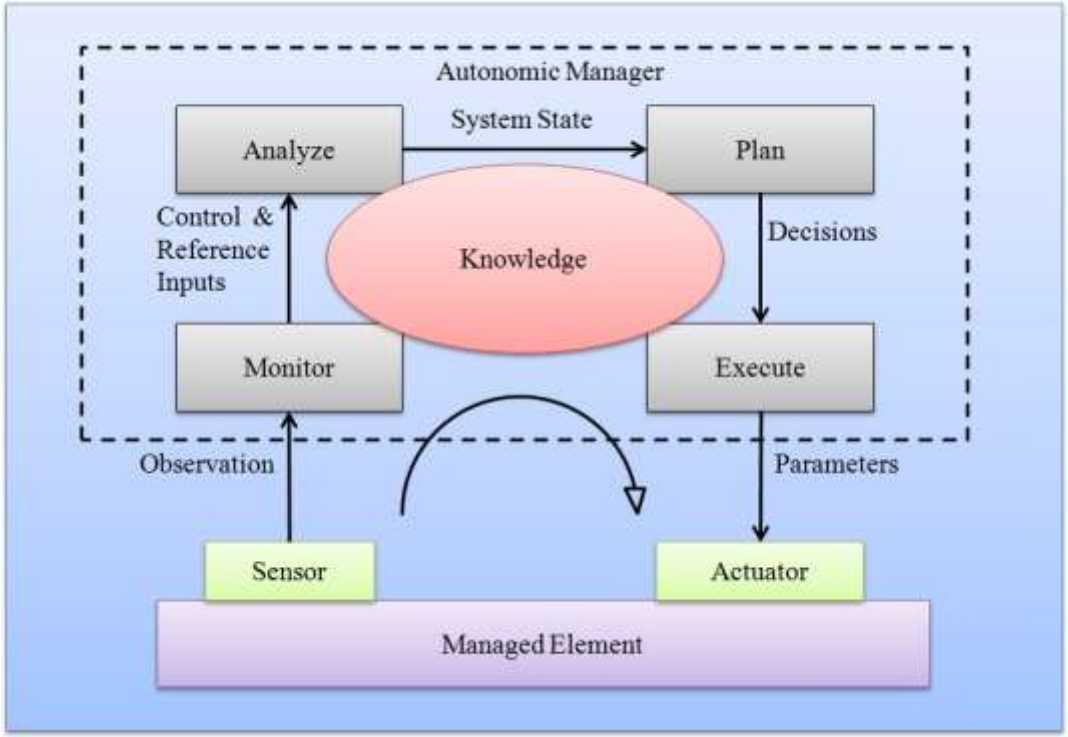


SCADA Architecture and Security Group Firewall

Autonomic Computing Overview

- Autonomic computing technology, inspired by biological autonomic nervous systems, aims to develop computational systems that are capable of configuring, optimizing, healing, and protecting themselves under different working conditions.

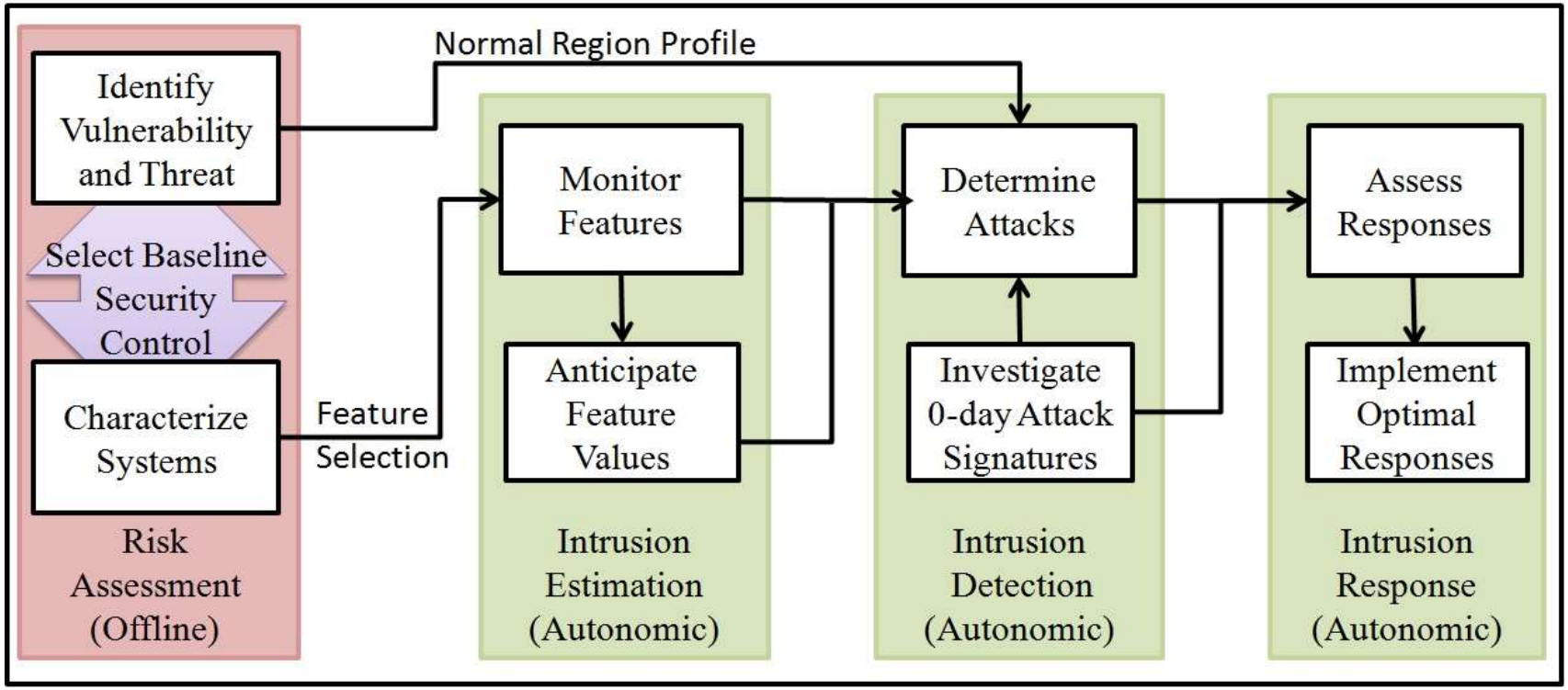
1. need to know itself
2. self configure and reconfigure under various circumstances
3. continuously optimize itself
4. recover from failure
5. self-protect from cyber attacks
6. know its environment and act accordingly
7. open environment
8. optimize and anticipate needed resources



Autonomic Computing Model

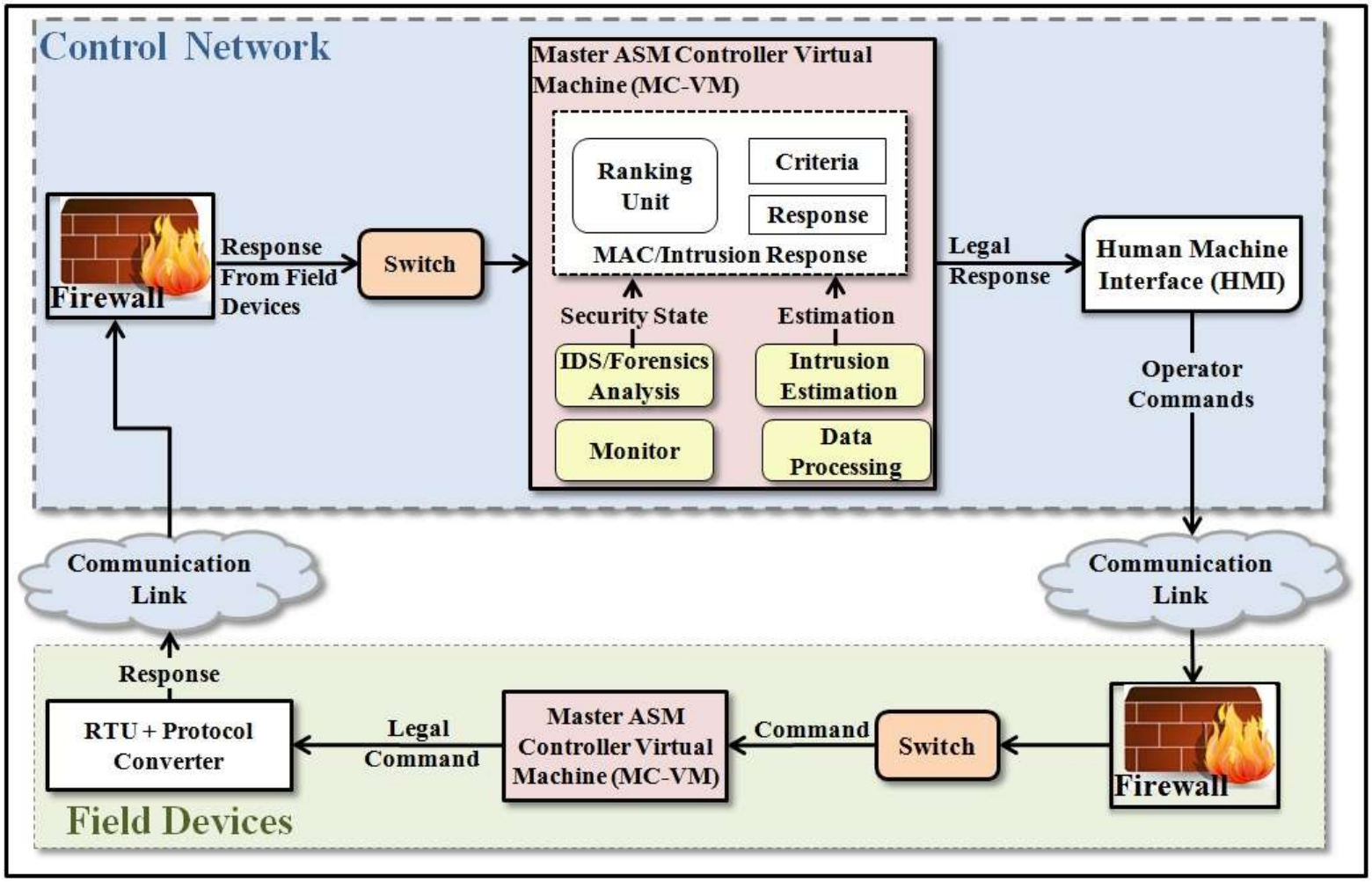
A Model of Self-Protection Functionality

- Self-protection systems anticipate and defend themselves as a whole from malicious activity

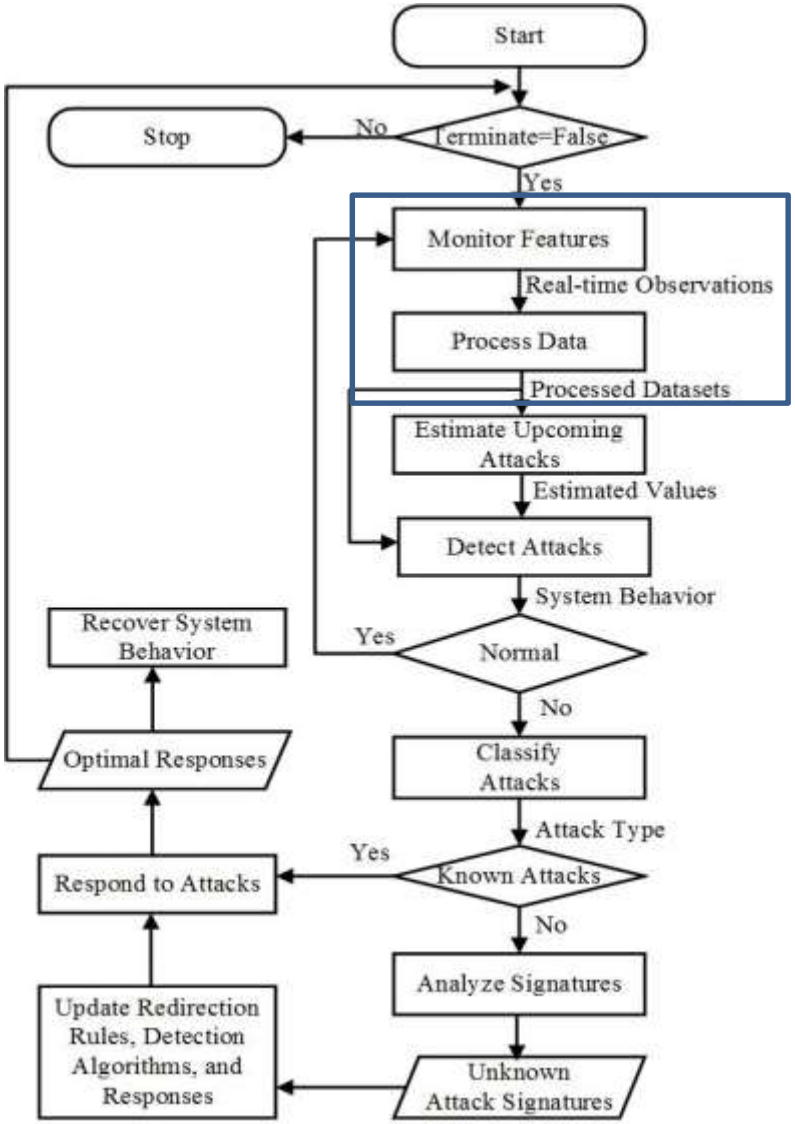


Autonomic Security Management Framework

ASM Approach for Industrial Control Systems



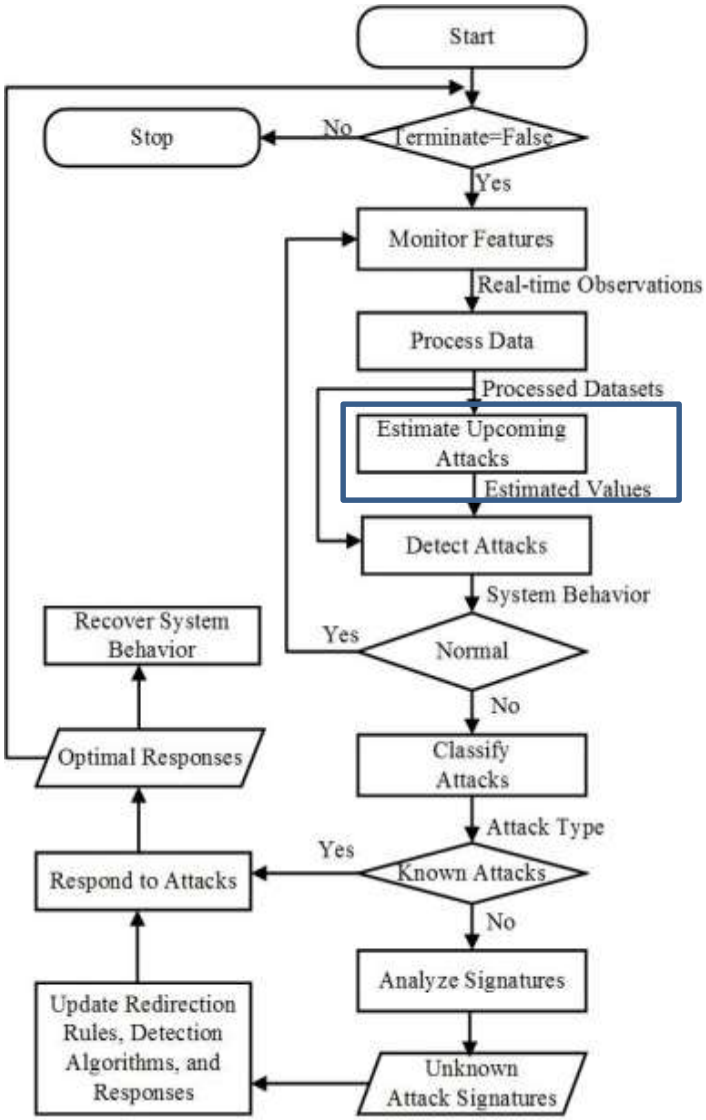
Data Sensors and Data Processing Module



Outline of the ASM Approach

- Data Sensor
  - Collect real-time observations of system performance, network utilization, and system security states
- Data Processing module
  - Process incomplete dataset
  - Formatted and Pre-processed real-time observations

Forecasting Module



Outline of the ASM Approach

- Autoregressive integrated moving average (ARIMA) model is applied to predict environmental parameters and security parameters.

$$(1-\sum_{i=1}^{p}\rho_i L^i)(1-L)^d y_t = \mu(1-\sum_{i=1}^{p}\rho_i) + (1-\sum_{j=1}^{q}\nu_j L^j)\epsilon_t$$

where *L* is the lag operator and defined as $L^1 y_t = y_{t-1}$; ρ_i and υ_j are weighting coefficients; μ is the intercept term, which is close to mean value of the time series; ϵ_t is the forecast expected error of $(y_t - \hat{y}_t)$; and $\hat{y}_t$ denotes a forecast of observation y_t

System Module

- The system module uses the estimated trending of both the environment and security parameters in conjunction with the current state of the host system to predict its future

$$\hat{x}(k) = f(x(k-1), \hat{\omega}(k), \hat{\lambda}(k), u(k))$$

Symbols

$\hat{x}(k)$: Future security state	$\hat{\omega}(k)$: Estimations of environment parameters
$\hat{\lambda}(k)$: Estimations of security parameters	$\mathrm{u(k)}$: Control/Protection mechanisms
k: Time step	*x(k-1)*: The system security state at time step k-1

Case Study: Linear Model

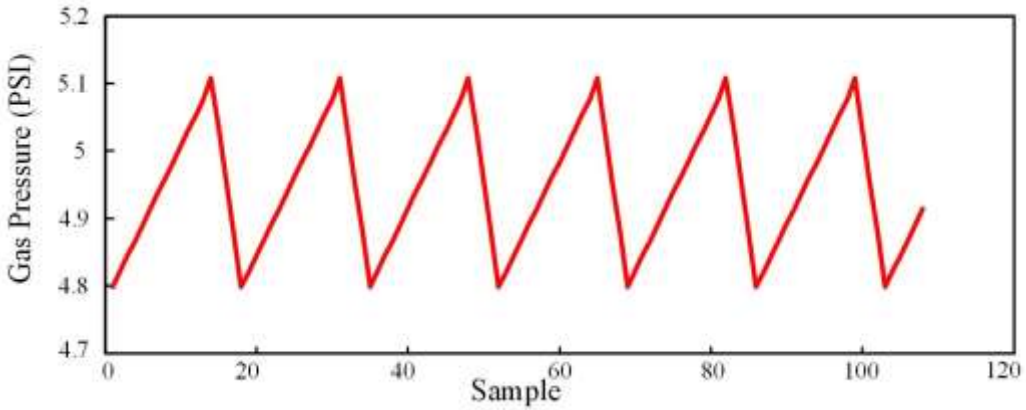
- Besides using the ARIMA method, a linear model has been constructed that captures the behavior of the physical system (gas pipeline and water storage tank systems).

$$\omega(k) = Ak + B$$

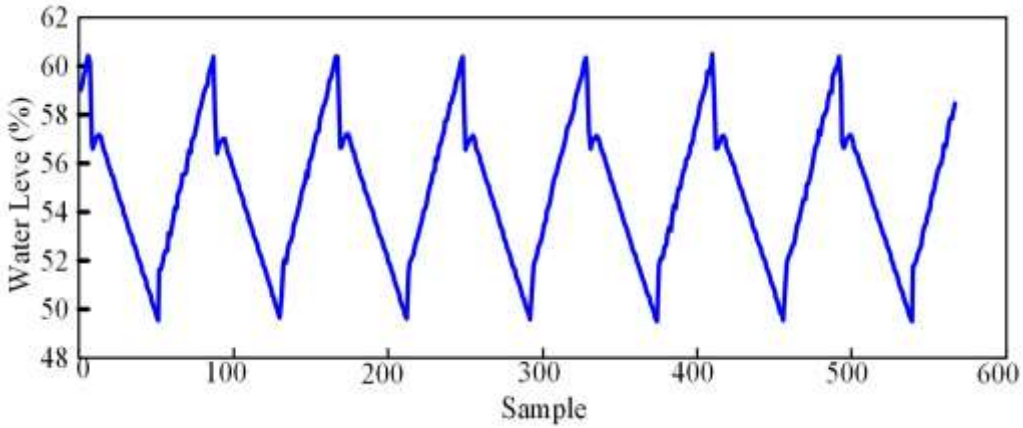
ω : the gas pressure/water level	$k \in \mathbb{Z}$: the number of samples.

Linear Model

- For a gas pipeline system (One period is 850 ms, 17 samples long)
  - A=0.0231, B=4.7752, when $1 \leq k \leq 13$
  - A=-0.0767, B=5.1850, when $14 \leq k \leq 17$
- For a water storage tank system (One period is 4000 ms, 80 samples long)
  - A = 0.256 and B = 51.181 when $1 \leq k \leq 35$
  - A = -1.976 and B = 62.090 when $36 < k < 45$
  - A= 0.0325 and B=56.718 when $46 < k < 80$



Normal Trend of Gas Pressure

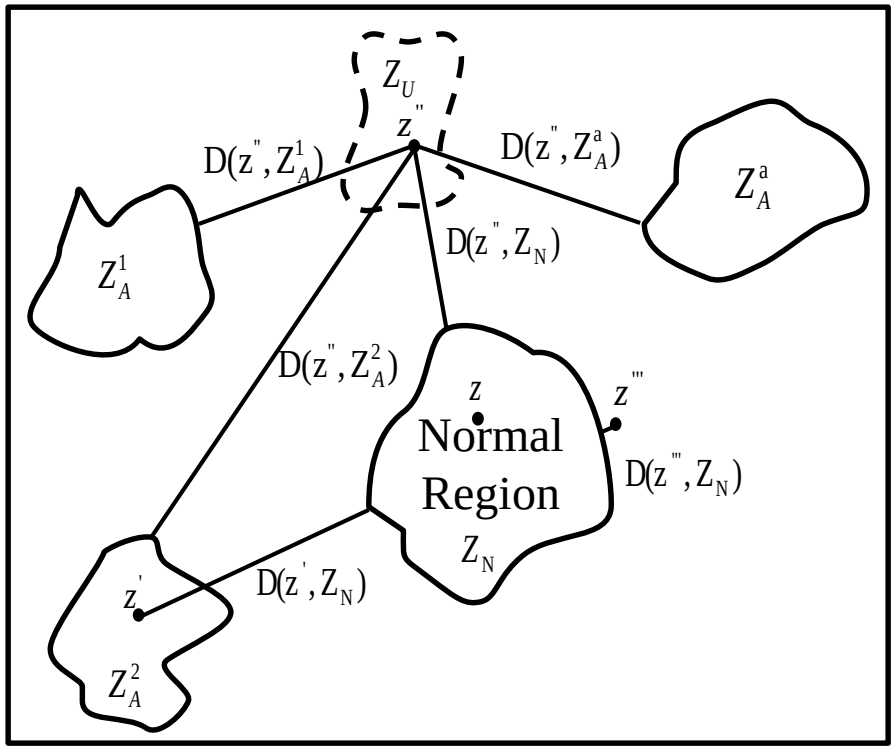


Normal Trend of Water Level

Future System Security State

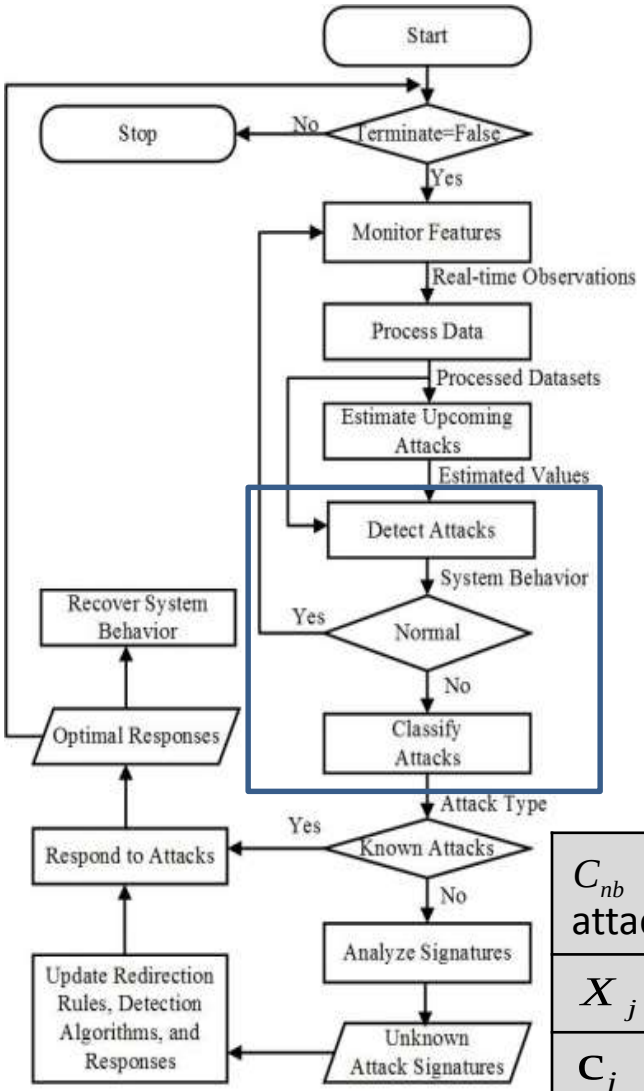
- auto.arima from the forecast package of R.
- ARIMA (p,d,q)
  - **p** is the number of autoregressive terms [0,5]
  - **d** is the number of non-seasonal differences
  - **q** is the number of lagged forecast errors in the prediction equation [0,5]

Z_N : The DS normal region	Z_A : Known attack regions
$z = [x\ \omega\ \lambda]$: DS system security	$D(z, Z_n) = \min_{z_n \in Z_N} \lVert z - z_n \rVert$: Distance between system Normal Region and system security state



Security States

Intrusion Detection System



Outline of the ASM Approach

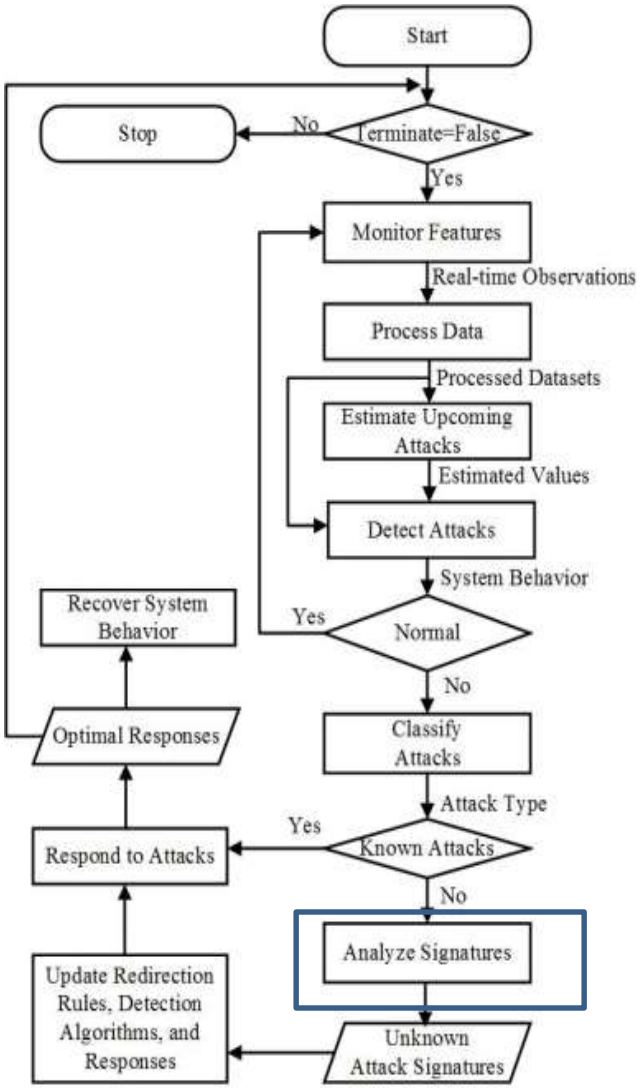
- The intrusion detection system (IDS) is a data mining tool that allows real-time event analysis. The goal of this tool is to provide accountability for intrusive activities
  - Performance-based IDS (PIDS)
    - Detect and both known and unknown attacks and classify known attacks
    - Naïve Bayesian

$$C_{nb} = \arg\max_{c_i \in C} P(c_i) \prod_{j=1}^{K} P(x_j|c_i)$$

$$P(x_j|c_i) = \frac{n_a + mp}{n + m}$$

C_{nb} : IDS output- the classified attack type	n_a : The number of examples for which c = C_i and x = x_j
X_j : PIDS Features j∈ [0,10]	p : A priori estimate for $P(x_j \mid c_i)$
C_i : Attack types i∈ [0,7]	m: The equivalent sample size
C : Seven Attacks	K=10

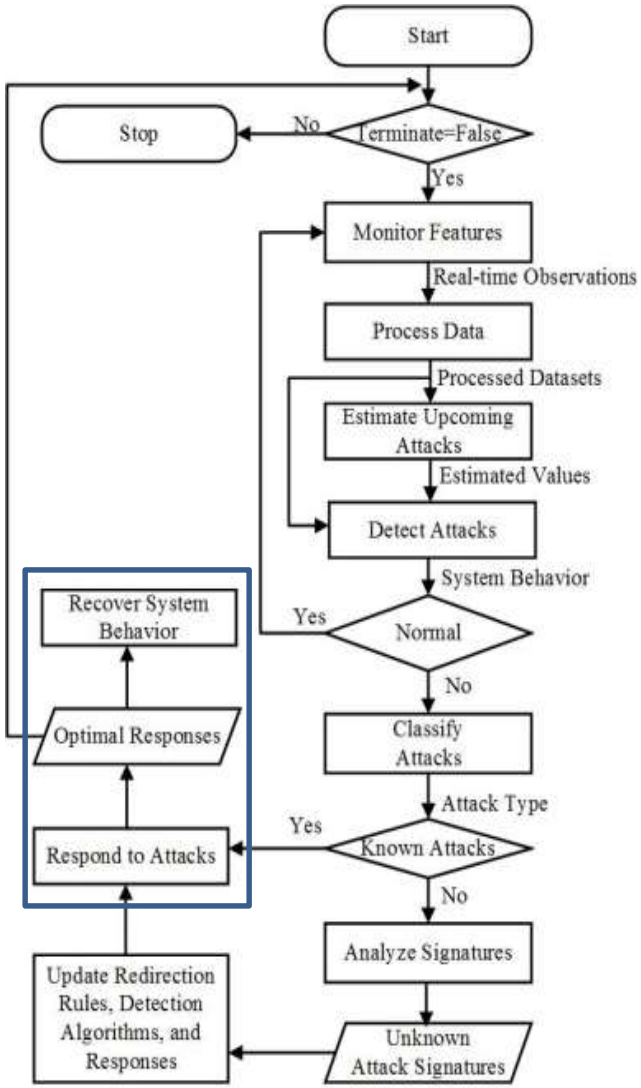
Learning Module



- Monitors and analyzes network traffic
- Derives and obtains the causes, impacts of novel attacks and unique attack signatures
- The learning module only captures and saves malicious packets to hard disk rather than logging all events to the hard disk to avoid exhausting system resources

Outline of the ASM Approach

Multi-criteria Analysis Controller (MAC)



Outline of the ASM Approach

- The MAC evaluates candidate responses as an intrusion response system (IRS) and initiates the most appropriate responses necessary to recover the system performance under two conditions
  - If estimations of the DS performance are abnormal
  - Real-time observations are identified as attacks
- Decision making methods
  - Fuzzy-logic
  - Preference Ranking Organization METHod for Enrichment Evaluations II (PROMETHEE II)

Candidate Responses

- Dropping Malicious Commands
- Packet Filtering
- Network Disconnection
- Serial Port Disconnection
- Replacement of Compromised Devices
- One time Authentication
- Termination of Physical Processes
- Isolation of Compromised Devices

Criteria for Assessing Effectiveness of Responses

- Enhancement of Security (C1): How fast/effective candidate responses could recover the system behavior of the compromised main host back to normal.
- Operational Costs (C2): The financial cost of implementing candidate mitigation responses.
- Maintenance of Normal Operations (C3): Whether the candidate mitigation responses interrupt normal operations
- Impacts on Property, Finance, and Human Lives (C4): How well the mitigation responses can reduce these impacts.

Fuzzy-logic Method

- Evaluate the efficiency of responses and the importance of relative weights of criteria
- Normalized these values to the range [0, 1]
- The cost of using a response $u \in U$ is defined as follows :

$$Cost(u) = \sum_{r \in R} w_r \, v(r, u)$$

- The best response u^* is the one corresponding to the smallest cost

$$u^* = \arg\min\{\sum_{r \in R} w_r v(r, u) | u \in U\}$$

R: A set of Criterion	*U*: A set of candidate responses
W_r : Weight of the criterion *r*	*V (r, u)*: is a map that assigns a value to each criterion

PROMETHEE II

- The PROMETHEE II approach is more complex but more precise than the fuzzy-logic method.
- It can be used when the fuzzy-logic approach fails to make an efficient decision.
- The procedures for this method are outlined as follows:
  - Step 1: Compute and compare preference degree
    - Determine the preference degree of each response with respect to others for each criterion.

$$(\forall u, u' \in U)(\forall r \in R) \quad d_r(u, u') = g_r(u) - g_r(u')$$

  - Step 2: Select a preference function (Usual, Quasi, V-Shape, Level, V-Shape with Indifference and Gaussian Criterion)
    - The preference degree between two actions:

$$P_r(u, u') = F_r\left(d_r(u, u')\right)$$

u and u' : Candidate responses	$d_r(u,u')$: pair-wise comparison
$g_r(u)$: Evaluation value of action *u* for criteria *r*	$F_r(d_r(u,u'))$: Preference function for criterion r

PROMETHEE II (Cont'd)

- Step 3: Calculate global preference index
  - Determine weights of each criterion.
  - The preference of protection mechanism u over u'

$$\pi(u, u') = \sum_{r \in R} w_r P_r(u, u')$$

- Step 4: Calculate positive and negative outranking flows
  - The positive (negative) outranking flow (ϕ^+ , ϕ^-) represents the extent to which a response is higher (lower) than other responses.
  - The net outranking flow ϕ is basically the difference (n: the number of candidate responses. j∈ [1,n])

$$\phi^+(u) = \frac{1}{n-1} \sum_{u_j \in U} \pi(u, u_j)$$

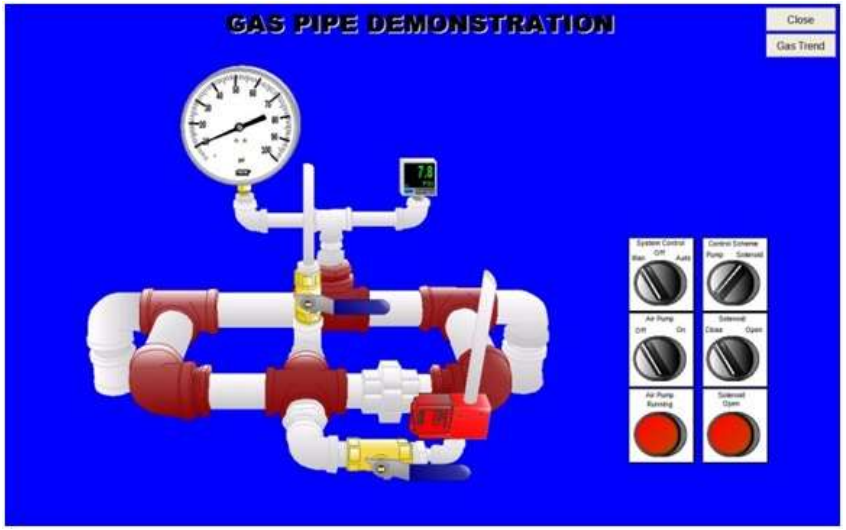
$$\phi^-(u) = \frac{1}{n-1} \sum_{u_j \in U} \pi(u_j, u)$$

$$\phi(u) = \phi^+(u) - \phi^-(u)$$

Validation Results

SCADA Testbed

- The virtual SCADA testbed presents a laboratory scale ICS containing:
  - Process simulator: model physical processes
  - Virtual device: model RTUs and MTUs
  - Configuration files: set/modify communication protocols and transmission speeds for process simulators and virtual devices
  - Data loggers: log gas pressure/water level.



Gas Pipeline HMI

Water Tank HMI

Validating Self-protection in SCADA System

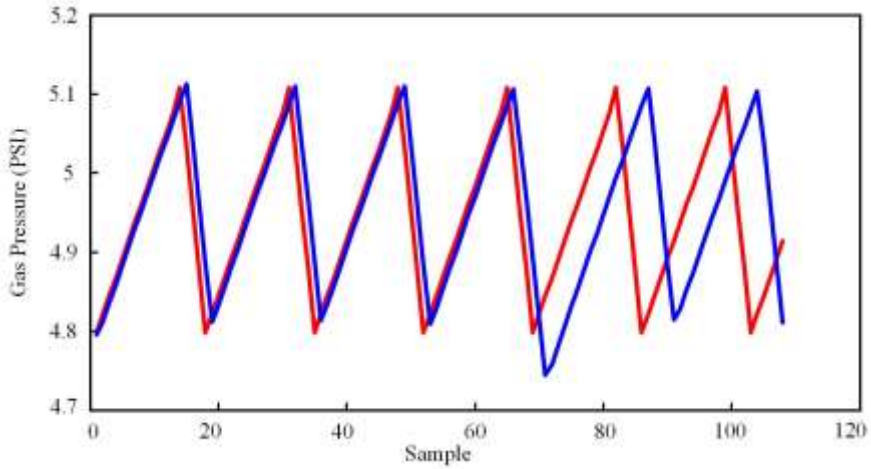
- Experiment One ***Function Code Scanning***: collects information of SCADA systems (the gas pipeline system) but has low impact on property damages .
- The most appropriate response is: ***Dropping Malicious Commands***

Assessment of Recommended Responses Example for FCS Attacks

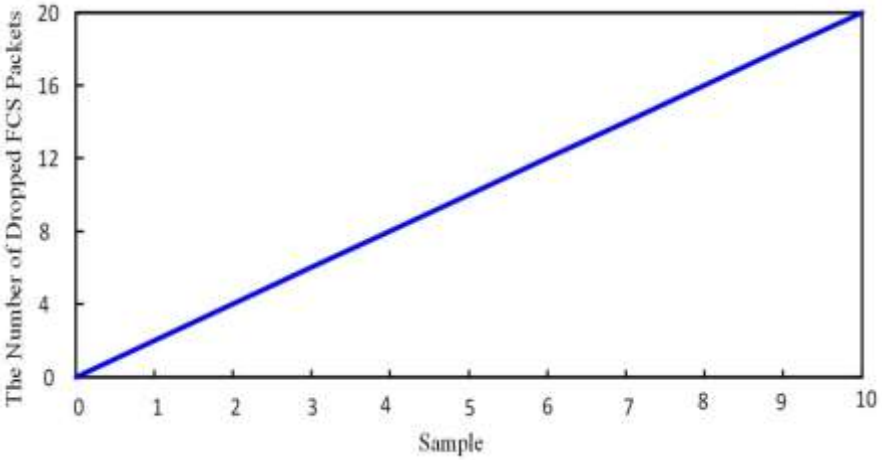
Ranking	Response	C1	C2	C3	C4	Total Value (Auto/ Semi-Auto)
1	Dropping Malicious Commands	0	0.3	0	0.2	0.165 (Auto)
2	Packet Filtering	0	0.5	0.4	0.2	0.077 (Auto)
3	Replacement of Compromised Devices	0	1	0	0.6	0.054 (Semi-Auto)
4	One time Authentication	0.5	0.5	0.2	0.2	-0.002 (Auto)
5	Serial Port Disconnection	0	0.8	0.8	1	-0.034 (Semi-Auto)
6	Network Disconnection	0	0.8	1	1	-0.058 (Semi-Auto)
7	Termination of Physical Processes	0	1	1	1	-0.089 (Semi-Auto)
8	Isolation of Compromised Devices	1	0.8	0.4	0.6	-0.113 (Semi-Auto)

Validating Self-protection in SCADA System (cont'd)

- All scanning packets have been dropped
- Attackers cannot obtain meaningful function codes via the attack, and the window of vulnerability is closed.



Observation and Prediction of Gas Pressure under FCS Attacks.



The Number of Dropped FCS Attack Packets.

Validating Decision Support in SCADA Systems

- Experiment Two: Malicious Parameter Command Injection (MPCI) Attacks (Gas Pipeline)

Experiment Two: MPCI Attacks

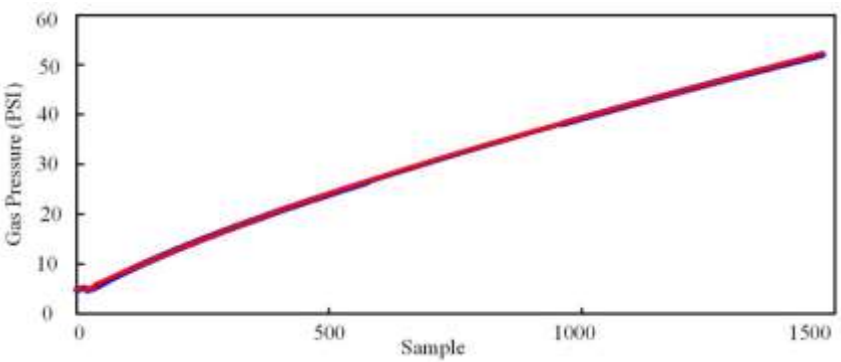
- MPCI attacks inject false commands to overwrite remote terminal registers, which may interrupt normal infrastructure operations or device communications.
- The gas pressure set point is changed from 5.00 to 2.93, and the PID gain is changed from 115 to 1.05610820582e-38.
- The most appropriate response is: ***Packet Filtering***

Assessment of Recommended Responses Example for MPCI Attacks

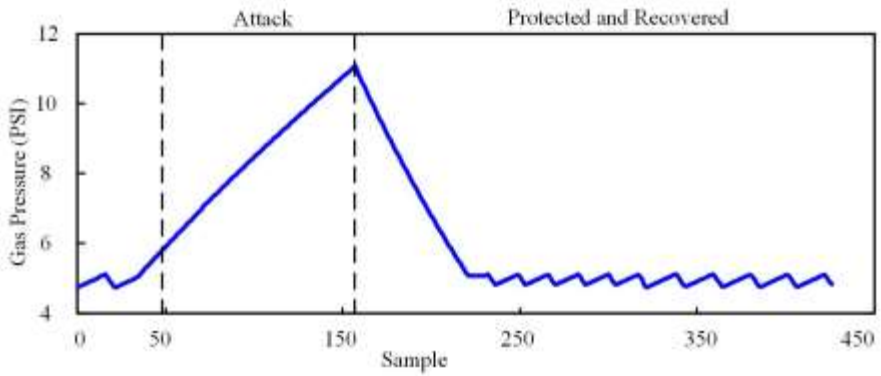
Ranking	Response	C1	C2	C3	C4	Total Value (Auto/Semi-Auto)
1	Packet Filtering	0	0.5	0.4	0.2	0.093 (Auto)
2	Replacement of Compromised Devices	0	1	0	0.6	0.069 (Semi-Auto)
3	Dropping Malicious Commands	0.5	0.3	0	0.2	0.040 (Auto)
4	One time Authentication	0.5	0.5	0.2	0.2	-0.016 (Auto)
5	Serial Port Disconnection	0	0.8	0.8	1	-0.018 (Semi-Auto)
6	Network Disconnection	0	0.8	1	1	-0.042 (Semi-Auto)
7	Isolation of Compromised Devices	0.3	0.8	0.4	0.6	-0.053 (Semi-Auto)
8	Termination of Physical Processes	0	1	1	1	-0.074 (Semi-Auto)

Experiment Two: MPCI Attacks (Cont'd)

- ***Packet Filtering*** is executed by the MAC to protect the system against future MPCI attacks from sample 155 onwards .
- The experienced system administrator regulates the set point and PID gain back to normal values at sample 220.
- After sample 230, the system is set back to "Auto", and the gas pressure returns back to the normal region even in the face of continuous and similar MPCI attacks.



Observation and Prediction of Gas Pressure under MPCI Attacks without Protection.



Protecting and Maintaining Gas Pressures by the ASM Approach

Experiment Three: Alerted Alarm Condition Attack

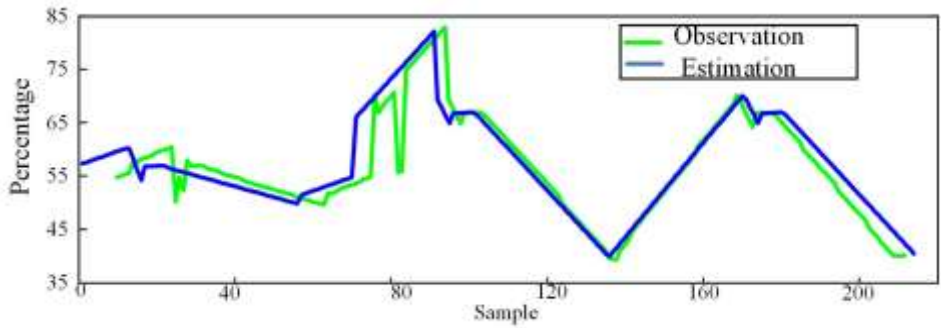
- AAC attacks modify the alarm condition for the water tank system:
  - L setpoint from 50.00% to 40.00%
  - alter H setpoint from 60.00% to 70.00%.
  - HH (the high high alarm) setpoint is modified to 80.00% from 70:00%
  - LL (the low low alarm) is changed to 10.00% from 20.00%.
- The most appropriate response is: ***Replacement of Compromised Devices***

Assessment of Recommended Responses Example for AAC Attacks

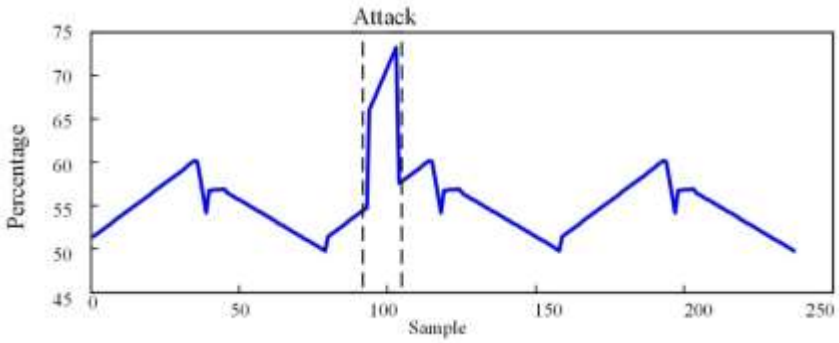
Ranking	Response	C1	C2	C3	C4	Total Value (Auto or Semi-Auto)
1	Replacement of Compromised Devices	0	0.5	0	0.5	0.141 (Auto or Semi-Auto)
2	Packet Filtering	0	0.5	0.4	0.2	0.085 (Auto)
3	Dropping Malicious Commands	0.5	0.3	0.1	0.2	0.049 (Auto)
4	One time Authentication	0.8	0.3	0.2	0.2	-0.010 (Auto)
5	Serial Port Disconnection	0	0.8	0.8	1	-0.026 (Semi-Auto)
6	Termination of Physical Processes	0	0.8	1	1	-0.052 (Semi-Auto)
6	Network Disconnection	0	0.8	1	1	-0.052 (Semi-Auto)
8	Isolation of Compromised Devices	0.5	0.8	0.8	0.6	-0.137 (Semi-Auto)

Experiment Three: Alerted Alarm Condition Attacks (Cont'd)

- At sample 94, the attack modified the alarm thresholds
- The water level is increased to an abnormal 65.99%.
- At sample 104 when "Replacement of Compromised Devices" is implemented, a replica RTU replies to the MTU who in turn sends commands to control the water level of this critical infrastructure.



Observations and Estimations of the Water Level Without applying the ASM Approach.



Protecting and Maintaining Water Level by the ASM Approach.

Conclusions

- The ASM framework
  - It has been proven usable, effective, and extensible.
  - This implementation realizes fully-autonomic functions, which are the first steps in developing a general framework to protect ICSs from known and unknown cyber attacks.
  - Decision support systems are used to protect high risk high impact systems from various cyber attacks.

Future Research

- Internet of Things-based (IoT-based) Ecosystems:
  - Only when strong security and privacy are in place will IoT be most feasible and practical.
  - One of the future trends of IoT is towards autonomic resources since it becomes impossible to manage the increasing complexity by system administrators.
- Autonomic Risk Assessment:
  - The autonomic risk assessment function does not already exist.
  - An autonomic risk assessment is an essential module for the successful implementation of a self-protecting system.
- Network Forensic Analysis:
  - The autonomic network forensic analysis tool is supposed to be composed of deep-analysis systems and flow-analysis systems.
  - Such tools have the ability to analyze log files, traffic flows, user activities, asset data, and vulnerabilities.

Questions and Comments

Thank you !